Assembly Language

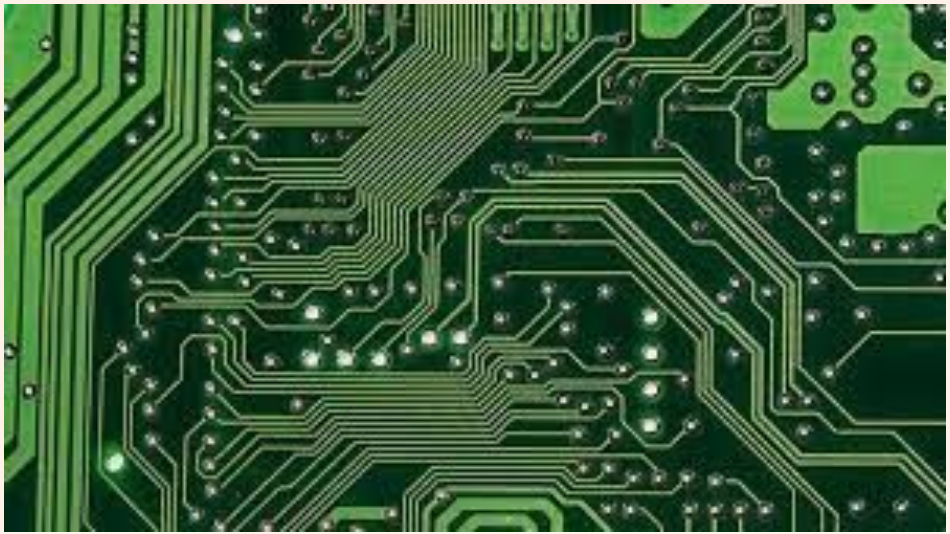
Marieke Thomas, Ben Eckley, Dave Ciolino-Volano, Ed Hawkins, Taylor Grant-Knight, Amanda Lee

slidesmania.com

What is Assembly Language?

Assembly language is a low-level computer language that communicates directly with a computer's hardware

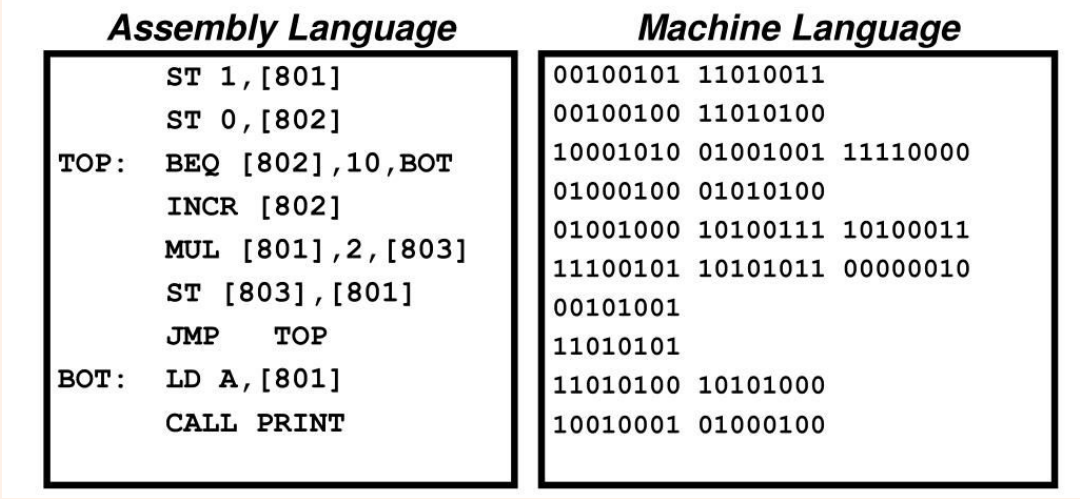
One line of assembly = one action by computer's hardware



slidesmania.com

What is Assembly Language?

This means that assembly language can be converted directly into the ones and zeros that flow through a computer's circuitry (called machine language)



slidesmania.com

Assembly is a low-level language

A low-level language is a programming language that provides little or no abstraction of programming concepts.

On the next slide, take a minute to compare the code to print “Hello, world!” in Python, a high-level language, and NASM Assembly, a low-level language. Which language would you rather program in?

slidesmania.com

Hello World! In Python

```python
print("Hello, World!")
```

Which language would you rather program in?

Hello World! In NASM Assembly

```
section .text
    global _start       ;must be declared for using gcc
_start:                 ;tell linker entry point
    mov edx, len    ;message length
    mov ecx, msg    ;message to write
    mov ebx, 1      ;file descriptor (stdout)
    mov eax, 4      ;system call number (sys_write)
    int 0x80        ;call kernel
    mov eax, 1      ;system call number (sys_exit)
    int 0x80        ;call kernel

section .data

msg db  'Hello, world!',0xa ;our string
len equ $ - msg         ;length of our string
```

slidesmania.com

High-Level Language

Ex. Python

Easier to code

Easier to read

Slower

Less efficient

Better for humans, worse for computers

Low-Level Language

Ex. 64-bit assembly, for Linux

Harder to code

Harder to read

Faster

More efficient– direct control over computer's memory + circuitry

Better for computers, worse for humans

slidesmania.com

Who uses Assembly?

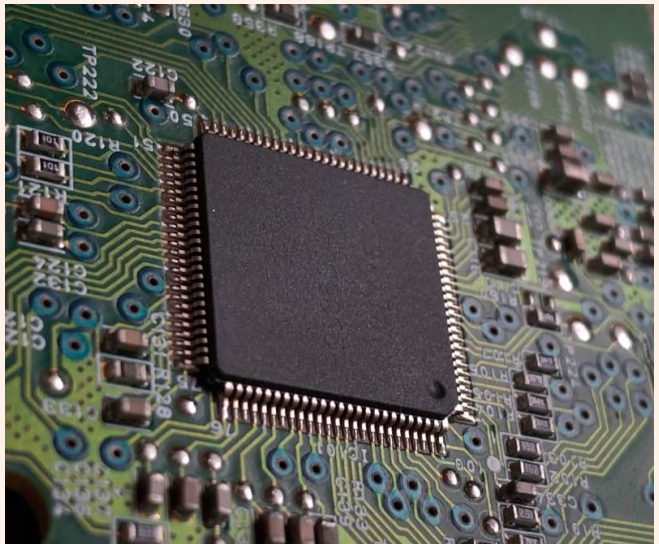
Assembly language is still used in specialized applications where performance demands take precedent (ex. a device with very low RAM) or low-level hardware access is required. Some examples of this are found in:

