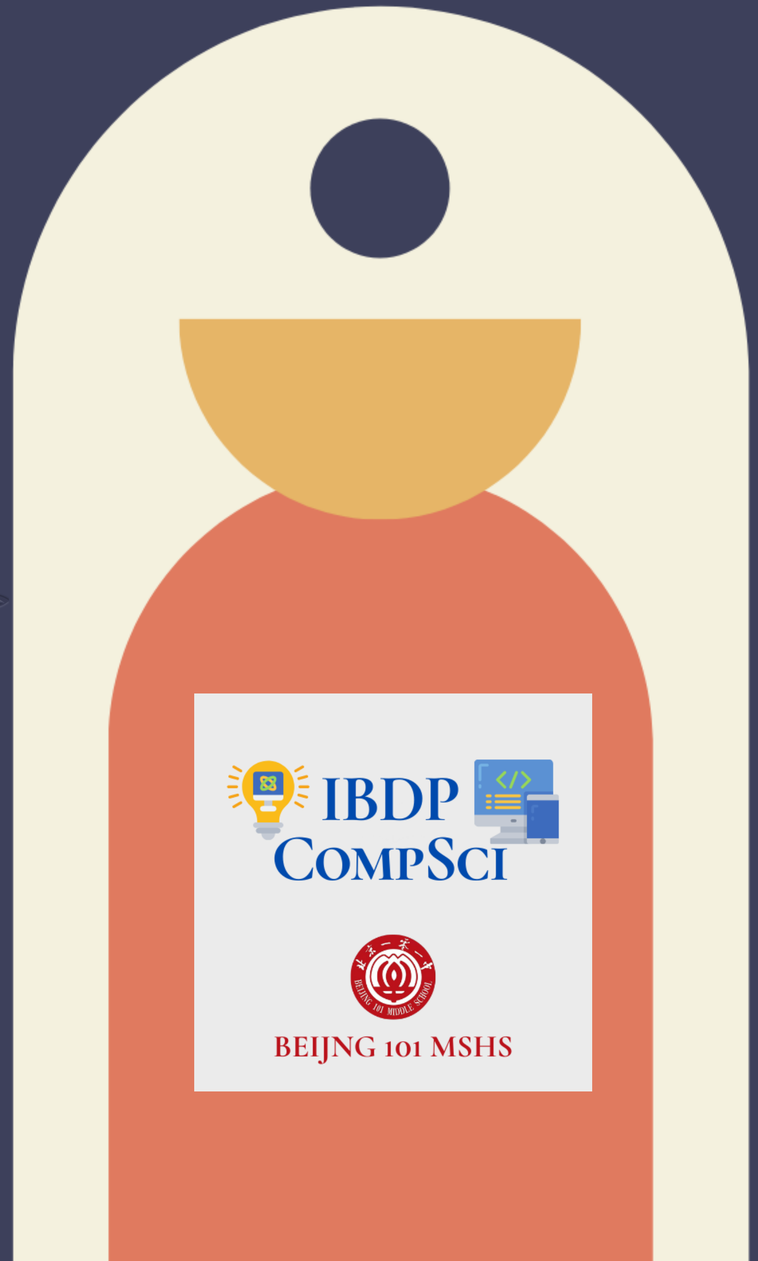
Welcome to Computer Science ***IBDP***

Beijing 101 Middle/High School



Highlights from Last time

♥ Logical Rule for a real world.

