

CS107e

Computer Systems from the Ground Up

**Chris Gregg, Philip Levis,
Maria Fernandez, Matt Trost,
Liana Keesing, Anna Misteleva**

Autumn 2021

<https://cs107e.github.io/>



Chris



Phil



Maria



Matt



Liana



Anna

Learning Goal I

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

OK

```
int counter;  
int calc() {...}
```

```
int a = 20;  
unsigned int b = 6;  
if (a < b) {...}
```

Not OK

```
int calc() {  
    int counter;  
    ...  
}
```

```
int a = -20;  
unsigned int b = 6;  
if (a < b) {...}
```

OK

```
int counter;  
int calc() {...}
```

Not OK

```
int calc() {  
    int counter;  
    ...  
}
```

```
int a = 20;  
unsigned int b = 6;  
if (a < b) {...}
```

```
int a = -20;  
unsigned int b = 6;  
if (a < b) {...}
```

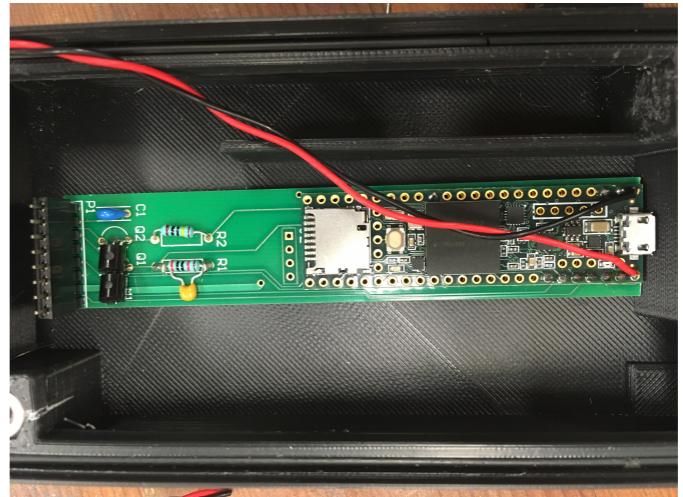
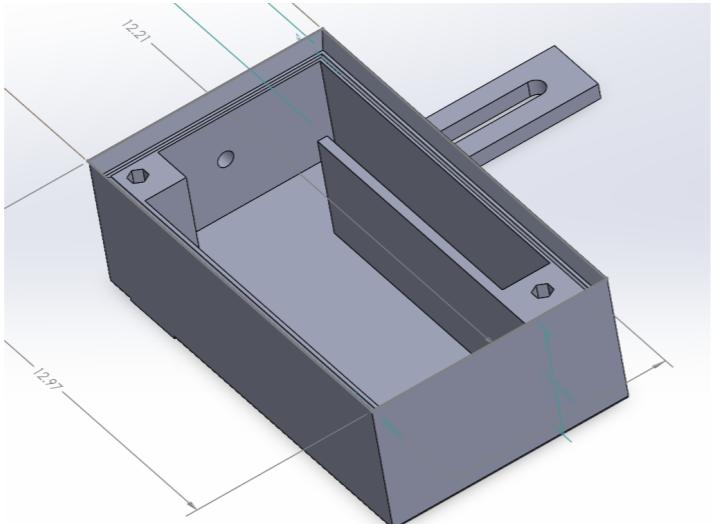
Why?

```
int main( ) {  
    ...  
}
```

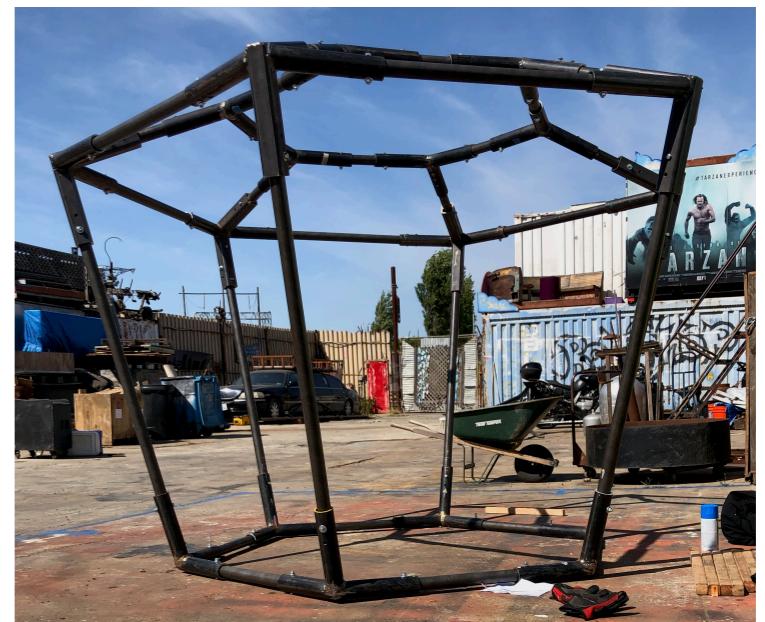
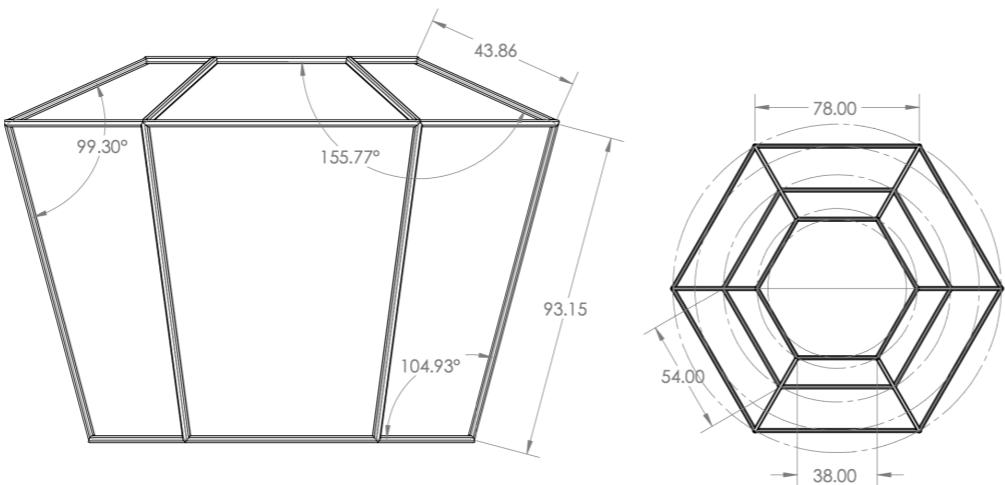
How does your program start
at the first instruction of
main()?
Or does it really?