- Embedded Systems – cars, medical and industrial devices. These systems call for low-level hardware access.
- Device Drivers - Printers, graphic and sound cards.
- Video Game Engines
- Operating System Development

slidesmania.com

Assembly is hardware-specific

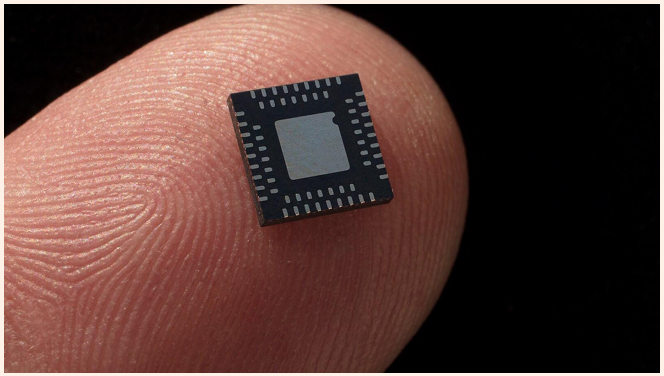
Since assembly code corresponds directly to the actions taken by the computer's physical circuitry, that means that different families of chips require different assembly code. There isn't one "assembly language" that can be run on every computer. We call this type of code **non-portable**.



slidesmania.com

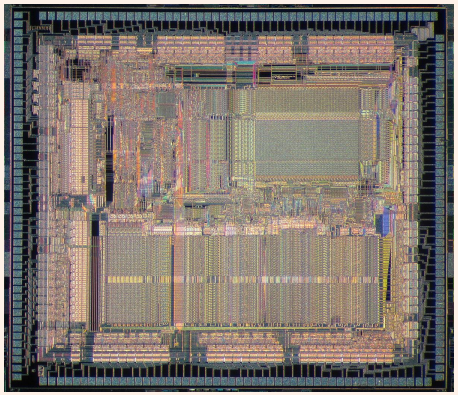
Assembly is hardware-specific

ARM



- Used in Smart Phones / Tablets
- Low power consumption and low heat
- Not as fast as x86 for computations

MIPS



- Used in game systems and Tesla's auto-driving features for its 3D rendering capabilities

x86

- Most commonly used in PCs
- Efficient and speedy mathematical computations

slidesmania.com

Since assembly is hardware-specific, it’s useful to have a basic understanding of how a computer’s hardware is put together (computer architecture)

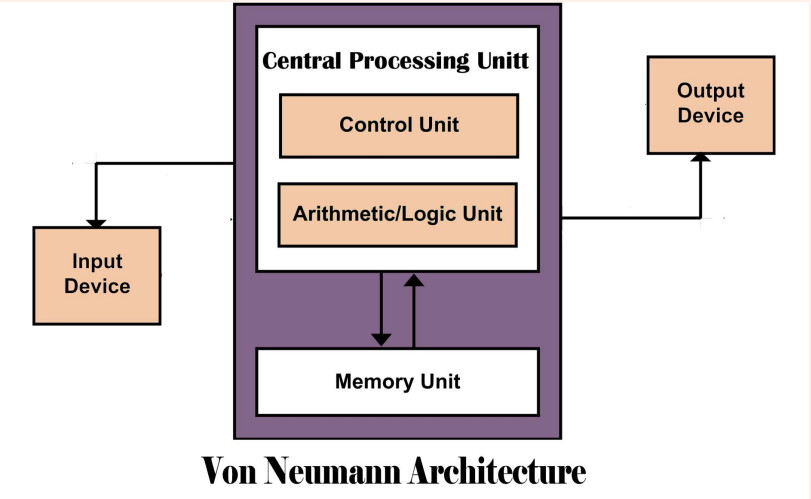
slidesmania.com

Computer Architecture: How is a computer put together?

Von Neumann Architecture, from a paper written in 1945 by John von Neumann, is often thought of as being the first model of what are contemporary processing units evolved from.

Control Unit: Tells the computer the order of the program and what operations to perform

Logic Unit: Does the computations and returns values to be tested by control unit.