♥ Topic 2-Computer Architecture

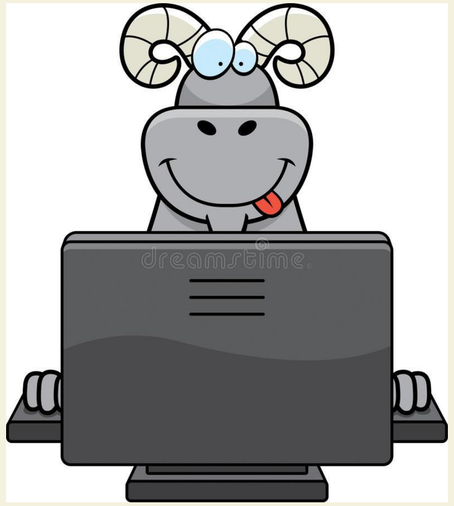
Today

- Ram & Rom
- Cache Memory
- Machine Instruction cycle

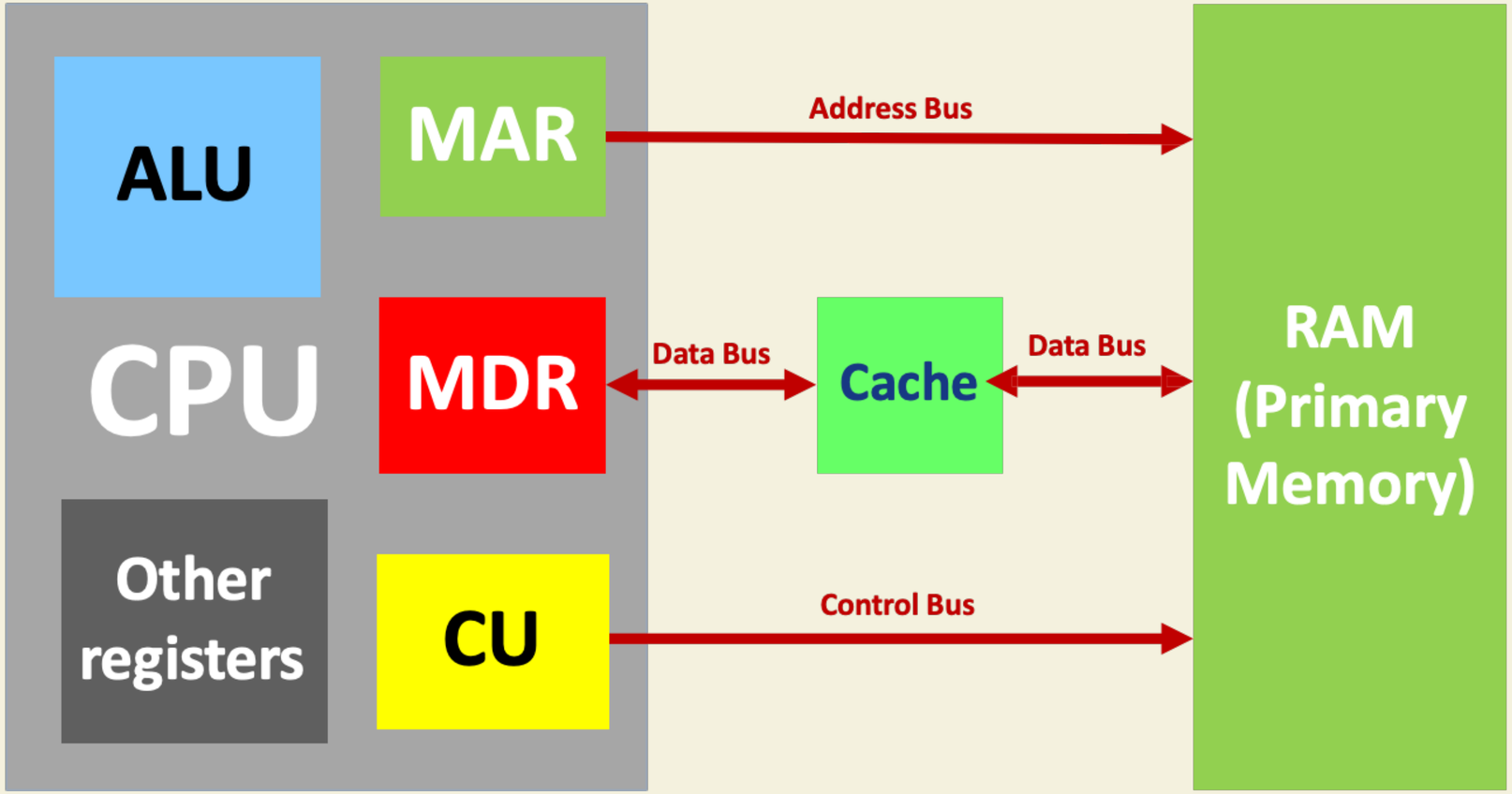
Task-Managebac Submission

- Describe the function of the data bus found in a PC
- Outline the function of the ALU (Arithmetic Logic Unit)

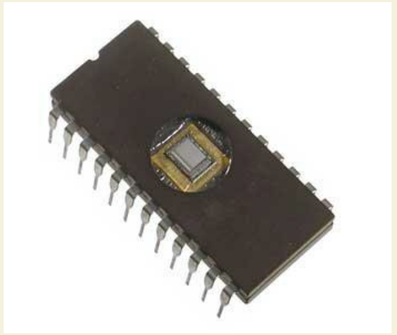
Topic 2.1.2
Describe primary memory.

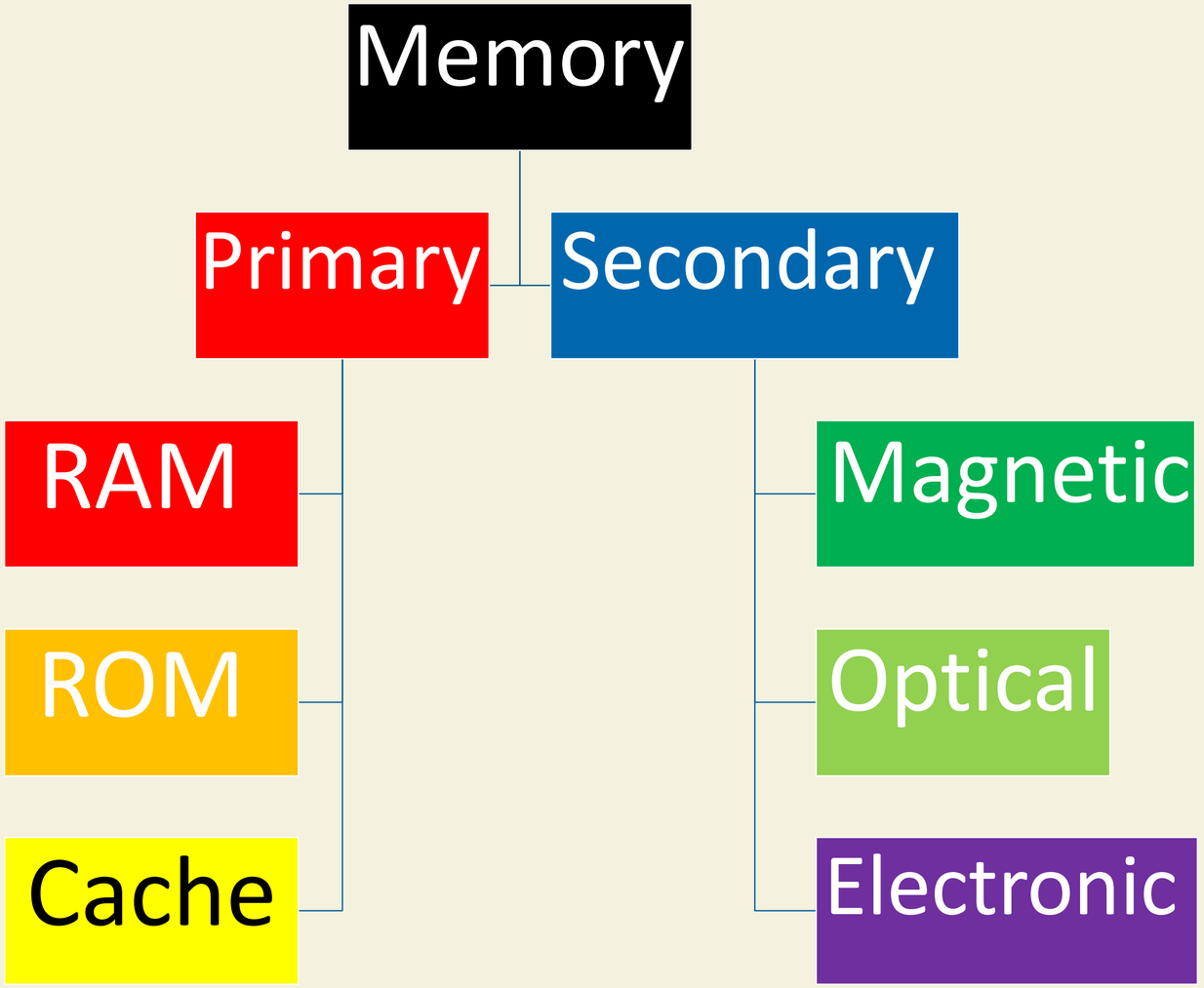


Simplified model: CPU, RAM



Simplified model: CPU, RAM





Primary memory = RAM

♥ As **RAM** is so important, it is often referred to as **primary memory** (even though it is actually only a branch of primary memory, alongside the cache and ROM).

In an exam/test, if you see *memory*, unless explicitly stated
otherwise, it would normally be referring to **RAM**.

RAM = Random Access Memory

- Contains the **data** and **instructions** the computer has loaded since starting up and everything the user has opened/loaded.
- Is **volatile** = loses its contents if power is lost
- Has a special link to the CPU (via busses)

In an exam/test, if you see *memory*, unless explicitly stated
otherwise, it would normally be referring to **RAM**.

ROM = Read Only Memory

- Originally its contents were static (hence ‘read only’) and could not be changed – not true any more (flash upgrades).

