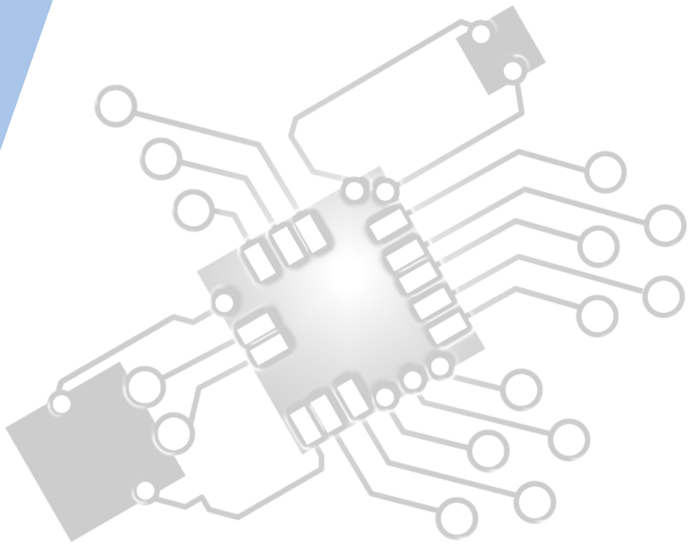
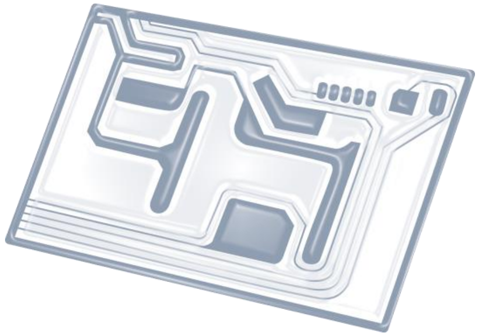
Computer Organisation

IB Computer Science



Content developed by
Dartford Grammar School
Computer Science Department

HL Topics 1-7, D1-4



1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

HL & SL 2 Overview

Computer architecture

2.1.1 Outline the architecture of the central processing unit (CPU) and the functions of the arithmetic logic unit (ALU) and the control unit (CU) and the registers within the CPU

2.1.2 Describe primary memory. 2 Distinguish between random access memory (RAM) and read-only memory (ROM), and their use in primary memory

2.1.3 Explain the use of cache memory

2.1.4 Explain the machine instruction cycle

Secondary memory

2.1.5 Identify the need for persistent storage

Operating systems and application systems

2.1.6 Describe the main functions of an operating system

2.1.7 Outline the use of a range of application software

2.1.8 Identify common features of applications

Binary representation

2.1.9 Define the terms: bit, byte, binary, denary/decimal, hexadecimal

2.1.10 Outline the way in which data is represented in the computer

Simple logic gates

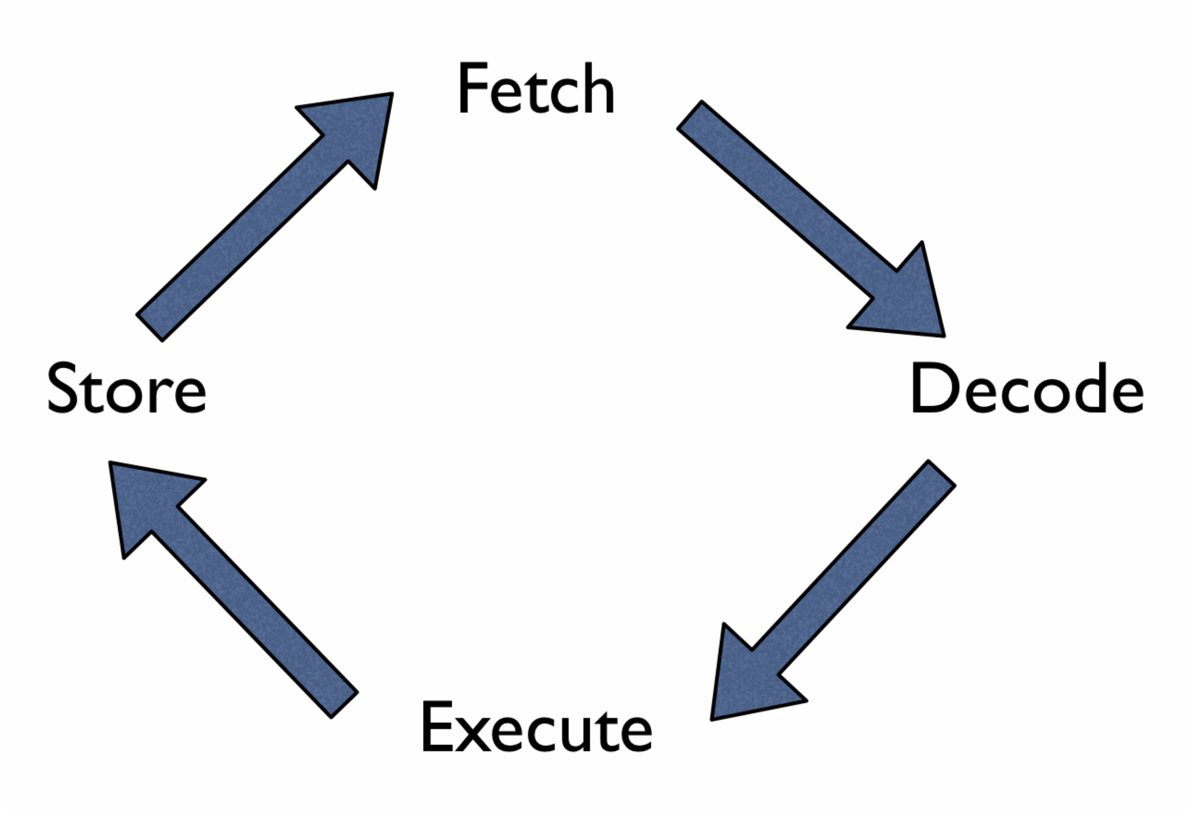
2.1.11 Define the Boolean operators: AND, OR, NOT, NAND, NOR and XOR

2.1.12 Construct truth tables using the above operators

2.1.13 Construct a logic diagram using AND, OR, NOT, NAND, NOR and XOR gates

Topic 2.1.4

Explain the **machine instruction cycle**

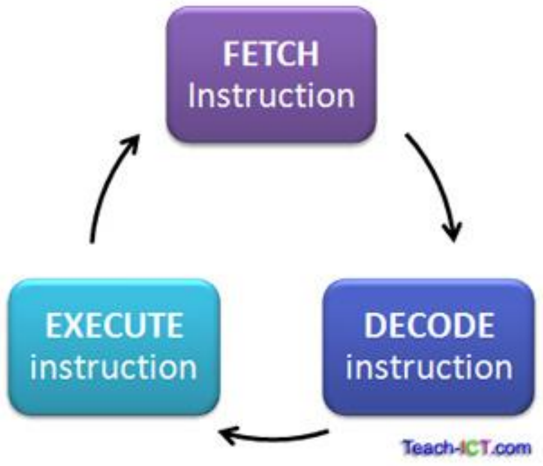


The Fetch-Execute cycle

The basic operation of a computer is called the ‘fetch-execute’ cycle (also called the ‘machine cycle’).

The computer fetches the instruction from its **memory** and then **executes** it.