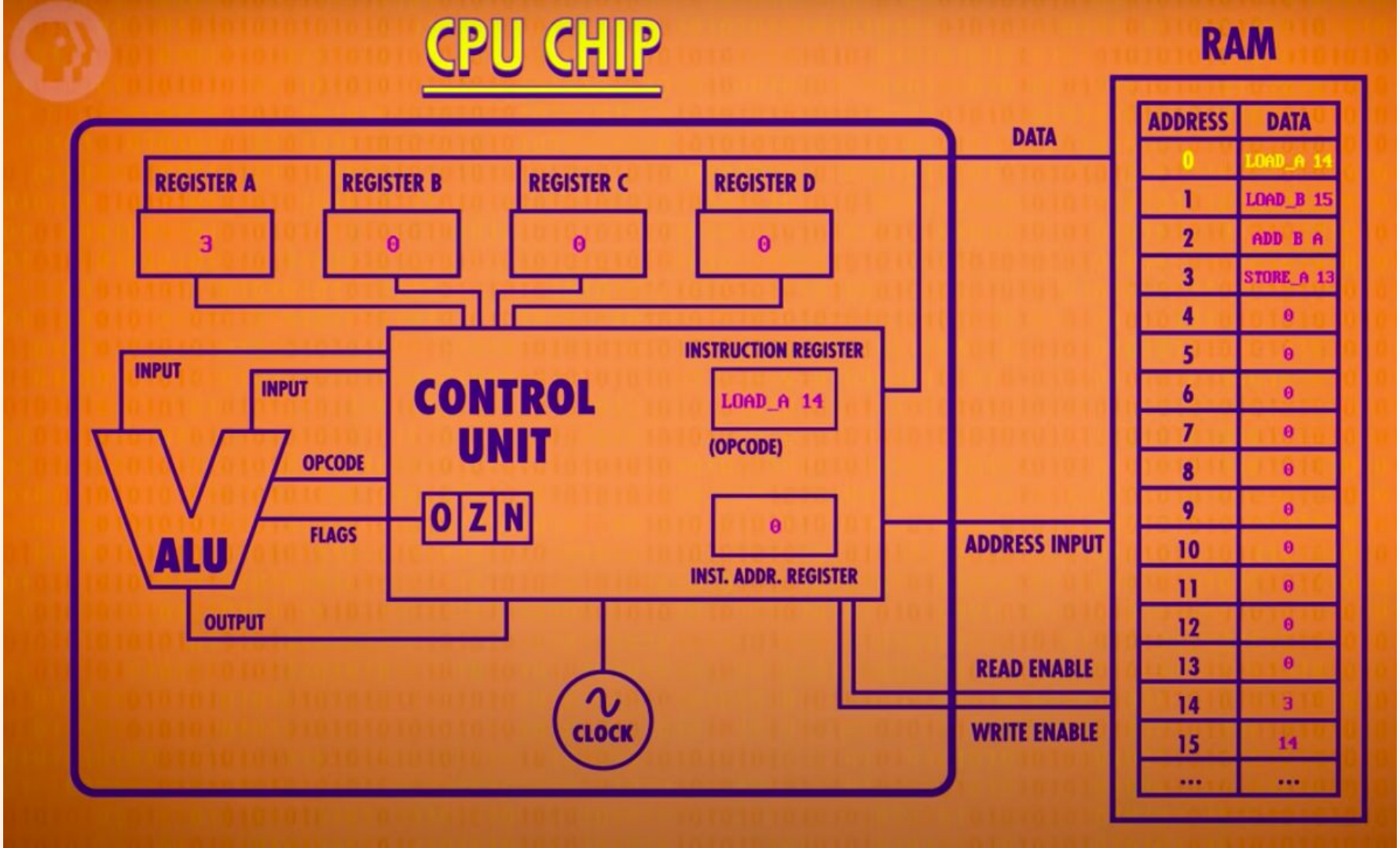


Von Neumann Architecture

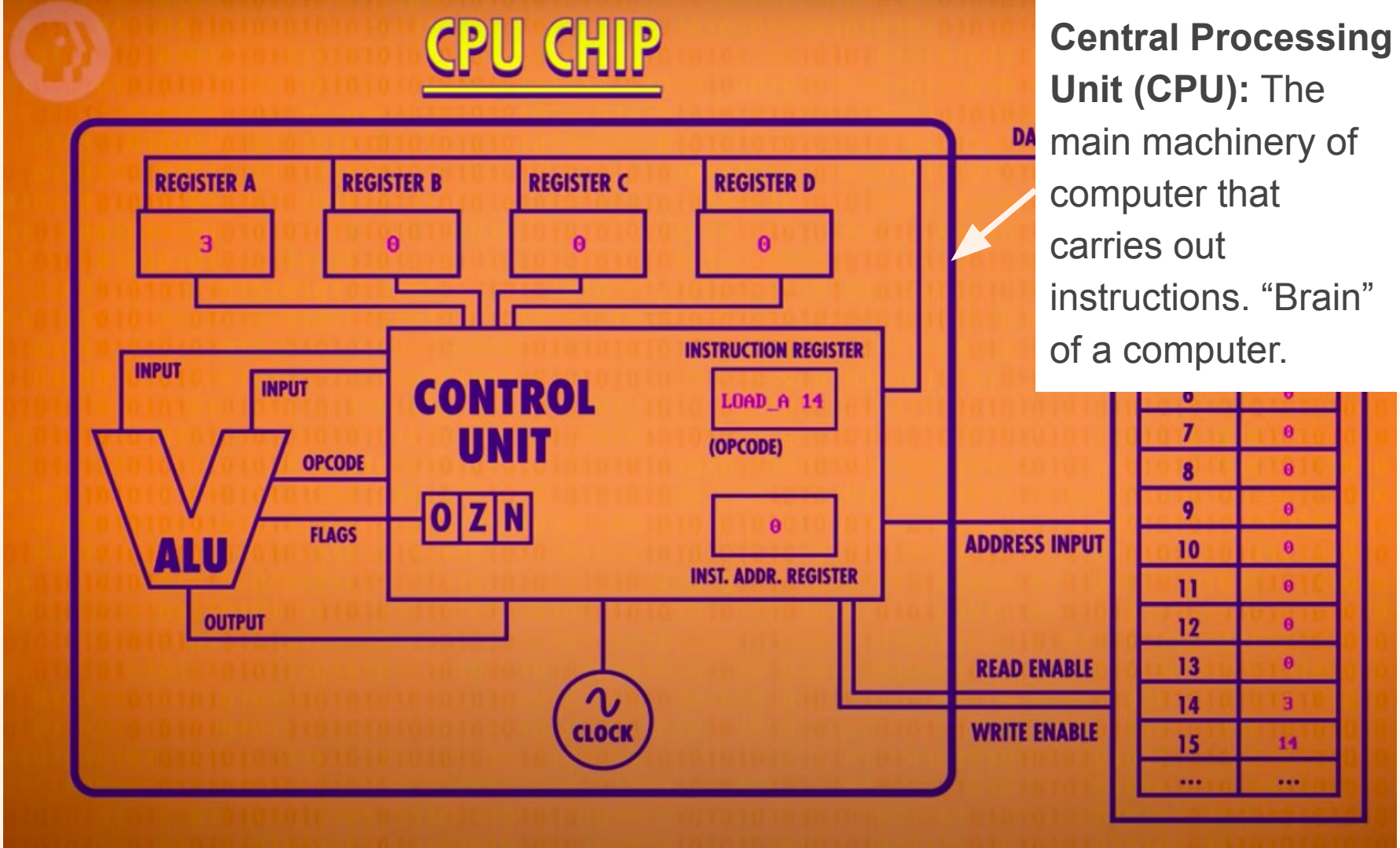
slidesmania.com

Let’s look at an example of simple computer architecture

slidesmania.com

