

COMPUTER SYSTEMS

GENERAL INFORMATION

Goals

Understand the computer organization.

Identify the computer architectures and the Instruction Set Architecture.

Summary

- 1.** |  Computer architecture
- 2.** |  Memory
- 3.** |  CPU
- 4.** |  Input / Output
- 5.** |  First Concepts
- 6.** |  Conclusions

Computer Functions and Components

1. Data processing (proceso de datos).

Multiple types of data and processing requirements.

↳ multiples tipos de datos y requisitos de procesamiento.

2. Data storage

- Short-term
- Long-term

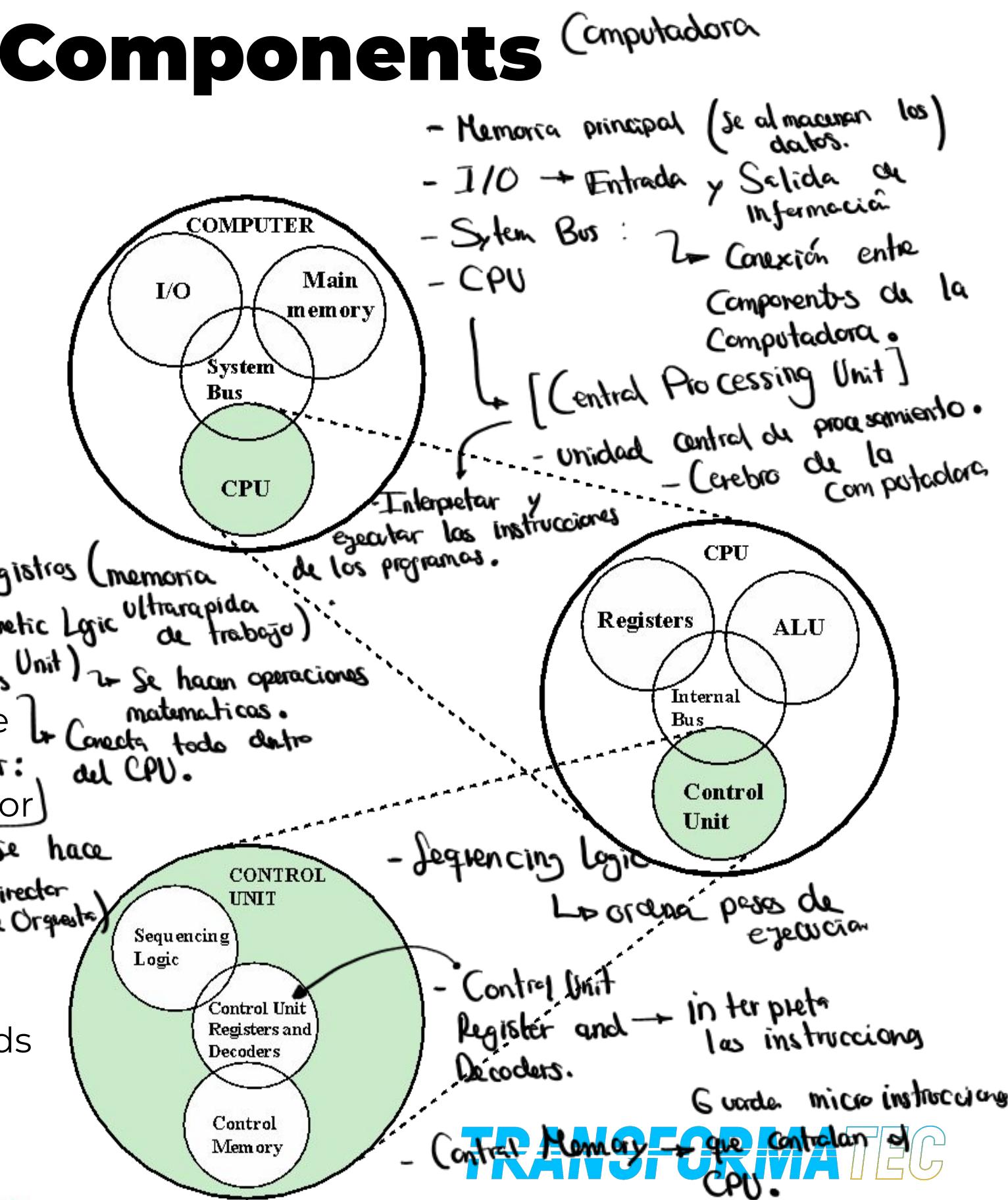
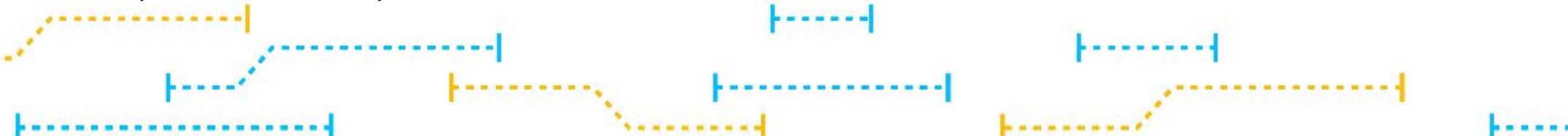
} término corto
término largo

3. Data movement

- Input-output (I/O): when data are received from or delivered to a device (peripheral) that is directly connected to the computer
- Data communications: when data are moved over longer distances, to or from a remote device

4. Control

A control unit manages the computer's resources in response to commands (instructions).



- Es de código abierto

RISC V Architecture

- Las Ventajas de utilizar RISC-V y del OpenSource es que muy probablemente sea la tecnología dominante en el área de arquitectura de computadoras.
- Esta libertad hace que multiples usuarios puedan contribuir con las bases de esta tecnología. Además es altamente modificable para el área de investigación y empresarial.

- First **open - source** instruction set architecture with broad commercial support.

→ hace 15 años

- Defined in **2010** by Krste Asanović, Andrew Waterman, David Patterson.

En la Universidad de California, Berkeley (UC Berkeley). Era estudiantes del laboratorio de arquitectura de Computadoras.

- **Comparable** in capabilities to commercial architectures such as ARM and x86.

Se puede comparar a los actuales arquitecturas comerciales. ↗

- Commercial **chips**: SiFive and Western Digital

SiFive es la primera startup basada completamente en RISC-V.

Lo que hace es dar servicios de implementación de los chips, mantenimiento, Soporte, diseños optimizados y personalización.

Muchas personas quieren usar SiFive pero no saben como empezar

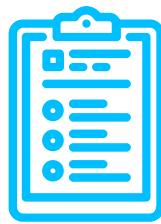
Una vez aprendida una arquitectura, esto es fácil de aprender otra.

Once you've learned one architecture, it's easier to learn a different one 😊

RISC V viene a ser una buena opción

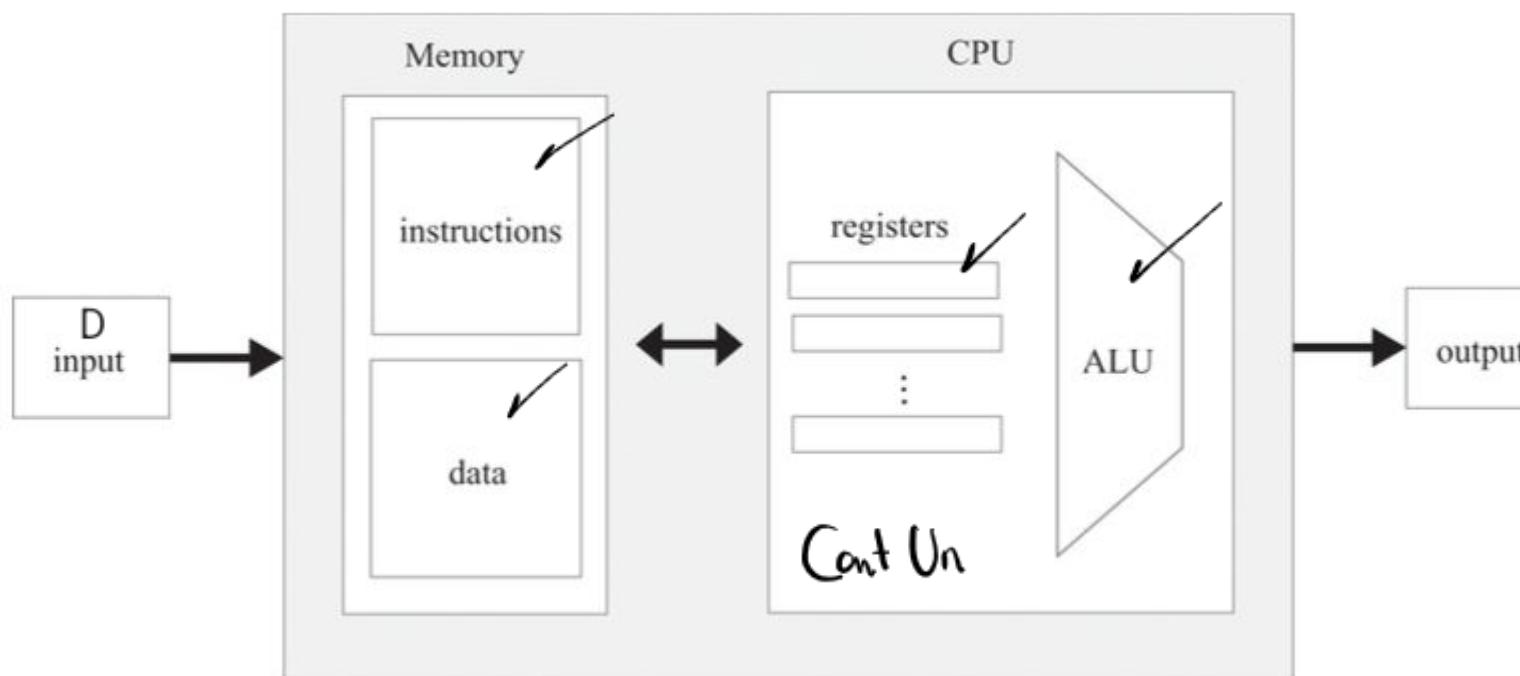
Western Digital (Multinacional de hardware). Adoptó RISC-V en masa y demuestra viabilidad comercial y escalabilidad.