Understanding is Empowering

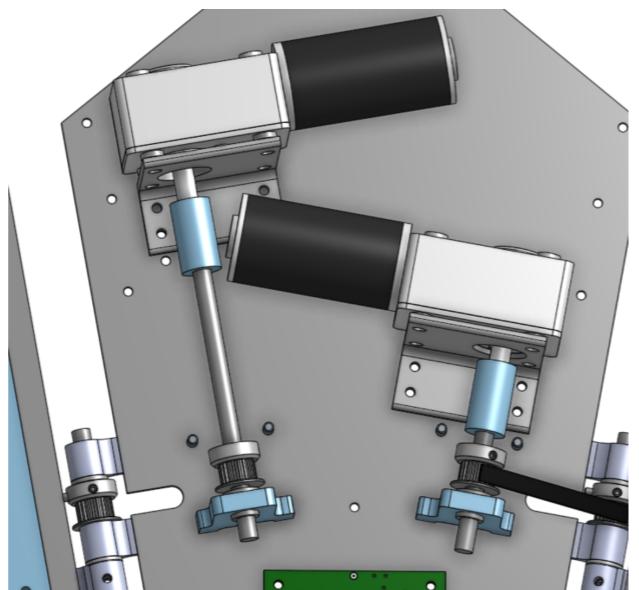
3D printing plastic



Machining steel



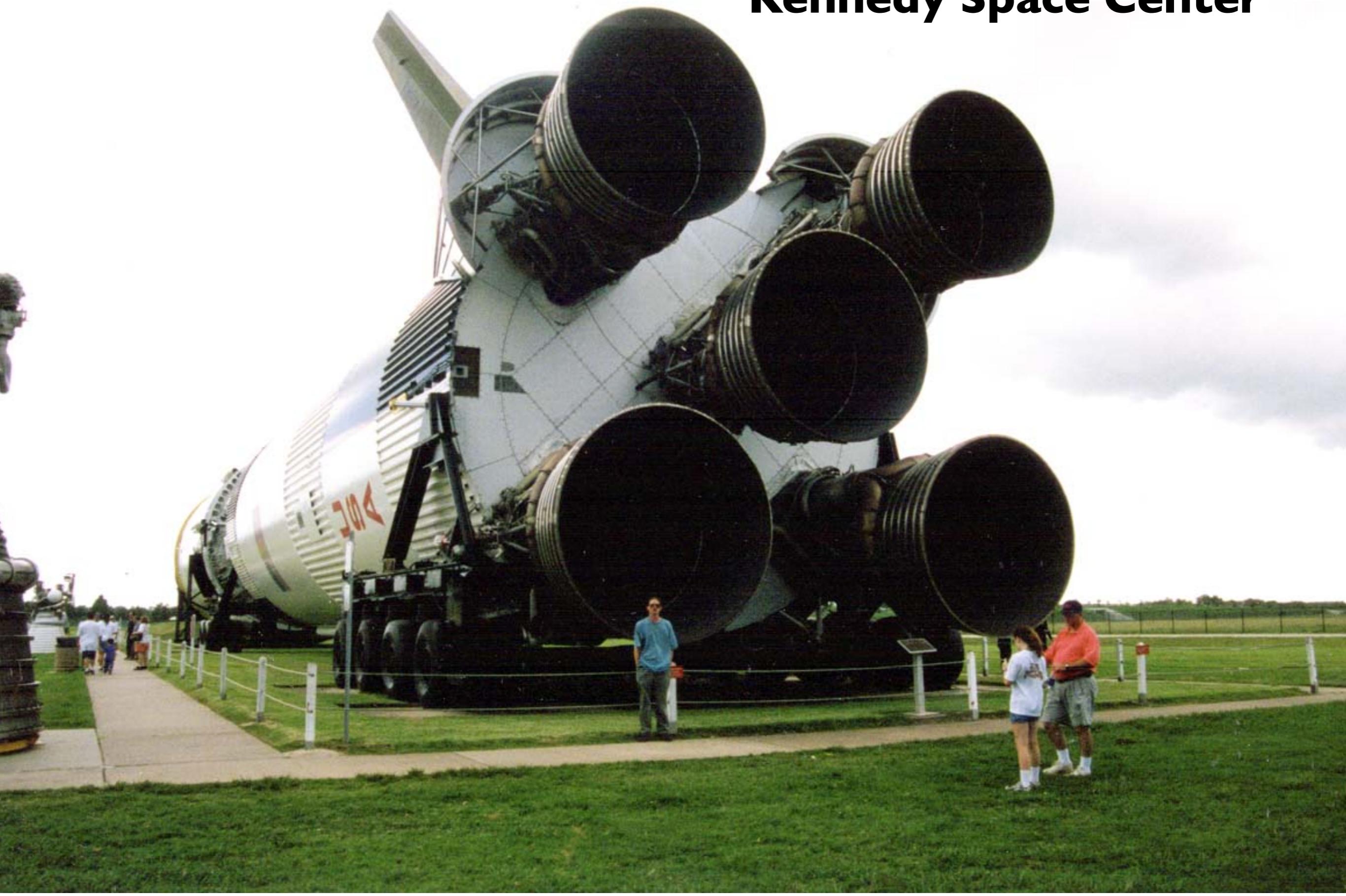
Designing mechanisms

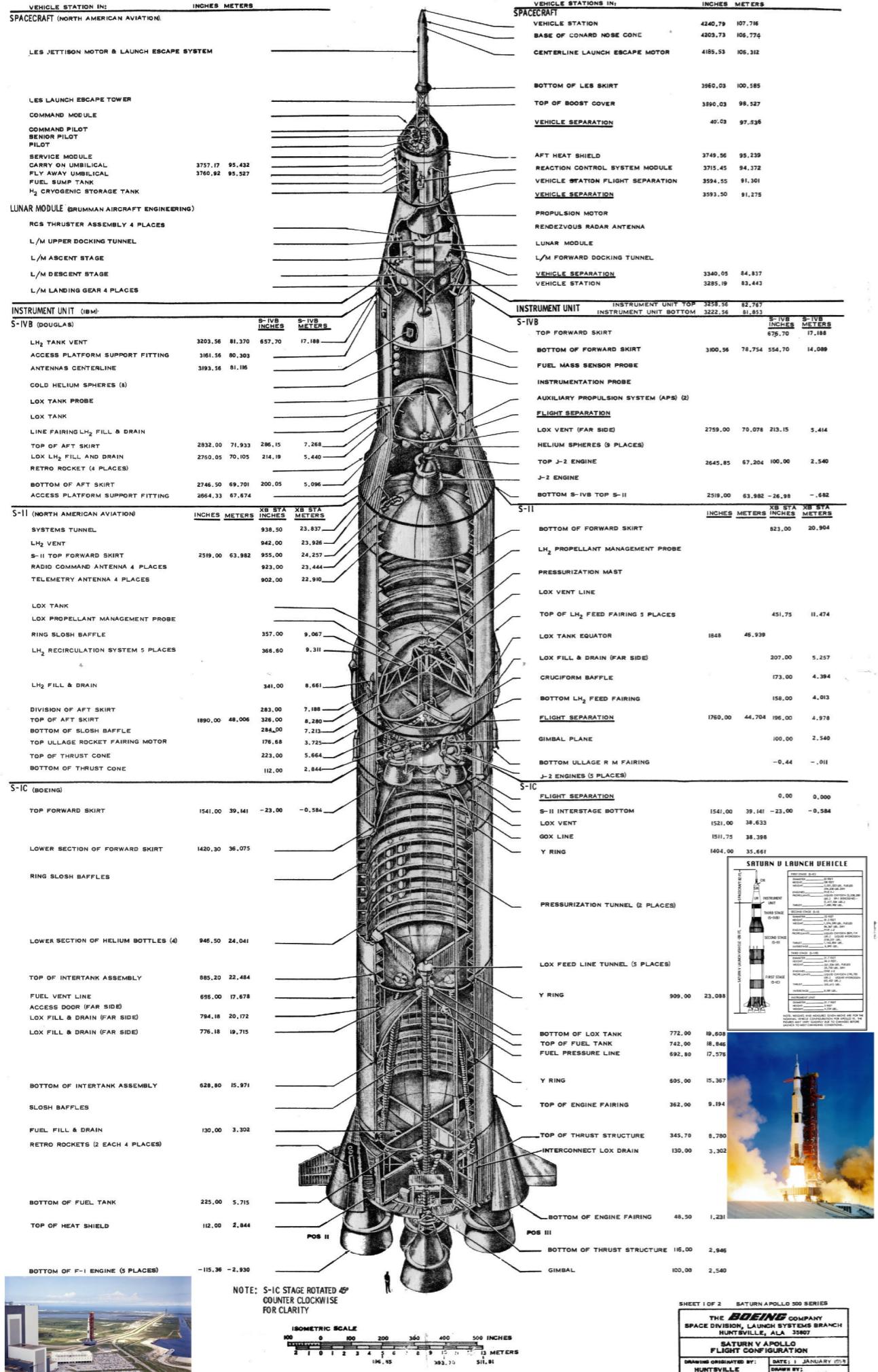


First steps are often the hardest

That's why we're here!

Saturn V Kennedy Space Center

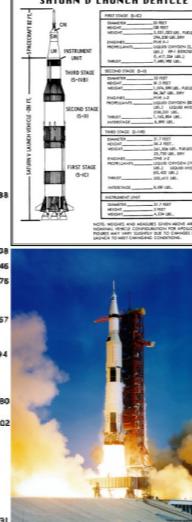




Command Module 64,000 lbs

Saturn V 6,200,000 lbs

Payload 1.5% of total weight



Falcon 9





Engineer for Excellence!

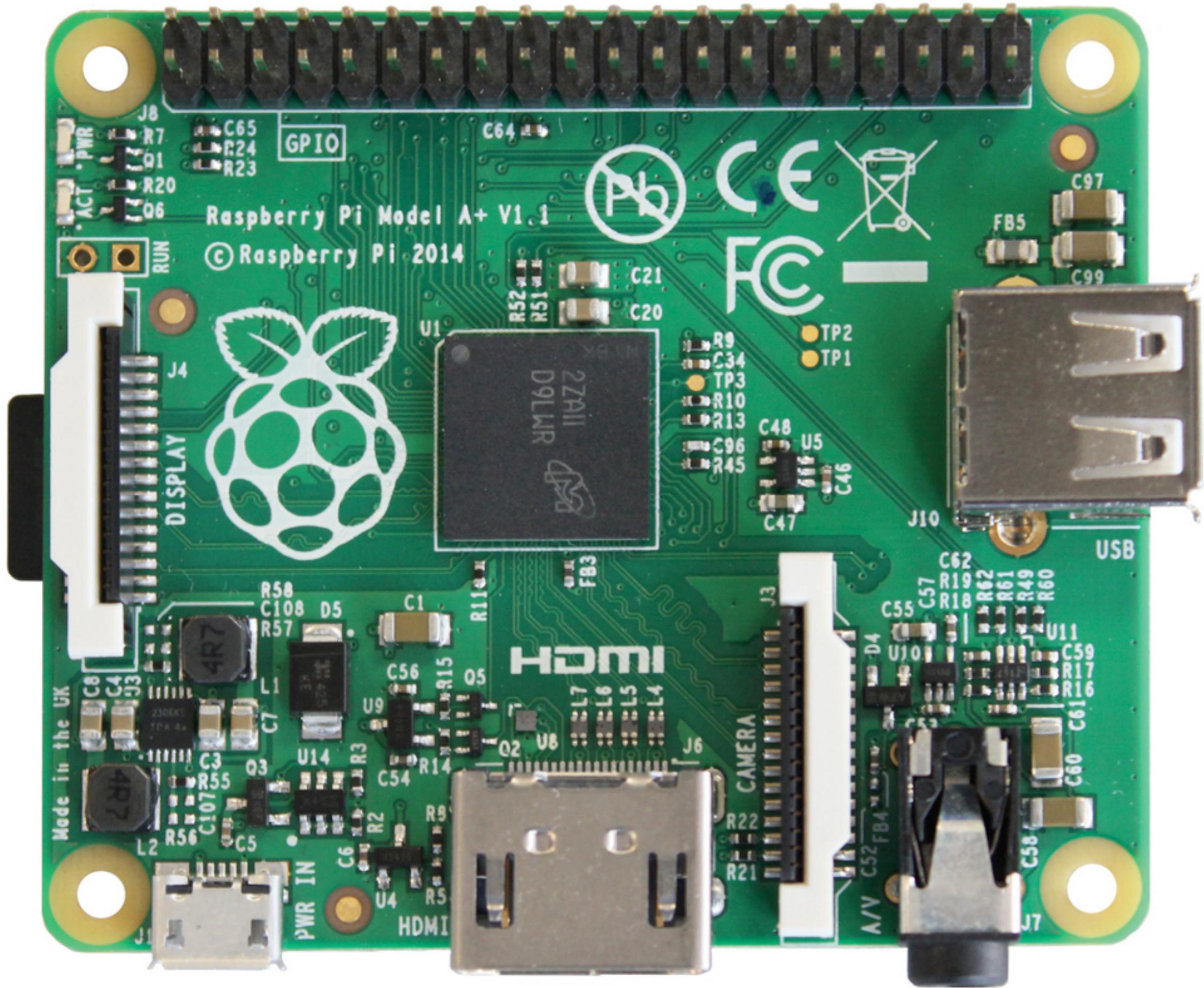
Perseverance!

Bare Metal on the Raspberry Pi

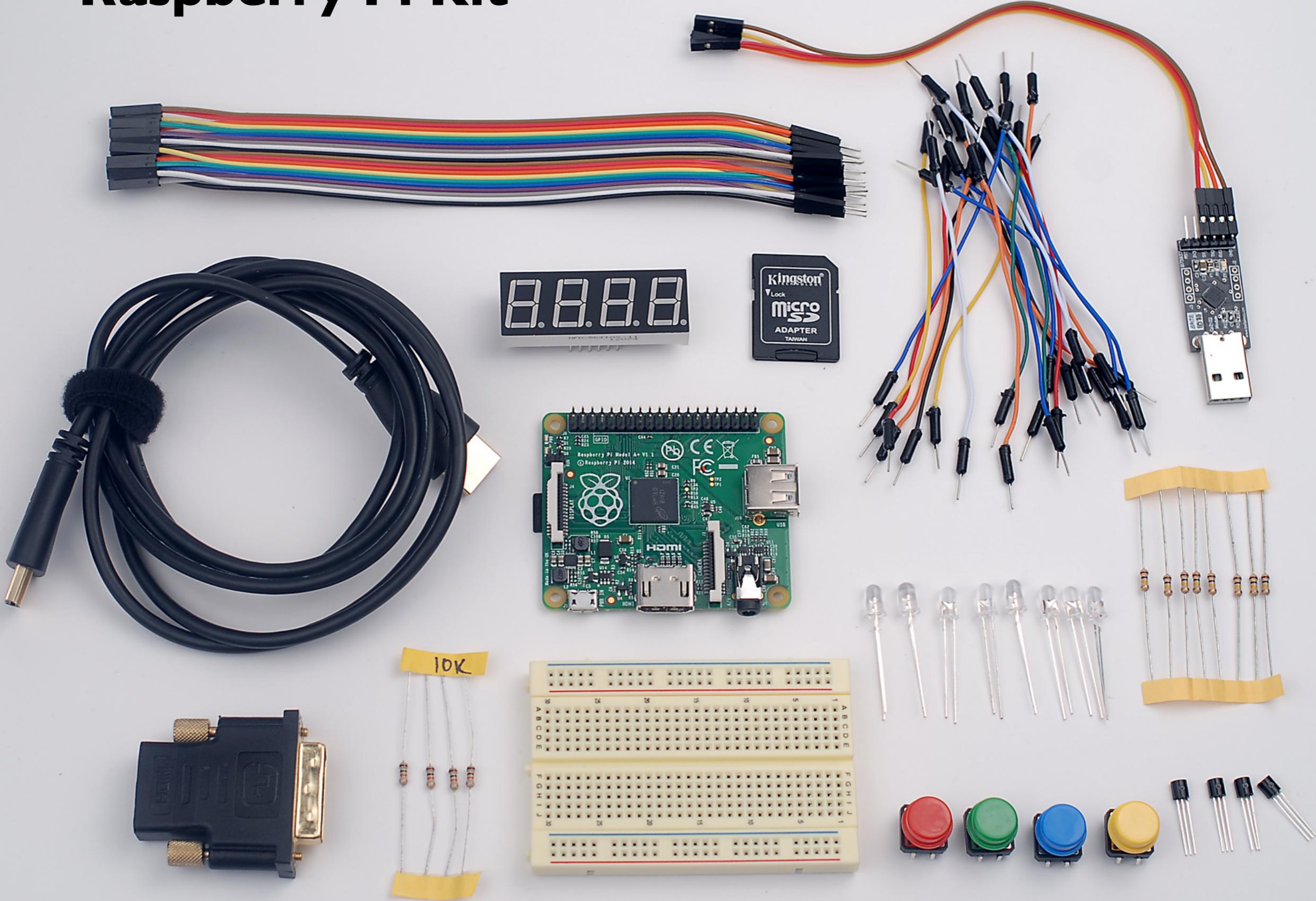
Definition: Bare metal programming involves no operating system (programmer constructs libraries)

Bare metal programs boot and startup on their own, and directly control peripherals

You'll understand every line of code in the system.



