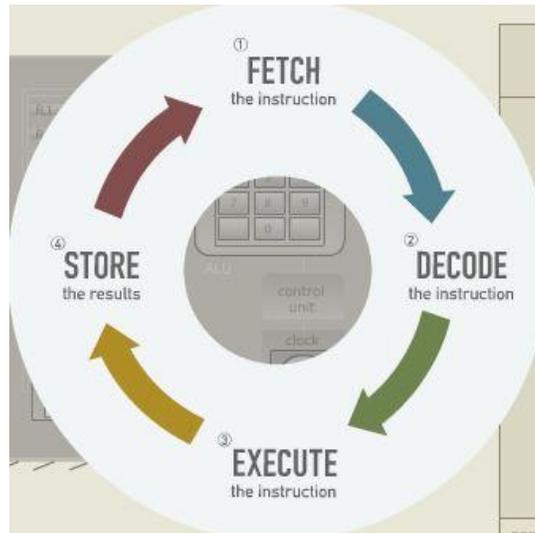


Fetch-Execute-Cycle

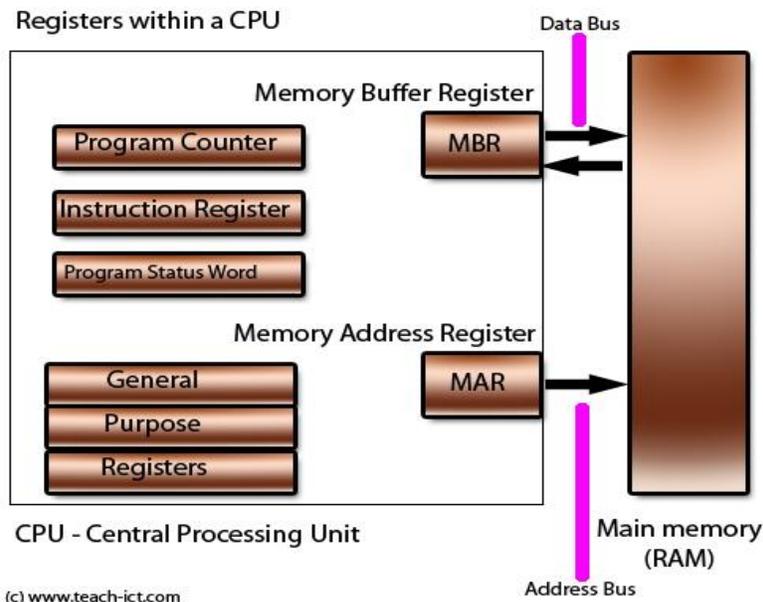
A CPU instruction is a line of code to perform some specific task.

The Fetch-Decode-Execute cycle describes the basic steps a CPU carries out to process an instruction.

A fetch execute decode cycle is shown below.



The picture below shows the general internal set up of the CPU



Special Purpose Registers:

The CPU contains a number of special-purpose registers:

- **Instruction Register (IR):** The instruction register holds the instruction currently being executed.
- **Memory Data Register (MDR):** The memory data register (also known as the memory buffer register or data buffer) holds the piece of data that has been fetched from memory.
- **Memory Address Register (MAR):** The memory address register holds the address of the next piece of memory to be fetched.
- **Program Counter (PC):** The program counter holds the location of the next instruction to be fetched from memory. It is automatically incremented between supplying the address of the next instruction and the instruction being executed.
- **Accumulator:** The accumulator is an internal CPU register used as the default location to store any calculations performed by the arithmetic and logic unit.

General Purpose Registers

General purpose registers are available to store any transient data required by the program.

For example, when a program is interrupted its state, ie: the value of the registers such as the program counter, instruction register or memory address register - may be saved into the general purpose registers, ready for recall when the program is ready to start again.

In general the more registers a CPU has available, the faster it can work.

Now, we will look at each stage in detail.

Fetch

Step 1

The Program Counter copies the address of the next instruction it contains into the Memory Address Register (MAR).

Step 2

The Memory Address Register places the address to be used on to the 'Address Bus'

Step 3

The Memory Address Register triggers a 'read' signal that causes main memory (RAM) to place the instruction being asked for on to the 'Data Bus'

Step 4

The instruction on the data bus is loaded into the Memory Data Register (also called Memory Buffer Register)

Step 5

The Memory Data register copies the instruction into the 'Instruction Register'

The Fetch stage is now complete

Decode

The CPU examines the instruction in the current instruction register (CIR) and 'decodes' it. This means a special part of the CPU called the 'decode' unit will make the rest of the CPU ready to carry out the instruction. It does this by issuing a series of 'micro-instructions'.

For example the instruction might say 'Add'. The decode unit understands what this means and gets the system ready to carry that instruction out.

Every CPU has an **instruction set** that defines what the decoder understands as legitimate commands.

All software eventually ends up as a set of commands from within the instruction set.

Execute then Reset

STAGE: Execute

The instruction within the instruction register is carried out (executed) by the CPU. The part that executes instructions is called the 'execute unit'.

STAGE: Reset

Now that the CPU is executing an instruction, the Program Counter can now be reset to point to the next instruction.

This is the Fetch-Decode-Execute cycle that is present in every sequential processing computer.