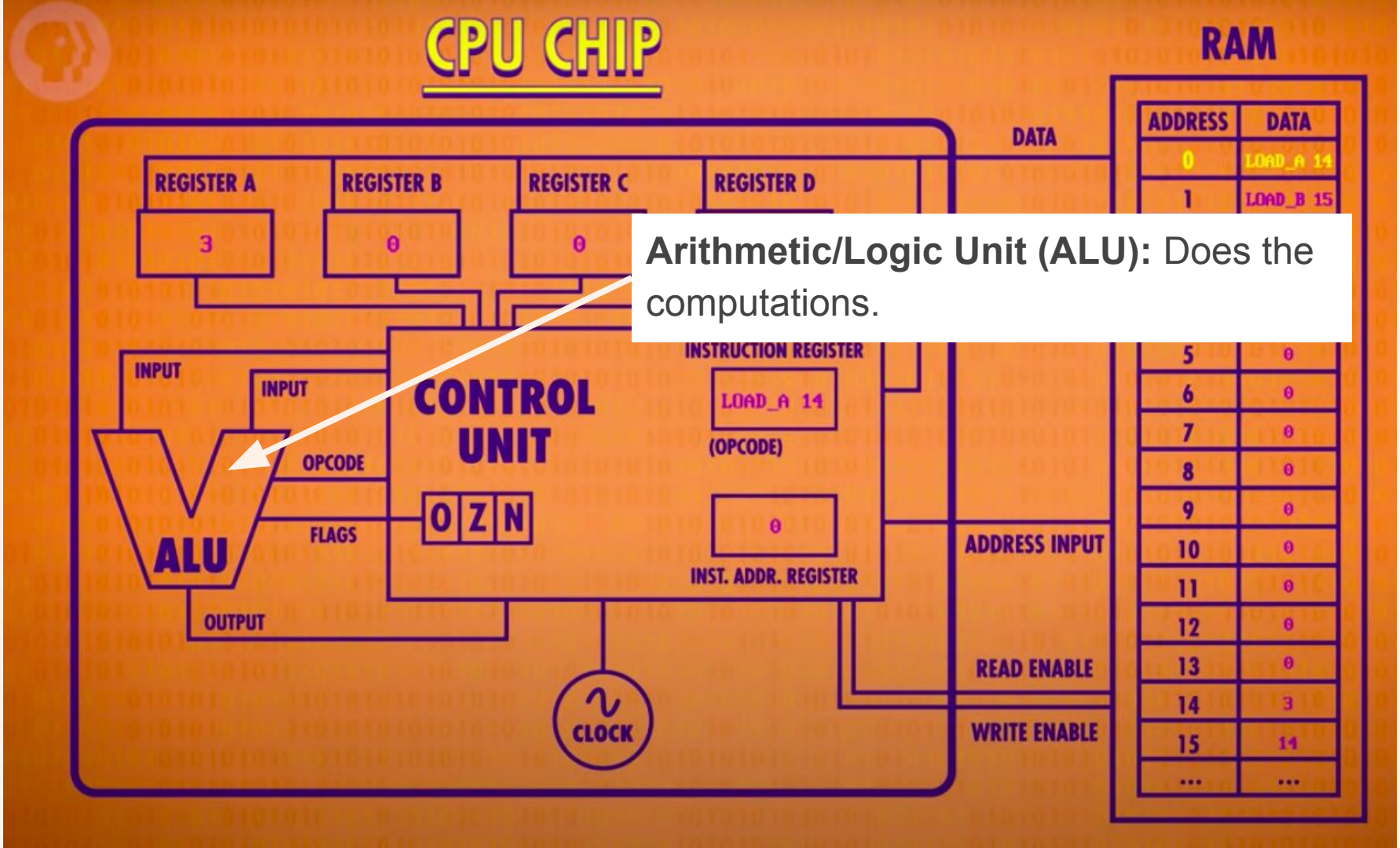


slidesmania.com



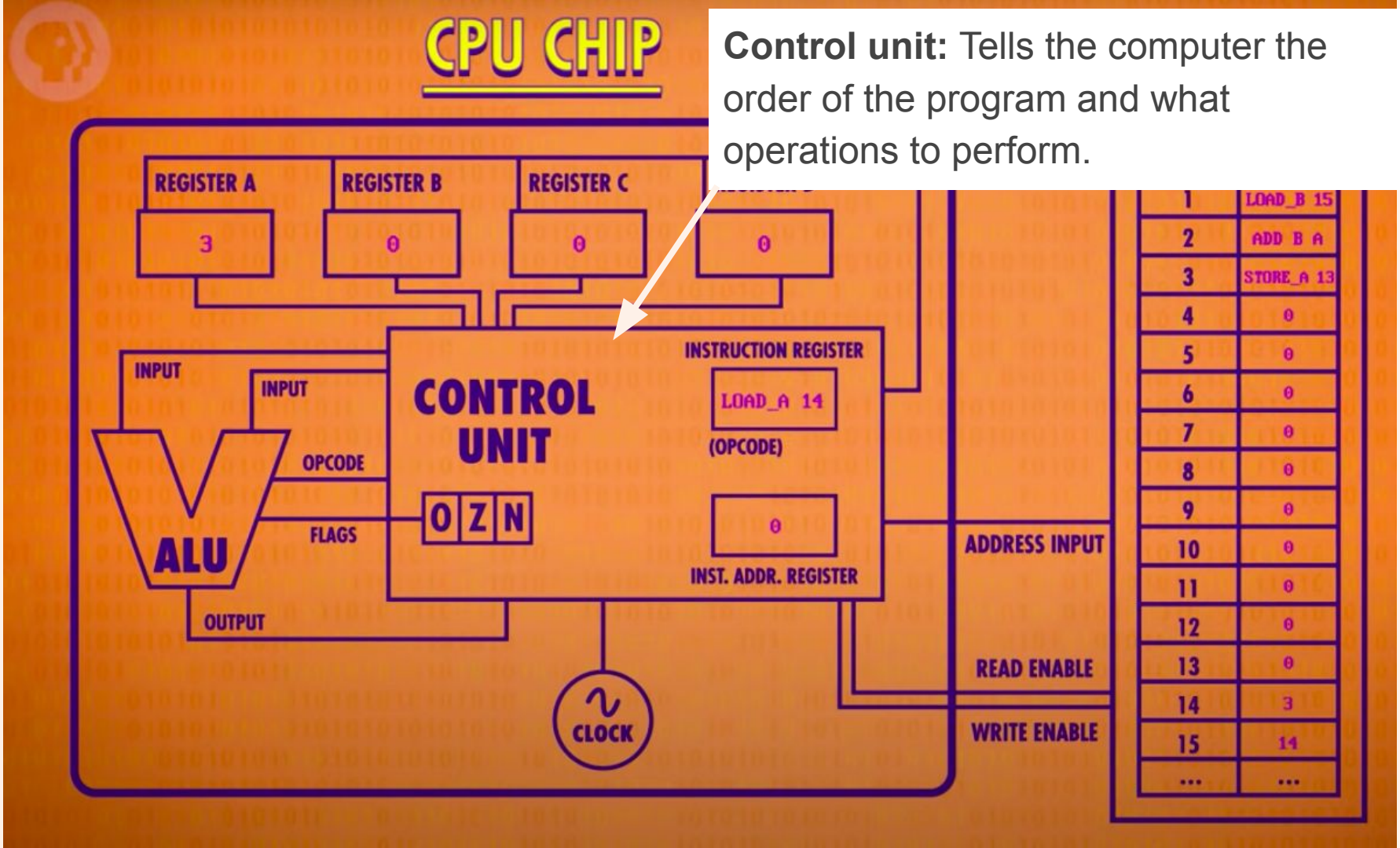
Central Processing Unit (CPU): The main machinery of computer that carries out instructions. “Brain” of a computer.

