

Glossary Part 1

Computer

Machine that can be programmed to carry out a series of arithmetical or logical operations.

Computer System

Elements that make a computer usable (Hardware, Software, Operating System and Peripherals).

Hardware

Physical parts of a computer. It includes CPU, RAM, Storage, Motherboard, Graphics Card...

Software

Collection of programs and data that tell a computer to perform specific tasks or instructions.

Operating Systems

Software that manages a whole Computer System, allocating resources to the Hardware and organizing the execution of the Software.

CPU

Central Processing Unit, the main logic arithmetic unit of the system. It is in charge of performing the computer operations according to the Software. It is an essential part of the Hardware.

CU

Control Unit, component of the CPU that directs the operations of the CPU. It uses a decoder of instructions that translates them into timings and signals that sends to other parts of the CPU or Computer like the ALU.

ALU

Arithmetical Logic Unit, component of the CPU that carries out the logical operations of the CPU.

Memory

Group of address that can store computer instructions or data. The CPU needs to access them in order to perform operations, either because it needs to know which instruction to run or which data to run an instruction with.

RAM

Random Access Memory, volatile Memory that can be read or write to in any order. It typically stores the memory or data that the processor is currently working on.

BUS

Communication system that transfers data between internal parts of a computer. For example there exists a BUS between the RAM and the CPU. There can exist either Data BUS or Control BUS, depending on which type of information they carry.

Register

Chunk of memory that is very fast and close to the processor. There are different types that can have different purposes. Usually they are an integral part of a computer operation or instruction.

Machine Cycle

Steps that a CPU takes to perform an instruction from Memory. It is the most basic CPU operation.

Instruction or Machine Code

Operation that the CPU can perform, usually composed of Operation and Operands. The CU is in charge of decoding this Instruction so that the CPU can execute it.

Instruction Operation

The actual operation that the CPU or ALU must perform. For example in an `add` instruction is going to use an adder (circuit that adds) from the ALU.

Instruction Operand

Data that the operation needs to be performed. There can be 3 types of operands depending on how the memory is addressed (see [Address Modes](#)), which are:

- Immediate: A number is directly given.
- Register: A register that has data is given.
- Memory Address: A memory address that contains data or another memory address is given.

Instruction Register (IR)

It holds the current instruction that the CPU is executing.

Program Counter (PC)

It holds the address where the next instruction to be executed is stored. Determines how the program must do next.

Accumulator

Register used to store intermediate (necessary) data that is used in multi step logical or arithmetical operations. It works as a temporal storage location.

Stack Pointer (SP)

Address that points to another address in which some data is stored. This data corresponds to the data stored by the Stack

Stack

Data structure that is integral to CU functionality. It is used to store data or instructions in an ordered manner. It requires a Stack Pointer in order to work.

Data Movement Instruction

Type of instruction that moves data from a Register to an Address or vice-versa. It can also load directly values into an Address or Register. What it uses depends on the Address Mode used. These operations are performed by the Control Unit (CU).

Arithmetical and Logical Instructions

Operations which involve math in some way. It includes: Integer Operations, Binary Operations, Logical Operations and Comparison. They are performed by the ALU.

Control Flow Operations

Depending on a given condition they change the PC. They are performed by the ALU and CU.

Routine

Set of Instructions that perform an specific task. A Routine can be called (like a functions) in order to execute the instructions that it contains. The Program Counter and the Stack Pointer are integral to the functionality of the Routine call.

Fetch (Cycle Step 1)

The CU requests the instruction that needs to be executed according to the PC. It is taken from Memory (RAM).

Decode (Cycle Step 2)

The received instruction is decoded and stored into the Instruction Register (IR). Needs to separate the operation code from the operands, breaking the instruction into its components. The amount of Operands that change from operation to operation.

Execution (Cycle Step 3)

Takes the decoded operation code and executes it with the corresponding operands. Increments the PC so that the next instruction is executed in the next Machine Cycle.

Instruction Set

Types of instructions that a particular CPU can perform. Each CPU architecture or family has its own Instruction Set.

Assembly

Usually each instruction set has its own Assembly language which is a human readable form to express Instructions. It is the lowest level of programming that a human can do. It is decoded into Machine Code to make the instructions machine readable.

Assembly Directive

Unit of operations expressed on Assembly code. Each operation in assembly has a Directive.

MOV

Assembly Directive (AD) used to move data from one operands to another.

ADD

AD used to carry out addition. Data movement operation.

SUB

AD used to carry out subtraction. Arithmetical operation.

MUL

AD used to carry out multiplication. Arithmetical operation.

SHL

AD that carries out "left shifting". Logical operation.

SHR

AD that carries out "right shifting". Logical operation.

LDA

AD that loads data into the Accumulator. Data movement operation.

PUSH

AD that stored data into the Stack. Data movement operation.

POP

AD that retrieves data from the Stack. Data movement operation.

CALL

AD that executes a Routine. Data Flow operation.

RET

AD that returns to original PC and returns value once a Routine **CALL** has finished. Control Flow operation.

JMP

AD that jump to given address or jump point. Control Flow operation.

JL

Jump if less than. Control Flow operation.

JLE

Jump if less or equal than. Control Flow operation.

JG

Jump if greater than. Control Flow operation.

JGE

Jump if greater or equal than. Control Flow operation.

JE

Jump if equal. Control Flow operation.