1.



Computer Architecture





Von Neumann Architecture

despues de la turing machine. (1936)

- After Turing Machine (1936).
- Memory
 - Data memory
 - Instruction memory
- CPU
 - Arithmetic Logic Unit
 - Registers
 - Control Unit



El Kernel es el que soporta las acciones en alto nivel.

► Reinventa el mundo ◀

Von Neumann Model

5 parts:

- Memory ✓
- Processing Unit CPU ✓
- Input ✓
- Output ✓
- Control Unit ✓

Problema: Cuello de botella de Von Neuman

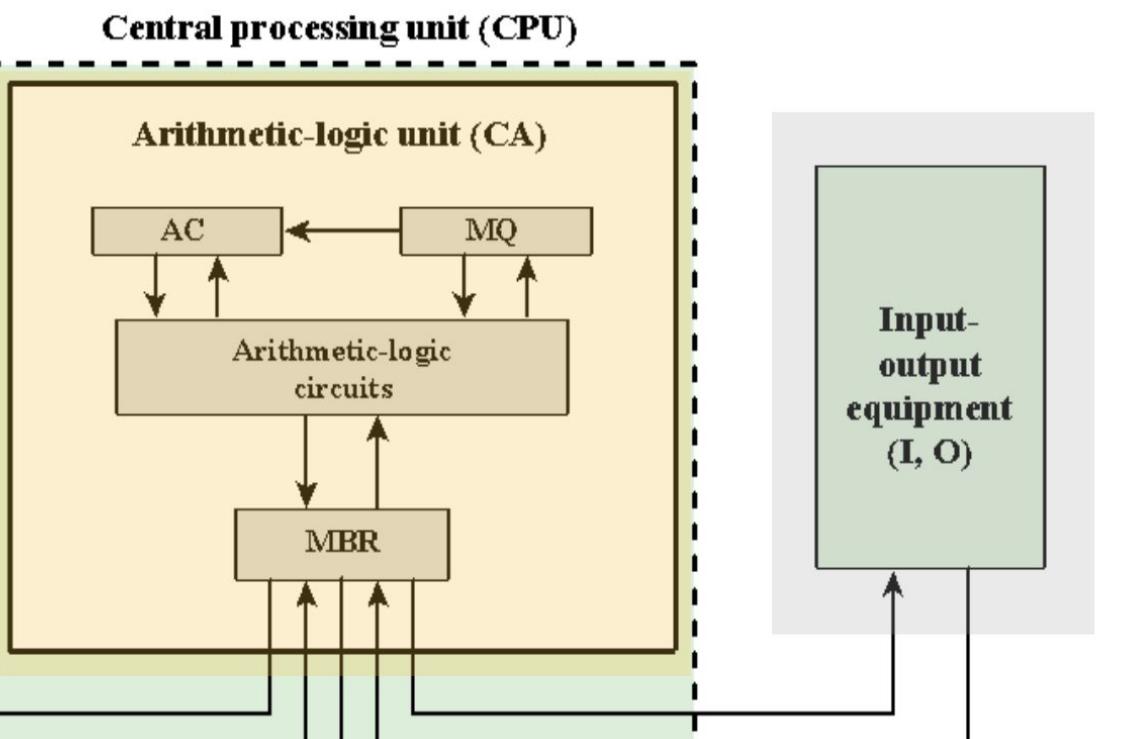
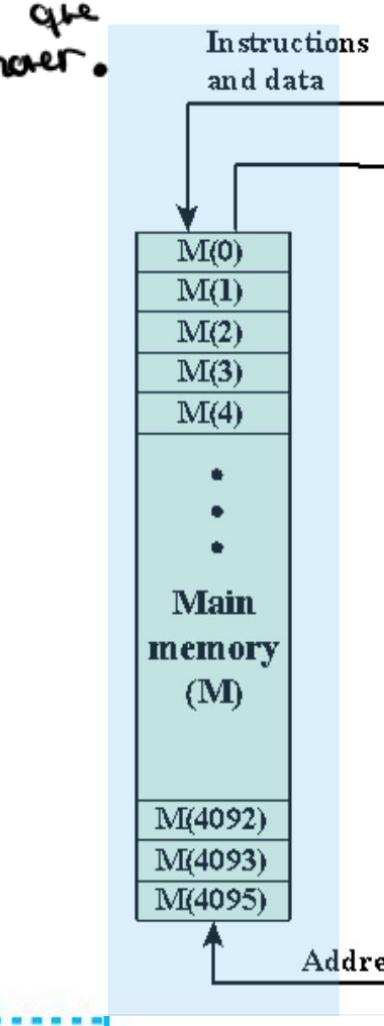
BottleNeck:

Tiempo acceso a data > tiempo en ejecución del programa.

Total time for access data can be higher than the execute program time.