slidesmania.com

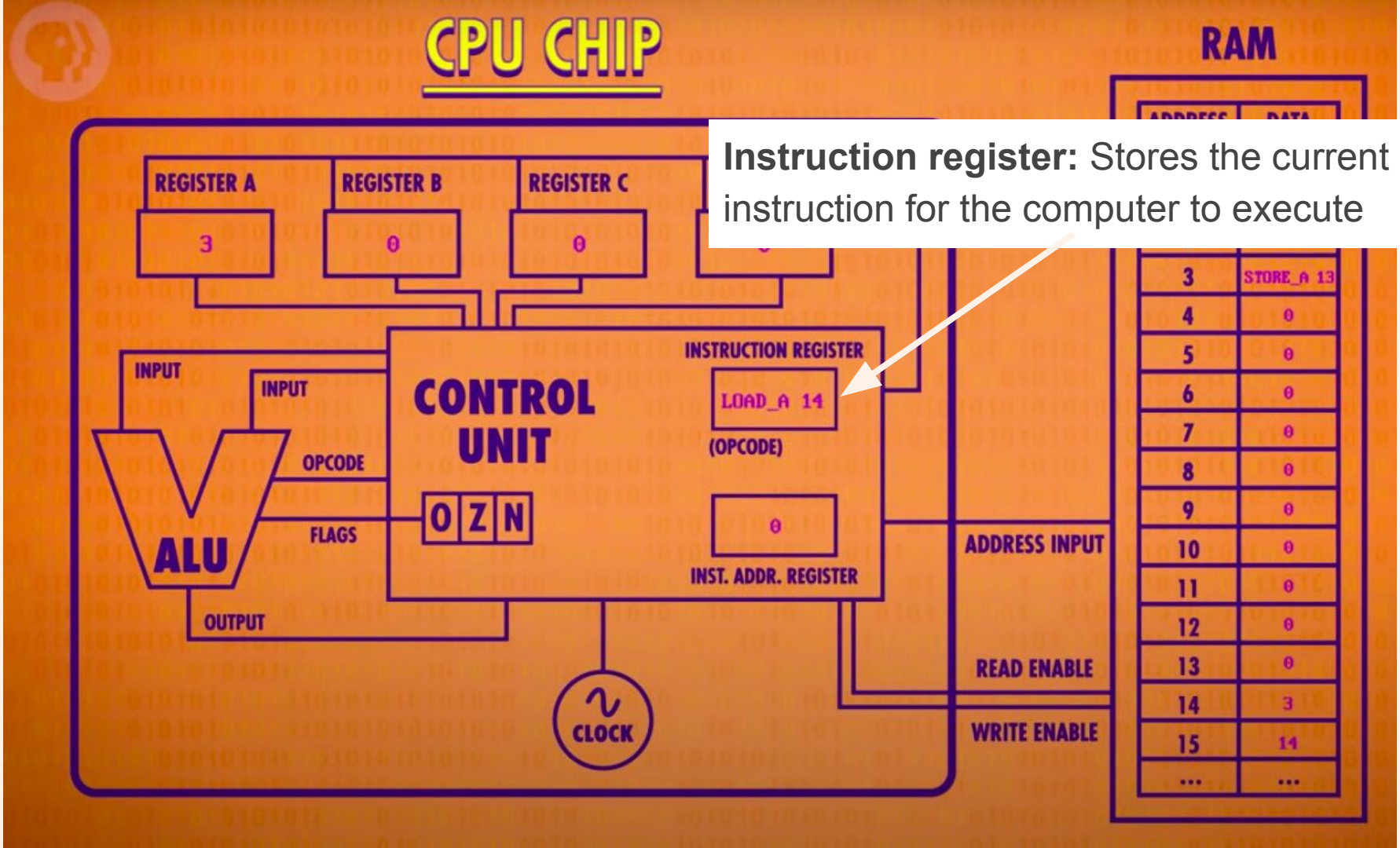


Arithmetic/Logic Unit (ALU): Does the computations.

