Chapter 7

Computer Architecture

Anybody who uses a computer ought to have an understanding of basic architectual components such as memory, registers, caches, and buses.

The actual data that a computer works with is stored as binary digits (0's and 1's) on memory chips. This collection of chips is often denoted *main memory* (or DRAM in the PC ads). The minimum number of bits (binary digits) that can be accessed by one request to memory is known as a *word*. Word sizes are almost always one *byte* (meaning eight bits) long. Single precision numbers are usually four words long, while double precision are eight words long. Computers access multiple words simultaneously.

If M is the total number of words available on a machine, then a unique word may be specified by an address from 0 to M-1.

If words are always eight bits wide, what is all this media fuss about 32 bit technology, or 64 bit technology? These numbers (32 or 64) represent $\log_2(M)$, i.e., with 64 bit technology, there are potentially $M = 2^{64}$ words available on the machine. The truth is that there are often fewer words actually available, but these many are at least potentially available. In a given United States area code, we might say we have seven digit "technology", but not all 10^7 numbers have been given away at any one time. It is always good to leave room for more. Furthermore, some memory addresses may not point to words in memory at all, but rather to other components where you may want to send bits.

To actually work with data, a processor must load it into its own *registers*. Registers are special storage areas on a processor that may be quickly accessed.

Words travel from memory to registers via *buses*. A bus is a piece of hardware that allows the various components of a computer to talk to each other.

It was well understood in the 1960s that the time taken to load data from memory into registers slowed down machine execution time. Most memory simply can not release the data fast enough for the processor to work with the data. Therefore a device designed to accelerate the average transfer time called a *cache* was invited.

Caches are small amounts of fast memory which contains copies of a small subset of the data in main memory. When a processor wants to load data from memory, the cache is checked to see if the data is there. If so we get a fast load which is called a *cache hit*. If not, we get a slow load called a *cache miss*. When a cache miss occurs, a contiguous block of data containing this word of data is loaded into cache anticipating that having the data and its neighbors in cache is likely to increase the chance of a cache hit the next time.

Therefore, it is of utmost important to design algorithms that exhibit this sort of locality whenever possible.