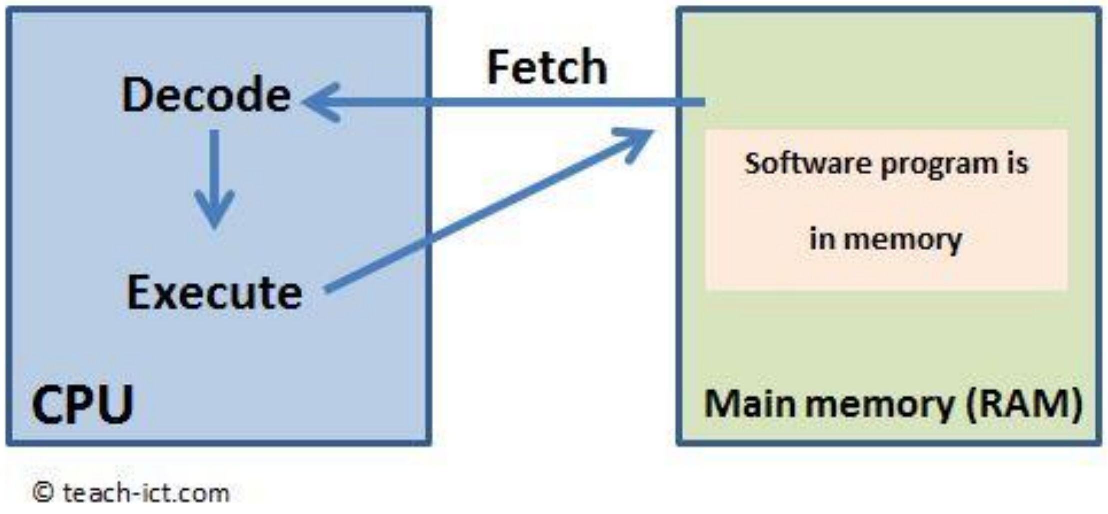
This is done repeatedly from when the computer is booted up to when it is shut down.



Step 1: Fetching the instruction

The first step the fetch-execute cycle carries out is **fetching** the instruction.

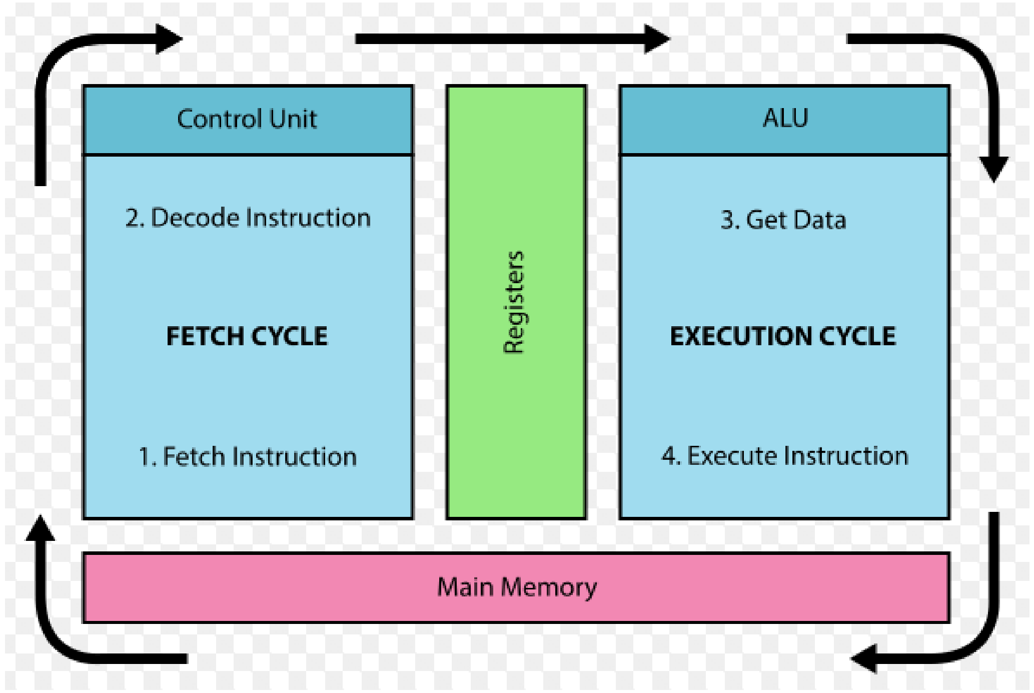
The CPU fetches this from the main memory (RAM) and stores it in the CPU temporary memory, called the registers.