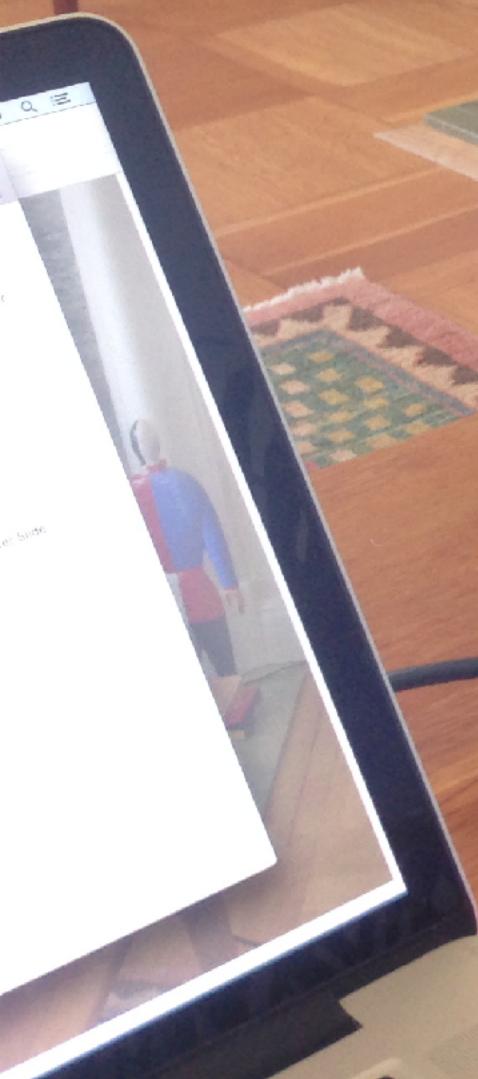
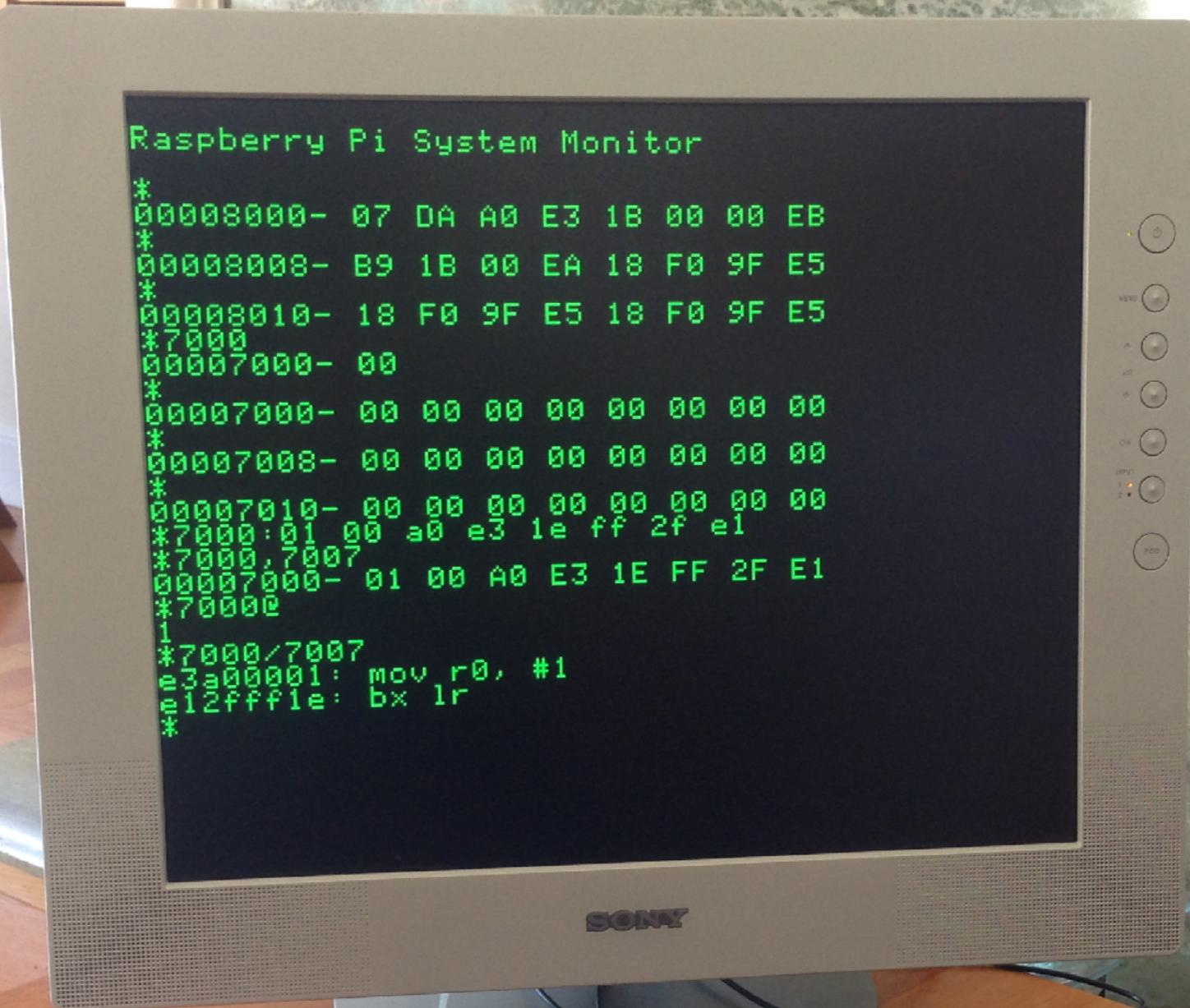
Raspberry Pi Kit



Raspberry Pi Shell

Raspberry Pi System Monitor

```
*00008000- 07 DA A0 E3 1B 00 00 EB
*00008008- B9 1B 00 EA 18 F0 9F E5
*00008010- 18 F0 9F E5 18 F0 9F E5
*7000
00007000- 00
*00007000- 00 00 00 00 00 00 00 00
*00007008- 00 00 00 00 00 00 00 00
*00007010- 00 00 00 00 00 00 00 00
*7000:01 00 a0 e3 1e ff 2f e1
*7000,7007
00007000- 01 00 A0 E3 1E FF 2F E1
*7000@
1
*7000/7007
e3a00001: mov r0, #1
e12fffffe: bx lr
*
```



**Almost every instruction
will be code you've written!**

Learning Goal 2

**Master your tools
Learn their value**

Software Tools

UNIX command line: bash, cd, ls, ...

Programming languages: C, ...

gcc

as

ld

binutils: nm, objcopy, objdump, ...

make

git and github.com

documentation: markdown

Different Tools for Different Jobs



<http://dans-woodshop.blogspot.com/>

Organized Development Environment



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

Don't Avoid Activation Energy

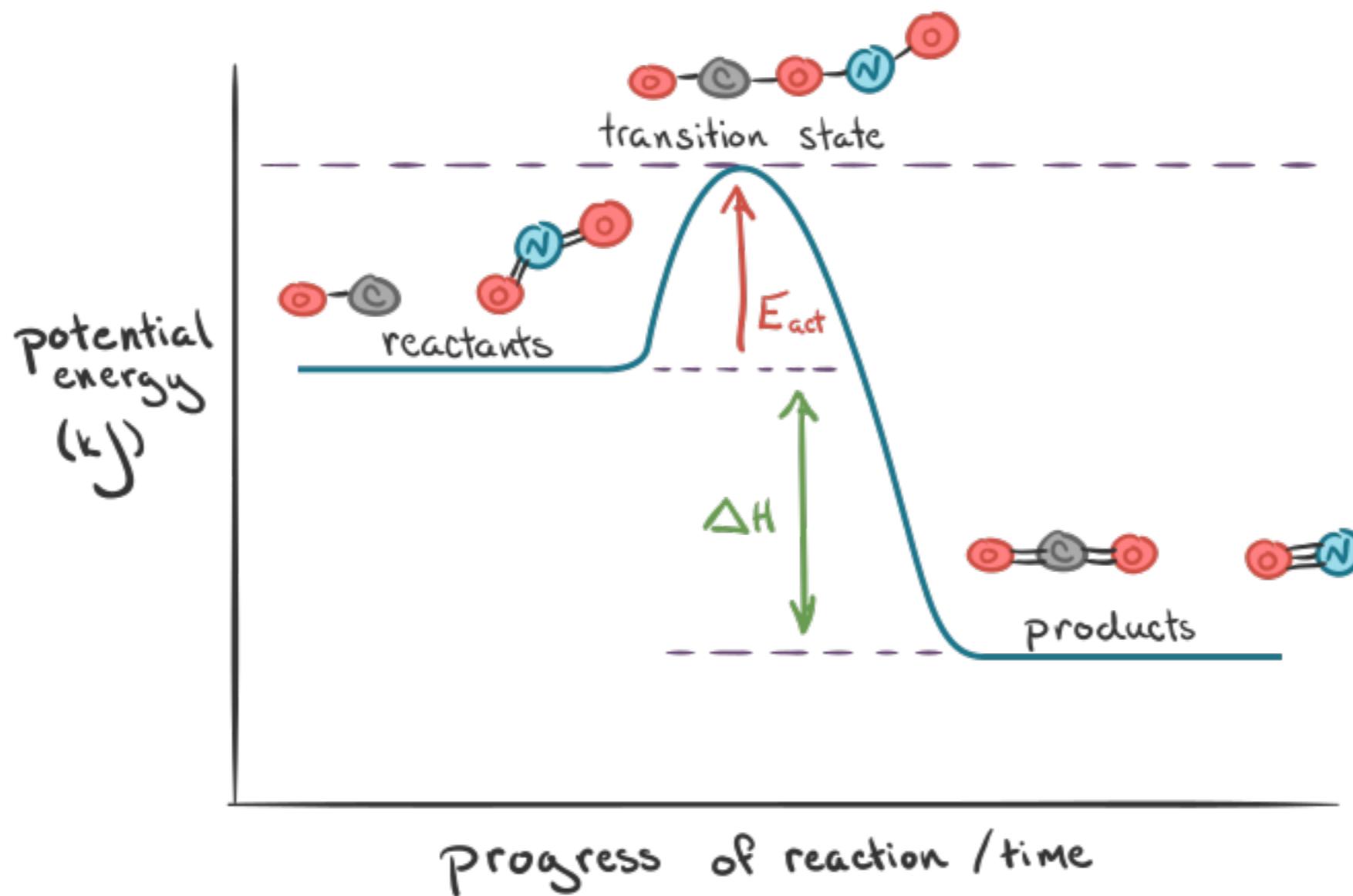


Figure from Khan Academy

<https://www.khanacademy.org/test-prep/mcat/chemical-processes/thermochemistry/a/endothermic-vs-exothermic-reactions>

Don't Avoid Activation Energy

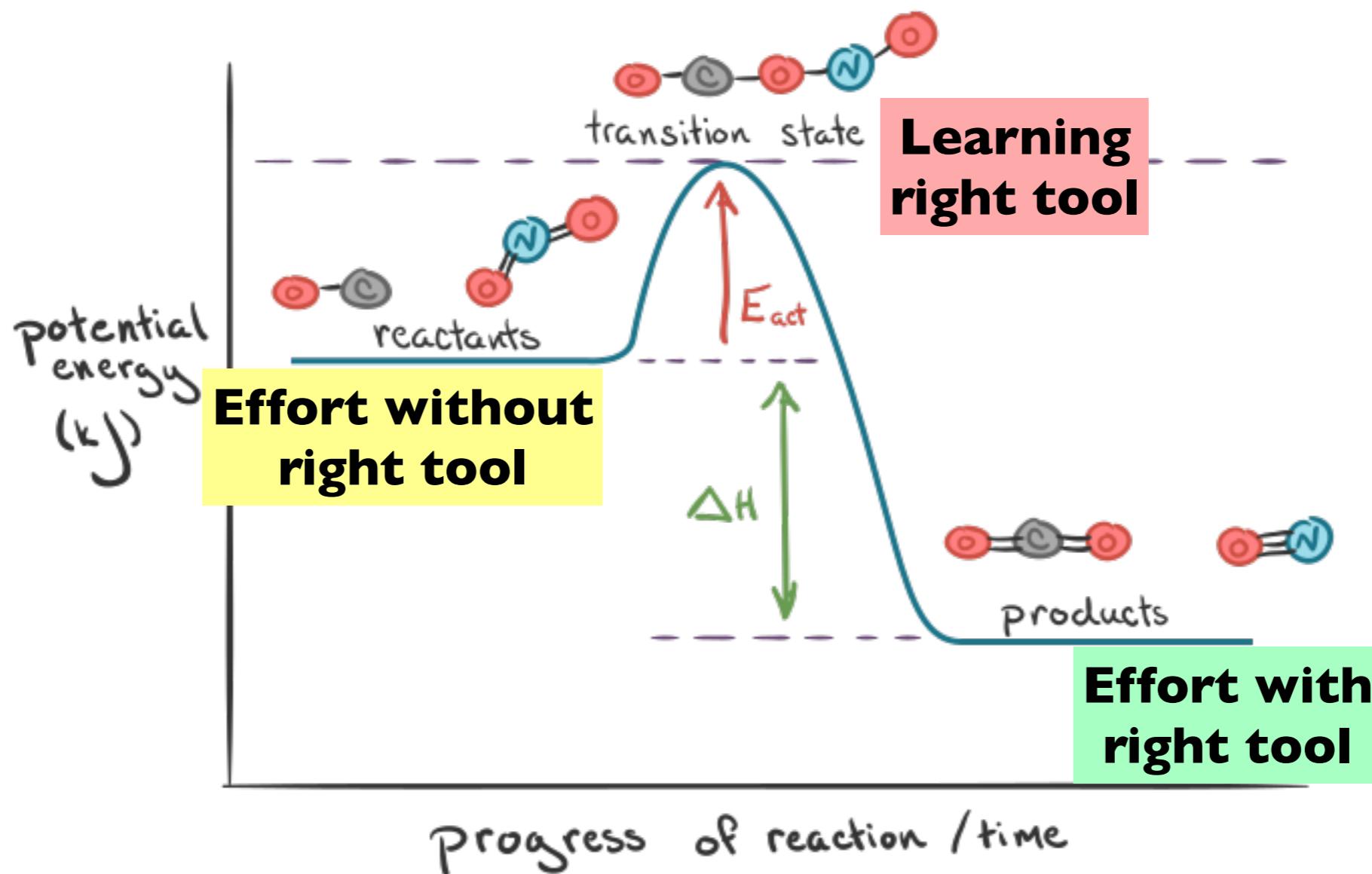


Figure from Khan Academy

<https://www.khanacademy.org/test-prep/mcat/chemical-processes/thermochemistry/a/endothermic-vs-exothermic-reactions>

