Chapter 1: Systems Architecture

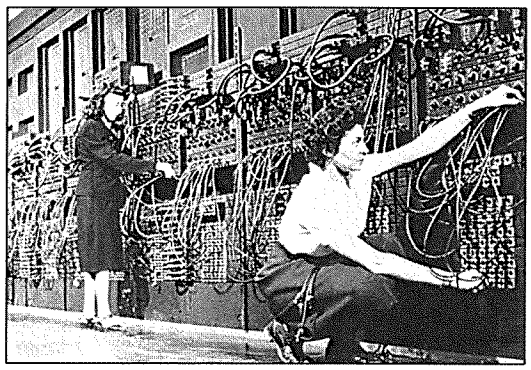
In this chapter you will learn:

- what is meant by a computer system and an embedded system
- what the CPU does
- what the different components are within the CPU
- what is meant by the fetch–execute cycle
- what can affect the performance of the CPU

What is a computer system?

A computer system is able to take in inputs, process them and output them; the output is sometimes stored. It does this using a variety of **hardware** and **software**. A computer system is comprised of the **CPU**, **memory** and other internal components, for example a **GPU**.

The first electronic general-purpose computer system was the Electronic Numerical Integrator and Computer (ENIAC); creation began in 1943 and was completed in 1946. ENIAC was designed by John Mauchly and J Presper Eckert at the University of Pennsylvania. ENIAC was able to solve large-scale numerical problems and was used by the US Army as part of their logistics.



ENIAC: the first electronic general-purpose computer

Computer systems have developed immensely since the invention of ENIAC and we now make them responsible for supporting many aspects of our daily lives.

Hardware – the physical parts of the computer

Software – the software and programs that are used by a computer

CPU – short for central processing unit; this is the main unit responsible for the processing instructions given to a computer

Memory – hardware used to store data

GPU – short for graphics processing unit; this is the main unit responsible for rendering images, animations and videos

Discussion point: Can you think of an activity in your daily life that you could not do without the use of a computer system? Do you think we have become reliant on computer systems?

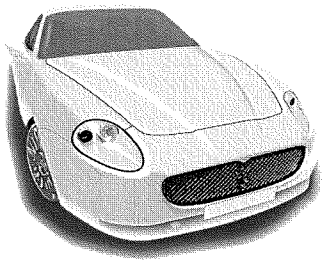
What is an embedded computer system?

OCR specification reference:

☑ embedded systems:
- purpose of embedded systems
- examples of embedded systems

An embedded computer system is normally built into a piece of machinery or device, hence the term 'embedded'. It is a computer system that is designed to provide very limited (normally one to two) and specific functions. It mostly consists of a small microprocessor that provides a similar function to a CPU; it takes in an input, processes it and provides an output as a result.

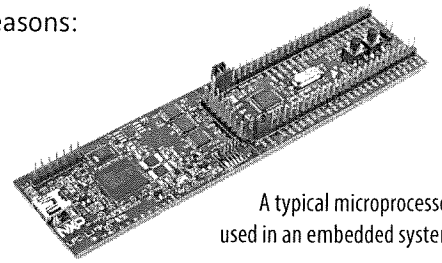
An embedded computer system does not look like a normal general-purpose computer; it does not have a keyboard, a mouse or a monitor. Embedded computer systems are built into many of the devices we use on a daily basis, such as a washing machine, a smoke detector or air conditioning. They are also used in many industries, for example medicine, in electronic devices such as blood pressure monitors and insulin pumps. Embedded computer systems can also be found in cars – in fact, they can have up to 50 embedded computers each performing a different task!



Discussion point: What embedded systems can you think of that exist in a car?

Manufacturers use embedded systems in their devices for a number of reasons:

- They allow the device to be programmed to automatically perform one or a small number of functions.
- They can reduce the size of the machinery or device. If a larger processor performing many functions is used, it would mean the device may need to be larger in size to accommodate this.



