Chapter 4: Introduction to Computer Architecture

A computer generally consists of three subsystems: the CPU, memory, and the input/output subsystem. These subsystems communicate with each other using buses. Physically, a bus is just a set of wires. The buses connect the various components of a computer together. Buses, in larger systems, become more efficient than direct connections between components, because of their smaller size and smaller energy drain on the system. Any given system may have a hierarchy of buses.

The processor sends out the address to be accessed in memory or the address of the I/O device via the address bus. The data bus carries data between the subsystems. Information sent on the control bus coordinates all data transfers.

To process an instruction, the processor goes through an instruction cycle. The **instruction cycle** is the procedure a microprocessor goes through to process an instruction. This, in turn, consists of cycles to fetch an instruction from memory, decode the instruction, and execute the instruction. All instructions are fetched and decoded in the same way, but the execute cycle is different for every instruction.

The CPU contains three sections. The register section is used for data storage. It contains the registers of the processor’s ISA as well as other registers not directly accessible by the programmer. The arithmetic/logic unit (ALU) performs computations on data with the CPU. Since it must complete its operations within a single clock cycle, it is constructed using only combinatorial logic. The control unit outputs signals to control the rest of the processor. By outputting the control signals in the correct order, it causes the processor to fetch, decode, and execute instructions.

There are several types of memory chips used in computer design. The data in read only memory chips, including **ROM**, **PROM** (programmable ROM), **EPROM** (erasable PROM), and **EEPROM** (E2PROM, or electrically erasable PROM), is fixed; their data does not change, and remains valid when the computer is turned off. A common usage of EEPROMs is the Basic Input/Output System, or BIOS, of personal computers. The computer can modify the contents of random access memory, SRAM (static RAM), and DRAM (dynamic RAM); its contents are lost when power is turned off. Dynamic RAM chips are like leaky capacitors. As it reads data, filling it, it slowly “leaks” out and would eventually be unreadable, but a refresh circuit reads the contents occasionally and rewrites it to its original place, restoring the memory to its maximum “charge”. Static RAM is more like a register. It writes its data, and it stays valid. SRAM is faster than DRAM, but it also more expensive. Cache memory in personal computers is constructed from SRAM.

Internally, all memory chips are organized as linear arrays or multi-dimensional arrays. The simplest organizations are linear organizations. Decoders select the memory location specified by the memory chip’s address inputs. As the number of locations increases, the size of the address decoder needed in a linear organization becomes prohibitively large. To remedy this, using a two-dimensional organization can be used.

Memory chips can be combined to increase the number of bits at an address and/or the number of addresses in the memory subsystem. These configurations come by the way of **high-order interleaving** and **low-order interleaving**. All memory locations within a chip are contiguous within system memory for high-order interleaving. Low-order interleaving can offer some speed advantages for pipelined memory access, and for CPUs that can read data from more than one memory location simultaneously.

Multibyte data organization comes in two types: **big endian** and **little endian**. In big endian format, the most significant byte of a value is stored in location X, the following byte in location X + 1, and so on. In little endian, the order is reversed. The least significant byte is stored in location X, the next byte in location X + 1, and so on. The same organizations can be used for bits within a byte. Neither organization has an impact on CPU performance, as long as the CPU is designed to use one over the other. One important thing to consider when designing for multibyte data is alignment. Most modern microprocessors can read multiple positions at once. It’s important to make sure multibyte data begins at the beginning read position where a processor will start reading data. Some CPUs require all data to be aligned, while others do not; they usually align data internally. Nonaligned CPUs have more compact programs, because no locations are left unused by alignment. Aligned CPUs can have better performance because they may need fewer memory read operations to fetch data and instructions.

A computer system can use one of two types of I/O access. Memory mapped I/O treats an I/O device as if it were a memory location. The same instructions are used to access both memory and I/O devices, but memory and the I/O device cannot both use the same address. Isolated I/O uses different instructions to access I/O devices and memory. It requires an additional control line to distinguish between the two, but this allows memory and I/O devices to both use the same addresses.