Architecture of the CPU

1.1.1 · Computer systems (Paper 1) · revision notes

The CPU (central processing unit) runs programs by fetching, decoding and executing instructions. You need to know the job of each component (ALU, CU, cache and registers), what the four Von Neumann registers (PC, MAR, MDR and accumulator) hold, and what happens at each stage of the fetch–execute cycle. It comes up as recall questions, tables to complete and 3 to 4 mark 'describe' questions on Paper 1.

Key facts

- CPU purpose: fetch, decode and execute instructions
- ALU: arithmetic and logical operations (calculations and comparisons)
- CU: decodes instructions and sends control signals; controls the cycle
- Cache: small, very fast memory for frequently used instructions and data
- PC: address of the next instruction to be fetched
- MAR: address of the location to be read from or written to
- MDR: data or instruction fetched from, or to be written to, memory
- ACC: results of calculations from the ALU
- Von Neumann: instructions and data stored together in the same memory
- Fetch: PC → MAR, instruction from RAM → MDR, PC incremented

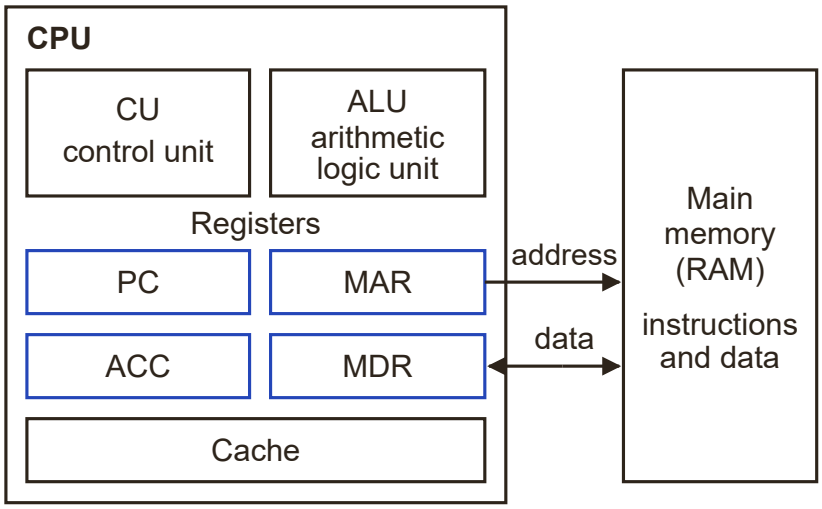
Notes

The CPU and its components

- The purpose of the **CPU** is to carry out the **fetch–execute cycle**: fetch an instruction from memory, decode it, execute it, then repeat.
- **ALU** (arithmetic logic unit): carries out calculations (e.g. addition and subtraction) and logical operations (e.g. comparisons such as 'is A greater than B?').
- **CU** (control unit): decodes instructions and sends control signals to coordinate the other parts of the computer, e.g. telling memory to read or the ALU to add. It controls the fetch–execute cycle.
- **Cache**: a small amount of very fast memory in or next to the CPU. It holds frequently used instructions and data, so the CPU waits for the slower RAM less often.
- **Registers**: tiny, extremely fast storage locations inside the CPU, each holding one item (an address, an instruction or data) that is in use right now.

Von Neumann architecture and its registers

- In the **Von Neumann architecture**, program instructions and data are both stored, in binary, in the same main memory (the stored program concept).

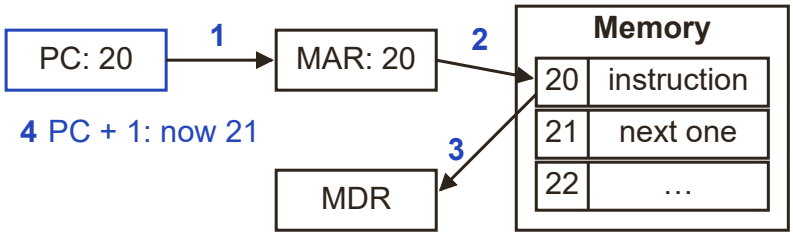


Von Neumann architecture: the CPU and one memory that holds both instructions and data

- **Program Counter (PC)**: holds the **address** of the next instruction to be fetched.
- **Memory Address Register (MAR)**: holds the **address** of the memory location that is about to be read from or written to.
- **Memory Data Register (MDR)**: holds the **data or instruction** just fetched from memory, or data waiting to be written to memory.
- **Accumulator (ACC)**: holds the **data** that results from calculations carried out by the ALU.
- An **address** is a location in memory; **data** is the value stored at that location.

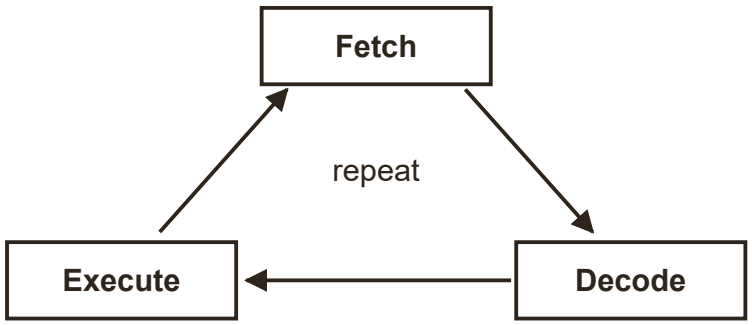
The fetch–execute cycle

- **Fetch**: the address held in the PC is copied into the MAR.
- The instruction stored at that address is copied from main memory (RAM) into the MDR.
- The PC is incremented (increased by 1), so it now holds the address of the next instruction.



