Computer Architecture and Functionality

Computer Architecture

A computer is made of at least a CPU and memory.

The memory stores the state of the computer while the CPU is the brains that is doing the work.

CPU

The CPU is comprised of a number of different parts with different functions.

The Arithmetic logic operator performs operations such as addition and XOR logic operations,

The Control unit contains the circuitry that coordinates the machine's activities. The control unit extracts the program from memory and then decodes and executes instructions sequentially.

The Registers store the data for inputs for the Arithmetic logic operator (ALO) to use. There a few types of registers but the general-purpose registers store most input data. The registers act as temporary memory only storing what is required for the execution of the next instruction.

The Bus connects the main memory to the CPU. The bus transfers data in parallel. The CPU can load and store from memory via the Bus. The Bus transfers data from the running program into the registers. The ALO operates on the data in the registers and can also store outputs in the registers. Outputs from operations can then be copied across to the memory.

Memory

The state of the computer is stored in the memory. The Memory is a volatile storage solution that requires a constant electrical connection to keep its state. External storage such as hard drives can be used to store permanent data that will be maintained after the machine restarts.

Data stored in memory is encoded in binary. Data has to be stored in binary as the smallest unit of storage, a flip-flip, can only store two possible values.

Flip-flops

A flip-flop is a component that can store its binary state as long as it is powered.

A Boolean operation is an operation that takes a Boolean input (a series of true or false/, 0 or 1s) and produces a Boolean output. Common Boolean operations include AND, OR and XOR. AND returns true if both inputs are true, otherwise, it returns false. OR returns true if either of the inputs are true. And XOR (exclusive or) only returns true if inputs are different. Hardware components that perform these logic operations are called logic gates. A NAND (not and) gate will return the opposite of the result of an AND gate.

A flip-flip is made of a pair of either NAND or NOR logic gates where the output of each feeds into the input of the other. As such the new state is based on the output of the other previous state and hence can store a value. Flip-flops can be active high or active low (a high or low current sets the value). When the value is set it is maintained until a reset signal is sent.

Storing data

A flip-flop can store a single bit of data, either a one or a zero. Data in memory is accessible by byte increments. A byte is a collection of 8 Bits. A bit can store 2^1 possible bits of information (1 bit), and a byte can store 2^8 bits of information (256 bits). Every byte is located at an address. Memory addresses are accessible as a continuous sequence 0,1,2,3,4,5..... . Memory being accessible, but not necessarily stored, sequentially allows the users to store information that requires more than 1 byte of memory. For example, by grouping data sequentially two bytes can be used for 16 bits of storage. As data can only be stored in binary, the addresses for memory locations can also only be stored in binary. This means that only addresses less than 2^n can be addressed. If an address pointer is 8 bits then even if the memory has capacity for more storage, only the first 256 bits can be accessed as the rest cannot be pointed too. This is why memory goes up in powers of 2. Memory is often referred to as Random-Access Memory (RAM) due to the fact any byte can be accessed in any order.

Mass Storage

The computers RAM cannot retain data without a power supply, Non-Volatile Memory is required. Most computers use a form of Mass Storage as Non-Volatile memory. As the name suggests Mass Storage devices can hold large amounts of data comparable to RAM.

Read and write operations to mass storage is slower than RAM. Mass storage devices connect to the system via the Bus. Multiple periphery devices connected to the Bus must take it in turns to communicate to the CPU. If too many devices attempt to communicate with the CPU at once the BUS can become overloaded and communication errors can occur.

There are a variety of Mass Storage device types. Two common storage types for computers and mobile devices are:

- Magnetic Disk Storage – These are common in desktops. This type of storage writes to spinning magnetic discs. MDS's are relatively slow to read and write due to their mechanical nature. This form of storage is cheap and has a high reliability if not knocked.
- Solid State Drives – SSD's are built on Flash-Technology. Flash tech requires no moving parts and as such is suited to laptops and mobile devices. Flash-Technology is fast to read and write to but is relatively expensive. Flash-Technology stores data using electrons in circuits to store data. Flash devices can store data for a few years without power, as such flash storage is not as reliable as Magnetic Disk Storage.
- Portable Storage Solutions – Many other storage types exist such as CD's, Cassettes and Punch cards.

Representing data with bits

In order to store data in bits, the data must be encoded. Different data types are encoded in different ways that are optimal for the specific type.

The Twos Complement encoding protocol is commonly used to store integers as it allows for convenient negative numbers and only has a single 0.

Unicode is commonly used to encode text; it is an extension of the ASCII code which supports more characters.

Data Protection

Data left in storage or transferred around the system can become corrupted. Static in wires can cause data corruption and flip-flops can lose the state of their bits.

A single flipped bit can cause unknown behavior.

To increase the stability of systems several fascinating techniques have been implemented.

Binary encoding techniques

The two techniques detailed here provide mechanisms to detect when binary data has become corrupted.

1. Parity Bits – Any sequence of bits will have either an odd or even number of 1's. As such by adding a single 1 or 0 we can ensure that all sequences have an odd or even number of 1's. If we know all sequences should have an even number of ones then we know that any sequence that has an odd number is corrupt. This additional bit is added to each byte and is called a parity bit. The usage of parity bits, however, cannot distinguish sequences where an even number of bits has flipped.
2. Error-Correcting Codes – An error-correcting code, is data encoded in such a way that means even if a small number of bits are flipped, the original message can still be recovered.