Hyperbolic Discounting



<https://medium.com/behavior-design/hyperbolic-discounting-aefb7acec46e>

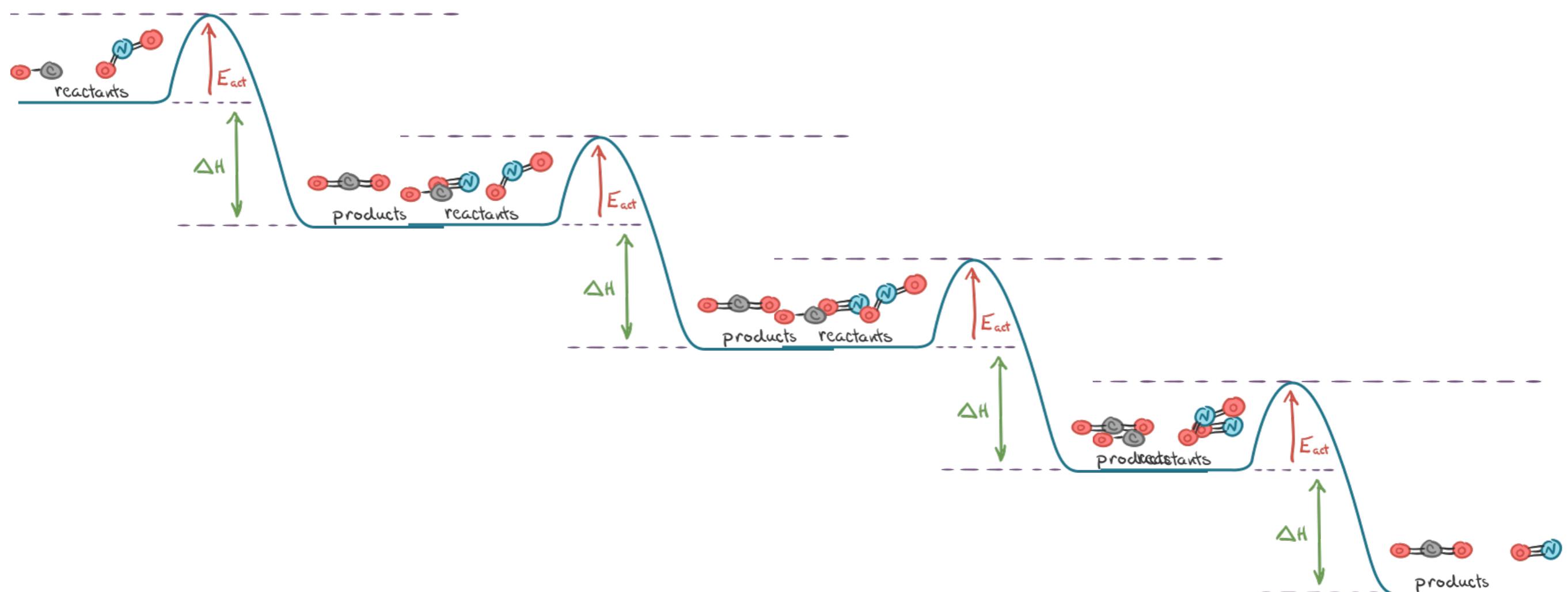
Hyperbolic Discounting



<https://medium.com/behavior-design/hyperbolic-discounting-aefb7acec46e>

A close-up photograph showing a person's hands working on a piece of wood. The person is using a chisel to shave off thin layers of wood, creating a pile of shavings on the workbench. The workbench is made of light-colored wood and shows signs of use. In the background, there are some tools and a bottle of water. The overall scene suggests a workshop or a woodworking environment.

Practice, Practice, Practice



It never ends... 1000x improvements possible!

Figure from Khan Academy

<https://www.khanacademy.org/test-prep/mcat/chemical-processes/thermochemistry/a/endothermic-vs-exothermic-reactions>

Debugging and Troubleshooting



Administration

Weekly Cadence

Each week has a focus topic

Pair of coordinated lectures on Fri and Mon

Mandatory lab on Wednesdays

Assignment handed out Wed evening (after lab),
YEAH hours on Thu, assignment due following Tue at
6 pm.

Laboratories

Attendance is **mandatory**

Do exercises and complete check-list

Leave lab ready for assignment: walks you through tricky bit (hardware/software interface) to get you started

Philosophy: lots-of-help, hands-on, collaborative

We will organize your lab into small (2-3 person) breakout groups so you can do the lab with mute off and chat/collaborate. Initially groups are randomly assigned, later in the quarter we will let you choose partners if you want to.

Assignments

7 assignments

- Build on each other

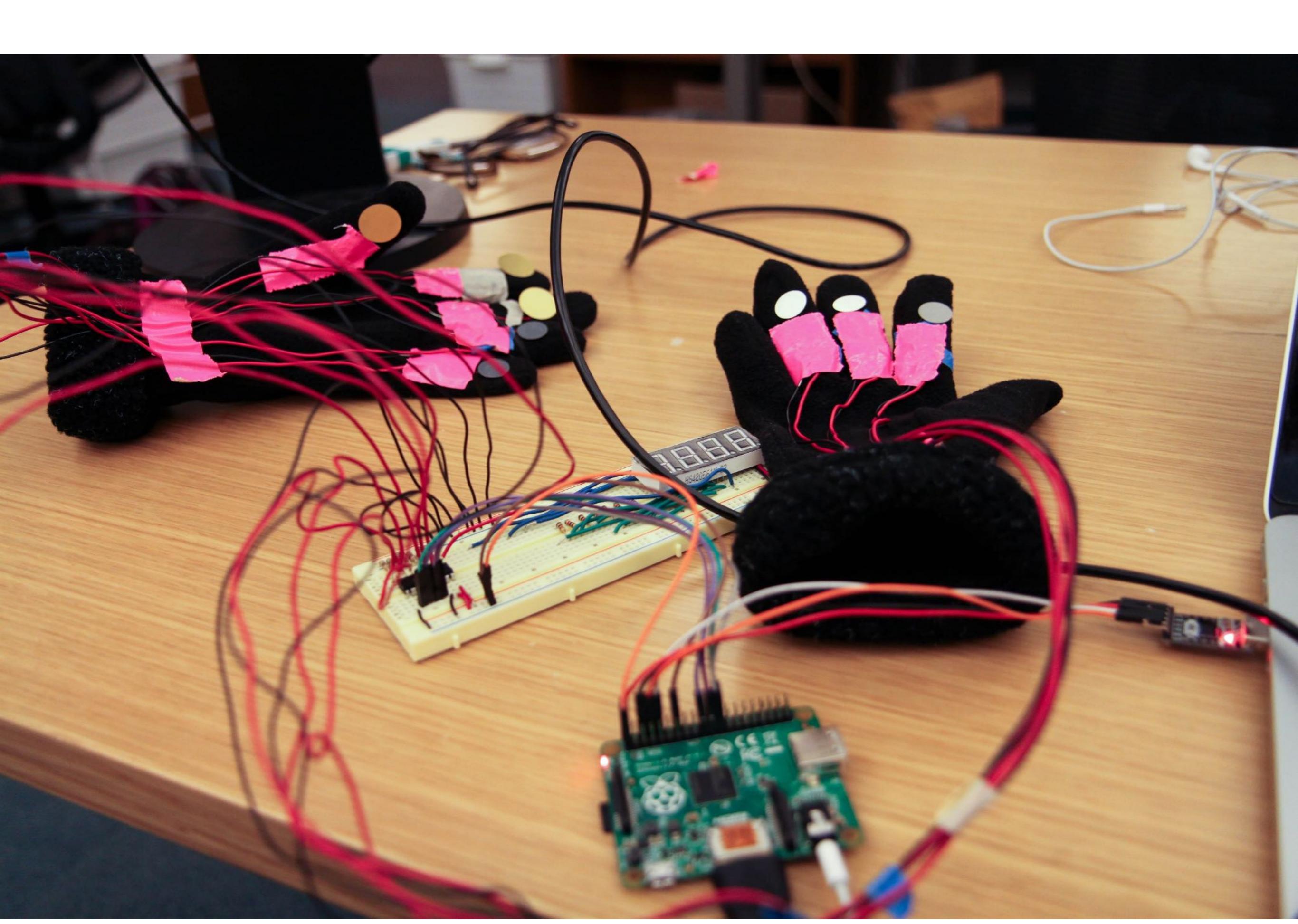
Two parts for each assignment

- Basic (required, tight spec, guided steps)
- Extension (optional, opportunity for your exploration/creativity)

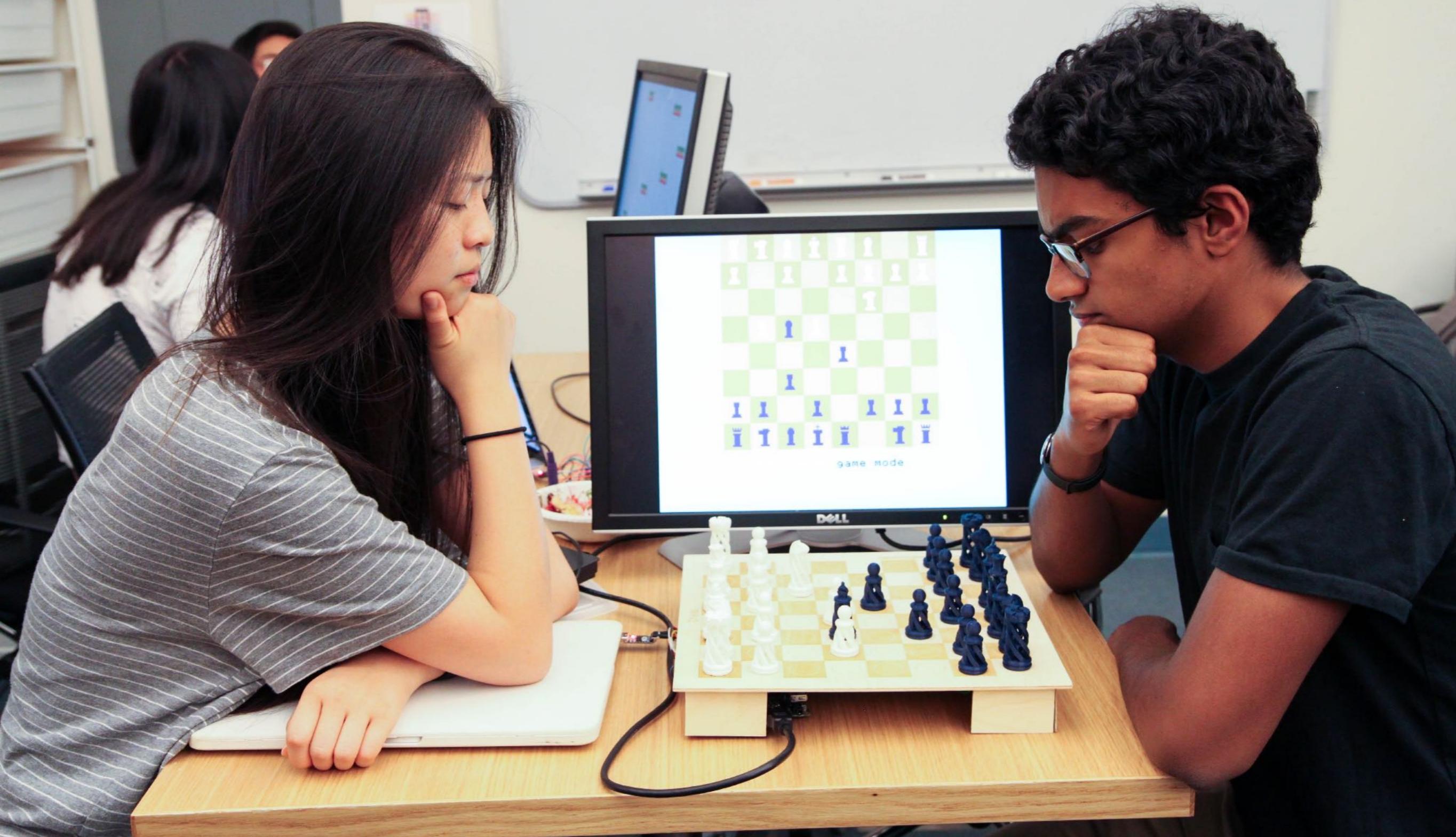
Final project demonstrations on Thursday, December 9

Scaled back due to lack of exam period, physical lab resources

- Encourage you to play with some hardware (e.g. sensors)