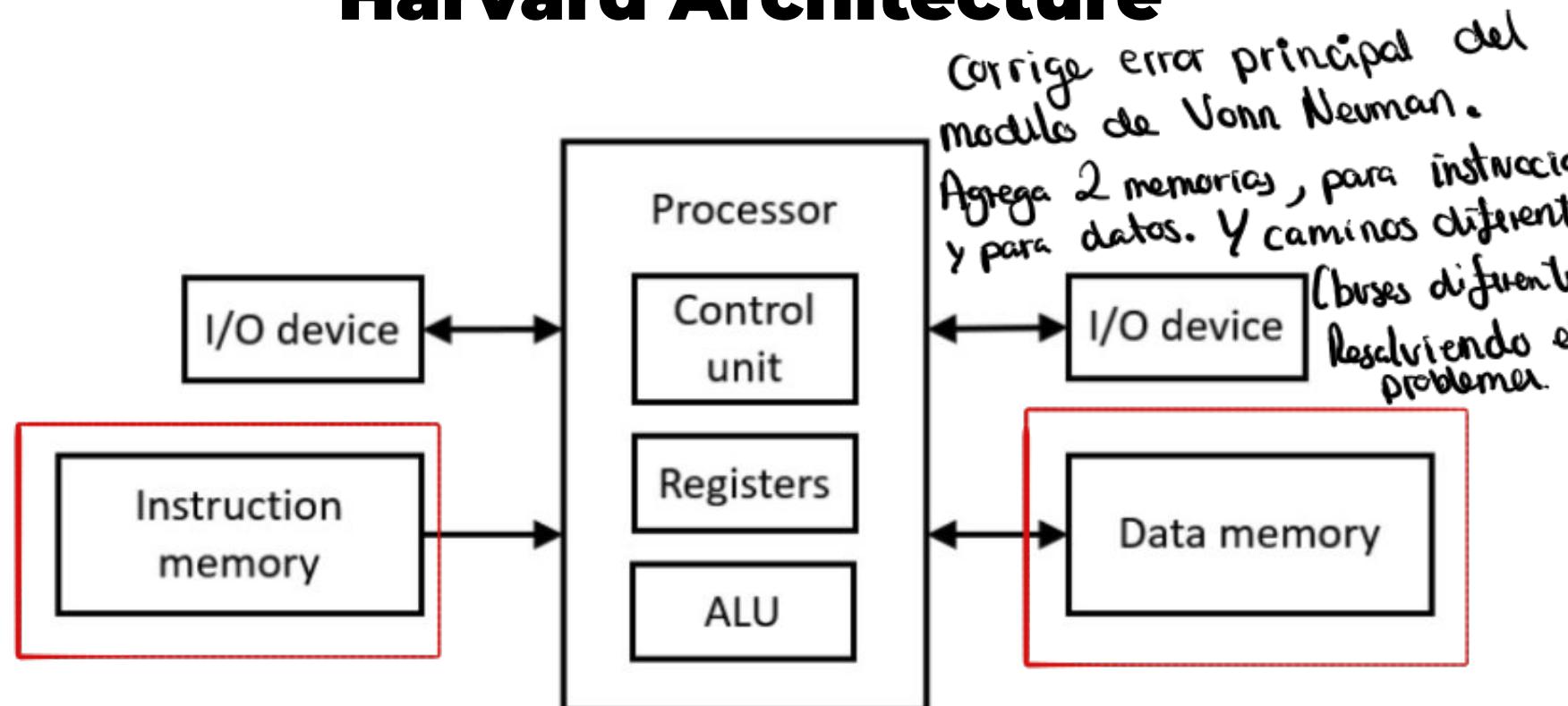


Uno de los problemas principales es que las instrucciones y los datos viajan por el mismo bus. Esto quiere decir que el CPU tiene tiempos muertos entre que espera el recorrido de las instrucciones y los datos. Esto genera un mayor consumo energético debido a los tiempos muertos. Desaprovecha el potencial y rendimiento del CPU ya que este no tiene la velocidad del bus a bits que puede mover.



AC: Accumulator register
MQ: multiply-quotient register
MBR: memory buffer register
IBR: instruction buffer register
PC: program counter
MAR: memory address register
IR: instruction register

Harvard Architecture

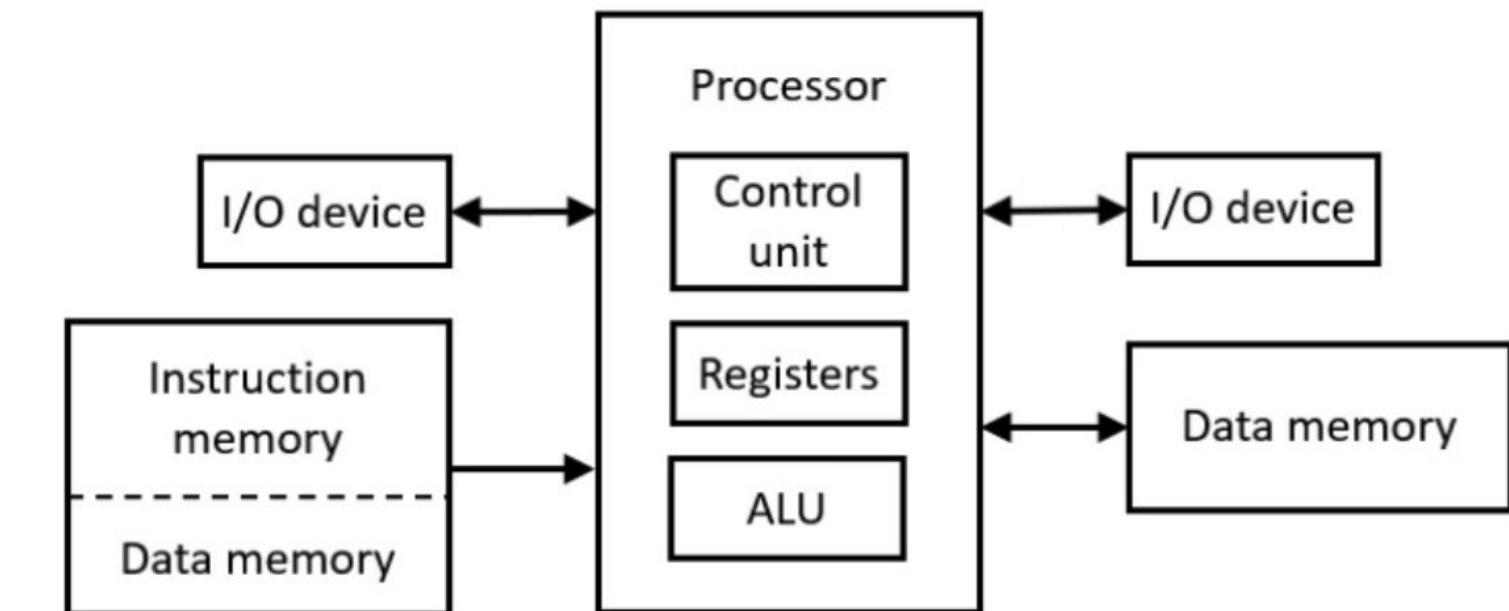


- Complete separation of Instruction and Data memory regions.
- **Rarely used in modern computers**

Desventajas:

- Es muy caro y complejo
- Requiere 2 memorias físicas distintas.
- No puedes tratar a la información como data fácilmente.

Modified Harvard Architecture



- Some degree of separation between Instruction and Data memory regions.
- **Modern Processors.**
 - . Incluido en la mayoría de procesadores modernos.
 - . Mantienen los doble buses para data y para instrucciones.

2.



Memory

TRANSFORMATEC



- La memoria es ordenada
- es un sistema computacional
- Tiene que ser ordenado
- Tiene tamaño fijo



Physically

Linear sequence of
REGISTERS
- Addressable.
- Fixed-size

Estantería
Cajones numerados

- Secuencia lineal, la memoria puede serse ordenada en forma de matriz.
- Cada memoria tiene un casillero único. En los PCs modernos, la memoria es byte-adressable, cada dirección apunta a un byte (8 bits).
- Fixed size: todos los espacios de memoria tienen el mismo tamaño, depende de la arquitectura.



Logically

To Store
- Data.
- Instructions.
HOW?

Sequence of bits

Todos son bits, 1's o 0's. La diferencia es que cada cadena de bits tiene la capacidad de ser interpretada como un dato o una instrucción.

Segura y lenta.

Acceso aleatorio.
Memoria dinámica.

ROM vs RAM

Read Only Memory:
Sequential occupation of locations..
Random Access Memory:
Every register can be reached instantaneously.

ROM: memoria que no cambia, almacena información importante e instrucciones.
RAM: memoria volátil, útil para almacenar información de uso continuo.



Componente computacional que almacena Registros.

Conjunto de registros.



Data Memory

- High - level program.
- Variables, arrays, etc.
- HW level:
Binary values
- Reading/Writing memory registers.

Memoria donde se encuentran nuestros datos en tiempo de ejecución.



Instruction Memory

- Before high - level.
- Low - level instructions.
- Binary values.
- Executable (binary)
from
- Load to disk for running a program.

el guion que sigue la PC paso a paso.

Dónde está el código máquina que el CPU va leyendo y ejecutando.

TRANSFORMA TEC

Memory

1. The memory stores

Data/Programs

La memoria RAM guarda dentro dos tipos de datos Datos/Programas.

En la memoria todo son bits (0, 1)

Se organiza mediante bytes o Words.

Las words pueden ser de \rightarrow bits según la arquitectura.

2. The memory contains bits

Bits are grouped into bytes (8 bits) and words (e.g., 8, 16, 32 bits)

3. How the bits are accessed determines the addressability

• Direccionalidad es como la CPU decide que cantidad de memoria corresponde a una dirección:

- E.g., word-addressable ①

- E.g., 8-bit addressable (or byte-addressable) ② → la más usada hoy.

The total number of addresses is the address space

Número total de direcciones

Espacio de direcciones

- In x86, the address space is 2^{32} y 2^{64} .