A typical microprocessor used in an embedded system

- They can reduce the cost of the machinery or device. If a larger processor performing many functions is used, most of the functionality would be wasted as only a small number of tasks are required. Therefore, by removing this wastage, the cost of manufacture can be reduced.

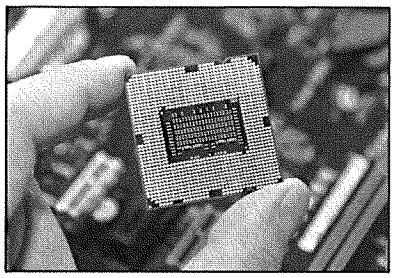
A computer system that can perform many different functions is called a general-purpose computer. This type of computer has a central 'brain' to allow it to perform all the different functions, which is called the CPU.

Discussion point: Is a smartphone an embedded system? Why?

What is the purpose of the CPU?

OCR specification reference:

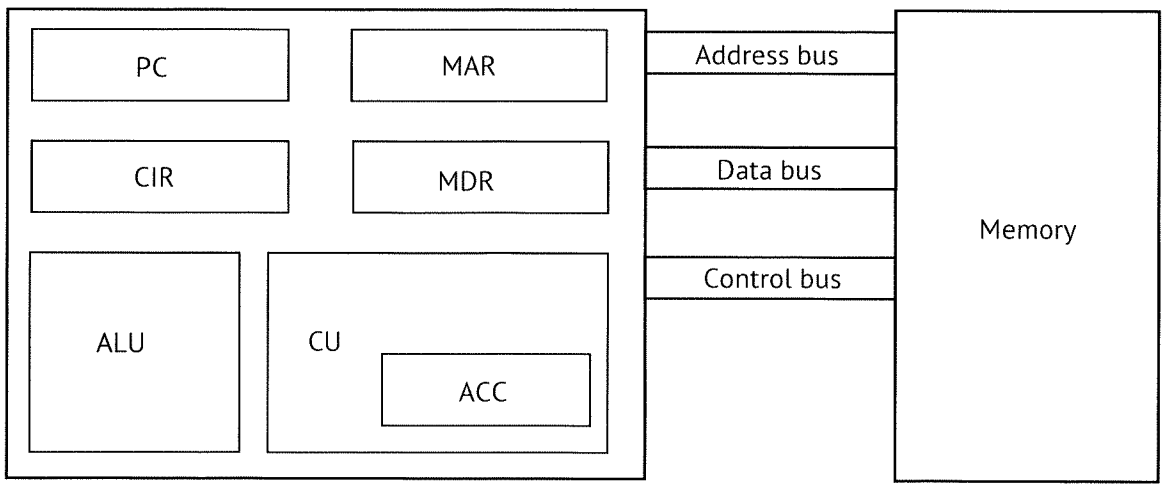
- ☑ the purpose of the CPU
- ☑ Von Neumann architecture:
  - MAR (Memory Address Register)
  - MDR (Memory Data Register)
  - Program Counter
  - Accumulator
- ☑ common CPU components and their function:
  - ALU (Arithmetic Logic Unit)
  - CU (Control Unit)
  - Cache



The purpose of the CPU is to process data and instructions. It is where the decision making takes place in the computer. Whatever you are using your computer for at any given time, be it playing computer games or doing your homework, the CPU is working tirelessly to allow you to do this. Data and instructions are transported around the CPU through the use of **buses**.

Bus – a channel of communication used to transport data and instructions in the CPU

The CPU is made up of a number of component parts that you need to know. There is a particular CPU structure that you need to focus on and that is the Von Neumann architecture.



A simple diagram of the Von Neumann architecture

Arithmetic and logic unit (ALU)

The ALU is the part of the CPU that carries out all the arithmetic and logical operations. Data is passed to the ALU to allow the required calculations to be carried out. The results of any calculations are then sent to be stored in a **register**.