- Non-volatile = does not lose its contents if power is lost

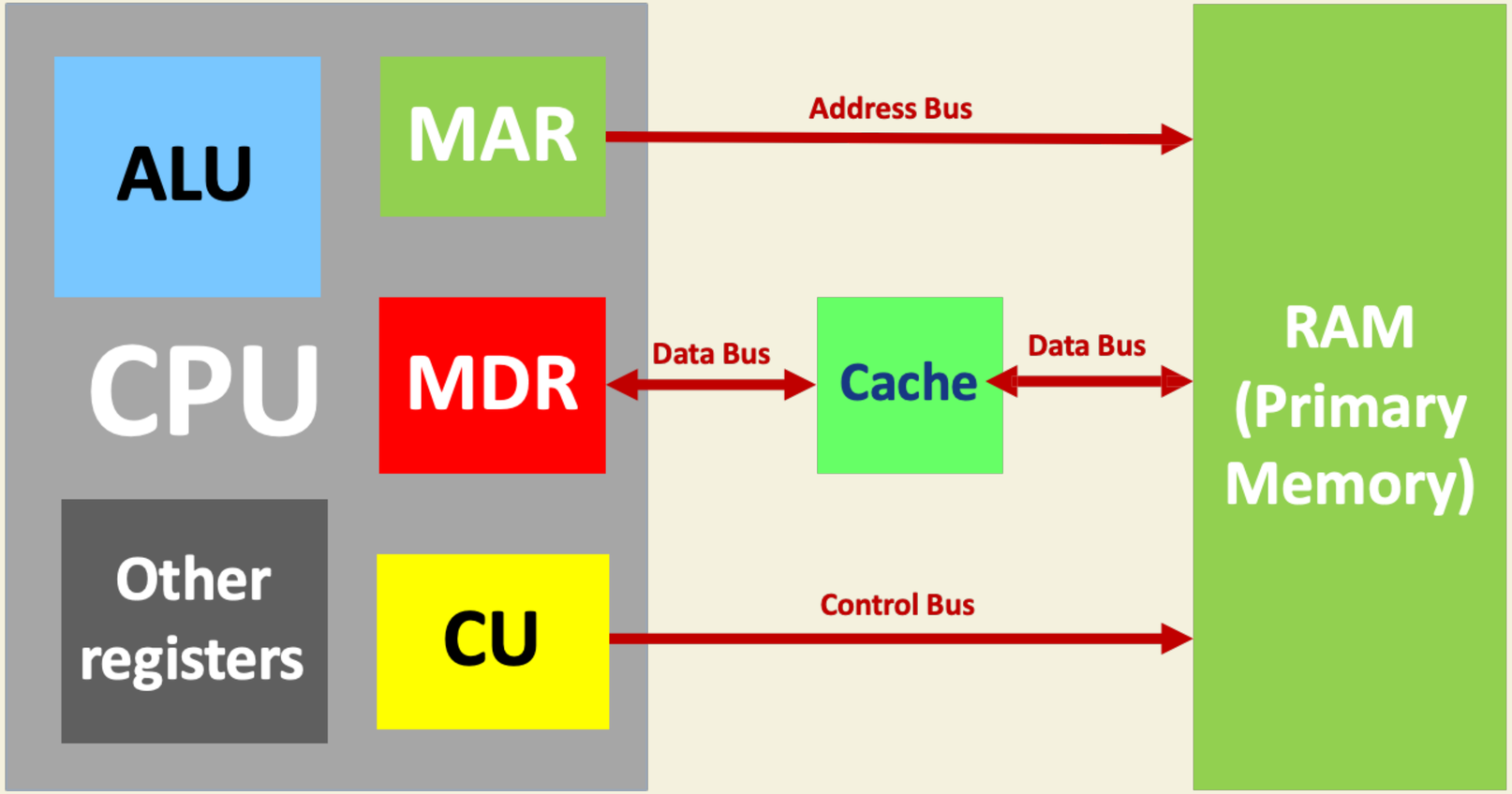
- Stores the BIOS (Basic Input Output System) – a small program that allows the computer to know what to do to find the operating system to ‘boot’ the computer after power is restored.

RAM - ROM

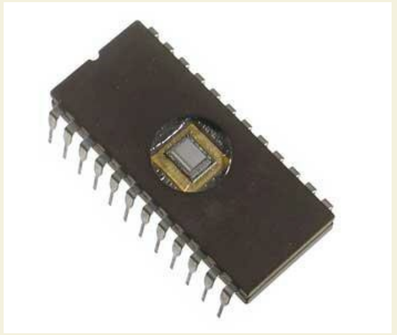
RAM	ROM
Volatile	Non-volatile
Contains user’s programs and data that has been loaded since ‘booting up’	Contains the BIOS
Usually upgradeable, can be increased	Usually part of motherboard, difficult to upgrade

