Lecture 00.03 Computer architectures

If we descend from high-level programming languages like Python to low-level languages like C, then continue to descend, considerations become more hardware-specific. *Computer architectures* are descriptions of this down to a fairly concrete level, stopping somewhere above the actual physics of the whole thing.

computer architecture

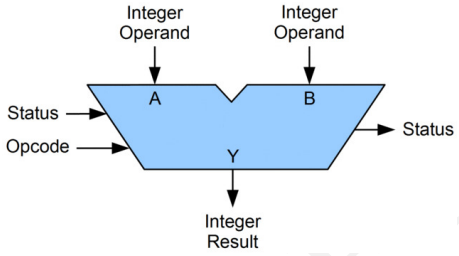

Figure 00.3: ALU block (Lamberson, 2018).

Embedded computer programming contacts the computer architecture much more than does personal computer programming. In fact, the C language gives us access to different aspects of it.

Consider the four primary components of a computer that follow.[2]

central processing unit (CPU) The *CPU* consists of a *control unit* (brain) and a *datapath* (brawn). The control unit receives machine code instructions from memory and controls the other components to perform corresponding tasks. The datapath consists of functional units such as registers (the fastest, smallest-capacity memory), buses (intra-computer communication systems), and *arithmetic logic units* (ALU) that can perform very basic arithmetic or logical operations (see Figure 00.3 for an illustration).

CPU
control unit
datapath
arithmetic logic unit (ALU)

memory *Memory* stores *data* (e.g. numbers, files, etc.) and *instructions*. Numerous forms exist, but the primary variable of interest is that of speed of access (read/write). Faster tends to mean less capacity and more expense. Permanence is another consideration: some types of memory, called *volatile*, lose their state when they lose power, while others, called *non-volatile*, do not. These considerations lead to complex optimization when, as is typically the case, different types of memory are used within the same system. Roughly speaking, data or instructions that are most likely to be read soon are moved to faster, smaller, more expensive memory.

memory
data
instructions
volatile memory
non-volatile memory

In addition to speed, volatility, and cost there is another important aspect of memory: its *rewritability*. This divides memory into two categories:

rewritability

[2] This is sometimes called the von Neumann architecture.

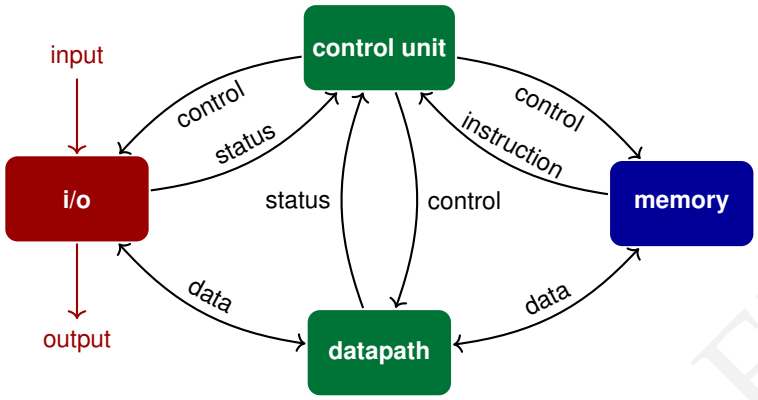

Figure 00.4: high-level schematic of the four primary components of a computer: ■ CPU (control unit and datapath), ■ memory, ■ input, and ■ output.

read-only-memory (ROM) of which the state "cannot" be altered, only detected, by the CPU and

read/write memory (RAM) that can be both read and written-to by the CPU.

There are several types of ROM, described as follows.

ROM is masked ROM that is programmed at manufacturing and cannot be altered. It is for mass production and is cheapest.

PROM is field-programmable; that is, it can be programmed after manufacturing, typically only once.

EPROM is erasable (usually with UV light) and reprogrammable after manufacturing, but has a short number of erases: on the order of 100 times.

EEPROM is electrically-erasable and non-volatile. It is often called *flash memory*.

flash memory

input

input *Inputs* are data from devices other than the computer, such as keyboards, sensors, and remotely communicated commands (via, say, the Internet).

output

output *Outputs* are data sent to devices other than the computer, such as displays, printers, and actuators (e.g. a motor).