- In ARM, the address space is 2^{32} . ✓

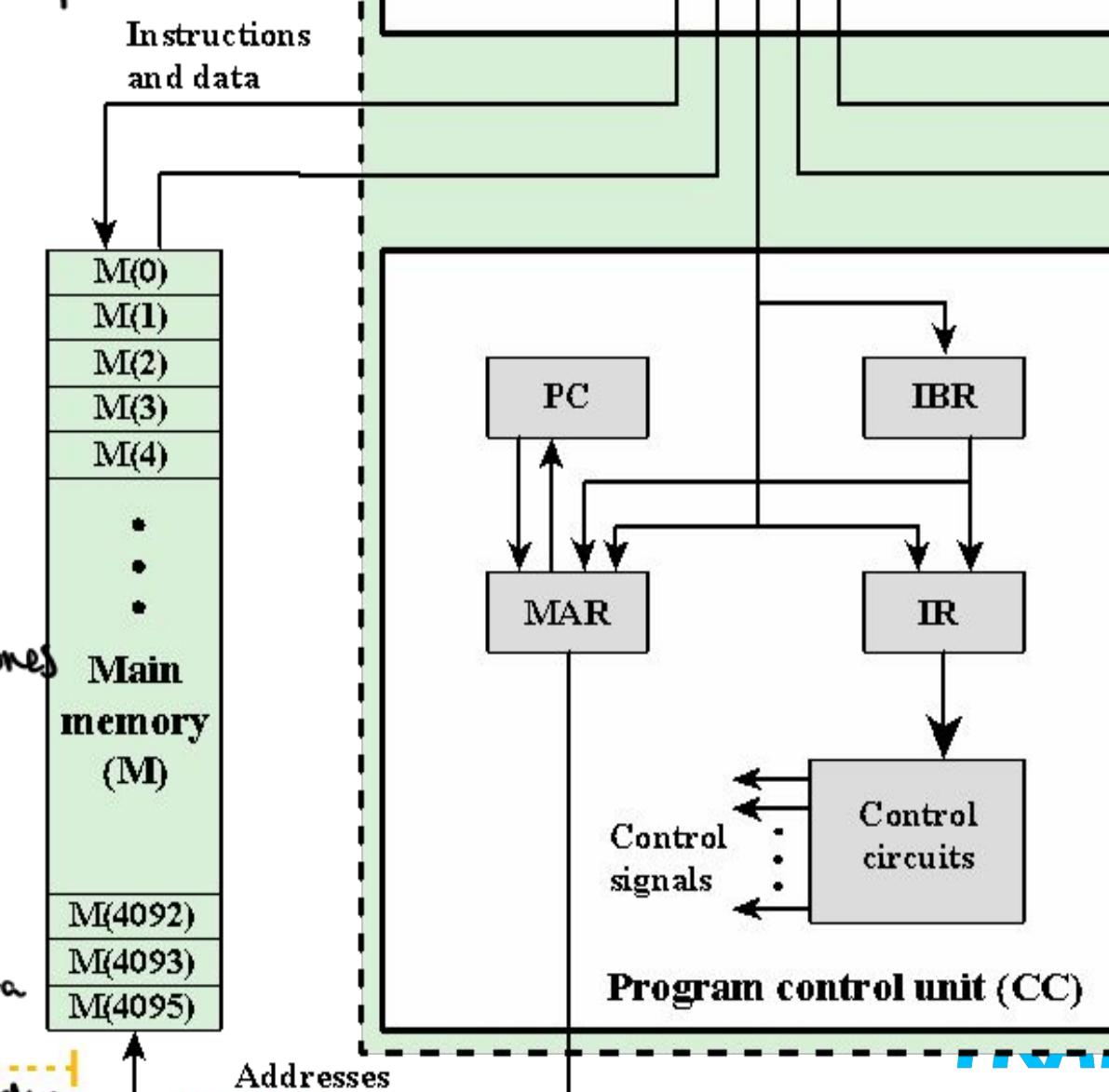
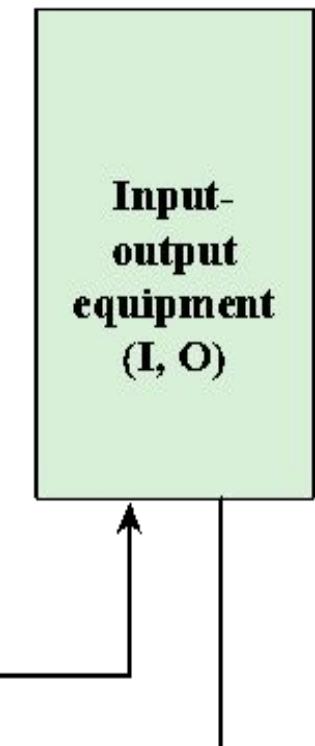
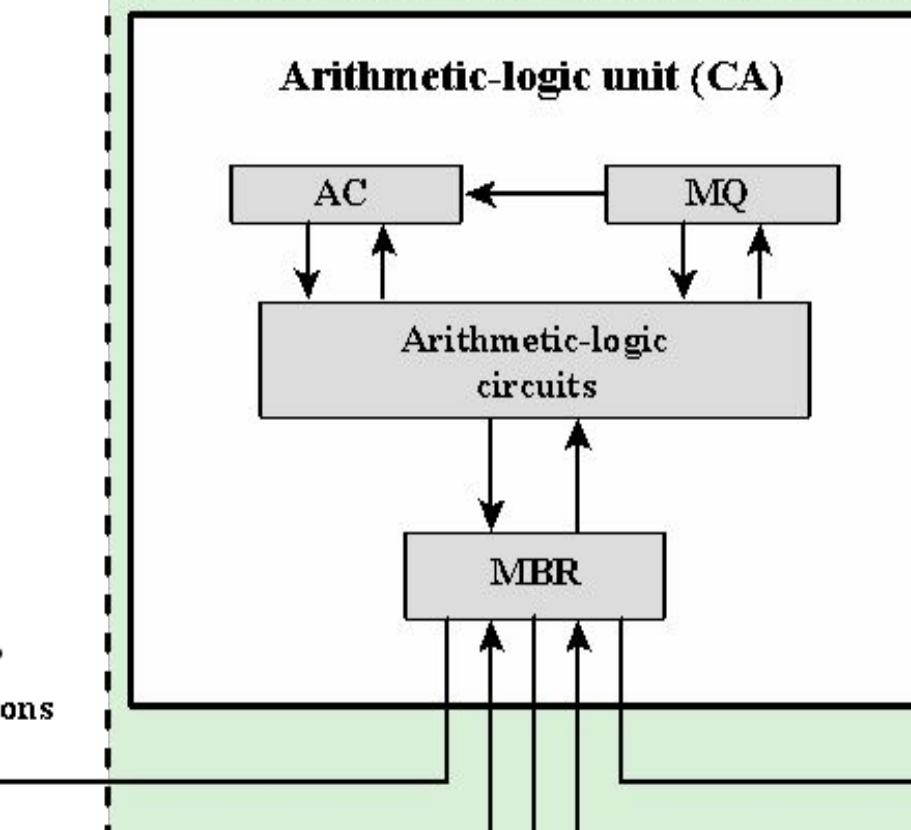
- 32-bit addresses

- In x86-64, the address space is (up to) 248.

$$2^{12} = 4095$$

Direccionalidad: tamaño de lo que representa una dirección.

Espacio de direcciones: cuántas direcciones se pueden generar con los bits de la dirección.



AC: Accumulator register
MQ: multiply-quotient register
MBR: memory buffer register
IBR: instruction buffer register
PC: program counter
MAR: memory address register
IR: instruction register

Accessing Memory

hay dos formas de acceder a la memoria.

1. There are two ways of accessing memory

- Reading or loading leer (traer información desde memoria).
- Writing or storing Escribir (mandar información desde CPU).

2. Two registers are necessary to access memory

- Memory Address Register (MAR) → Guarda la "dirección de memoria" que se quiere leer o escribir
- Memory Data Register (MDR) → guarda el "dato" que viaja entre la CPU y la memoria