Register – a holding place to temporarily hold data and instructions as they are being processed

Typically, the ALU includes a register that it uses to store intermediate results of calculations while the full calculation is being carried out. This register is known as the accumulator.

The ALU is responsible for number crunching

Accumulator (ACC)

The accumulator is the register where intermediate arithmetic and logic results are stored. It accumulates the results of the calculation, hence the term 'accumulator'. If a CPU did not have an accumulator, it would need to write the results of each stage of the calculation it is carrying out to the main memory, then read it back into the CPU again to perform the next stage. This would make the process much slower.

Control unit (CU)

The control unit is responsible for directing the flow of all inputs and outputs to the CPU. It controls the different stages of the execution of an instruction and it guides the flow of data through the different parts of the CPU. It also controls and monitors the hardware attached to the computer. It tells the ALU, the computer's memory and the hardware devices how to respond to an instruction. The CU communicates with the rest of the components by using the control bus.

Memory address register (MAR)

Each position within a computer's memory has a memory address. This address is used to locate the data or instruction that is stored there. The MAR holds the address location of the next piece of data that is to be fetched into the CPU, or the address where a piece of data will be sent to be stored. The address location is sent to and from the MAR using the address bus.

Memory data register (MDR)

The MDR holds the actual data that is being fetched into the CPU, or the data that is being sent to be stored in the computer's memory. The data is sent to and from the MDR using the data bus. The MDR can also be known as the memory buffer register (MBR).

Program counter (PC)

The program counter is a register that holds the address of the next instruction that is to be carried out. It is designed to indicate where the CPU is up to in a program sequence. The program counter can also be known as the instruction pointer (IP) or the instruction address register (IAR). There is also a register that holds the address of the current instruction that is being processed; this is called the current instruction register (CIR).

Cache

The cache is a small section of memory that is built into the CPU. It stores the most commonly used instructions and data to allow quick and easy access to them. There are three 'levels' of cache, with each level being progressively further away from the CPU.

- The first type of cache memory is L1 cache, typically operating at speeds close to the CPU
- The second type is L2 cache, which is slightly slower and further away from the CPU
- Finally, L3 cache is the cache memory furthest away from the CPU, which runs the slowest.

In order to process an instruction, the CPU carries out a process called the fetch–execute cycle.

What is the fetch-execute cycle?

OCR specification reference:

☑ the function of the CPU as fetch and execute instructions stored in memory

The fetch-execute cycle is the process a CPU goes through in order to process an instruction. At the start of the fetch-execute cycle, both the data and the instruction are loaded into the computer's main memory so they are ready to be processed. It is the computer's operating system that loads them into place.

What does an instruction look like?

The instructions that are processed by the computer come in a specific form of binary bits. This is another agreed-upon protocol; when the computer receives an instruction it will always know what to do, and where to find the things it needs. The actual instruction itself is split into two parts, the **opcode** and the **operand**. The opcode tells the CPU what is to be done, and the operand tells the CPU where to do it.

Opcode – part of an instruction that tells the CPU what operation to do

Operand – part of an instruction that tells the CPU where to perform the operation

01101101	11101100
opcode	operand

We can use a set of commands called mnemonics to create computer programs. The example opcode might correspond to the INCREMENT command, with the operand telling the CPU which memory address to add one to.

Fetching an instruction

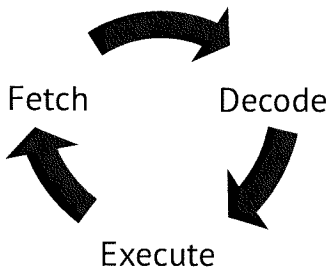
The first step the CPU carries out is to fetch the instruction and data into the CPU.

- A value is placed in the PC to indicate the address of the instruction that is to be carried out.
- This value is then passed to the memory address register so that the instruction or data stored there can be located
- The value is then sent to the memory to locate the address, this is done using the address bus
- Any data stored in the memory address is then sent to the MDR using the data bus

Decoding an instruction

In order to process an instruction, the CPU needs to understand the instruction.

- This process is called decoding an instruction
- In order to decode an instruction, the CPU uses an instruction set.
- An instruction set is a set of commands that are written in machine language that the CPU understands



Once a cycle is complete, the CPU is ready to carry out the next cycle

Executing an instruction

This is the part of the cycle where any actions required by the instruction take place, such as any calculations that need to be carried out using the data. The ALU may be used at this stage to carry out the calculation.

What can affect the performance of the CPU?

OCR specification reference:

☑ how common characteristics of CPUs affect performance: clock speed, cache size, number of cores

The performance of a computer can be dependent on the performance of the CPU within it. By the performance of the CPU we mean how quickly it can carry out instructions. There are three main factors that can affect the performance of a CPU; these are the clock speed, the cache size, and the number of cores available to the CPU.

Clock speed

The speed at which the CPU can carry out instructions is called the clock speed. The clock speed is measured in cycles per second. This refers to one cycle of the fetch–execute cycle. Clock speed is measured in hertz and one cycle per second would be 1 hertz. A typical CPU could have a clock speed of 2 gigahertz (GHz) per second. This means it is capable of carrying out two billion cycles per second – that's one busy computer! The greater the clock speed of the CPU, the more instructions it can process in a second. The more instructions per second it can process, the greater the performance of the CPU.

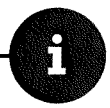
Number of cores

The number of cores a CPU has can vary. If a CPU has a single core it is capable of processing one instruction at a time. If a CPU has two cores, referred to as dual core, it can process two instructions simultaneously. This effectively doubles the number of instructions it is capable of processing in a second.

A single-core 2 GHz processor can process two billion instructions per second; however, a 2 GHz dual-core processor can process four billion instructions per second. Most modern computers have dual-core or quad-core processors; and as of 2016 octa-core processors are becoming more common – for example both the Playstation 4 and Xbox One consoles have 8 cores. To run a multi-core processor a computer requires a great deal more power than to run a single-core processor; they are also more expensive to manufacture.

Cache

The cache is the part of the CPU where the most commonly used instructions and data are held. This allows quick and easy access to these instructions. If a CPU has a larger cache, it can store more commonly used instructions and data and has a greater range of them available for quicker access. If the CPU has a greater range available, this can increase the performance of the CPU for the more common tasks it performs.



Chapter Summary

- An embedded system is a type of processor designed to perform limited functions, normally one or two. They are often built into electronic machinery or devices.
- The purpose of the CPU is to process instructions given to a computer system. It carries out the instructions and any data that requires storage as a result is stored in the main memory.
- The CPU has a number of components that are involved in carrying out an instruction; these are the ALU, PC, CU, ACC, MAR and MDR, and the cache.
- The fetch–execute cycle is the name given to one whole cycle of processing and instruction.
- The clock speed of a CPU is measured in hertz. One hertz is the equivalent of one cycle per second. The greater the number of hertz available, the greater the performance of the CPU.
- A single-core processor can process one cycle at a time. A dual-core processor can process two cycles simultaneously. The more cores a CPU has, the more cycles it can process simultaneously and the greater the performance of the CPU.
- The cache is a small part of the CPU where commonly used instructions and data are stored. The more commonly used instructions and data that can be stored, the quicker the CPU can access these and the greater the performance.

Practice Questions

1. Describe what is meant by an embedded system. [1]
2. Explain the difference between hardware and software. [2]
3. Explain how the CPU processes an instruction. [3]
4. Describe two functions of the Control Unit. [2]
5. Janey has the following spec for her computer:
   - 2 GHz single-core processor
   - 4 GB RAM
   - 500 GB HDD

   Describe two ways Janey could upgrade the performance of her computer. [2]