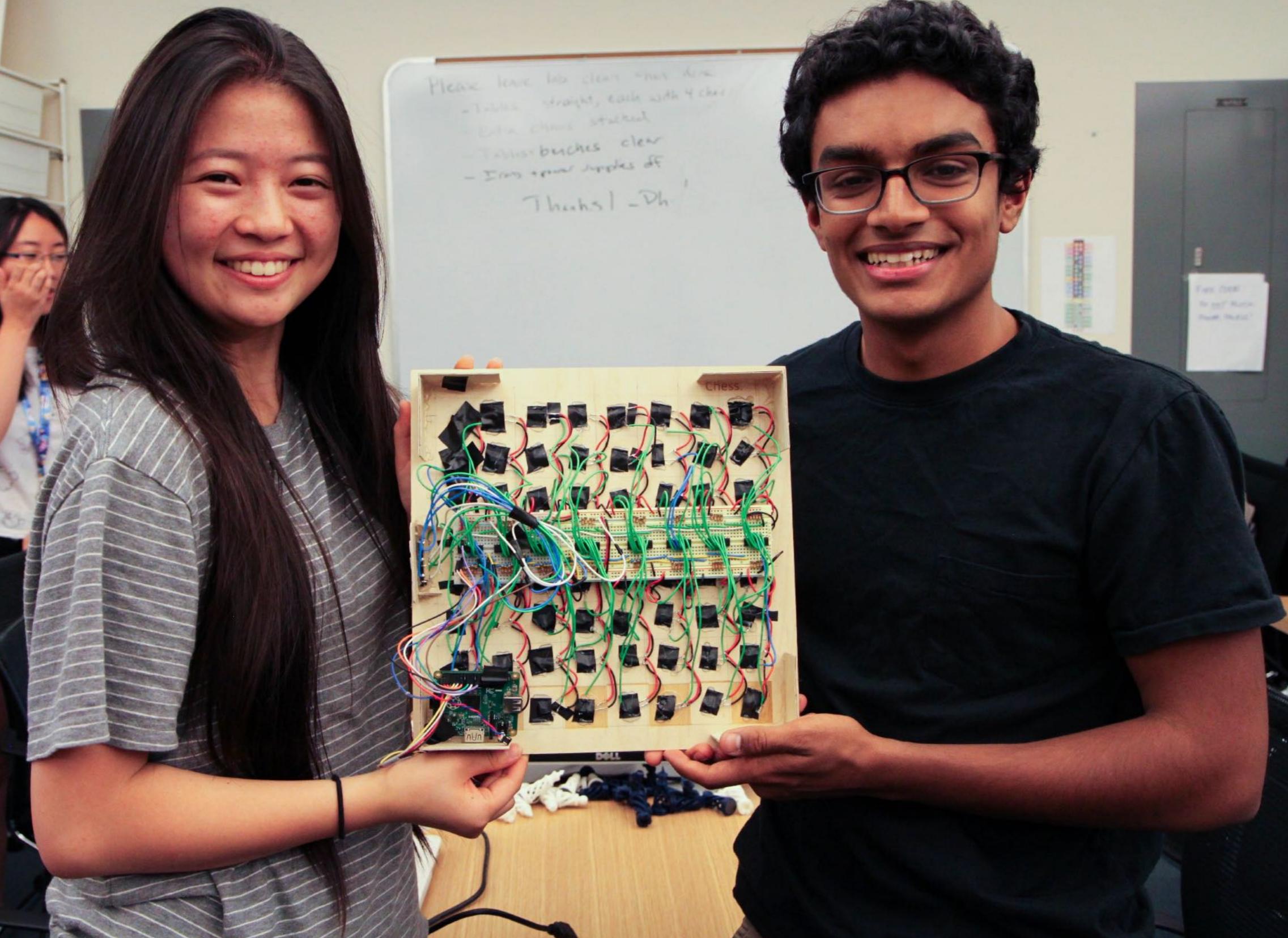
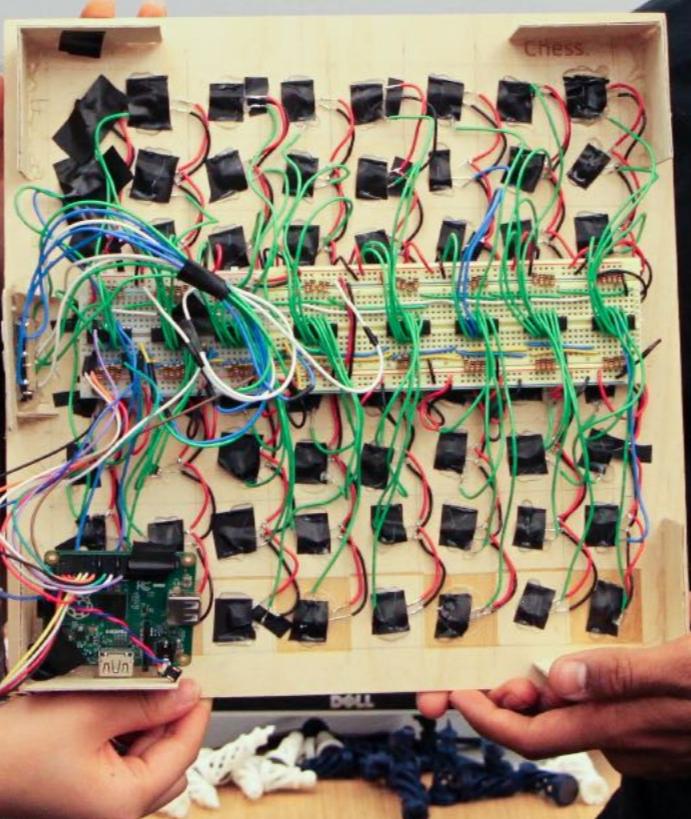


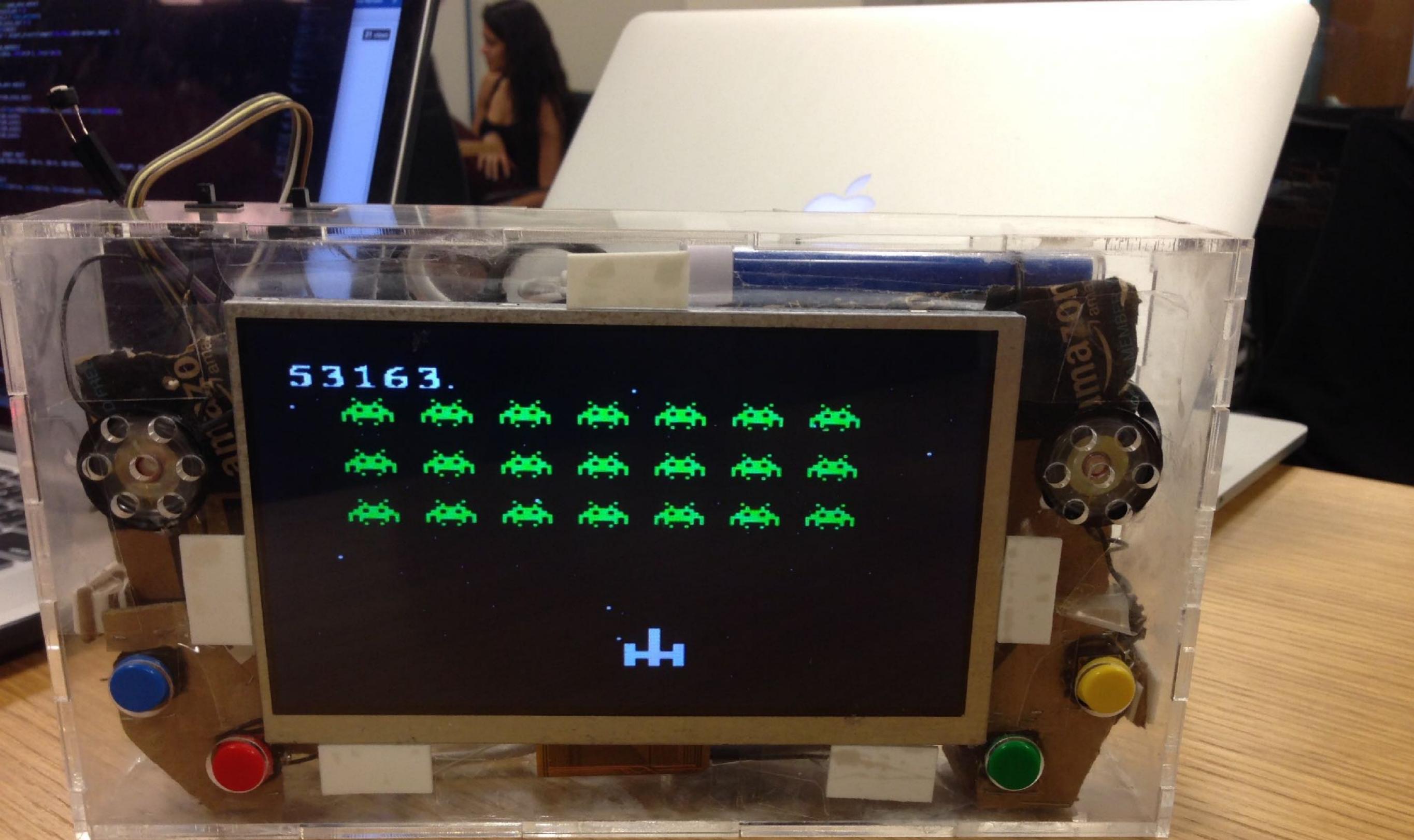




Please leave tea clean - no tea
- Tables straight, each with 4 chairs
- Extra chairs stacked
- Paper bushes clear
- Dirs equal widths of

Thanks! - Dh





First Week

Today

Fill out course application if you haven't already

Assignment 0

- Join forum [https://edstem.org/us/courses/12375/
discussion/](https://edstem.org/us/courses/12375/discussion/)
- Read and understand our guides on basic topics:
electricity, numbers, and UNIX
- Create github account and send us your GitHub id
- Install/setup your development environment
- Fill out lab time poll (will be posted on EdStem)

Class Meeting Protocol I

Class attendance is expected. The slides are not intended to be a replacement for being in class. There are a lot of conceptually difficult topics, and having you there to ask us to stop and clarify is important.

In our experience teaching this class, students who stop regularly attending lectures struggle greatly and spend the last two weeks trying to earn more points back by fixing bugs and resubmitting.

Class Meeting Protocol 2

However, we acknowledge that this is a crazy time! We have recordings of lectures from during remote learning. If you have to miss a lecture due to self-isolation, safety concerns, or an emergency, email -staff and we will send you a link to the video with no questions asked. The videos might differ slightly from the current lectures, but should be pretty close (except for Sound).

Dealing with COVID

The screenshot shows a news article from The Washington Post. At the top, there are navigation links for 'Search' and 'Sections' with a menu icon. The main header reads 'The Washington Post' and 'Democracy Dies in Darkness'. On the right, the author's name 'Philip Levis' is displayed with a small profile icon. Below the header, the main title of the article is 'Parenting a child under 12 in the age of delta: ‘It’s like a fire alarm every day’'. A subtitle below the title reads 'Routine outings become tricky decisions for the youngest Americans, who are still ineligible for coronavirus vaccines'. The word 'Screenshot' is visible at the bottom left of the image.

Both Chris and I have small kids (not vaccinated yet)

We are delighted to be teaching in person but need to be careful

Stanford requires masks are worn inside at all times

- Please wear them well and carefully! Even if you are alone in lab.

What is a computer?

You've taking CSI06A / 106B, and possibly some hardware-esque classes (e.g., ENGR 40M, EE 101A/B, EE108, etc.), but you probably haven't taken a course where you discussed what a computer really is.

Almost all computers you use today (including the Raspberry Pi, your phone, and your laptop/desktop, but not FPGAs) are built using the *von Neumann architecture*, which was laid out by John von Neumann et al. in a 1945 paper on the EDVAC computer.

The von Neumann architecture

The original architecture designed by von Newmann has the following features (see Wikipedia)

- A processing unit that contains an arithmetic logic unit and processor registers
- A control unit that contains an instruction register and program counter
- Memory that stores data and instructions
- External mass storage
- Input and output mechanisms

A critical part is the idea that instructions and data are both kept in the same memory