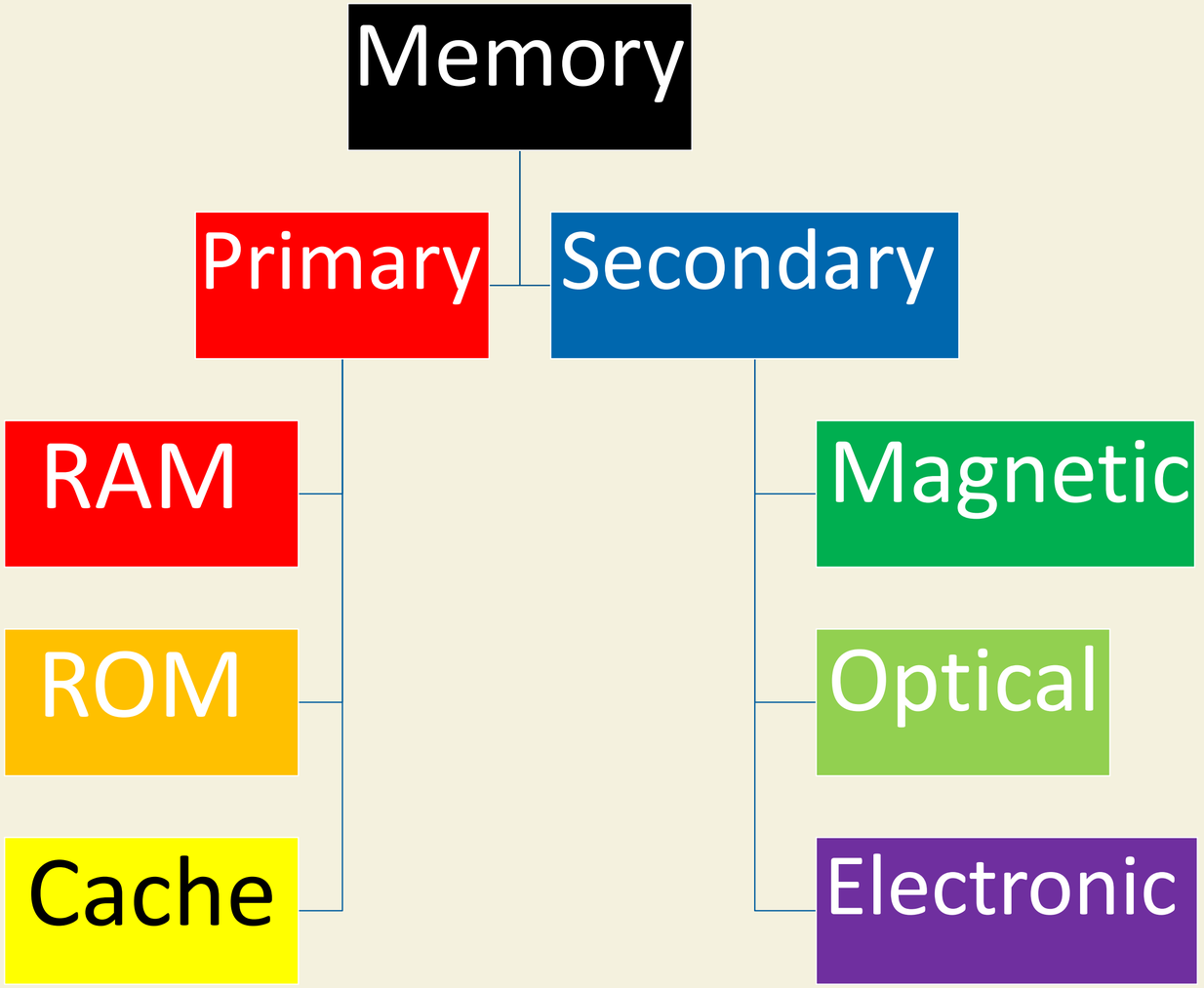
Topic 2.1.3
Explain the use of cache memory.

Simplified model: CPU, RAM



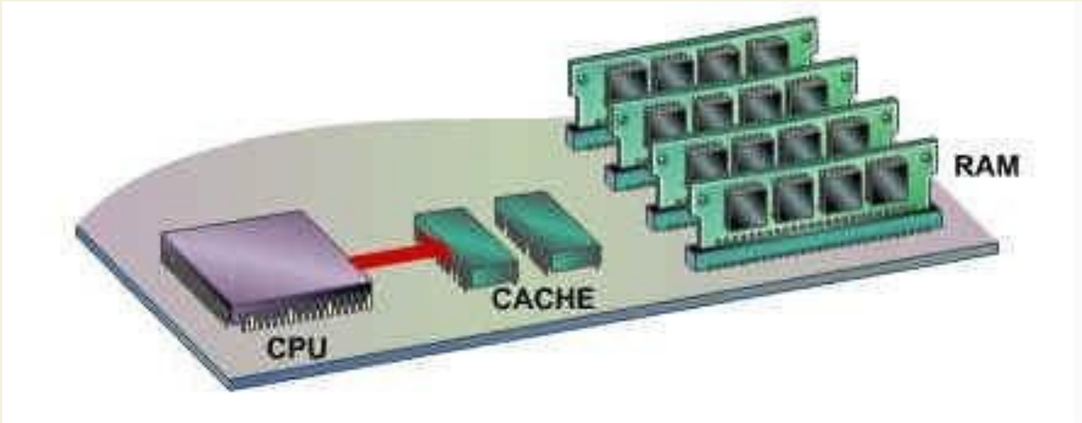
Simplified model: CPU, RAM





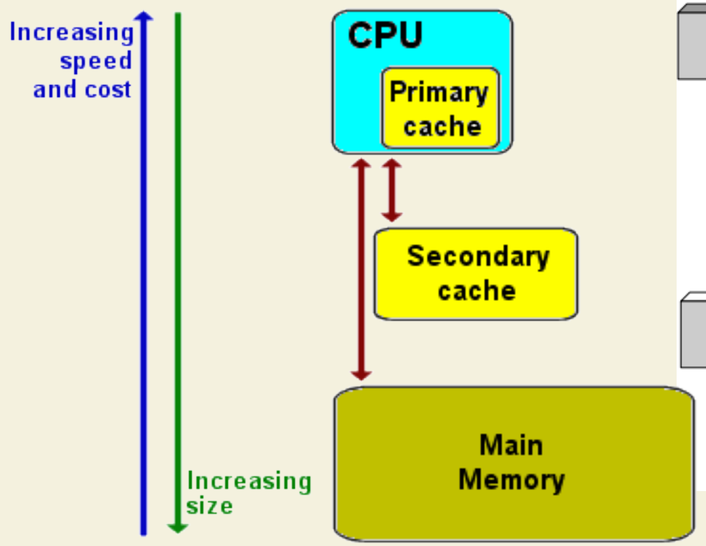
Definition: cache

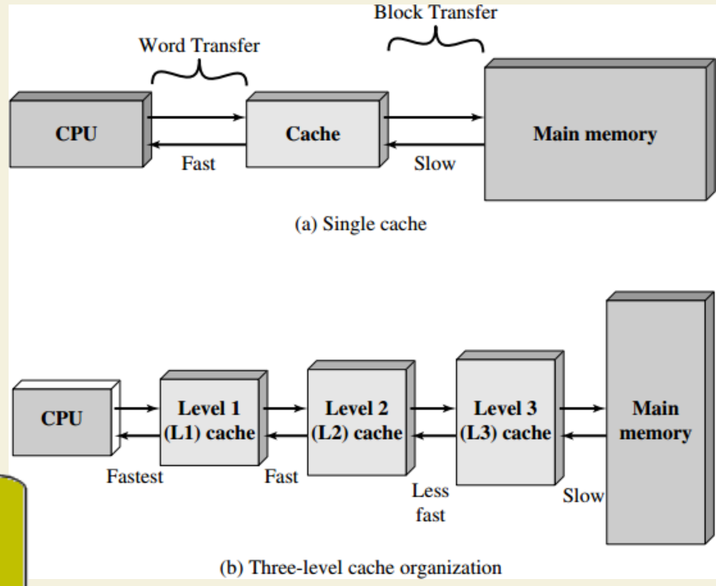
A type of **small**, **high-speed** memory **inside** the CPU used to hold **frequently used data**, so that the CPU needs to access the much slower RAM less frequently



Cache levels*

*Although not examined, it is good to know that cache actually exists in levels/stages in modern computers

