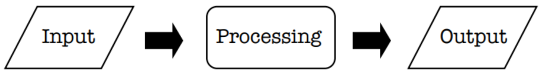
Week 2: System basics

Week 2: Hardware components

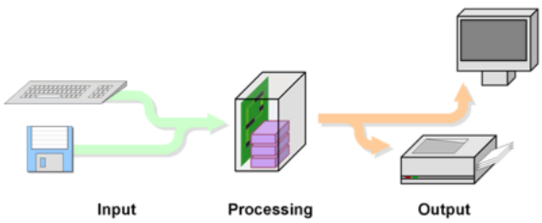
There is a complex interaction between the primitive level of computation that the hardware offers, and the end-user benefits that computer software provides. We'll look at an overview of these processing layers here, and see them in action in the later Operating Systems and Application Software chapters.

At its simplest level, a computer system merely performs basic logic calculations on binary data. These basic operations are performed billions of times a second, allowing the system to complete operations of significance, as well as create an accessible means for people to perform these tasks.

- A computer processes information according to a set of instructions



- We would thus expect a computer to have hardware for collecting input, performing ‘processing’ and producing output



Processing

The bulk of computer processing is done by the "brain" of the system; a chip known as the Central Processing Unit (CPU). This chip receives a steady stream of instructions (and data for the instructions to process), completes those instructions and returns their results.

They are generally the most well-known parts of the computer, and feature prominently in marketing material. Typical computer processors in modern desktop machines are Intel's [Core (http://en.wikipedia.org/wiki/Core_(microarchitecture))](http://en.wikipedia.org/wiki/Core_(microarchitecture)) family and AMD's [various (http://www.amd.com/us/products/desktop/processors/Pages/desktop-processors.aspx)](http://www.amd.com/us/products/desktop/processors/Pages/desktop-processors.aspx) processor families.

Input

In order for the CPU to process data, there must be a source for it all -- this is the role of input hardware. Data input (from the perspective of the CPU) can come from many sources, and at various speeds; a computer keyboard might feed it with data at a few dozen words per minute, or a high-speed disk system could bombard the CPU with millions of data records each second.

Output

Once calculations are completed, they must be stored -- either permanently or temporarily. Usually, computer programs store the results of their operations in memory, as it is the fastest storage medium. In the long term, such results are then stored to a more permanent location like a hard disk.

Sometimes the output isn't intended for permanent storage; for example, the computer might be calculating how to display a window on screen, or how to render a document onto a piece of paper. In this case, the results of that calculation would be fed directly to the screen (via the graphics processor), or to the printer.

Specialist hardware

In most computers, some processing burden is relieved from the CPU and moved towards chips more suited to the task known as *co- processors*. This provides two benefits; not only is the CPU more free to execute other instructions, but the specialist chips often do a much better job of these operations.

A historical perspective

Early CPUs were not very fast at complicated mathematics known as *floating- point* operations. The *math co-processor* was a processing unit that could only process complicated mathematics, but perform them much faster than the CPU could.

By installing a math co-processor into a computer system, such tasks were diverted to the co-processor for a significant overall speed gain. Modern CPUs include circuitry to perform these complicated "floating-point" mathematics tasks that were once left to a co-processor.

The most common example of co-processing in today's computers is in graphics processing: add-in boards containing GPUs (**Graphics Processing Units (http://en.wikipedia.org/wiki/Graphics_processing_unit)**) can process 3D visual effects and other graphics operations much faster than any general-purpose CPU. Common examples of such chips in use today include nVidia's **GeForce (http://www.nvidia.com/page/desktop.html)** and AMD's **RADEON (http://www.amd.com/us/products/Pages/graphics.aspx)** families of graphics processors.

Other recent examples of co-processing are **Physics Processing Units (http://en.wikipedia.org/wiki/Physics_processing_unit)** , a processor tailored to the task of calculating physics problems. The most common application of these co-processors are in modern video game consoles and high end gaming-specific graphics cards.

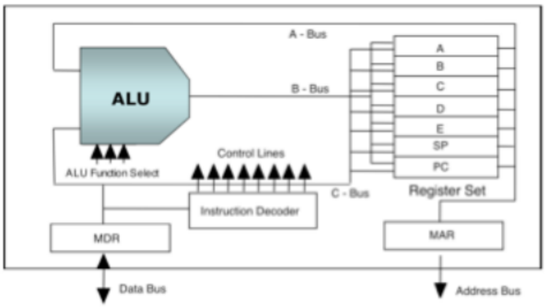
CPU

- Central Processing Unit (CPU) fetches, decodes and then executes instructions that perform

  – arithmetic

– logic comparisons

- e.g. “is number1 equal to number2 ?” – other operations
- e.g. “skip the next 50 instructions”



- Arithmetic Logic Unit (ALU) performs arithmetic and logic functions
- Registers - high speed‘scratch pad’to store data currently being processed
- Memory Buffer Register (MBR) – stores data just received from, or about to be written to memory
- Memory Access Register (MAR) – stores address of memory to be accessed next
- Program Counter (PC) – stores address of next instruction

Memory

- Each byte of memory has an address

  – numbered sequentially

  – individual bits not addressable, just (usually) bytes

- An address length of N bits can express 2^N numbers $(0..2^N-1)$

  – so maximum size of memory limited by length of address