# 2.1 Algorithms – L2 Activity 1

## Diagram – Internal Components of a Computer System

## High Ability – Label the following Diagram

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

INP

Add 5

INP

STO 8

INP

ADD 8

1. Find definitions of the following

|  |  |
| --- | --- |
| **Processor** |  |
| **MAR** |  |
| **MDR** |  |
| **CIR** |  |
| **Main Memory** |  |
| **Address Bus** |  |
| **Data Bus** |  |
| **Accumulator** |  |
| **PC (Program Counter)** |  |
| **ALU** |  |

1. Now label where you believe they should be in the diagram.

## Medium/Low Ability task with some complete

Address Bus

INP

Add 5

INP

STO 8

INP

ADD 8

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

MAR

ALU

1. Find definitions of the following

|  |  |
| --- | --- |
| **Processor** | Performs calculations in the computer – Executes Instructions |
| **MAR** |  |
| **MDR** |  |
| **CIR** | The CIR is the Current Instruction Register. This hold the instruction that has just been fetched from main memory and is about to be executed. |
| **Main Memory** |  |
| **Address Bus** |  |
| **Data Bus** |  |
| **Accumulator** |  |
| **PC (Program Counter)** |  |
| **ALU** | Arithmetic Logic Unit – performs arithmetic (Add / Subtract) and logical (AND, OR) operations |

1. Now label where you believe they should be in the diagram.

## High level – Stretch & Challenge

1. What is meant by the Fetch-Execute Cycle?
2. Describe in words the stages of the fetch-execute cycle using the diagram that has just been completed.
3. Watch the video located: <https://www.youtube.com/watch?v=xfJbpCJSpd8> to check your answers.

**Answers:**

Main Memory

Address Bus

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

INP

Add 5

INP

STO 8

INP

ADD 8

MAR

PC

Data Bus

MDR

CIR

ALU

Accumulator

|  |  |
| --- | --- |
| **Processor** | The processor processes data and executes instructions. The diagram above shows the internal working of the processor (often known as the CPU). The processor contains registers (very small/fast storage areas). The registers can have a dedicated purpose or can be general purpose. In the diagram above dedicated registers are used which are discussed below. The diagram shows how the processor is internally linked to Main Memory (RAM) using what are called System Busses. |
| **MAR** | MAR – This is the Memory Address Register. It is a very small area of memory that is used to hold a memory address. The memory address is provided by the Program Counter. The Memory Address is the next address in main memory to address. E.g. in the diagram above – if the next instruction to fetch is STO 8 – then 0100 would be held in the MAR ready to be transferred along the Address Bus. |
| **MDR** | Once data is retrieved from Main Memory (RAM) it travels along the Data Bus to the Memory Data Register. Data/Instructions are held in the MDR temporarily. Any instructions will then be passed to the CIR. |
| **CIR** | The CIR is the Current Instruction Register. This hold the instruction that has just been fetched from main memory and is about to be executed. The Current Instruction is split into an Op Code (the operation to be performed) and the Operand (the data in which to perform the operation on).  On our previous example of STO 8 – STO would be the Op Code and 8 would be the Operand. |
| **Main Memory** | All Data and Instructions currently in use are held here. When the power is turned off, all data/instructions are lost. (Volatile Memory) |
| **Address Bus** | Carries addresses from the Processor (MAR) to Main Memory |
| **Data Bus** | Carries Data/Instructions from Main Memory back to the Processor (MDR) |
| **Accumulator** | The Accumulator is used to hold the result of a calculation or an operation. When a value is inputted into the computer it will be held in the accumulator. If a value is added – then the result is temporarily held in the accumulator.  The Accumulator is another Register. |
| **PC (Program Counter)** | Hold the address of the next instruction that will be fetched. |
| **ALU** | Arithmetic Logic Unit – performs arithmetic (Add / Subtract) and logical (AND, OR) operations |

We’d like to know your view on the resources we produce. By clicking on ‘[Like](mailto:resources.feedback@ocr.org.uk?subject=I%20liked%20the%20OCR%20GCSE%20(9-1)%20Computer%20Science%20Challenges%20(Computational%20Logic)%202.1%20Algorithms%20–%20L2%20Activity%201%20)’ or ‘[Dislike](mailto:resources.feedback@ocr.org.uk?subject=I%20disliked%20the%20OCR%20GCSE%20(9-1)%20Computer%20Science%20Challenges%20(Computational%20Logic)%202.1%20Algorithms%20–%20L2%20Activity%201%20)’ you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click ‘Send’. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: [www.ocr.org.uk/expression-of-interest](http://www.ocr.org.uk/expression-of-interest)

**OCR Resources**: *the small print*OCR’s resources are provided to support the teaching of OCR specifications, but in no way constitute an endorsed teaching method that is required by the Board, and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources.   
© OCR 2016 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work.

OCR acknowledges the use of the following content: n/a

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: [resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk)