

Exercise walk-through

3.1

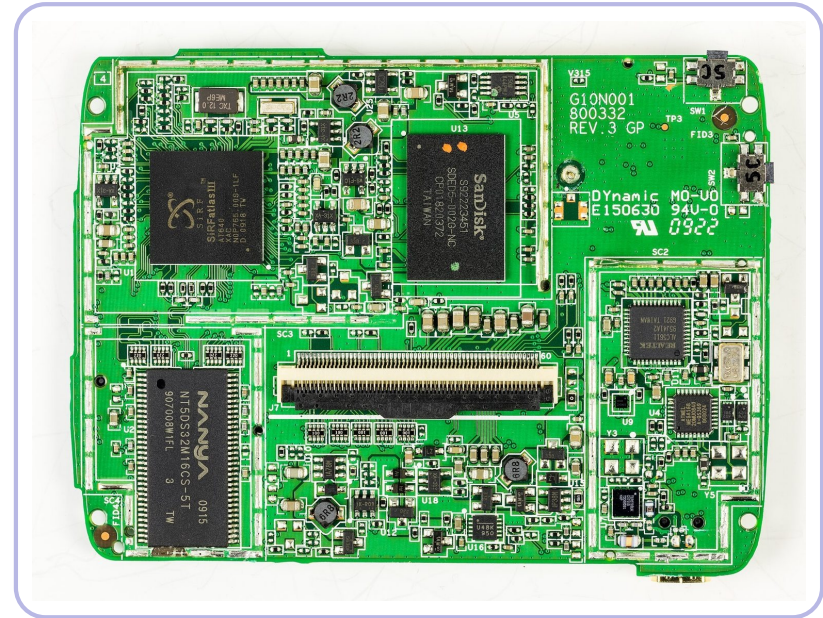
Computer architecture

IGCSE Computer Science

Embedded systems

An **embedded system** is a small computer built *inside* a larger device to do **one fixed job**.

- dedicated to that task, small and low-cost
- **no** general operating system
- a washing machine, a microwave, a set-top box



The mainboard from inside a satnav

Virtual memory, and the two kinds of address

**virtual mem-
ory**

when RAM fills, the computer uses part of **secondary storage** as extra, pretend RAM

the cost

secondary storage is far slower, so heavy use makes the machine crawl

**MAC ad-
dress MAC**

a **permanent** hardware address, set by the manufacturer, unique to the network card

IP address IP

assigned by the network, and it **changes** when the device moves

IPv4

32-bit, four denary numbers 0–255, for example 192.168.0.1

IPv6

128-bit, eight groups of hexadecimal – far more addresses, because IPv4 ran out

MAC identifies the **device**; IP identifies **where it is on the network**. A letter needs both the person and the address.

Exercises 2.1-2.3

2.1 State the function of the ALU.

[1]

2.2 Name the three stages of the fetch–execute cycle.

[3]

2.3 Name the register that holds the address of the next instruction.

[1]

Exercises 2.1-2.3

2.1 State the function of the ALU. [1]

2.1

- ALU stands for Arithmetic and Logic Unit
- it carries out **calculations and logical operations**

2.2 Name the three stages of the fetch–execute cycle. [3]

2.2

- the cycle has three named stages
- **fetch, decode, execute**

2.3 Name the register that holds the address of the next instruction. [1]

2.3

- the next instruction's address is held in the **program counter (PC)**

Exercises 2.4-2.5

2.4 Identify the bus that carries data in both directions, and the bus that carries addresses in one direction only. [2]

2.5 Explain how a larger cache improves the performance of a computer. [2]

Exercises 2.4-2.5

2.4 Identify the bus that carries data in both directions, and the bus that carries addresses in one direction only. [2]

2.4

- both directions: **data bus**
 - one direction (CPU → memory): **address bus**
-

2.5 Explain how a larger cache improves the performance of a computer. [2]

2.5

- cache is very fast memory holding frequently used instructions and data close to the CPU
- the CPU makes fewer slow fetches from RAM, so it **spends less time waiting**

Exercise 3.1

3.1 Cache memory speeds up the CPU because it:

[1]

☐

A stores data permanently

B is very fast memory holding frequently used data

C has no power needs

D is part of secondary storage

Exercise 3.1

3.1 Cache memory speeds up the CPU because it:

[1]

☐

A stores data permanently

B is very fast memory holding frequently used data

C has no power needs

D is part of secondary storage

3.1 B

- cache is **very fast memory holding frequently used data**
- A is ROM/storage; C is false; D is secondary storage

Exercise 3.2 (a)

3.2 A computer is upgraded.

(a) State two changes that would improve CPU performance.

[2]

Exercise 3.2 (a)

3.2 A computer is upgraded.

(a) State two changes that would improve CPU performance.

[2]

(a)

- any **two** of: **more cores** / **larger cache** / **higher clock speed**

Exercise 3.2 (b)

3.2 A computer is upgraded.

(b) State what is meant by an embedded system, with one example.

[2]

Exercise 3.2 (b)

3.2 A computer is upgraded.

(b) State what is meant by an embedded system, with one example.

[2]

(b)

- a small computer built into a device to do **one fixed job**
- e.g. a **washing machine** or **microwave**

Setup for exercise 3.3

3.3 A computer has a Von Neumann architecture with a single processor.

(a) Complete the table by giving the full name of each register and what it stores. [4]

Abbreviation	Full name	What it stores
PC		
MAR		
MDR		
ACC		

(b) Describe the fetch stage of the fetch–execute cycle, using the register names from part (a). [4]

(c) Identify the bus that would be used to send a “write” signal to memory, and explain why this bus is bidirectional. [2]

Exercise 3.3

(d) The computer is upgraded from two cores to four. Explain why this does not always make a program run twice as fast. [2]

(e) Give one example of an embedded system and describe what makes it different from a general-purpose computer. [2]

Exercise 3.3

(d) The computer is upgraded from two cores to four. Explain why this does not always make a program run twice as fast. [2]

(e) Give one example of an embedded system and describe what makes it different from a general-purpose computer. [2]

3.3

(a)

- **PC**: program counter – address of the next instruction

Full solution 3.3 (1/3)

3.3

(a)

- **PC**: program counter – address of the next instruction
- **MAR**: memory address register – address being read or written
- **MDR**: memory data register – data or instruction from that address
- **ACC**: accumulator – result of a calculation

Full solution 3.3 (2/3)

(b)

- address in the PC is copied to the MAR
- the PC is incremented
- the instruction at that address is fetched into the MDR along the data bus
- it is copied from the MDR into the CIR ready for decoding

(c)

- a “write” signal uses the **control bus**
- it is bidirectional because the CPU sends commands while devices send back signals

Full solution 3.3 (3/3)

such as an interrupt request

(d)

- not all programs can be split into parts that run at the same time
- cores still share memory and wait for each other, so extra cores are **not always fully used**

(e)

- e.g. a **washing-machine controller**, traffic lights, a microwave, or a car engine unit
- it is built into a larger device and runs **one dedicated program** the user cannot change