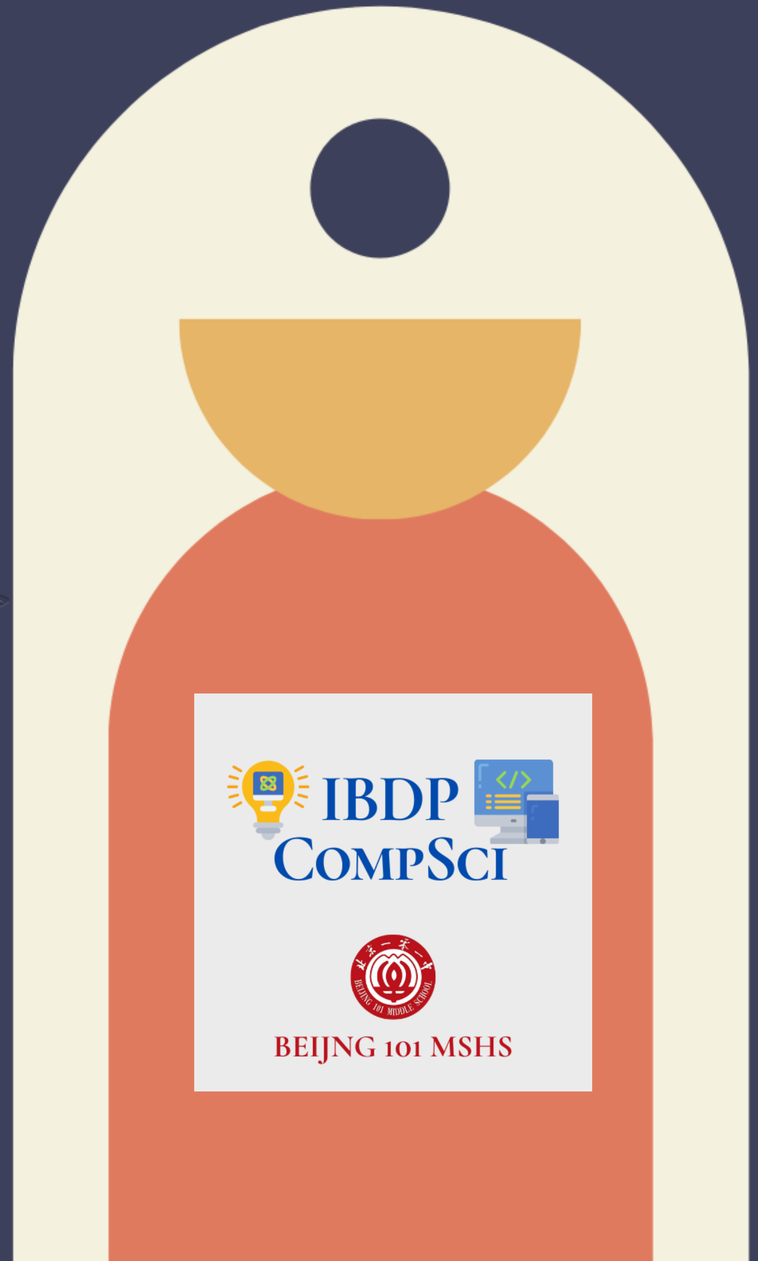




How Does Cache Work?

Welcome to Computer Science ***IBDP***

Beijing 101 Middle/High School



Highlights from Last time

- Ram & Rom
- Cache Memory

Today

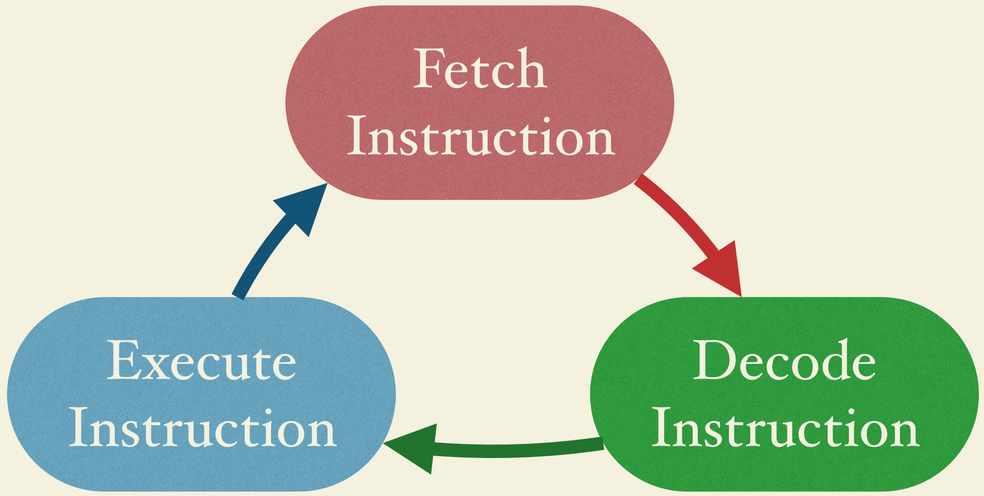
- Machine Instruction cycle

Task-Managebac Submission

♥ Based on your understanding, design a block diagram of a CPU and describe the flow of activities happening.

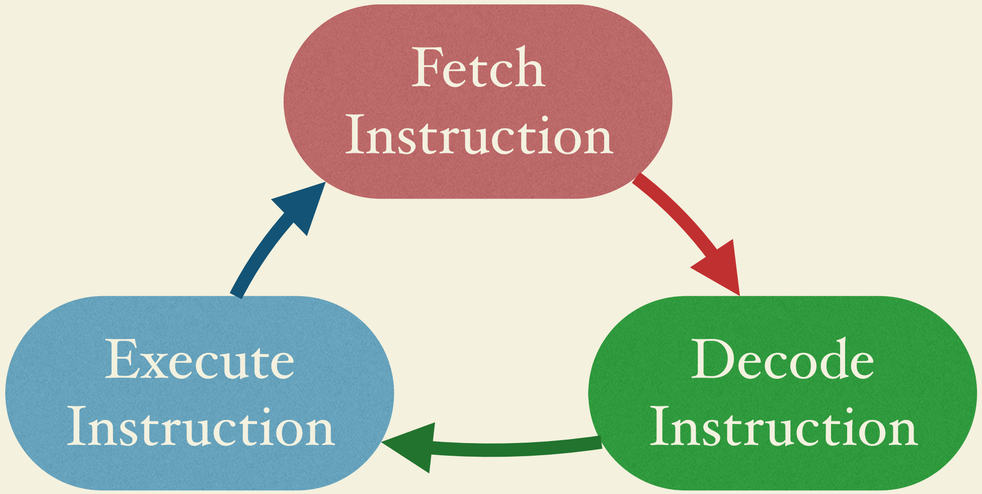
Submit on Managebac when done.

