



## F9.4 The CPU and fetch-execute

Logic and computer systems · GCSE · OCR J277 1.1.1, AQA 8525 3.4.5,  
Edexcel 1CP2 3.1.1 · about 20 min

---

Name \_\_\_\_\_

Class \_\_\_\_\_

Date \_\_\_\_\_

### What this lesson is about

---

The ALU, control unit, registers and buses, and tracing the fetch-decode-execute cycle.

- Inside the CPU
- Buses
- The fetch-decode-execute cycle
- Instructions are numbers too

### Questions

6 marks in all

1. Which register holds the address of the next instruction to fetch? [1 mark]

- ☐ A Program counter
- ☐ B Memory data register
- ☐ C Accumulator
- ☐ D Memory address register

2. What does the ALU do? [1 mark]

- ☐ A Carries out arithmetic and logic operations
- ☐ B Decodes instructions
- ☐ C Holds the address to read from
- ☐ D Sends the clock pulse

3. Put these steps of the cycle in order. [1 mark]

Number the lines 1 to 4 to put them in the right order.

- \_\_\_ The instruction is executed
- \_\_\_ The address in the PC is copied to the MAR
- \_\_\_ The instruction at that address is copied to the MDR
- \_\_\_ The control unit decodes the instruction

4. Which bus carries addresses from the CPU to memory? [1 mark]

- ☐ A Address bus
- ☐ B Data bus
- ☐ C Control bus
- ☐ D USB

5. What does the memory data register hold?

[1 mark]

- ☐ A The data or instruction just read from memory, or about to be written
- ☐ B The address of the next instruction
- ☐ C The result of the last calculation only
- ☐ D The clock speed

6. What does this program print?

[1 mark]

```
pc = 4
mar = pc
pc = pc + 1
print(mar, pc)
```

---

---

---

## The task: trace the cycle

Complete the fetch-decode-execute loop so it runs the program in `memory` and supports `LOAD`, `ADD`, `SUB`, `STORE`, `OUT` and `HALT`. During each fetch, print `fetch <MAR>: <instruction>` (for example `fetch 0: LOAD 6`). The program outputs `output: 35`.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

memory = ["LOAD 6", "SUB 7", "ADD 8", "STORE 9", "OUT 9", "HALT", 40, 12, 7, 0]
pc = mar = acc = 0
mdr = None
```

Plan your program here, then type it in and press Run.

---

---

---

---

---

---

---



Do it on the robot

[www.bugbotlab.com/learn/f9-4-the-cpu-and-fetch-execute/](http://www.bugbotlab.com/learn/f9-4-the-cpu-and-fetch-execute/)

The simulator checks it and tells you when it passes. Nothing to install, no account.

## Challenges

1. Add a `JUMPZ a` instruction that jumps to address `a` if the accumulator is 0. Write a countdown program with it.
2. Count how many fetch-execute cycles the program takes, and how many would run in one second at 1 GHz.

3. Encode each instruction as an 8-bit number, with a 4-bit opcode and a 4-bit address, and print the memory as binary.