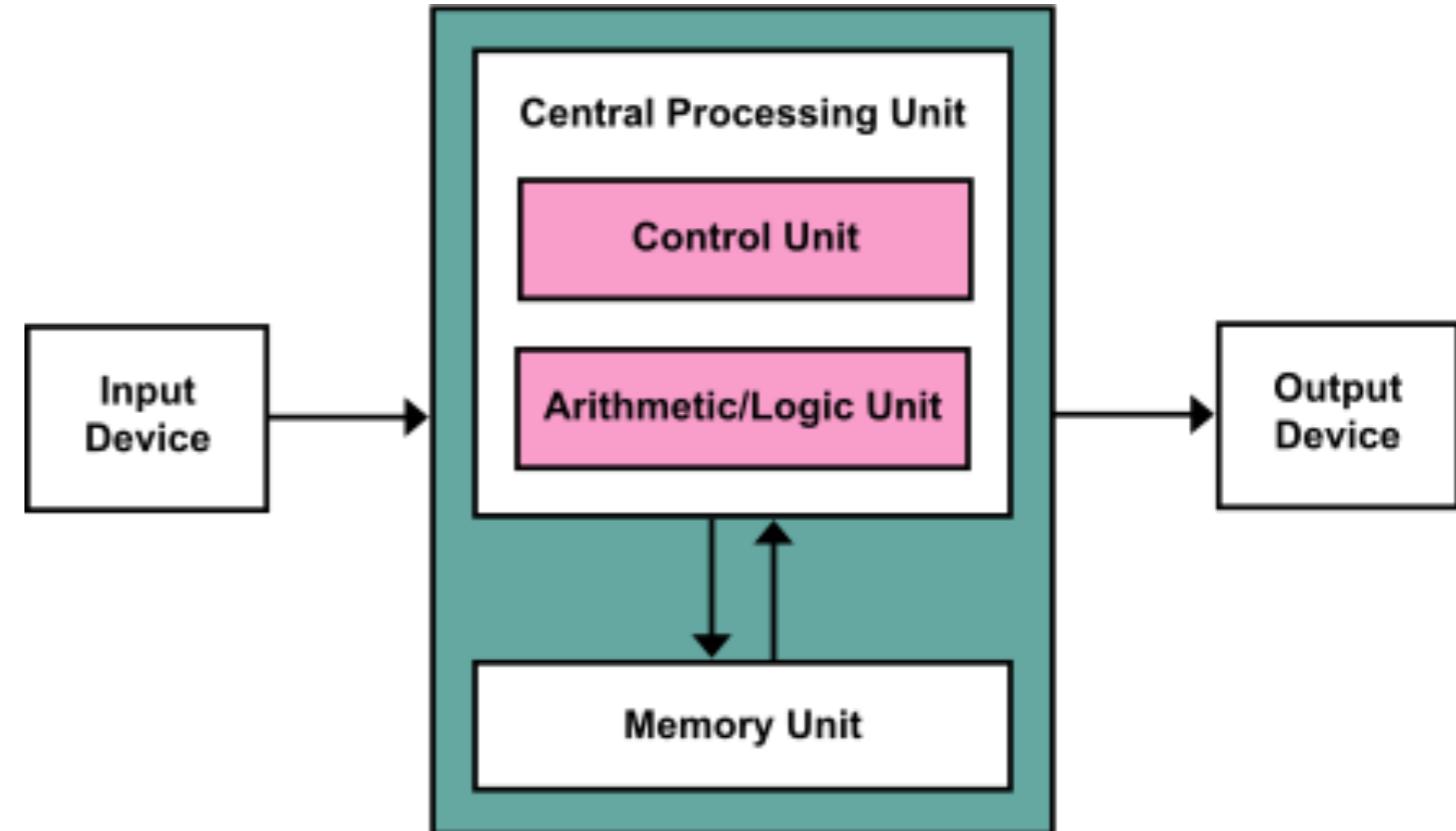
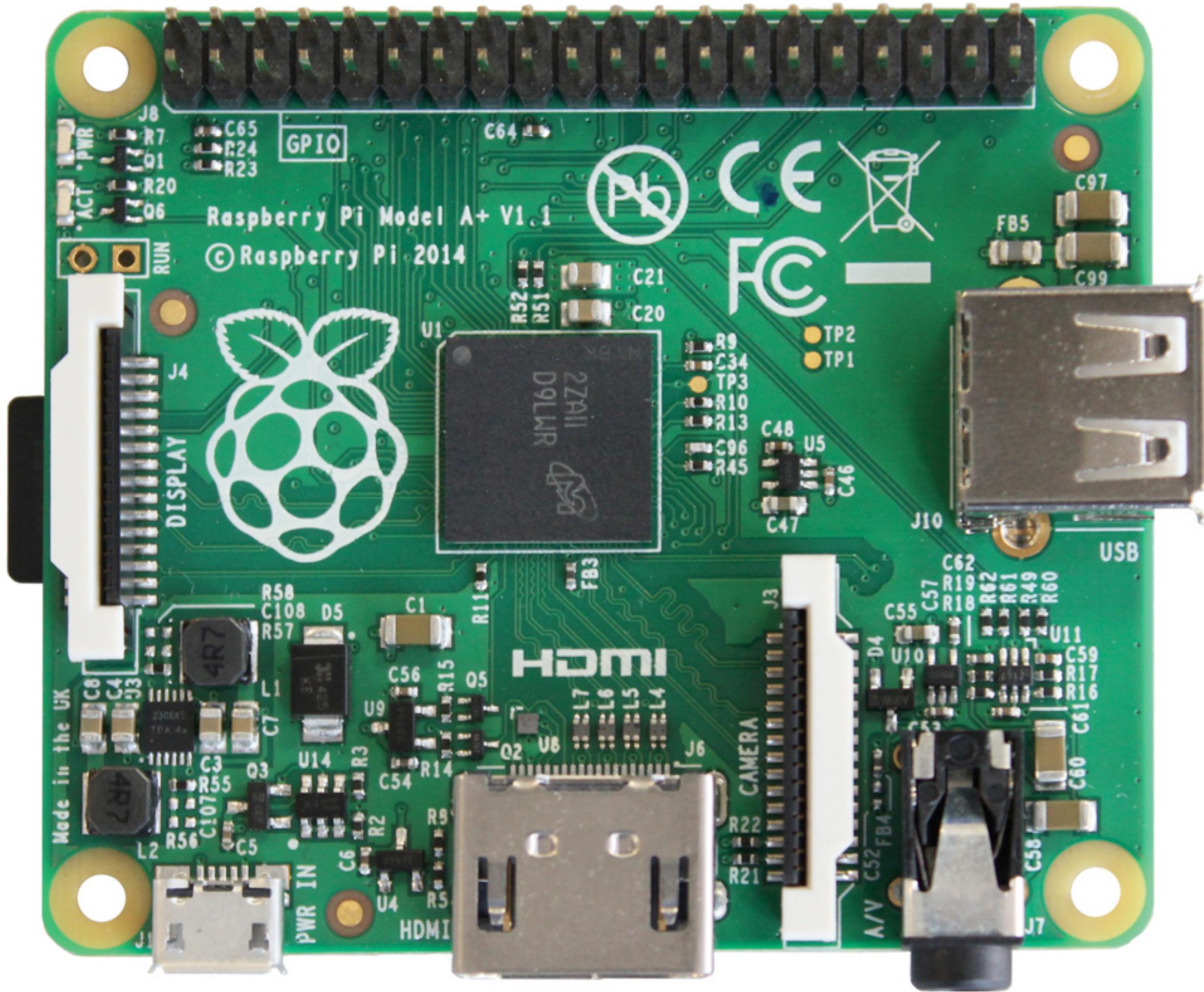


Image: https://en.wikipedia.org/wiki/Von_Neumann_architecture#/media/File:Von_Neumann_Architecture.svg

How is our Raspberry Pi a von Newmann architecture machine?

- A processing unit that contains an arithmetic logic unit and processor registers
 - The Raspberry Pi has an ARM processor with processor registers
- A control unit that contains an instruction register and program counter
 - The ARM processor has an instruction register (kind of — it is not something we can get access to) and a program counter ("r15")
- Memory that stores data and instructions
 - The Raspberry Pi A+ has 512MB of RAM
- External mass storage
 - The Raspberry Pi A+ has an SD card reader
- Input and output mechanisms
 - The Rasperry PI A+ has USB (we won't use it), HDMI (we will use it), and GPIO pins (you better believe we will use them)



