**Introduction to Computer Organization and Architecture**

Computer architecture refers to those attributes of a system that have a direct impact on the logic execution of a program. Computer organization refers to the operational units and their interconnections that realize the architectural specifications.

A computer is essentially a device that has some components that allow it to perform some functions. There are four basic functions a computer can perform: data processing, data storage, data movement and control. Its main components include the central processing unit (CPU), the main memory, input and output devices and interconnections. The CPU further consists of the control unit (CU), the processing unit (PU), registers for internal data storage and interconnections.

**How Programs Are Executed**

The instructions provided in a higher-level language like C or C++ are converted to assembly language and then to machine language so that it is possible to execute the instructions. For example, take the instruction print(a+5) in python, assuming that a has already been declared with a value of 10. This instruction will be executed in multiple steps.

Firstly, we need to retrieve the value of a. a actually holds the value of an address in the RAM. Every bit in RAM has a specific address, but with the huge number of bits in RAM these days, it is easier to indicate addresses as bytes of 8 bits each. So, the first thing the CU must do is retrieve the address a holds and LOAD it onto its cache. The cache contains a table and a specific row is designated to hold the address of a. It gets the value held at that address and puts it in the register. Note that the register is used temporarily and can only store a very small amount of data for immediate calculations.

The next instruction is simple enough. Just ADD 5. The CU must also store the result in the register so that it can be used to execute the next command.

Finally, the last command is to print the result. The simple one-line instruction has resulted in a multi-line set of instructions that are stored in assembly language. The assembly language file looks somewhat like this:

LOAD a

ADD 5

OUTPUT d1

Note that an extra keyword, d1, has been added. The command OUTPUT just tells the system that some value needs to be given as output. It does not indicate how the data should be processed, whether it is visual or audible or something else. To resolve this issue, the CU gives an output device code along with this command. Every output device connected to the computer, be it speakers or monitors, has a specific key that the CU knows. Thus, the CU can pass along that key with the OUTPUT command depending on what type of output must be given, which in our case is visual. When the instruction will be executed, that output device will be used. Note that the keyword d1 used is just an example and this is not an actual key.

Next the assembly language file is converted to machine language. Each instruction from the assembly language file is converted to a single line of 1’s and 0’s by a decoder inside the CU. The actual execution occurs one by one. Each line is pasted in the RAM and executed.