3. To read

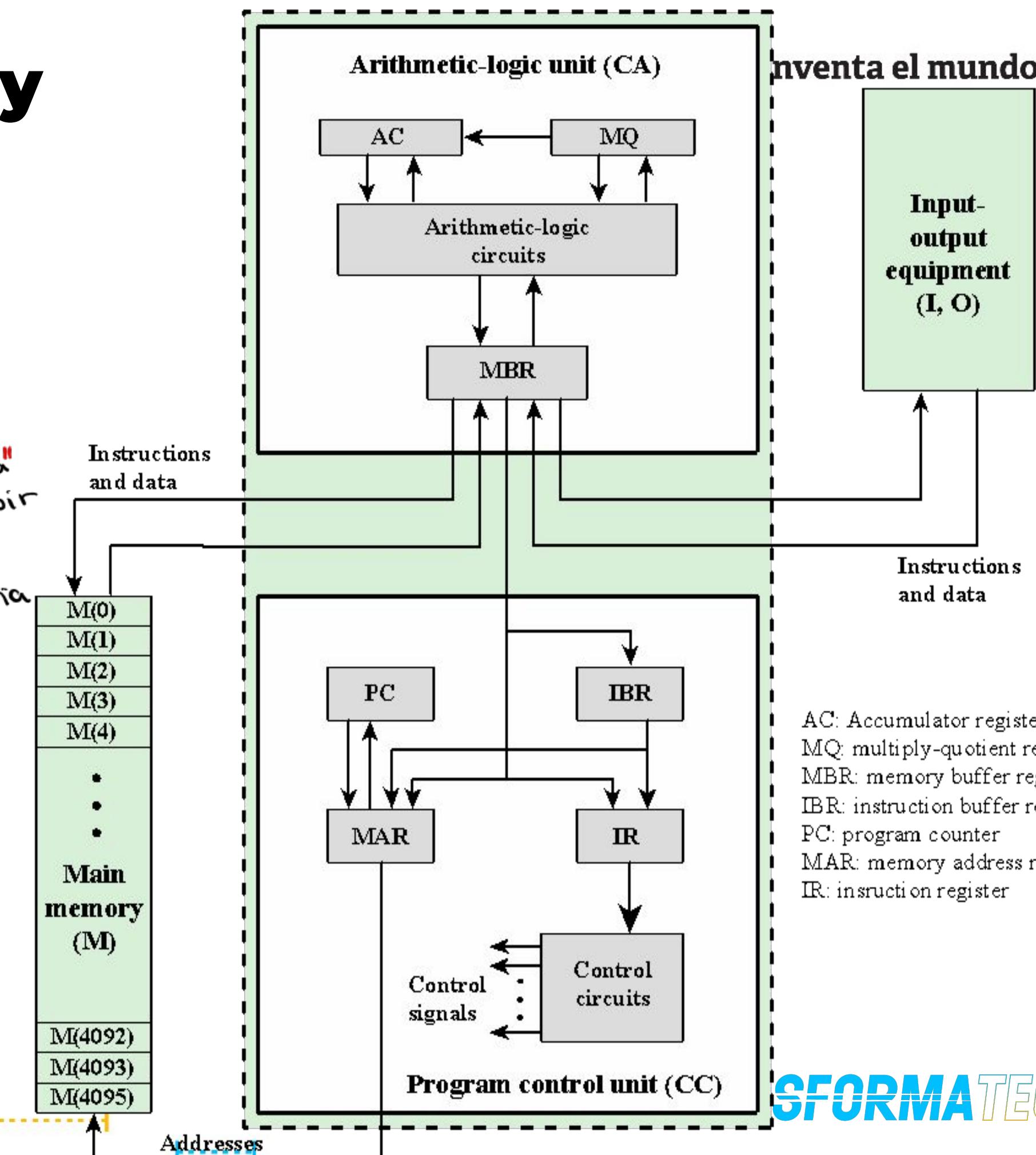
- Step 1: Load the MAR with the address
- Step 2: Data is placed in MDR

Para leer, la CPU manda la dirección en MAR y guarda el valor MDR de la memoria y lo trae al CPU.

4. To write

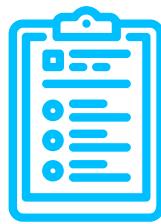
- Step 1: Load the MAR with the address and the MDR with the data
- Step 2: Activate Write Enable signal

Guarda la dirección en MAR, Guarda el valor en MDR y Activa el Write Enable.



AC: Accumulator register
 MQ: multiply-quotient register
 MBR: memory buffer register
 IBR: instruction buffer register
 PC: program counter
 MAR: memory address register
 IR: instruction register

3.



Control Processing Unit



Central Processing Unit

Execute the instructions:

Instructions

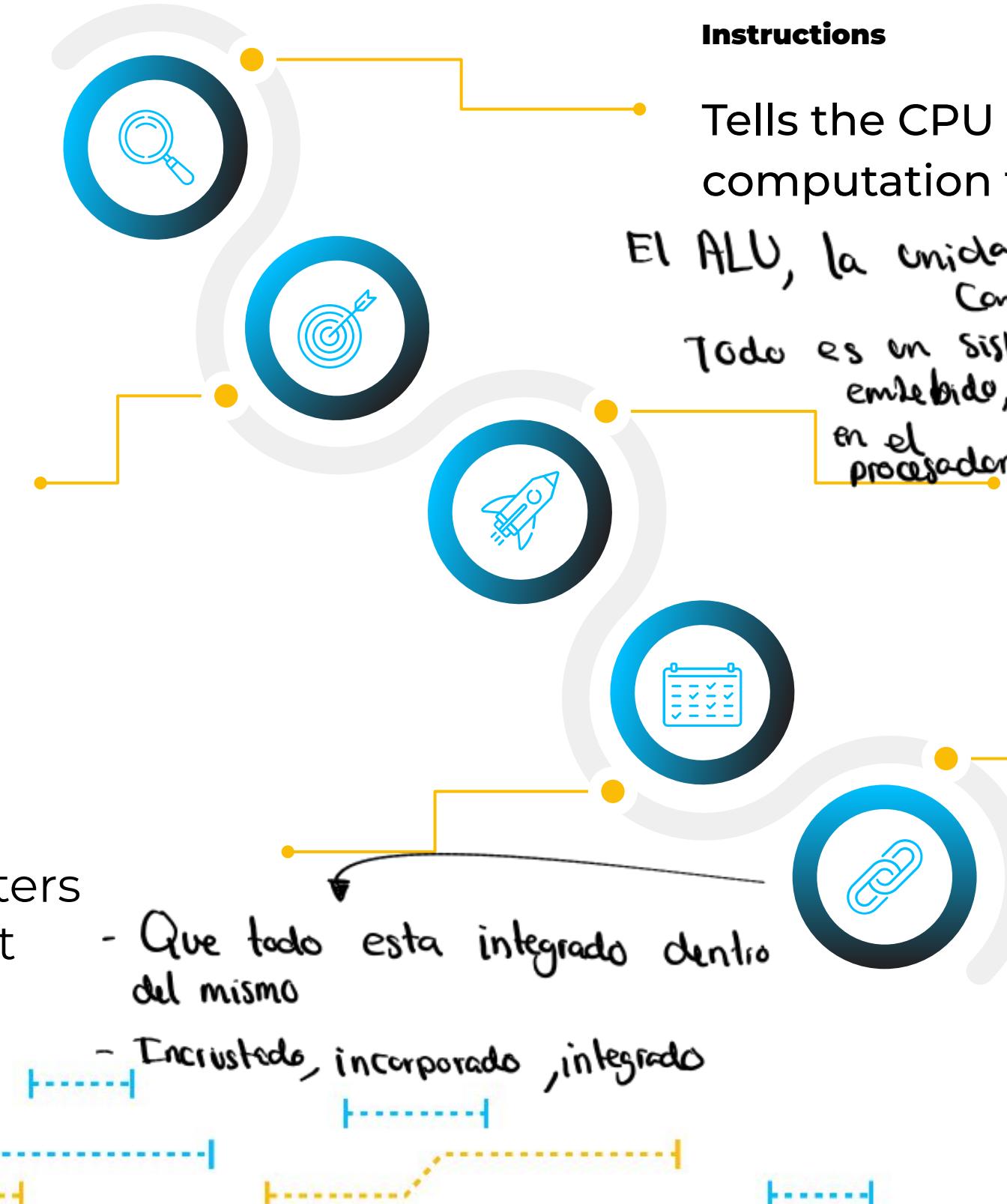
Tells the CPU which registers to access

Main Elements

ALU

Set of registers

Control Unit



la unidad Central de procesamiento está encargada que registros debe utilizar.

- Se le dice a la CPU cual es la que operación se debe realizar.
- Le le indica al CPU cual es la siguiente ejecución. lo ejecuta el ALU.
-

Sistema embebido
que estan conectados y sirven para activar y desactivarse juntas con otros sistemas

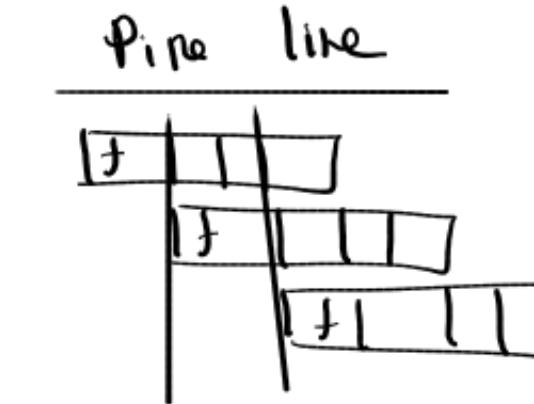
All Embedded

Processing Unit

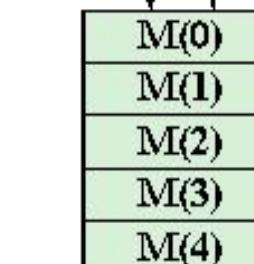
- The processing unit can consist of many functional units.
- We start with a simple Arithmetic and Logic Unit (ALU), which executes computations
 - ARM: add, sub, mult, and, nor, ...
- The ALU processes quantities that are referred to as words
 - Word length In ARM it is 32 bits
- Temporary storage: Registers:
 - E.g., to calculate $(A+B)*C$, the intermediate result of $A+B$ is stored in a register

Registros temporales.