Step 2: Decoding the instruction

Once the instruction has been fetched, the CPU will need to understand the instruction to action it.

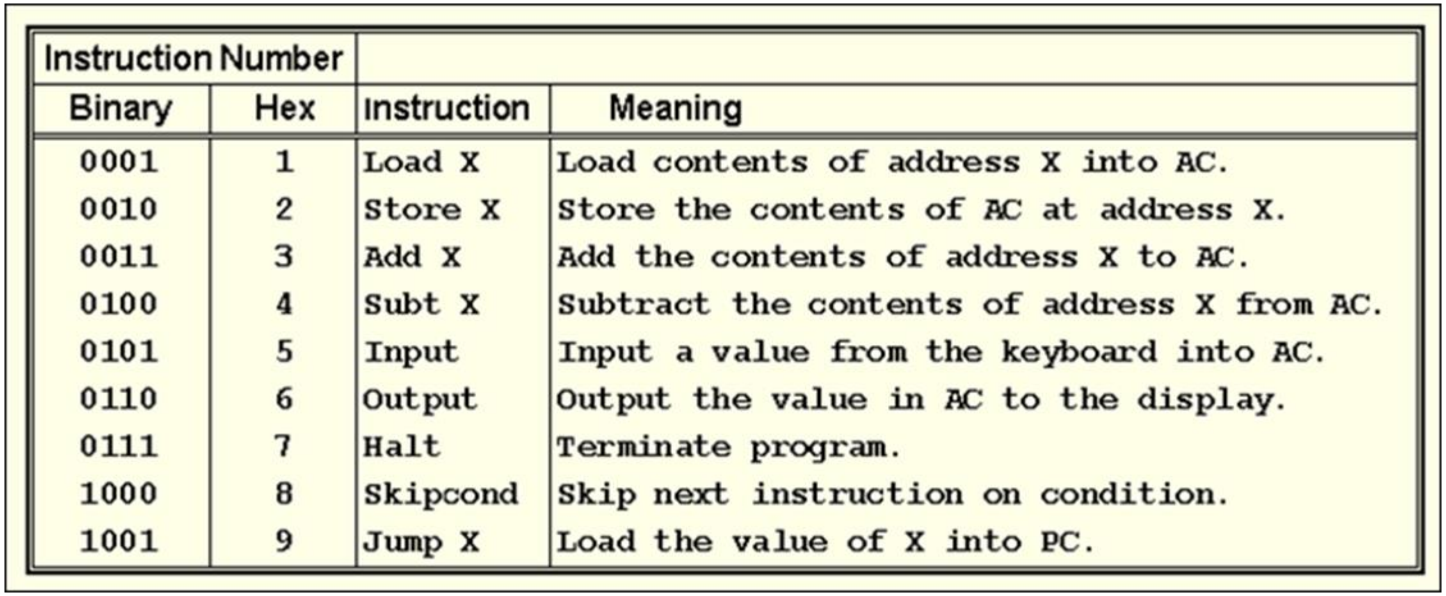
This is called **decoding**.



Step 3: Executing the instruction

When the instruction has been decoded, the CPU can carry out the action that is needed.

This is called **executing** the instruction. The CPU is designed to understand a set of instructions - the instruction set.



Instruction Number			
Binary	Hex	Instruction	Meaning
0001	1	Load X	Load contents of address X into AC.
0010	2	Store X	Store the contents of AC at address X.
0011	3	Add X	Add the contents of address X to AC.
0100	4	Subt X	Subtract the contents of address X from AC.
0101	5	Input	Input a value from the keyboard into AC.
0110	6	Output	Output the value in AC to the display.
0111	7	Halt	Terminate program.
1000	8	Skipcond	Skip next instruction on condition.
1001	9	Jump X	Load the value of X into PC.