Topic 2.1.4
Explain the machine instruction cycle



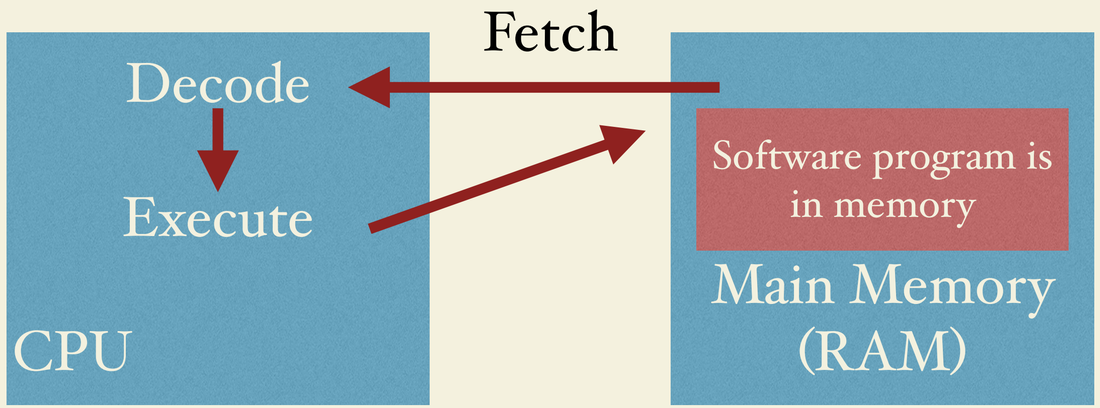
The Fetch-Execute cycle

- The basic operation of a computer is called the ‘fetch-execute’ cycle (also called the ‘machine cycle’).
- The computer fetches the instruction from its memory and then executes it.
- This is done repeatedly from when the computer is booted up to when it is shut down.



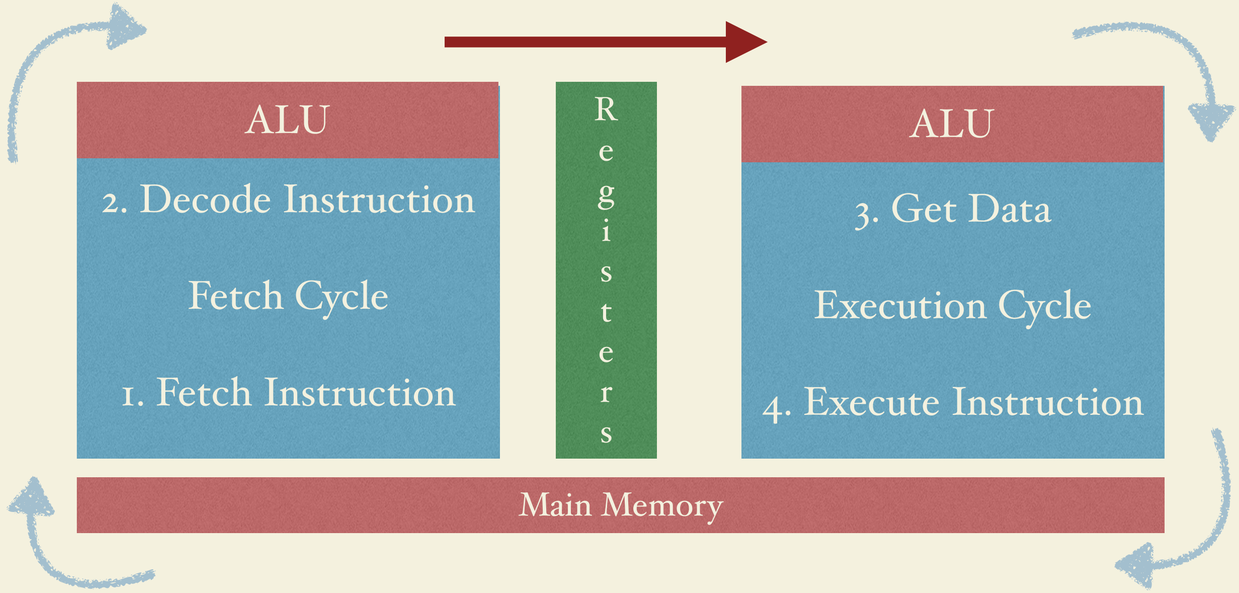
Step 1: Fetching the instruction

- The first step the fetch-execute cycle carries out is fetching the instruction.
- The CPU fetches this from the main memory (RAM) and stores it in the CPU temporary memory, called the registers.



Step 2: Decoding the instruction

- Once the instruction has been fetched, the CPU will need to understand the instruction to action it.
- This is called decoding.



Step 3: Executing the instruction

- When the instruction has been decoded, the CPU can carry out the action that is needed.
- This is called executing the instruction. The CPU is designed to understand a set of instructions - the instruction set.

Instruction Number			
Binary	Hex	Instruction	Meaning
0001	1	Load X	Load contents of address X into AC.
0010	2	Store X	Store the contents of AC at address X.
0011	3	Add X	Add the contents of address X to AC.
0100	4	Subt X	Subtract the contents of address X from AC.
0101	5	Input	Input a value from the keyboard into AC.
0110	6	Output	Output the value in AC to the display.
0111	7	Halt	Terminate program.
1000	8	Skipcond	Skip next instruction on condition.
1001	9	Jump X	Load the value of X into PC.

Example

A single piece of program code might require several
instructions. Look at this Java code:

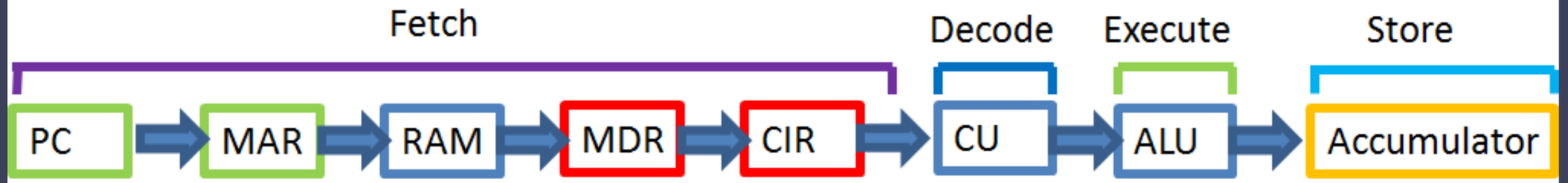
```
area = length * width
```

First, the computer needs to load in the value of the variable
`length` into the immediate access store (registers).
Next it needs to load in the value of the variable **`width`**.

Then it needs to multiply the two numbers together, and finally it needs to store the result in the variable **`area`**.