ARMKCE MC1
V-OF3
1439 1-6

PP10
PP13



MICRO SD CARD

090604
JW406AAC

J9

C66

C2 R1 PP8 PP4

PP9

PP7 PP1

PP2

TRST_N
T01 T00 TMS TCK GND

J5

PP32 PP29 PP34

PP33 PP31

R12 C40 C17
C95 FB2
C94 C36
C51 C18 C69
C14 C12 C37
R25
C50 C9 C13 C35
C45 C29 C30

PP21 C23 C26
R31 C26 C42 C39
C19 C41 C63
C43 C67 C31
C16 C33 C22

C44 C16
C25

C27

C10

X1

C11

C38

C24

C19

PP15

PP18

PP14

PP16

PP17

PP5

PP20

PP39

PP40

PP38

PP37

PP30

PP32

PP29

PP34

PP33

PP31

PP22

PP23

PP26

PP25

PP24

PP40

PP39

PP38

PP37

PP30

PP32

PP29

PP34

PP33

PP31

PP35

PP27

PP12

PP25

PP24

PP6

PP26

PP25

PP24

PP40

PP39

PP38

PP37

PP30

PP32

PP29

PP34

PP33

PP31

PP1

PP2

PP3

PP4

PP5

PP6

PP7

PP8

PP9

PP10

PP11

PP12

PP13

PP14

PP15

PP16

PP17

PP18

PP19

PP20

PP21

PP22

PP23

PP24

PP25

PP26

PP27

PP28

PP29

PP30

PP31

PP32

PP33

PP34

PP35

PP36

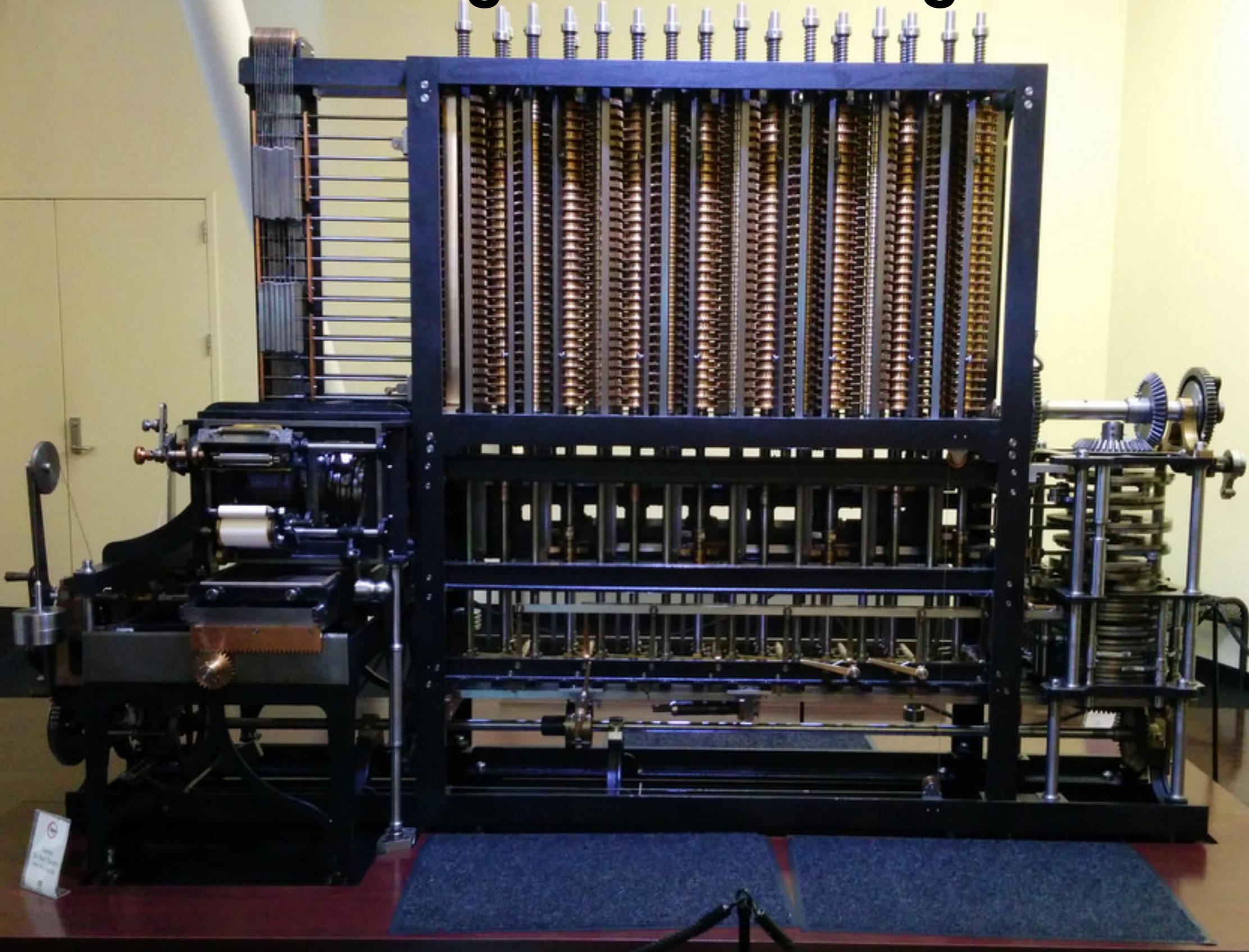
PP37

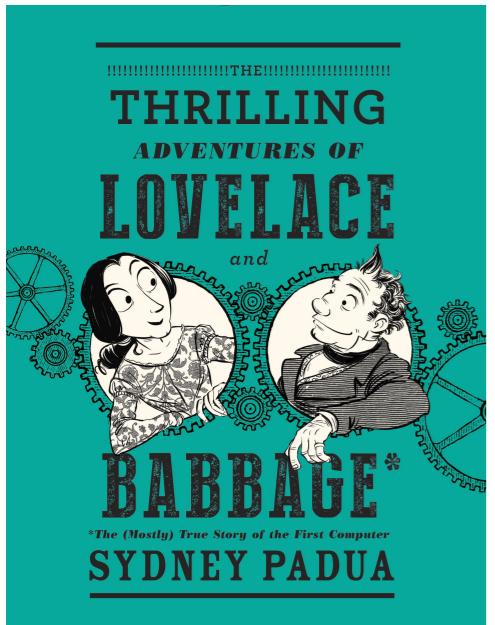
PP38

PP39

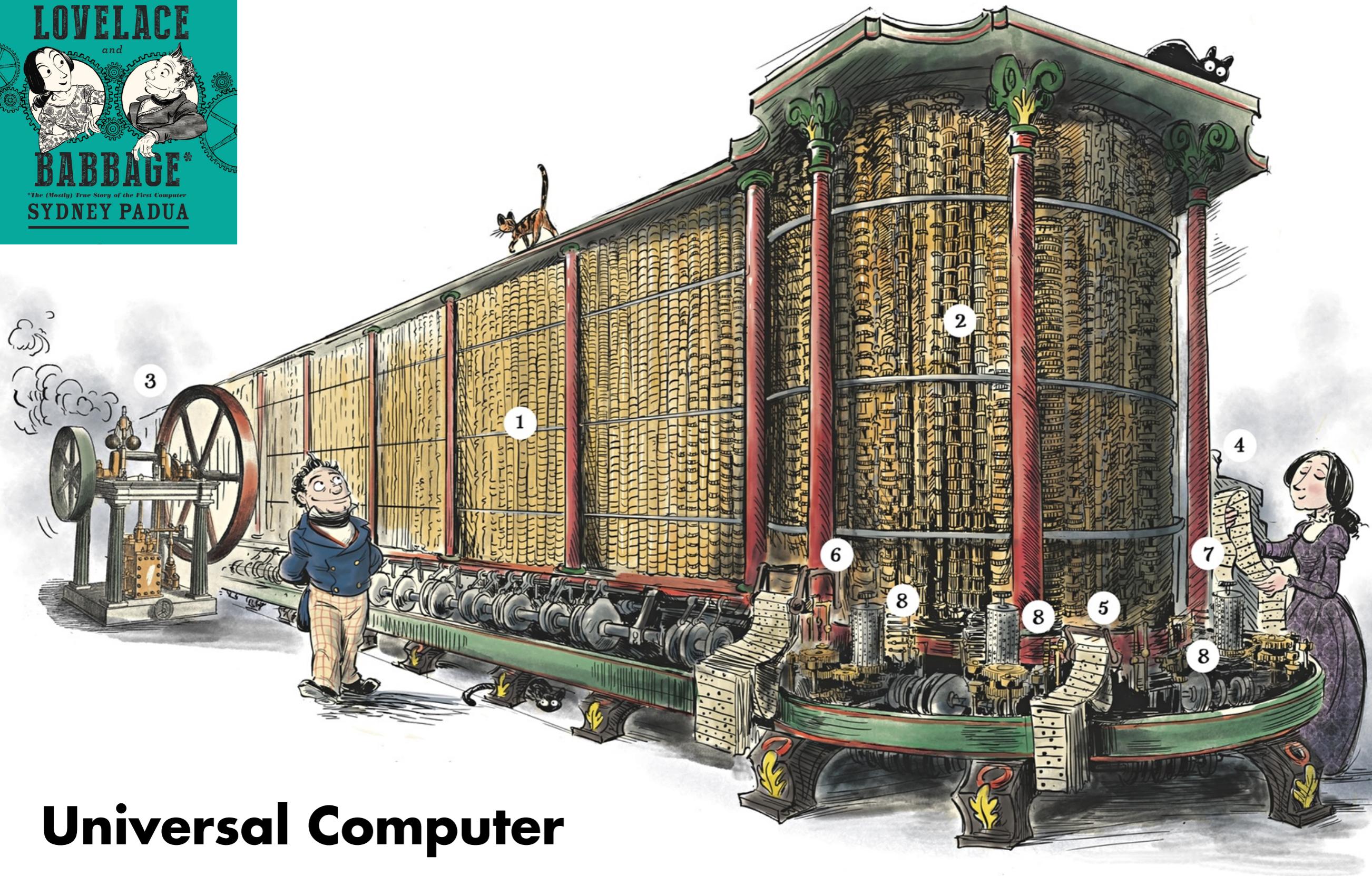
PP40

Babbage Difference Engine



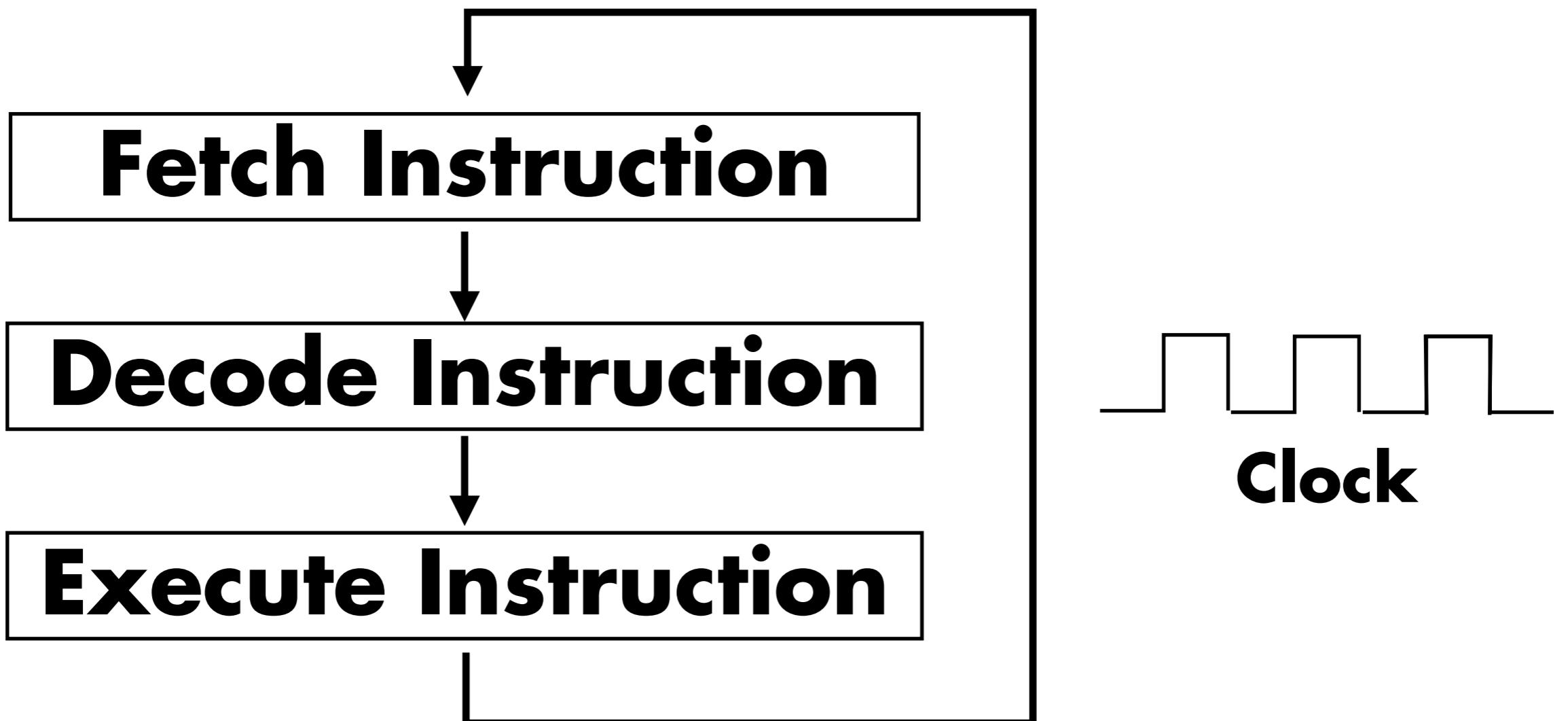


Analytical Engine



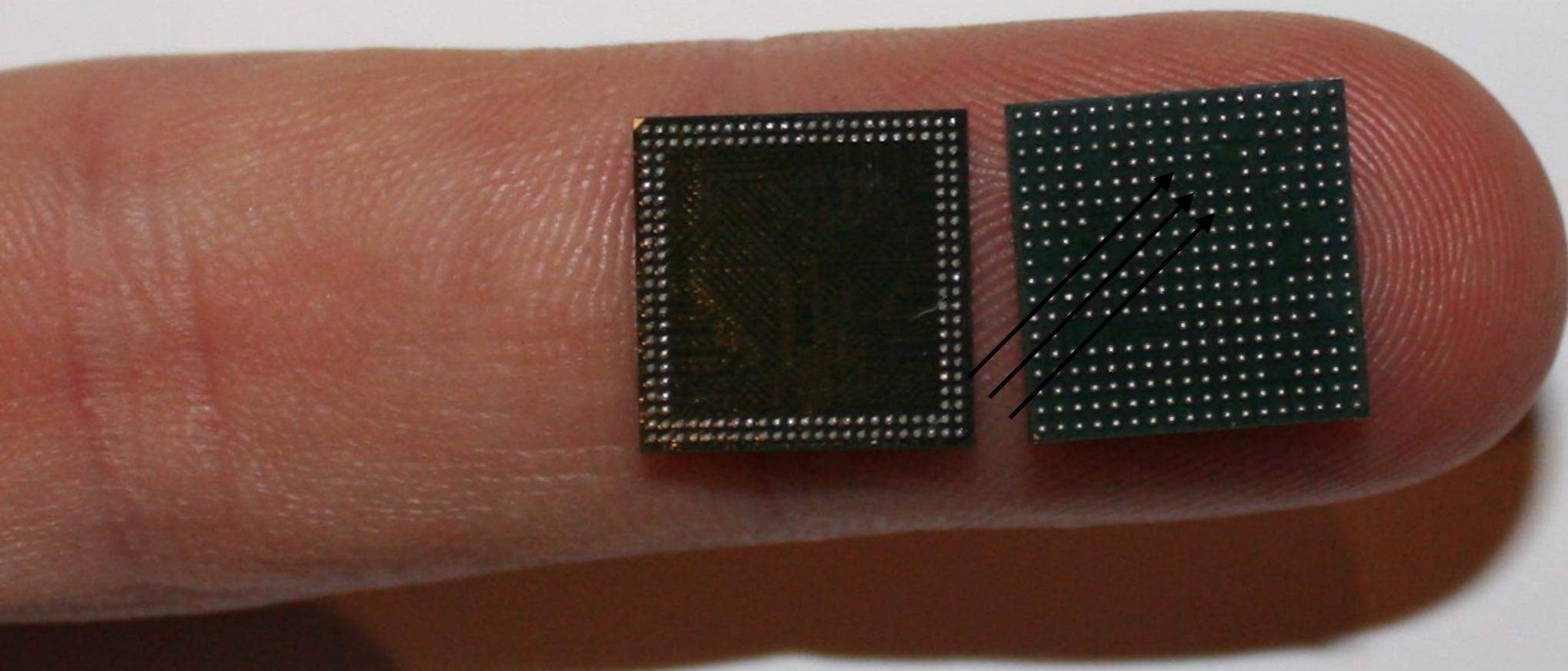
Universal Computer

Running a "Program"



Package on Package

Broadcom 2865 ARM Processor



Samsung 4Gb SDRAM

10000000_{16}

Memory used to store both instructions and data

Storage locations are accessed using 32-bit addresses

Maximum addressable memory is 4 GB

Address refers to a byte (8-bits)