Example

A single piece of program code might require several instructions. Look at this Java code:

```
area = length * width
```

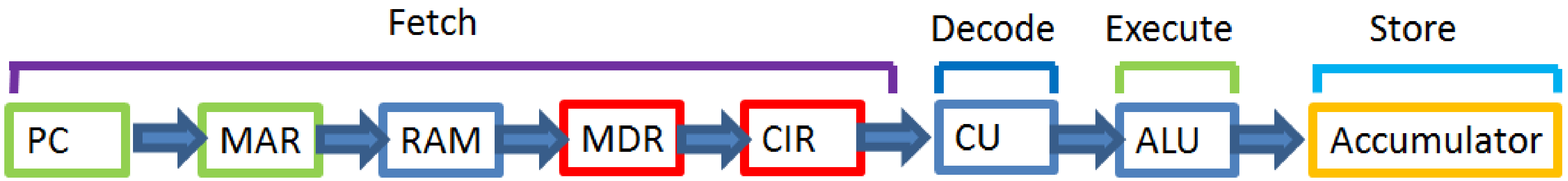
First, the computer needs to load in the value of the variable **`length`** into the immediate access store (registers).

Next it needs to load in the value of the variable **`width`**.

Then it needs to multiply the two numbers together, and finally it needs to store the result in the variable **`area`**.

Simplified model

Stages of the Machine Instruction Cycle



The address of the next instruction is copied onto the MAR from PC

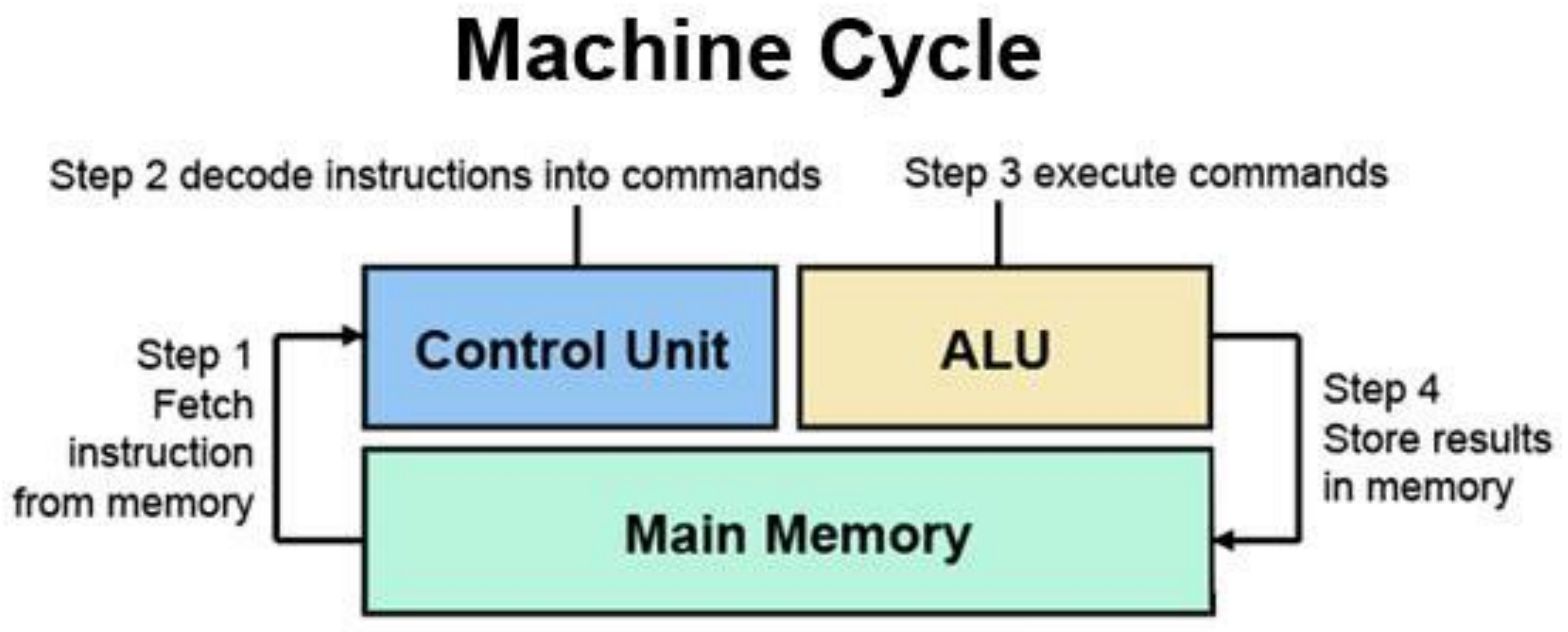
The Data is fetched using the address from the MAR and sent to the MDR

