The Fetch-Execute Cycle

# Stored Program Concept

* ***A running program is resident in main memory***
* When a program is run, its instructions and data are loaded into main memory

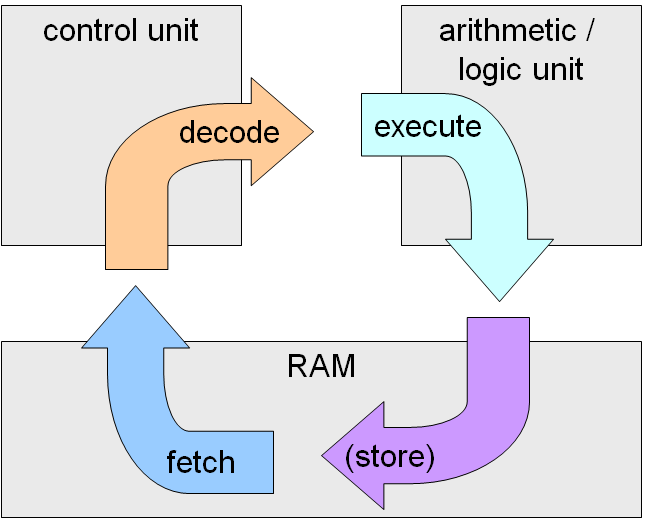
## Von Neumann Architecture

* This architecture is flexible; instructions and data can occupy the same memory

## Harvard Architecture

* Separate buses are used for data and instructions, which access different parts of memory
* This is much more efficient for devices with a fixed amount of code, such as embedded systems (e.g. a washing machine); if instructions never change, they can be stored in ROM (which is cheaper)

# Fetch-Execute Cycle

* **Fetch:** The processer fetches the next instruction from main memory
* **Decode:** The processor determines the purpose of the binary code
* **Execute:** The processor carries out the instruction, which may include reading or storing data, or performing a calculation.

## The Control Unit

* Supervises the fetch-execute cycle, ensures data is processed and routed properly

## The ALU

* Carries out mathematical functions and compares values
* It is sent the op-code (instructions) and operands (data)

## System Clock and Internal Clock

* There is a system clock (on the motherboard) and the CPU’s internal clock, which runs faster than the system clock, as it needs to perform operations faster
* It takes the system clock and multiplies it

## Interrupts

* Can be sent from hardware, software, or a processor
  + Interrupt sent to processor
  + Processor halts current thread
  + Current thread state saved to stack
  + Interrupt handler executed
  + Previous thread popped from stack
  + Thread continues