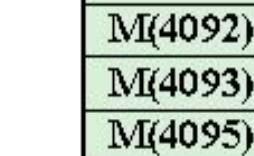
*Mini memorias rápidas
dentro del procesador.*



Instructions and data



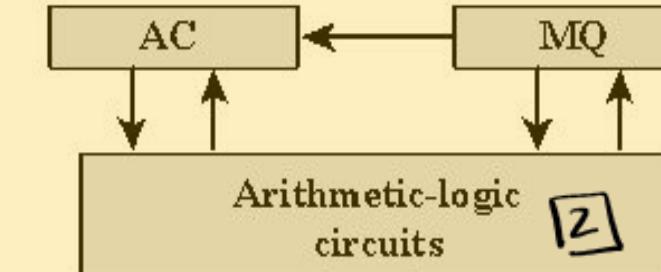
Main memory (M)



Addresses

Central processing unit (CPU)

Arithmetic-logic unit (CA)



*hay mas operaciones,
tene 4 operacion*

MBR

Input-output equipment (I, O)

Instructions and data

AC: Accumulator register

MQ: multiply-quotient register

MBR: memory buffer register

IBR

IBR: instruction buffer register

PC

PC: program counter

MAR

MAR: memory address register

IR

IR: instruction register

Program control unit (CC)

Control signals

Control circuits

SFORMATEC

REGISTERS

- **Memory** is big but slow
→ grande pero lenta

- **Registers**

Registers are faster than memory
mas rápidos que la memoria

Ensure fast access to operands
aseguran el acceso rápido a operaciones

Typically one register contains one word
Contiene palabras

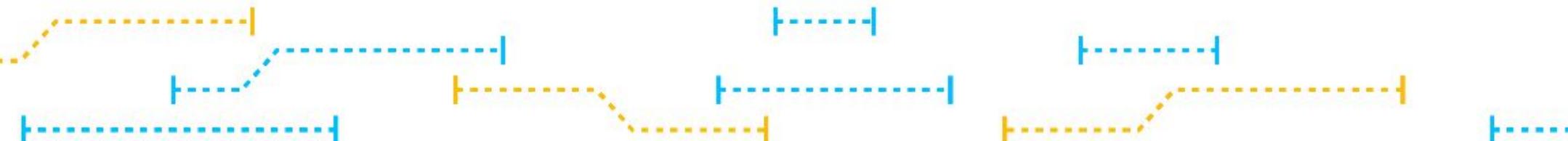
- **Register set or file**

RISC - V has 32 registers

Each register is 32 bits (RV32I)

RISC-V tiene
32 registros en total,
Cada uno con 32 bits
de tamaño.

Register	ABI Name	Description
x0	zero ✓	Hard-wired zero
x1	ra ✓	Return address
x2	sp ✓	Stack pointer
x3	gp ✓	Global pointer
x4	tp ✓	Thread pointer
x5–7	t0–2 ✓	Temporaries
x8	s0/fp ✓	Saved register/frame pointer
x9	s1 ✓	Saved register
x10–11	a0–1 ✓	Function arguments/return values
x12–17	a2–7 ✓	Function arguments
x18–27	s2–11 ✓	Saved registers
x28–31	t3–6 ✓	Temporaries



4.



Input / Output



Input and Output

- Many devices can be used for input and output
- They are called peripherals
- Input
 - Keyboard
 - Mouse
 - Scanner
 - Disks
 - Etc.
- Output
 - Monitor
 - Printer
 - Disks
 - Etc.

Muchos dispositivos pueden ser utilizados como medio de entrada y salida de información a estos los llamamos periféricos.

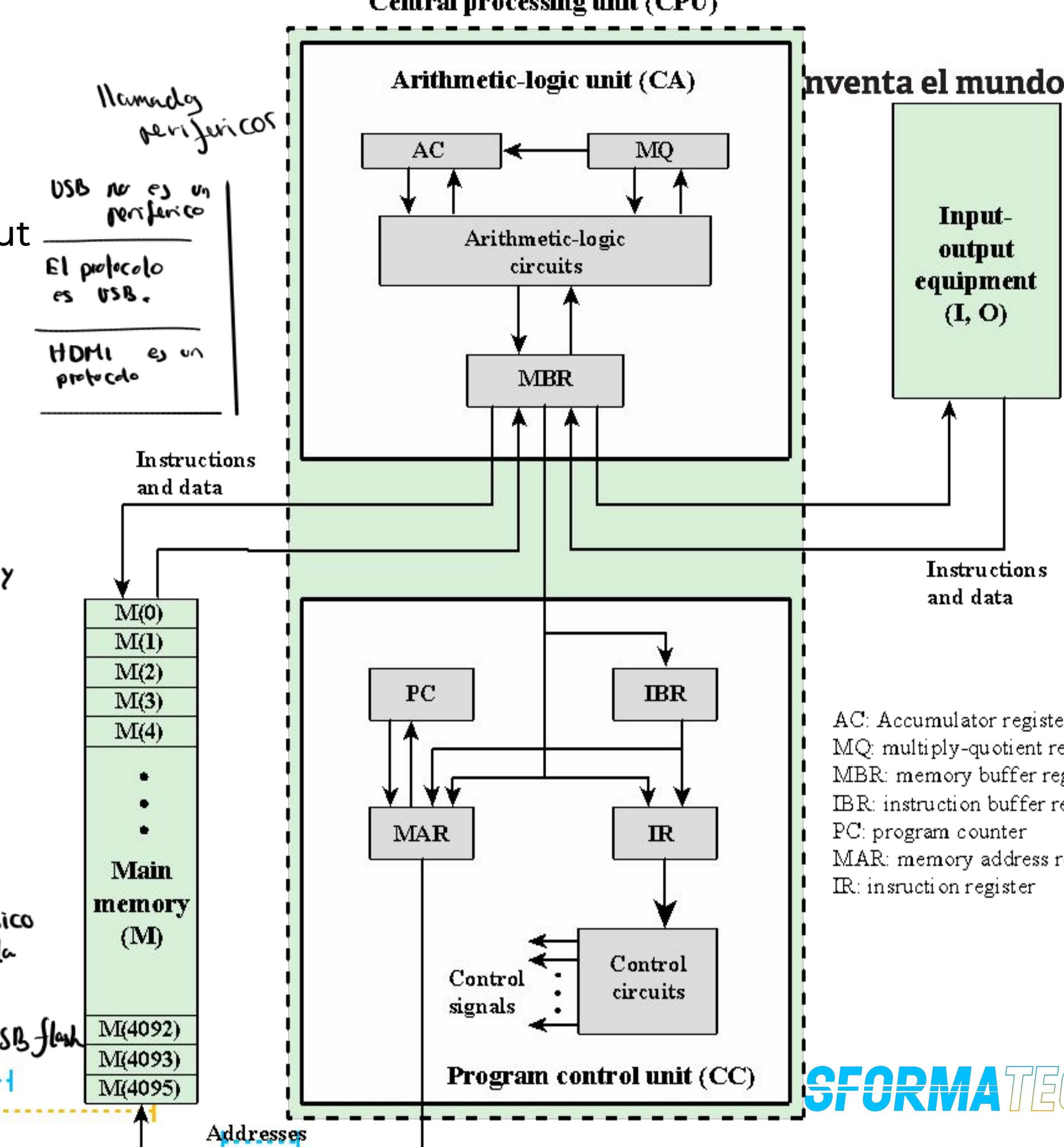
- hay una diferencia entre protocolo y periférico.

Un protocolo es el conjunto de reglas y normas que define como se transmiten o reciben los datos entre el CPU y un periférico.

Ejemplo: USB, HDMI, SATA, PCIe

Un periférico, es el dispositivo físico conectado a una computadora para la entrada o salida de datos.

Ej: teclado, mouse, impresora, pantalla, USB flash, Disco Duro.