slidesmania.com

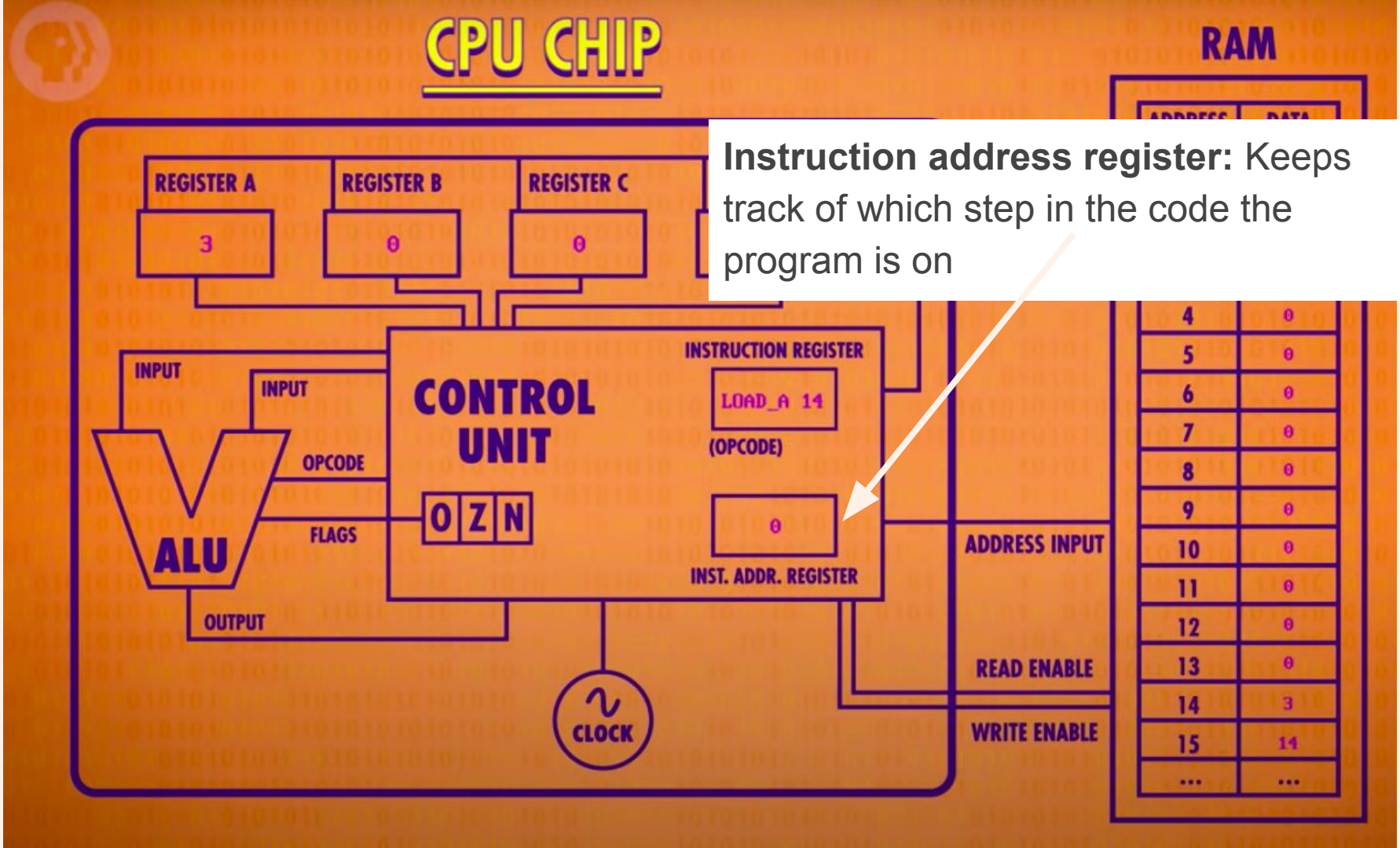


slidesmania.com



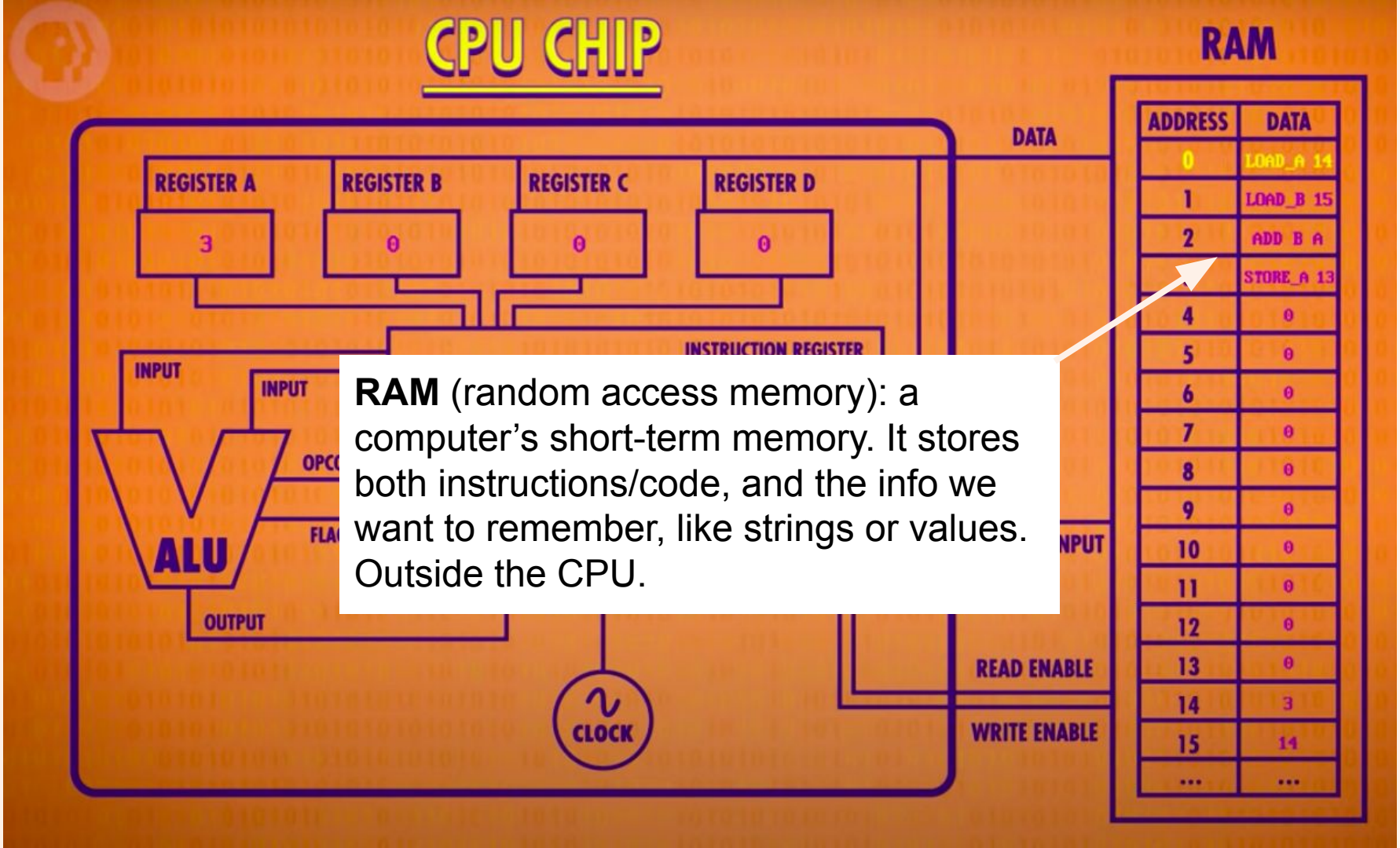
Instruction register: Stores the current instruction for the computer to execute

