



## CPU & Fetch-Decode-Execute Cycle

Revision notes by **FutureLogic Education** covering the CPU, its components, the Fetch-Decode-Execute cycle, and key registers for IGCSE Computer Science.

### What is the CPU?

The **CPU (Central Processing Unit)** is the brain of the computer. It processes instructions, performs calculations, and controls all operations.

**KEY FACT:** The CPU processes every instruction the computer executes.

### Main Parts of the CPU

Component	Role
<b>Control Unit (CU)</b>	Manages the flow of data. Directs and controls the Fetch-Decode-Execute cycle.
<b>Arithmetic Logic Unit (ALU)</b>	Performs arithmetic calculations (add, subtract) and logical operations (comparisons).
<b>Registers</b>	Small, extremely fast storage locations inside the CPU used during processing.
<b>Cache Memory</b>	Stores frequently used data and instructions so the CPU can access them faster than RAM.

*Teacher note: Students often confuse the CU and ALU. A simple rule: CU = controller, ALU = calculator.*

### The Fetch-Decode-Execute Cycle





The FDE cycle is the process the CPU repeats continuously to process instructions. It happens millions of times per second.

<b>1 FETCH</b>	The next instruction is fetched from RAM using the address stored in the <b>Program Counter (PC)</b> . It is placed into the <b>Current Instruction Register (CIR)</b> . The PC is then incremented to point to the next instruction.
<b>2 DECODE</b>	The <b>Control Unit</b> decodes the instruction in the CIR to determine what operation needs to be carried out and which data is needed.
<b>3 EXECUTE</b>	The decoded instruction is carried out by the relevant part of the CPU — either the <b>ALU</b> (for calculations) or another component. Results are stored in registers.

**EXAM TIP:** Always describe the FDE cycle in order: **Fetch** → **Decode** → **Execute**. Missing the **Decode** stage is the most common exam error.

*Teacher note: Ask students to recite the three stages from memory before describing them. Order matters — examiners expect the stages in sequence.*

## Important Registers

Register	Abbreviation	What it stores
<b>Program Counter</b>	PC	Address of the <b>next</b> instruction to be fetched.
<b>Memory Address Register</b>	MAR	Address of the memory location being read from or written to.
<b>Memory Data Register</b>	MDR	Data that has just been fetched from, or is about to be written to, memory.
<b>Current Instruction Register</b>	CIR	The instruction currently being decoded and executed.





<b>Accumulator</b>	ACC	Stores the result of calculations performed by the ALU.
--------------------	-----	---

**EXAM TIP: MAR holds an ADDRESS. MDR holds DATA. Mix these up and you lose marks instantly.**

*Teacher note: Memory hook: MAR = Address (both have "A"). MDR = Data (both have "D"). Drill this distinction — it appears almost every year.*





## Cache Memory & Clock Speed

Cache Memory	Clock Speed
Stores <b>frequently used</b> data and instructions close to the CPU.	Measured in <b>GHz</b> (gigahertz).
Faster to access than RAM — reduces time the CPU spends waiting.	Controls how many FDE cycles occur per second.
More cache = faster performance.	Higher clock speed = more instructions processed per second.

**EXAM TIP:** Cache improves performance because frequently used data is accessed faster than RAM, therefore the CPU spends less time waiting.

*Teacher note: Cache and clock speed are the two most common "explain why performance improves" questions. Practise the Point → Because → Effect formula for both.*

## Common Exam Mistakes

**X Mixing up MAR and MDR.**

MAR = Address. MDR = Data. Learn these precisely.

**X Confusing the CU and ALU.**

CU = controls the cycle. ALU = performs calculations. Different jobs.

**X Forgetting the Program Counter.**

The PC stores the address of the NEXT instruction — not the current one.

**X Missing the Decode stage.**

Students often jump from Fetch straight to Execute. Decode is always the middle step.





**X Vague FDE descriptions.**

Name the registers involved — PC, CIR, MAR, MDR. Naming them earns marks.

## Examiner Tips

**✓ Learn the role of each register precisely.**

Examiners expect accurate register names — not just "memory".

**✓ Name components in FDE answers.**

Say PC, CIR, MAR, MDR. Vague answers like "stored in memory" lose marks.

**✓ Explain each stage in order.**

Fetch → Decode → Execute. Never skip or reorder stages.

**✓ Use the explain formula for performance.**

"Cache improves performance because... therefore the CPU..."

## Quick Quiz

1.	What does the ALU do?
2.	Which register stores the address of the next instruction?
3.	Why is cache memory important for CPU performance?
4.	What happens during the Decode stage of the FDE cycle?
5.	What is the difference between MAR and MDR?

*Answers on the next page →*





## Quick Quiz — Answers

1.	The ALU performs arithmetic calculations (addition, subtraction) and logical operations such as comparisons.
2.	The Program Counter (PC) stores the address of the next instruction to be fetched.
3.	Cache stores frequently used data and instructions close to the CPU. Because it is faster to access than RAM, the CPU spends less time waiting, therefore performance improves.
4.	The Control Unit decodes the instruction in the CIR to determine what operation must be carried out and which data is needed.
5.	MAR (Memory Address Register) stores the memory address being accessed. MDR (Memory Data Register) stores the data that has been fetched from or is being written to that address.

## Final Summary

Topic	Key Point to Remember
<b>CPU</b>	Processes all instructions. Made up of CU, ALU, registers, and cache.
<b>CU</b>	Controls the FDE cycle and manages data flow.
<b>ALU</b>	Performs arithmetic and logical operations.
<b>PC</b>	Stores the address of the NEXT instruction.
<b>MAR</b>	Stores the memory ADDRESS being accessed.
<b>MDR</b>	Stores the DATA fetched from or being written to memory.
<b>CIR</b>	Stores the CURRENT instruction being decoded/executed.





<b>Cache</b>	Stores frequently used data for faster CPU access than RAM.
<b>Clock Speed</b>	Measured in GHz. Higher speed = more FDE cycles per second.
<b>FDE Cycle</b>	Fetch → Decode → Execute. Always in this order.

