How a CPU Works http://www.hardwaresecrets.com/how-a-cpu-works/

The CPU (Central Processing Unit) – which is also called microprocessor or processor – is in charge of processing data. How it will process data will depend on the program. The program can be a spreadsheet, a word processor or a game: for the CPU it makes no difference, since it doesn't understand what the program is actually doing. It just follows the orders (called commands or instructions) contained inside the program. These orders could be to add two numbers or to send a piece of data to the video card, for example.

When you double click on an icon to run a program, here is what happens:

1. The program, which is stored inside the hard disk drive, is transferred to the RAM memory. A program is a series of instructions to the CPU.
2. The CPU, using a circuit called memory controller, loads the program data from the RAM memory.
3. The data, now inside the CPU, is processed.
4. What happens next will depend on the program. The CPU could continue to load and executing the program or could do something with the processed data, like displaying something on the screen.

Figure 1: How stored data is transferred to the CPU.

In the past, the CPU controlled the data transfer between the hard disk drive and the RAM memory. Since the hard disk drive is slower than the RAM memory, this slowed down the system, since the CPU would be busy until all the data was transferred from the hard disk drive to the RAM memory. This method is called PIO, Processor I/O (or Programmed I/O).

Nowadays data transfer between the hard disk drive and the RAM memory in made without using the CPU, thus making the system faster. This method is called bus mastering or DMA (Direct Memory Access). In order to simplify our drawing we didn't put the north bridge chip between the hard disk drive and the RAM memory on Figure 1, but it is there. If you'd like to learn more about this subject, we've already written a tutorial on that.

Clock: So, what is clock anyway? Clock is a signal used to sync things inside the computer. Take a look on Figure 2, where we show a typical clock signal: it is a square wave changing from "0" to "1" at a fixed rate. On this figure you can see three full clock cycles ("ticks"). The beginning of each cycle is when the clock signal goes from "0" to "1"; we marked this with an arrow. The clock signal is measured in a unit called Hertz (Hz), which is the number of clock cycles per second. A clock of 100 MHz means that in one second there is 100 million clock cycles.

Figure 2: Clock signal.

In the computer, all timings are measured in terms of clock cycles. For example, a RAM memory with a "5" latency means that it will delay five full clock cycles to start delivering data. Inside the CPU, all

instructions delay a certain number of clock cycles to be performed. For example, a given instruction can delay seven clock cycles to be fully executed.
PARTE 2:

Block Diagram of a CPU

On Figure 6 you can see a basic block diagram for a modern CPU. There are many differences between AMD and Intel architectures and we plan to write specific articles about them in the near future. We think that understanding the basic block diagram of a modern CPU is the first step to understand how CPUs from Intel and AMD work and the differences between them.

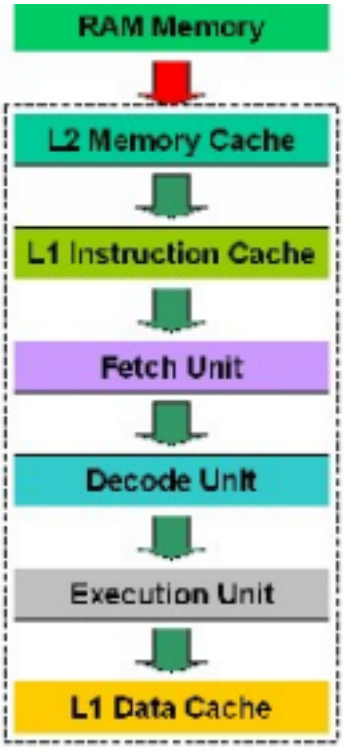

Figure 6: Basic block diagram of a CPU.

The dotted line on Figure 6 represents the CPU body, as the RAM memory is located outside the CPU. The datapath between the RAM memory and the CPU is usually 64-bit wide (or 128-bit when dual channel memory configuration is used), running at the memory clock or the CPU external clock, which one is lower. The number of bits used and the clock rate can be combined in a unit called transfer rate, measured in MB/s. To calculate the transfer rate, the formula is number of bits x clock / 8.

For a system using DDR400 memories in single channel configuration (64 bits) the memory transfer rate will be 3,200 MB/s, while the same system using dual channel memories (128 bits) will have a 6,400 MB/s memory transfer rate. For more information on this subject, read our tutorial Everything You Need to Know About DDR Dual Channel.

Memory Cache

Memory cache is a high performance kind of memory, also called static memory. The kind of memory used on the computer main RAM memory is called dynamic memory. Static memory consumes more power, is more expensive and is physically bigger than dynamic memory, but it is a lot faster. It can work at the same clock as the CPU, which dynamic memory is not capable of.

Since going to the “external world” to fetch data makes the CPU to work at a lower clock rate, memory cache technique is used. When the CPU loads a data from a certain memory position, a circuit called

memory cache controller (not drawn on Figure 6 in the name of simplicity) loads into the memory cache a whole block of data below the current position that the CPU has just loaded. Since usually programs flow in a sequential way, the next memory position the CPU will request will probably be the position immediately below the memory position that it has just loaded. Since the memory cache controller already loaded a lot of data below the first memory position read by the CPU, the next data will be inside the memory cache, so the CPU doesn't need to go outside to grab the data: it is already loaded inside in the memory cache embedded in the CPU, which it can access at its internal clock rate.