slidesmania.com



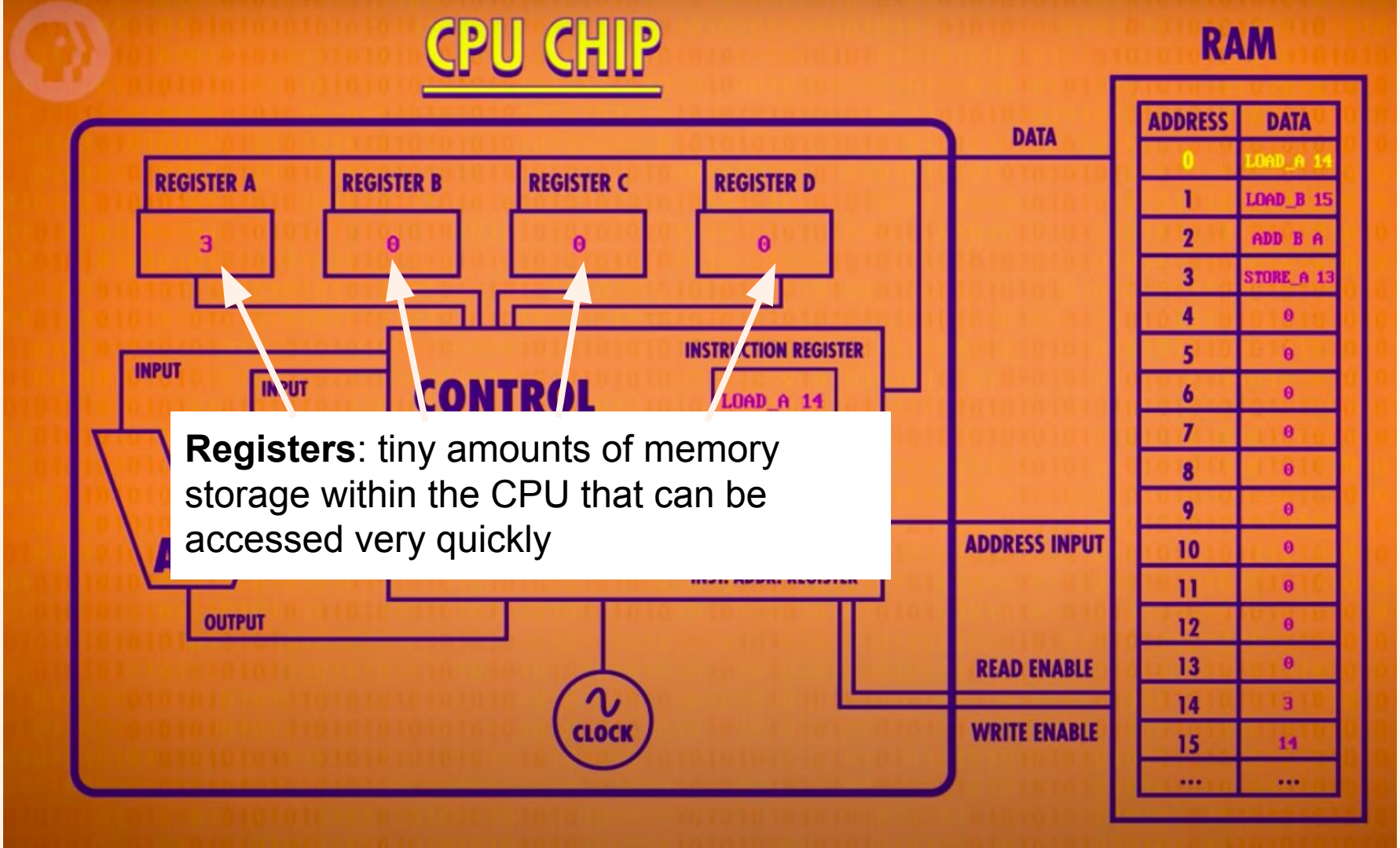
Instruction address register: Keeps track of which step in the code the program is on

slidesmania.com



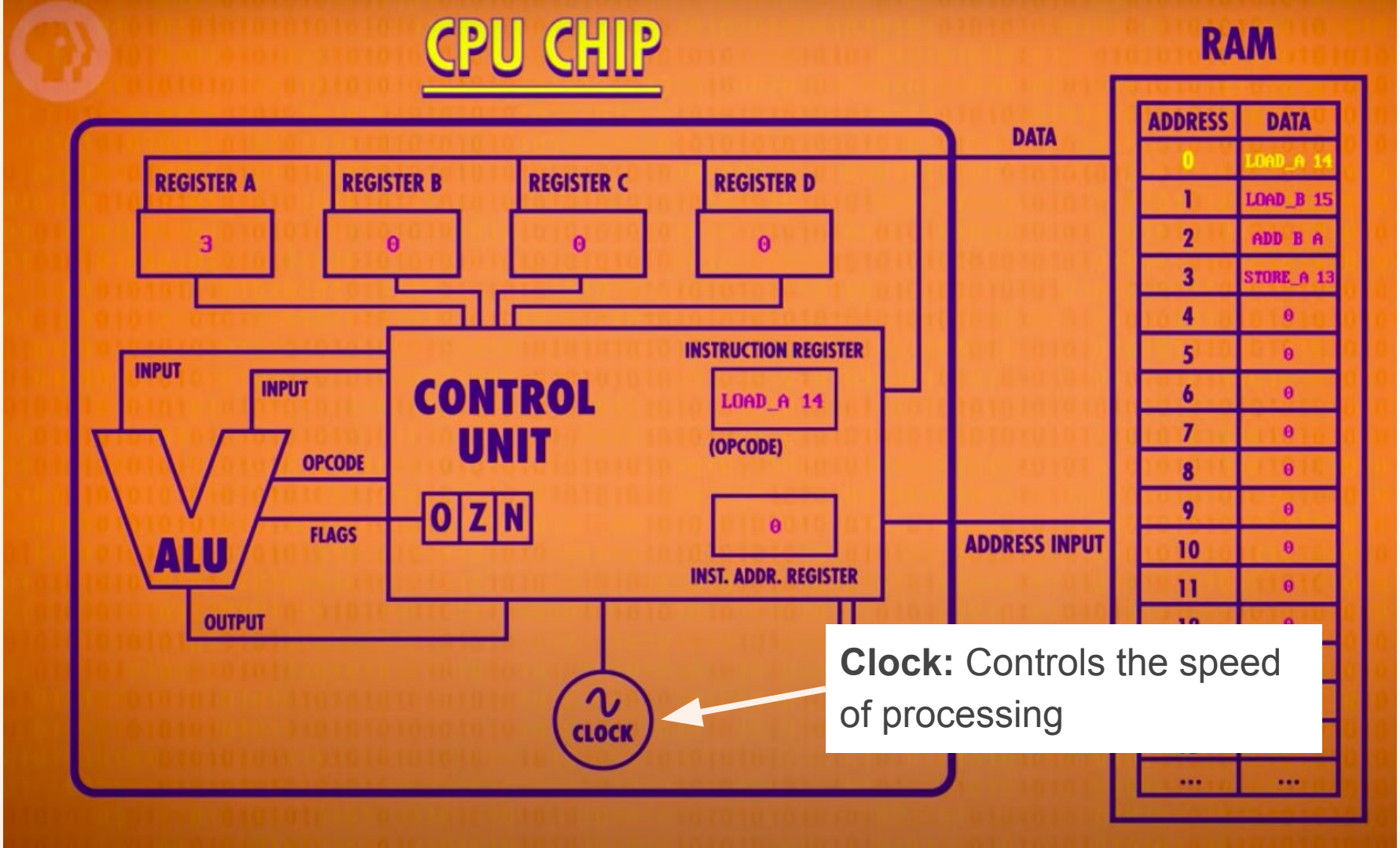
RAM (random access memory): a computer's short-term memory. It stores both instructions/code, and the info we want to remember, like strings or values. Outside the CPU.