Memory Map

00000000_{16}

10000000₁₆

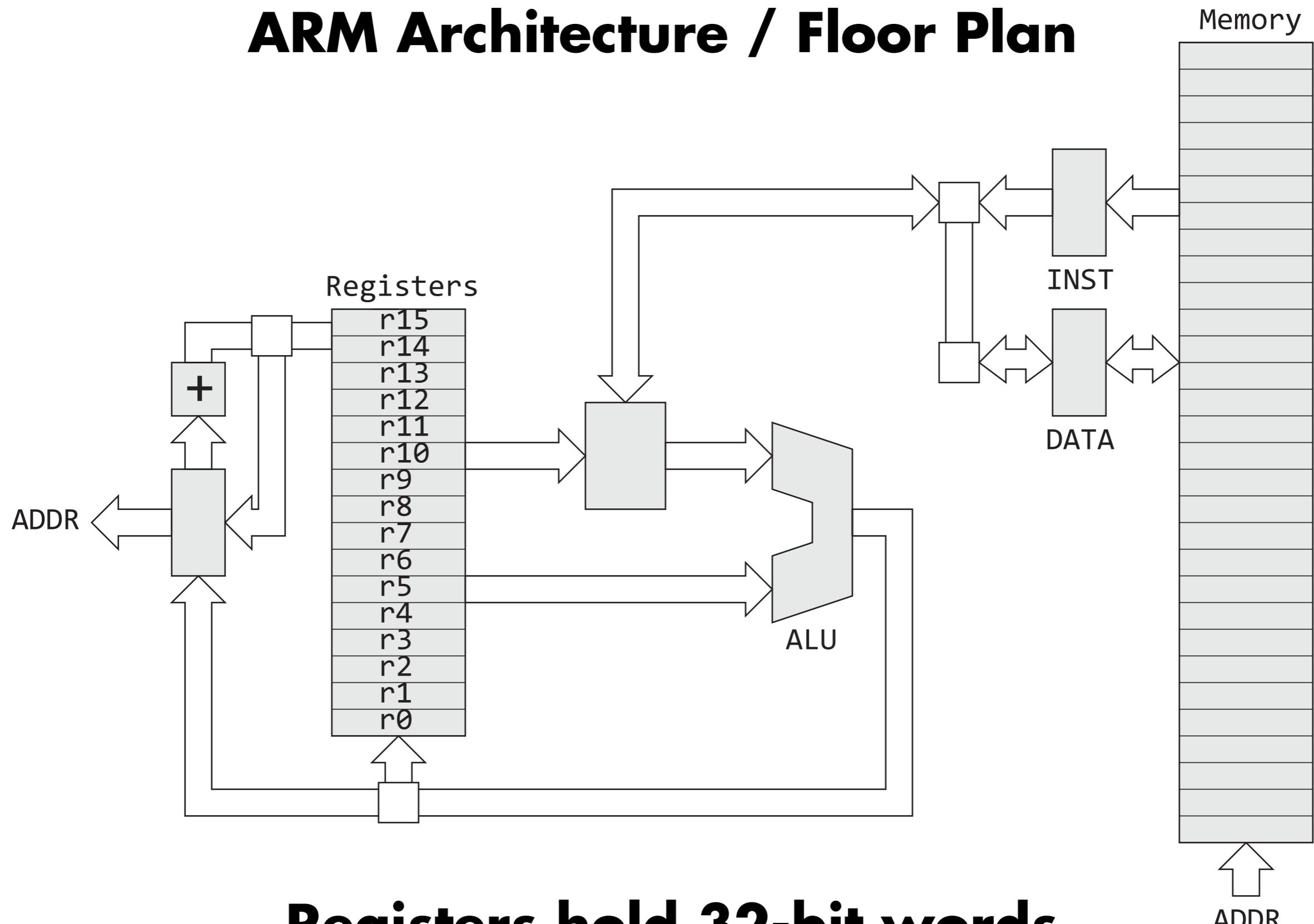
Memory Map

02000000₁₆

512 MB Actual Memory ➔



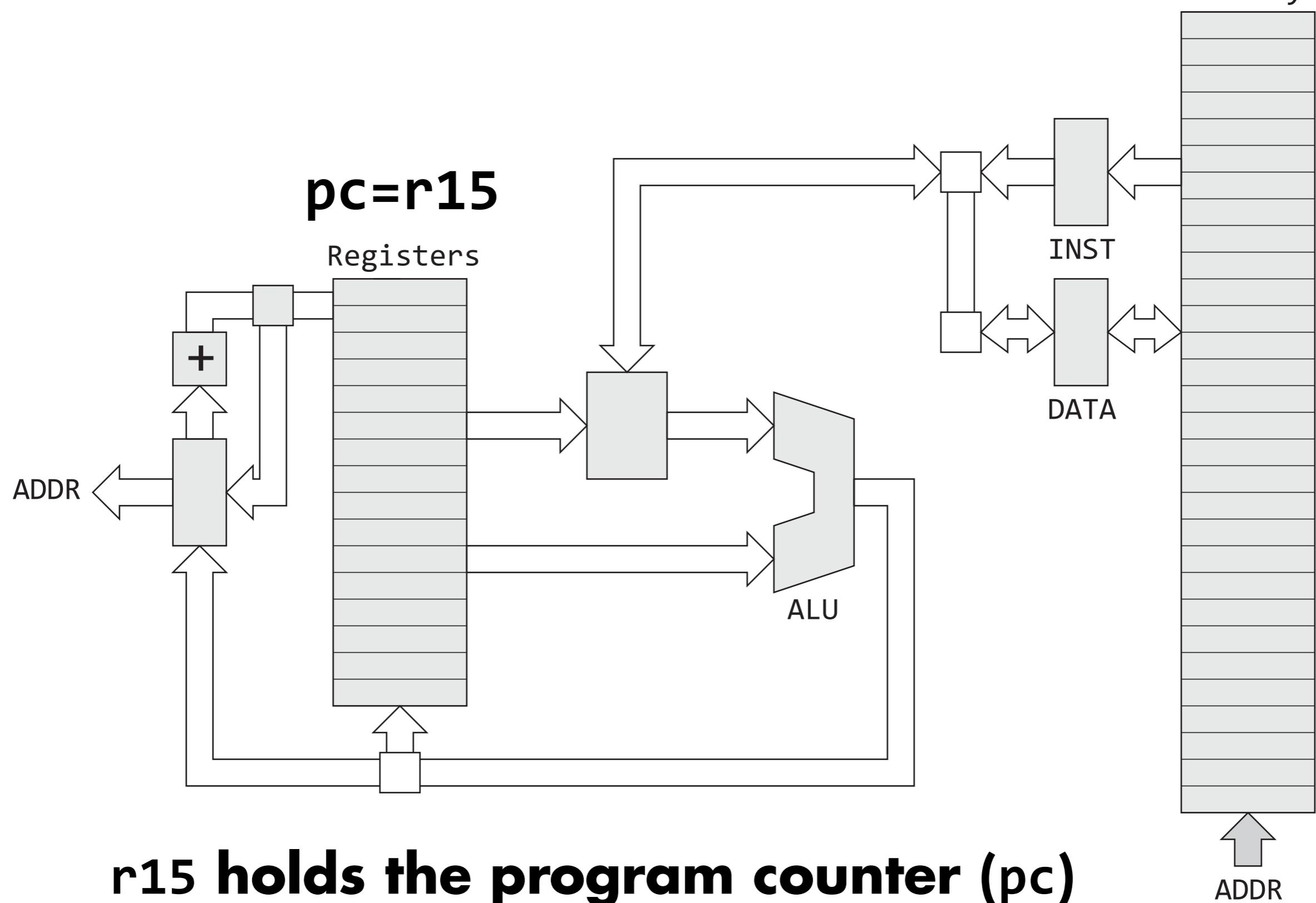
ARM Architecture / Floor Plan



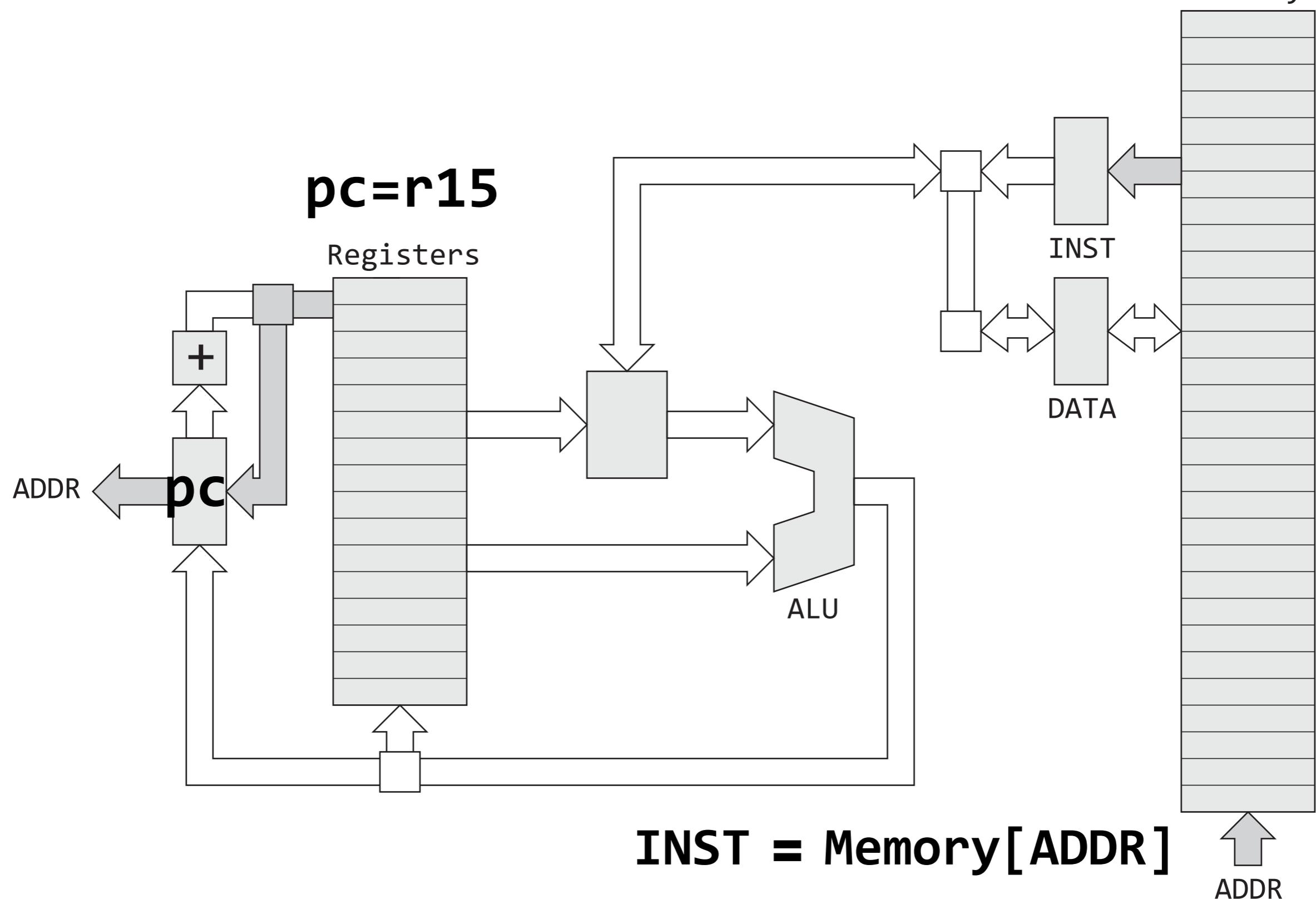
Registers hold 32-bit words

Arithmetic-Logic Unit (ALU) operates on 32-bit words

Instruction Fetch

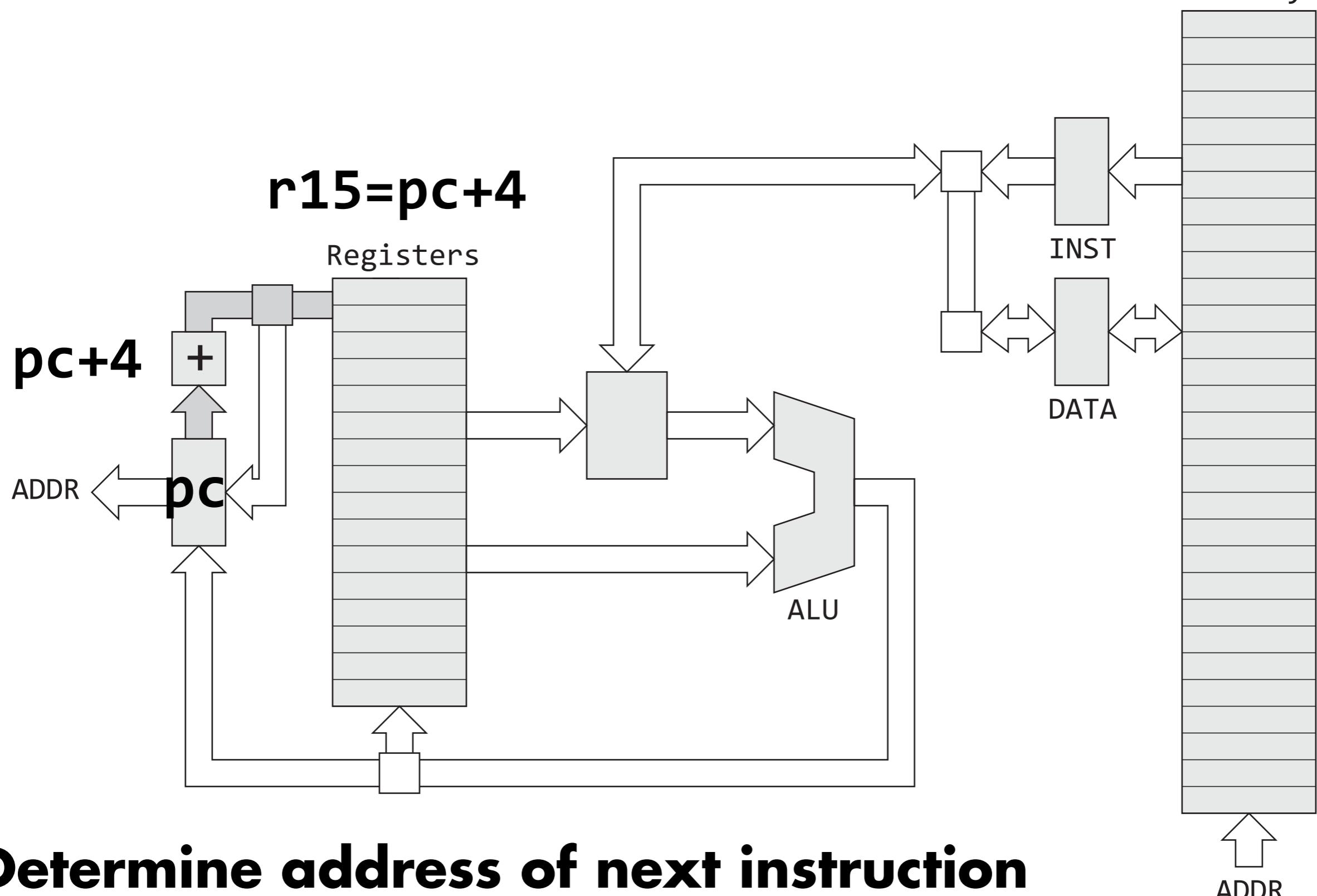


Instruction Fetch



Addresses and instructions are 32-bit words

Instruction Fetch



Determine address of next instruction

Why pc+4?