Simplified model

Stages of the Machine Instruction Cycle



The address of the next instruction is copied onto the MAR from PC

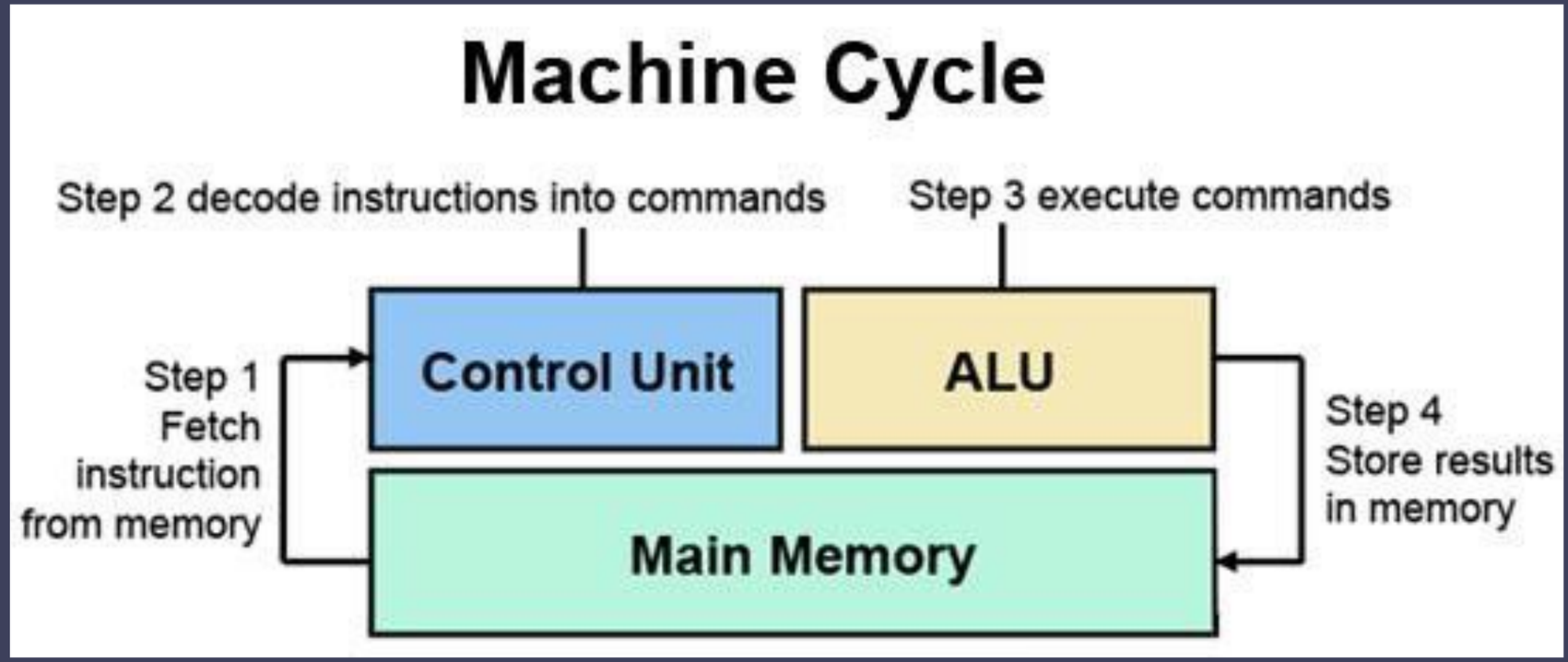
The Data is fetched using the address from the MAR and sent to the MDR

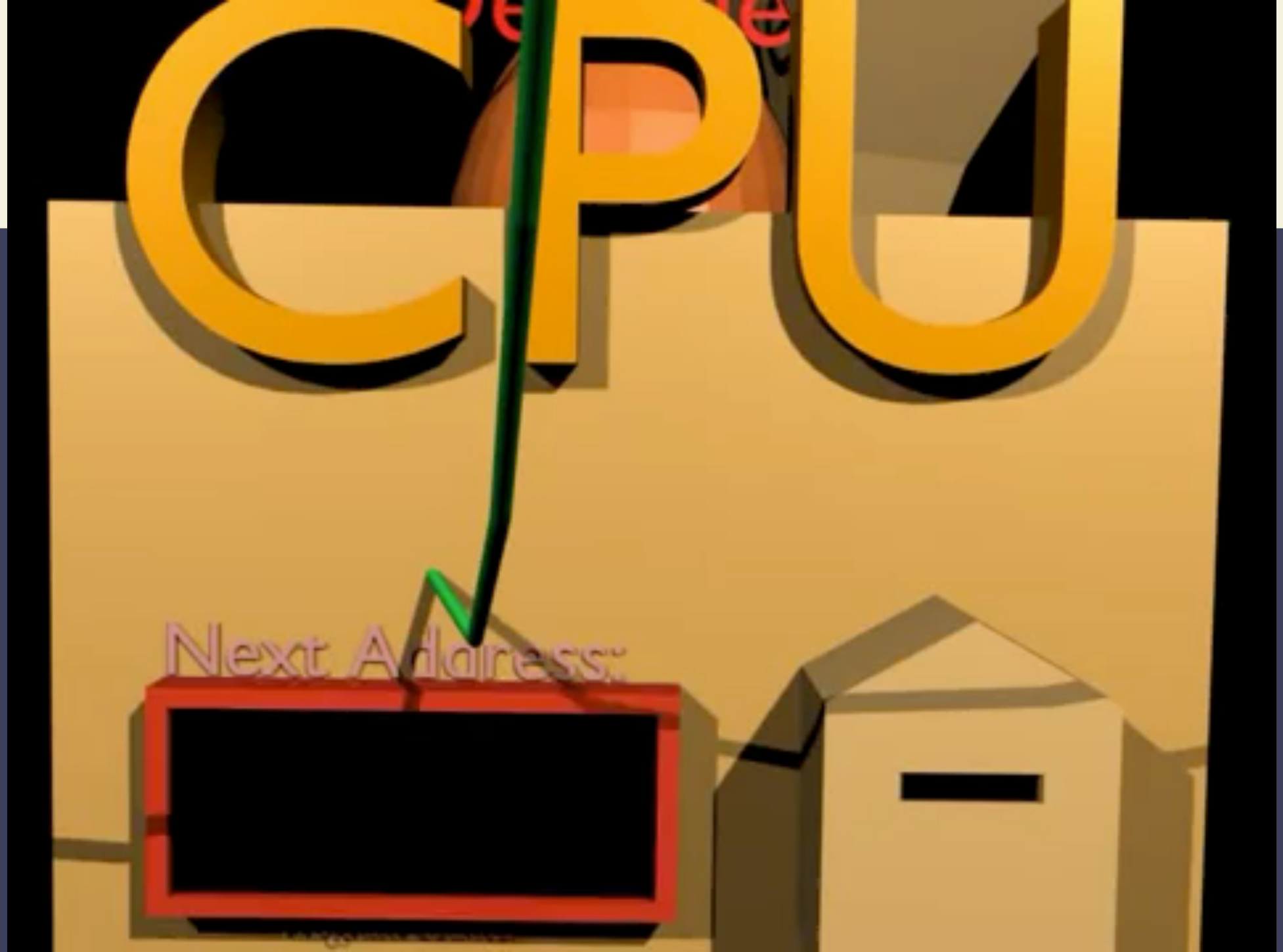
The instruction is stored in the MDR which is then copied into the CIR