slidesmania.com



Registers: tiny amounts of memory storage within the CPU that can be accessed very quickly

slidesmania.com



slidesmania.com

Assembly Code in Action

slidesmania.com

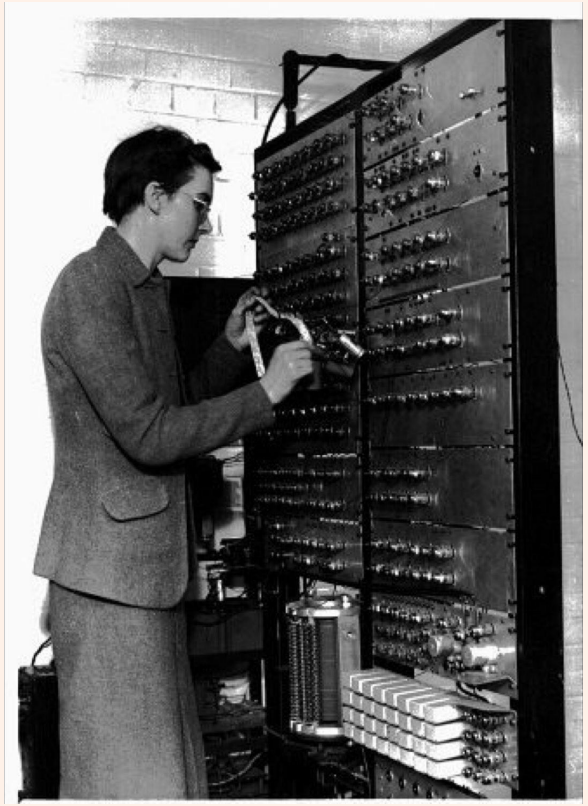
History: the First Computers

- The first computers, such as ENIAC and UNIVAC used binary code as its primary source code and were very difficult to program.

- The process was tedious and time-consuming, requiring the programmer to manually enter every instruction into the machine.

slidesmania.com

History:the First Computers



Kathleen Booth (1922 -2022) - Assembly Language Theory Creator and Rockstar

- Wrote first assembler for code she created to convert alphanumeric code into binary.
- Along with husband Andrew she co-founded one of the first computer science programs at Birkbeck College, University of London.
- Helped create 3 new computer designs.

slidesmania.com

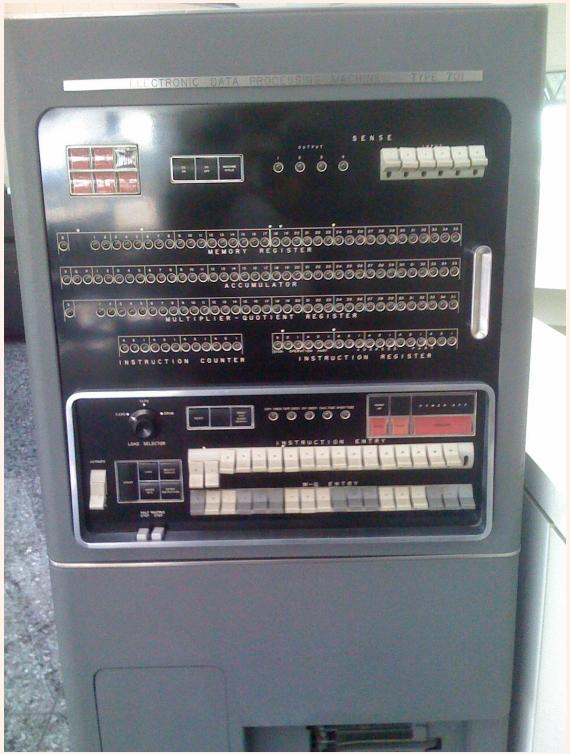
Kathleen Booth - Assembly Language Creator

slidesmania.com

History: First Assembly Language

- In the early 1950s, the first assembly languages were developed to simplify the programming process.

- The first assembly language was developed for the IBM 701 computer.



slidesmania.com

History: Advancements

During the 60’s and 70’s, assembly languages advanced to be more user-friendly and more powerful as new instructions were added to make the language more expressive.

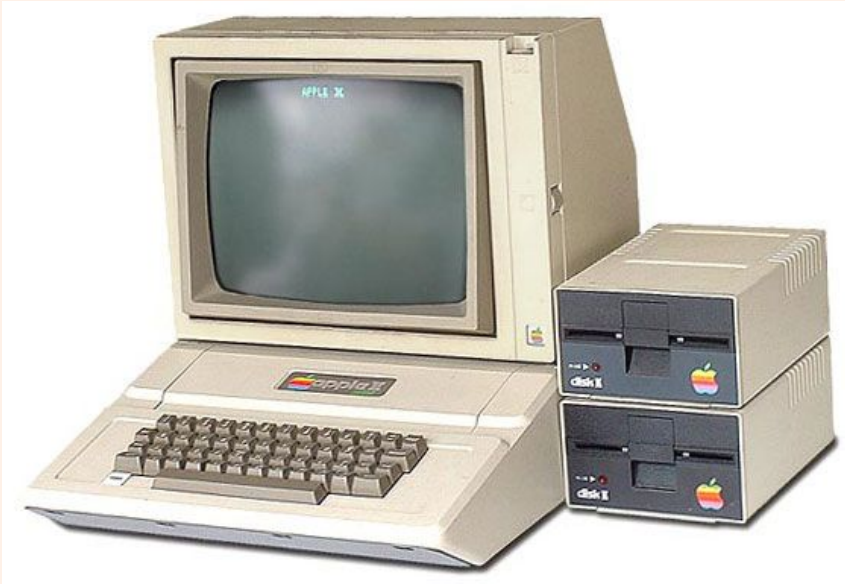
This also made assembly language programs faster and more efficient, which made them widely used on mainframe computers of the day.

slidesmania.com

History: Assembly Languages and Personal Computing

In the 80s and 90s, assembly language was used to write programs for personal computers, such as IBM PC and the Apple Macintosh.

As higher-level programming languages became more popular, assembly language became less common.



slidesmania.com

Assembly Language wins a Grammy

In 1978, at the request of his employers, Donald Fagen and Walter Becker, Recording audio engineer Roger Nichols created one of the earliest electronic drum machines using a COMPAL-80 and the 8080 assembly language.

slidesmania.com

Assembly Language wins a Grammy

Nichols named his creation Wendel. Wendel received performance credit on several tracks on the 1980 album “Gaucho”.

In 1982, Gaucho (and Wendel) won a grammy for “Best Engineered Non-Classical Recording”.

slidesmania.com

Thank you for reading our pre-work presentation!

You rock!

slidesmania.com