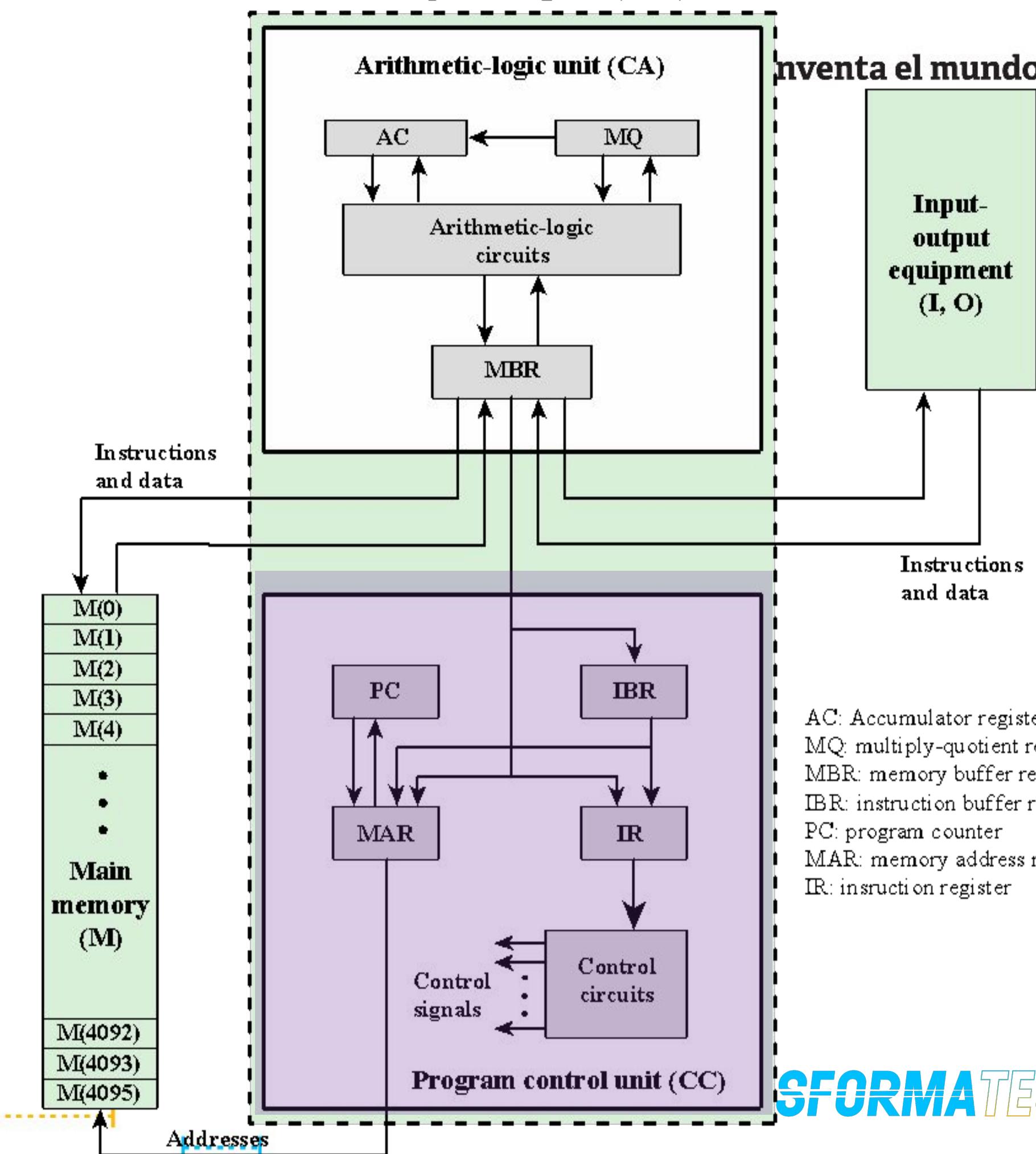


Control Unit

- The control unit is similar to the conductor of an orchestra
- It conducts the **step-by-step** process of executing (every instruction in) a program
- It keeps track of the instruction being executed with an **instruction register** (IR), which contains the instruction
- Another register contains the address of the next instruction to execute. It is called **program counter** (PC) or **instruction pointer** (IP)

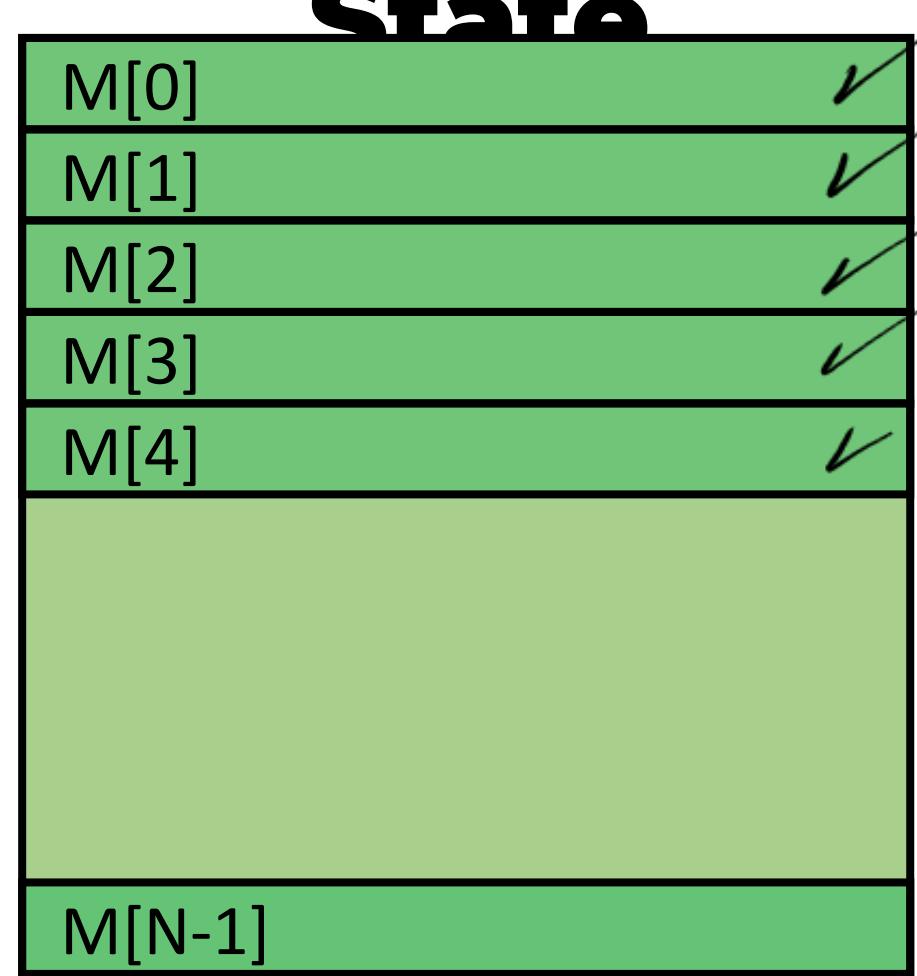
IR (Registro de instrucciones): guarda la instrucción que se está ejecutando en ese momento.

PC (program Counter): Lleva la dirección de la próxima instrucción a ejecutar.



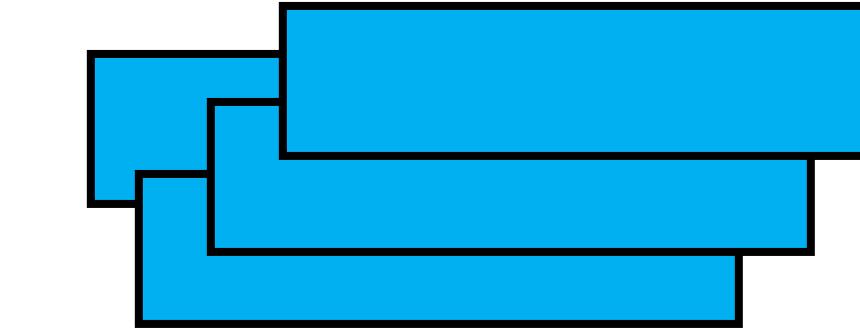
AC: Accumulator register
 MQ: multiply-quotient register
 MBR: memory buffer register
 IBR: instruction buffer register
 PC: program counter
 MAR: memory address register
 IR: instruction register

Programmer Visible (Architectural) State



Memory

Array of storage locations
indexed by an address



Registers

- given special names in the ISA (as opposed to addresses)
- general vs. special purpose

hay registros especializados para cierto propósito.

Program Counter

memory address

of the current instruction

→ registro especial que, guarda la dirección de la siguiente instrucción.
encargado de decirme donde estoy actualmente.

TRANSFORMATEC

Instruction Set
Architecture.

→ Instructions (and
programs) specify how
to transform the values
of programmer visible
state

↳ las instrucciones son
las acciones. Cambia
Continamente.