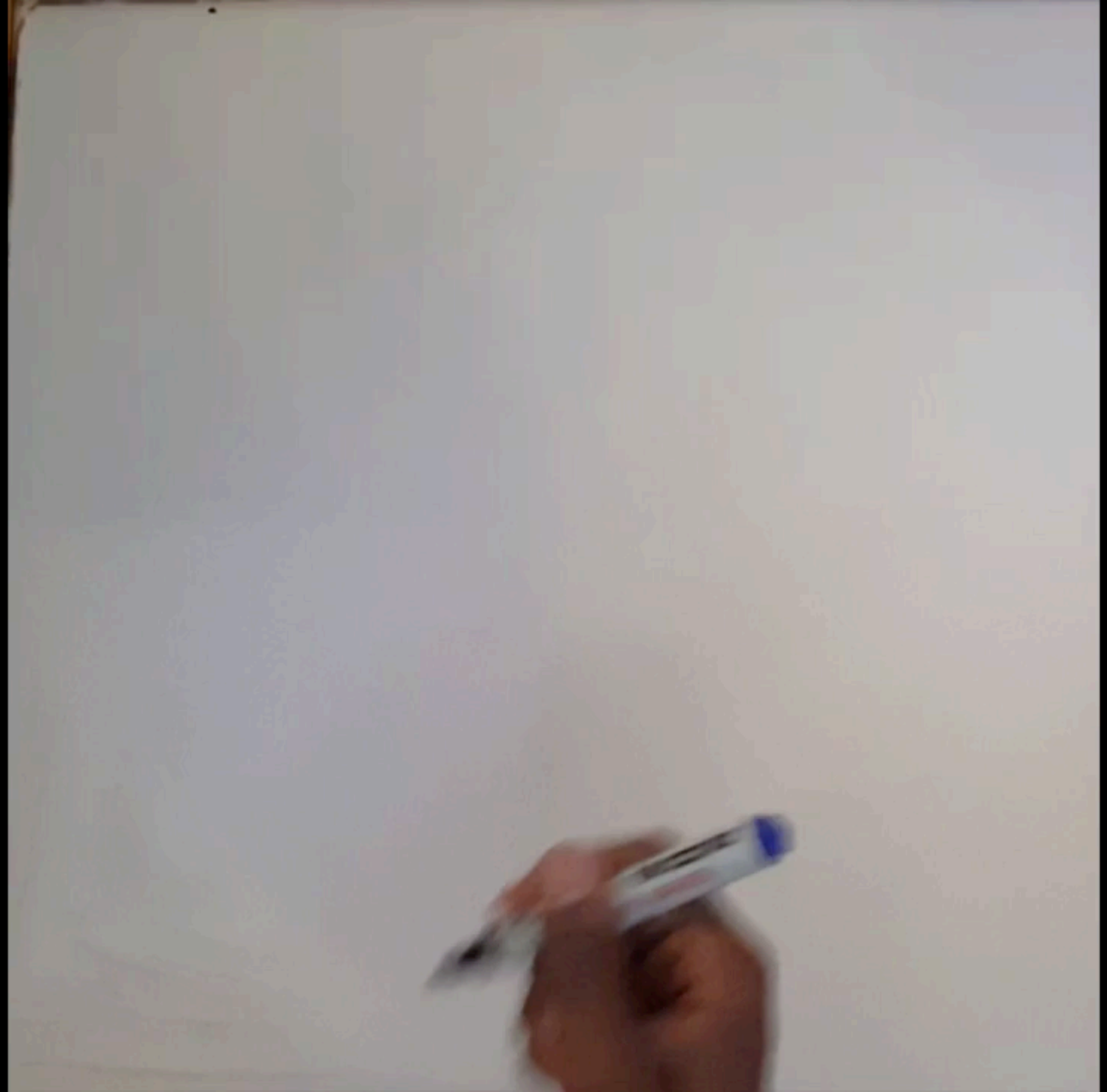
The instructions are then sent to the CU which makes changes to the ALU. The ALU then executes the instruction and stores them in the Accumulator if needed.

The Accumulator stores the results of the executed instruction.

Simplified model 2



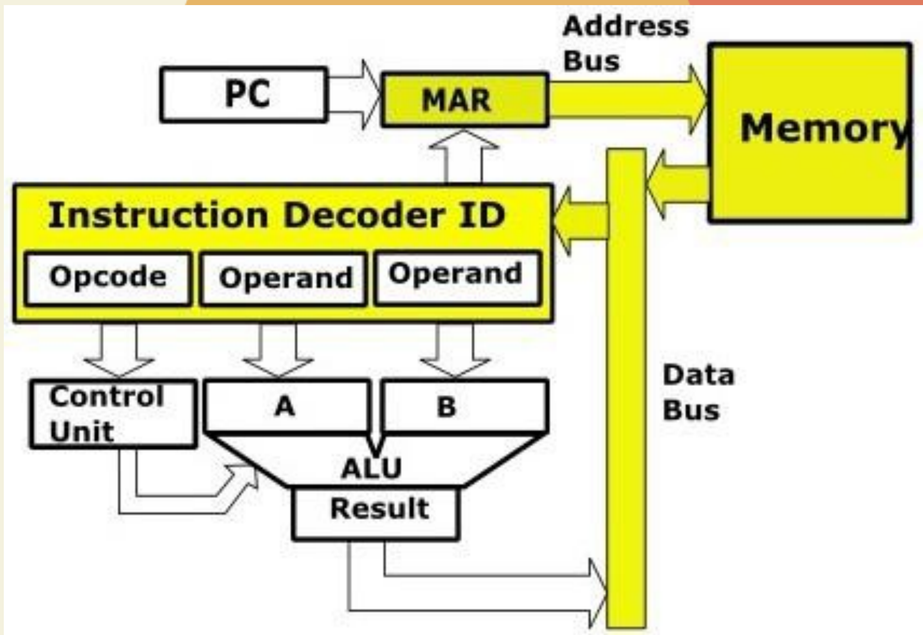

CPU
Next Address:



Exam note!

- This curriculum point requires you to describe the role of the data and address busses in the cycle.

- Think about what information they carry, from where to where and what they connect to at each end.



Thank you and see you next time.