Lesson Code: 1.1.1 **Lesson Title:** Architecture of the CPU **Year Group:** 10 - 11

LESSON OVERVIEW

Students explore how the Control Unit, ALU, registers and cache work together during the Fetch-Decode-Execute cycle. They apply accurate OCR terminology to interpret a CPU fault scenario, explain the evidence and make a justified technical recommendation.

LESSON OBJECTIVES

1. Identify the main components of the CPU.
2. Explain the purpose of the Control Unit, ALU, registers and cache.
3. Describe the Fetch-Decode-Execute cycle in the correct sequence.
4. Apply CPU knowledge to interpret evidence from a fault scenario.
5. Justify which CPU component is responsible for a fault using technical evidence.

KEY VOCABULARY

- Control Unit
- Arithmetic Logic Unit
- Registers
- Cache
- Fetch-Decode-Execute cycle

LESSON PLAN (60 mins)

TIME	STAGE	STUDENT ACTIVITY	RESOURCES
10 mins	Starter	CPU Component Charades: in teams, one student secretly receives a CPU component or cycle stage and acts out its function without speaking. The team earns one point for naming it and a second point for explaining its purpose accurately. Teams earn an additional point when they challenge an answer respectfully and replace it with stronger technical reasoning.	Printed component cards from Lesson Activities; timer; mini whiteboards
20mins	Activity 1	Using the lesson slides, the teacher introduces and explains the Control Unit, ALU, registers, cache and the Fetch-Decode-Execute cycle, modelling the reasoning students will need before guided practice. models the Fetch-Decode-Execute cycle using the Lesson Presentation, pausing after each stage for students to predict what happens next before revealing and explaining the answer. Students then complete a CPU detective carousel. At four stations they examine short clues about the Control Unit, ALU, registers and cache, then record the component, its function and the evidence.	Lesson Activities station cards; Student Workbook; Vocabulary Sheet
20 mins	Activity 2	Using the lesson slides, the teacher revisits the essential knowledge and demonstrates how it applies to diagnosing a processor fault. models how to analyse one item of evidence from the CPU fault scenario, thinking aloud to show how an incorrect component can be ruled out and how a justified recommendation is constructed. Students then complete the Stretch and Challenge fault investigation, analyse the failed decode stage, rule out components that are working.	Stretch and Challenge worksheet; Student Workbook; Knowledge Organiser
10 mins	Plenary	Human CPU Relay: groups arrange themselves as Fetch, Decode and Execute. A data card is passed through the cycle and each student must state what happens at their stage. Finish with one student explaining how the Control Unit coordinates the process. Students then.	Stage cards; instruction cards; mini whiteboards

ASSESSMENT OPPURTUNITIES

Formative (ongoing):

- Use hinge questions after each CPU component to check whether students can distinguish function from location.
- Listen for correct use of component names and cycle stages during the detective carousel.
- Check recommendations during the investigation for a clear claim, relevant evidence and technical reasoning.

Summative (end of lesson):

- Completed Student Workbook activities.
- Completed Stretch and Challenge investigation with a justified recommendation.
- OCR exam-style responses marked using the supplied answers.

HOMEWORK / FOLLOW UP

Complete the portal Homework task: annotate the Fetch-Decode-Execute cycle and answer the extended question explaining why the Control Unit is essential. Use the Knowledge Organiser and Vocabulary Sheet for support.

RESOURCES REQUIRED

Lesson Activities cards; Student Workbook; Vocabulary Sheet; Stretch and Challenge worksheet; Knowledge Organiser; mini whiteboards

REAL–WORLD EXAMPLES

A hospital diagnostic computer may appear to start normally while a fault in one CPU stage prevents applications from processing instructions correctly. Engineers use evidence from each stage to isolate the failed component.

TEACHER TIPS

Keep asking students to link each component to an action in the Fetch-Decode-Execute cycle. Correct vague statements such as 'the CPU does the work' by requiring the named component and precise function.

DIFFERENTIATION

Support

Provide labelled component cards, a partially completed CPU diagram, vocabulary prompts and sentence frames such as 'The evidence suggests... because...'.

Challenge

Remove the sentence frames and require students to compare two plausible components before rejecting one using evidence.

Stretch

Ask students to explain how the cycle would be affected by two simultaneous faults and to design one additional diagnostic test.

LESSON REFLECTION

NEXT LESSON

1.1.2 CPU Performance