Fetch: 1 PC copied to MAR, 2 memory location read, 3 instruction copied into MDR, 4 PC incremented

- **Decode**: the CU works out what the instruction means.
- **Execute**: the instruction is carried out, e.g. the ALU does a calculation and the result goes in the accumulator, or data is loaded from or stored in memory.



The cycle repeats for every instruction

How to answer each type of question

Identify a register or component (often a table to complete)

1 mark each | Grade 4

1. Learn each register and component as a name plus a one-line job.
2. Match the key words: 'address' → PC or MAR; 'data or instruction from memory' → MDR; 'result' → accumulator; 'decodes' → CU; 'calculations' → ALU.
3. Full names or the standard abbreviations are both accepted, but spell them correctly.

Example. A CPU has a Von Neumann architecture.
(a) Identify the register that holds the data that has just been read from memory. [1]
(b) Identify the register that is incremented during the fetch stage. [1]
(c) Identify the component of the CPU that compares two values. [1]

Model answer.

(a) Memory Data Register / MDR (1)
(b) Program Counter / PC (1)
(c) Arithmetic Logic Unit / ALU (1)

Explain the difference between two registers

2 marks | Grade 5

1. Say what the first register holds, and whether it is an address or data.
2. Say what the second register holds, and whether it is an address or data.
3. Make the contrast clear, e.g. with 'whereas'.

Example. Explain the difference between the contents of the Program Counter and the contents of the accumulator. [2]

Model answer. The Program Counter holds an address: the memory location of the next instruction to be fetched (1). The accumulator holds data: the result of a calculation carried out by the ALU (1).

Describe the fetch stage of the fetch–execute cycle

3 to 4 marks | Grade 6

1. Write the steps in order, one step per sentence.
2. Name the register used at each step (PC, MAR, MDR).
3. Say where the instruction comes from: main memory (RAM).
4. Finish with the PC being incremented.

Example. Describe how the CPU fetches an instruction during the fetch–execute cycle. [4]

Model answer. The Program Counter holds the address of the next instruction (1). This address is copied into the MAR (1). The instruction stored at that address is copied from RAM into the MDR (1). The Program Counter is incremented so it holds the address of the next instruction (1).

Don’t lose marks

- Saying the instruction is fetched from the hard drive. It is fetched from main memory (RAM).
- Mixing up the MAR and the MDR: the MAR holds an address, the MDR holds the data or instruction.
- Saying the PC 'holds the next instruction'. It holds the address of the next instruction.
- Saying the CU 'does the calculations'. The ALU calculates; the CU decodes and controls.
- Forgetting that the PC is incremented in a description of the fetch stage.
- Describing cache as 'storage for files'. It holds frequently used instructions and data while programs run.

More tips

Memory tricks

- Only two registers hold addresses: the PC (address of the next instruction) and the MAR (address being read from or written to now). The MDR and the accumulator hold data.
- Fetch in four: PC → MAR → MDR → PC + 1.
- The CU is the conductor of an orchestra: it plays nothing itself but tells every other part when to act. The ALU is the calculator.
- Speed order, fastest first: registers, cache, RAM.

Exam technique

- In 'describe the fetch stage' questions each correct step is usually worth a mark, so name the register at every step.
- When a question says 'address' or 'data', check which register fits: the difference is tested often.
- Use the exact register names (Program Counter, Memory Address Register, Memory Data Register, Accumulator). Vague names such as 'the memory register' do not score.
- For 'explain' questions about a component, give its job and its effect, e.g. 'cache holds frequently used instructions, so the CPU fetches from the slower RAM less often'.

What each grade needs

What you need to be able to do, from the first marks up to the top grade.

Grade 3 — **State the purpose of the CPU**
The CPU carries out the fetch–execute cycle: it fetches, decodes and executes instructions, over and over, to run programs.

Grade 4 — **Describe what the ALU and CU do**
The ALU carries out arithmetic and logical operations; the CU decodes instructions and sends control signals to coordinate the other components.

Grade 4 — **Name the four Von Neumann registers**
Program Counter (PC), Memory Address Register (MAR), Memory Data Register (MDR) and Accumulator (ACC).

Grade 5 — **Know which registers hold data or addresses**
The PC and MAR hold addresses; the MDR holds data or an instruction; the accumulator holds data (the results of calculations).

Grade 6 — **Describe the fetch stage in order**

The address in the PC is copied to the MAR, the instruction at that address is copied from RAM into the MDR, and the PC is incremented.

Grade 7

Explain each component's role during the cycle

Link each register and component to exactly what it stores, manages or controls at the fetch, decode and execute stages.

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.