Von Neumann Model

Also called stored program computer (instructions in memory). It has two key properties:

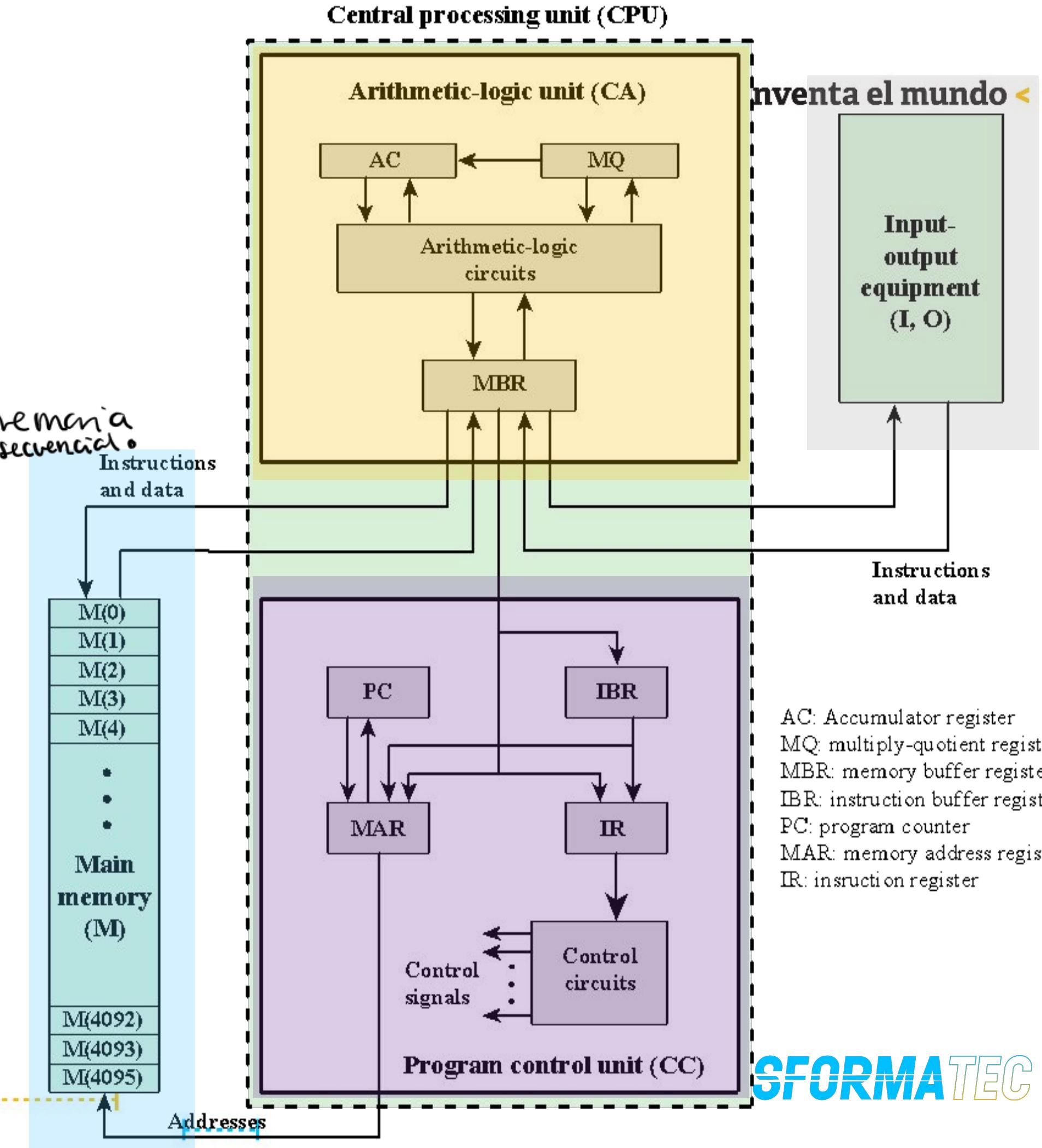
1. Stored program

- Instructions stored in a linear memory array
- Memory is unified between instructions and data.
- The interpretation of a stored value depends on the control signals

las instrucciones están guardadas en una memoria como lista secuencial.
la memoria está unificada entre instrucción e información.
todo está almacenada en la memoria y cada posición puede ser interpretada como dato o como instrucción.

2. Sequential instruction processing

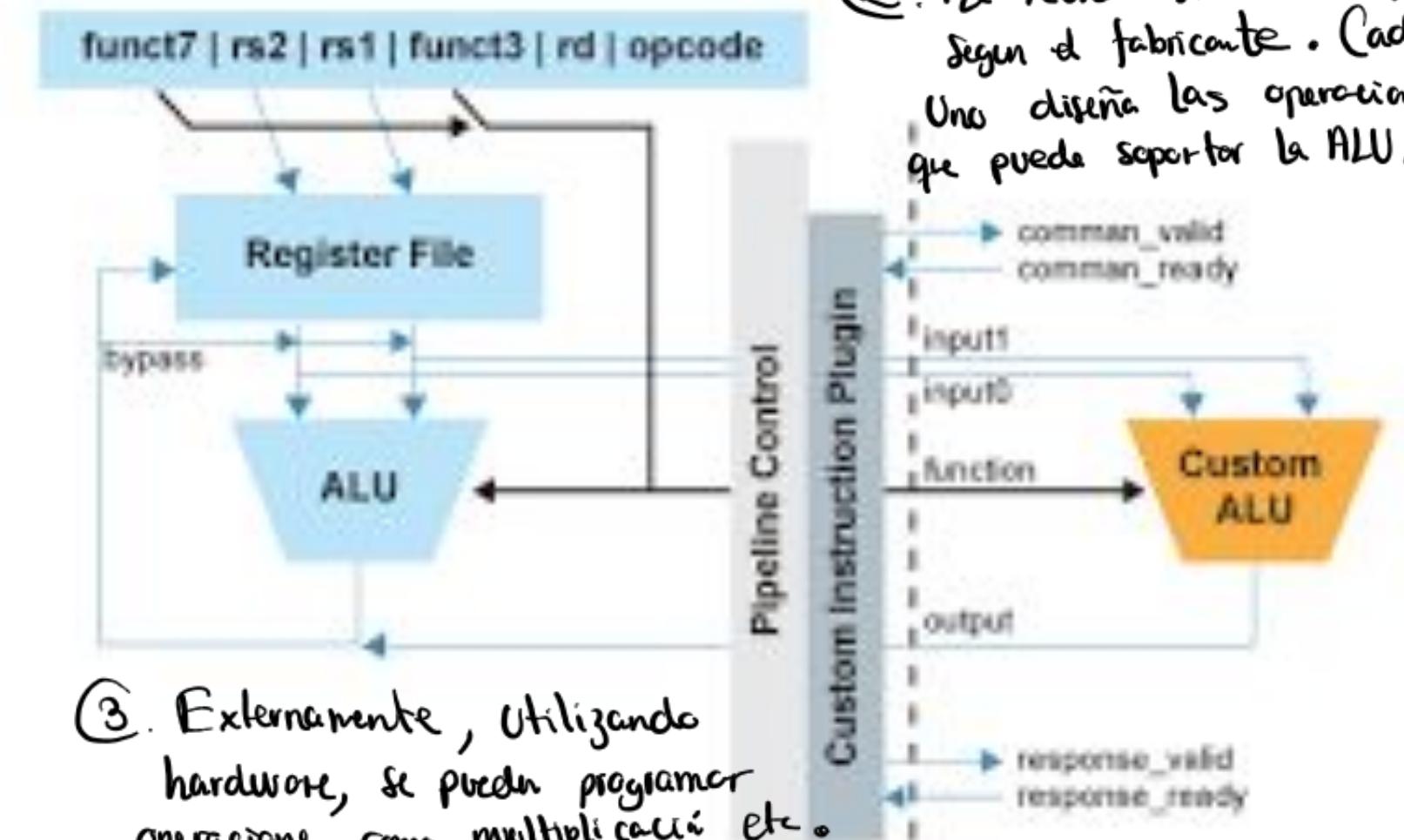
- One instruction processed (fetched, executed, completed) at a time
- Program counter (instruction pointer) identifies the current instruction → dice que instrucción toca ejecutar.
- Program counter is advanced sequentially except for control transfer instructions



Arithmetic Logic Unit

- ① Low - level arithmetic and logical operations featured by the computer.
- ② ALU functionality: Manufacturer design.
- ③
 - HW Implementations: Efficient but more expensive hardware.
 - SW Implementations: Inexpensive and less efficient.

④ Es barato implementarlo pero se vuelve muy lento ya que para crear una operación como multiplicación combinas varias sumas y restas.



- ① La ALU en su mayoría trabaja con operaciones de bajo nivel (Sumas, Restas, corrimiento de bits).
- ② De hecho esto varía según el fabricante. Cada uno diseña las operaciones que puede soportar la ALU.
- ③ Externamente, utilizando hardware, se pueden programar operaciones como multiplicación etc.
 - Esto hace que el ALU no trabaje en estos cálculos pero sea más eficiente.
 - Esto es muy costoso.

Reference Books

- ➔ Patterson, D. A., & Hennessy, J. L. (2020). Computer Organization and Design RISC-V Edition: The Hardware Software Interface. Morgan Kaufmann
- ➔ "The elements of computing systems: building a modern computer from first principles" Nisan, N., & Schocken, S. (2021). MIT press
- ➔ Silberschatz, A., Gagne, G., & Galvin, P. B. (2015). Operating system concepts (9th edition, international student version). John Wiley & Sons Inc.
- ➔ "Digital Design and Computer Architecture, RISC-V Edition". Morgan Kaufmann. Harris, S., & Harris, D. (2021).

Questions?