The instruction is stored in the MDR which is then copied into the CIR

The instructions are then sent to the CU which makes changes to the ALU. The ALU then executes the instruction and stores them in the Accumulator if needed.

The Accumulator stores the results of the executed instruction.

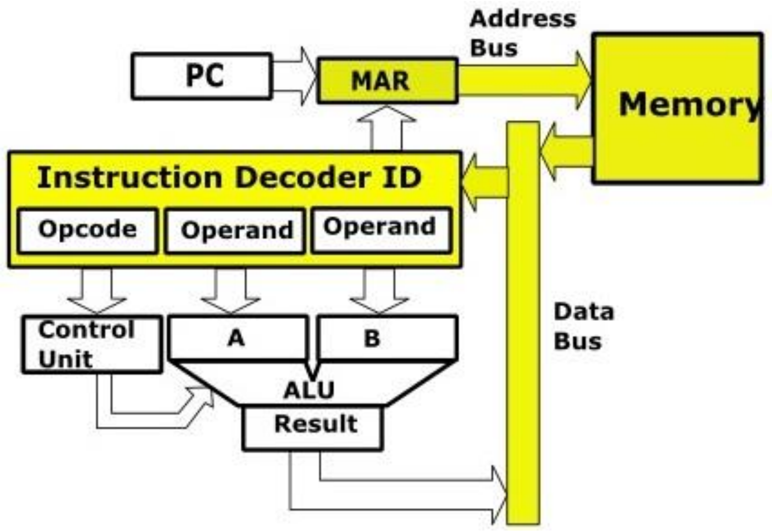
Simple Model #2


Machine Cycle
Step 2 decode instructions into commands
Step 3 execute commands
Step 1 Fetch instruction from memory
Control Unit
ALU
Step 4 Store results in memory
Main Memory

Exam note!

This curriculum point requires you to **describe** the role of the **data** and **address busses** in the cycle.

Think about what **information** they carry, **from** where **to** where and what they **connect to** at each end.



Two videos to explain

Video 1:

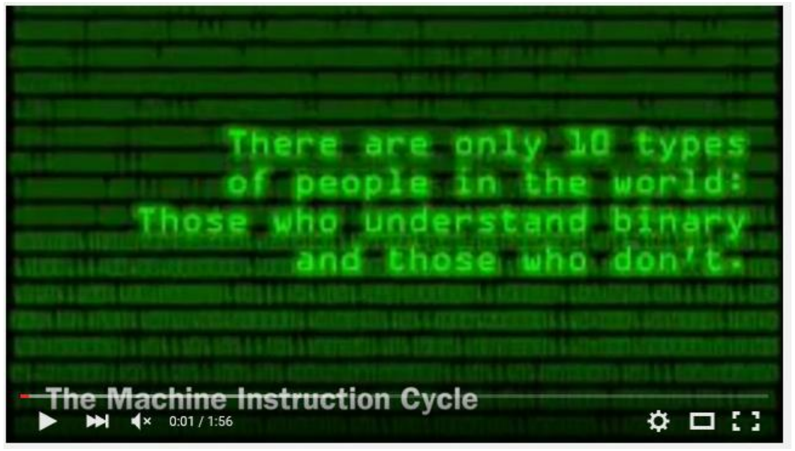
Video 2:

THE FETCH - EXECUTE CYCLE

Theodore Messinezis

International Baccalaureate Student

www.theomessin.com


There are only 10 types of people in the world: Those who understand binary and those who don't.

The Machine Instruction Cycle