CSCI 120 Introduction to Computation
Bits... and pieces (draft)

Saad Mneimneh
Visiting Professor
Hunter College of CUNY

1 Yes No Yes No... I am a Bit

You may recall from the previous lecture that the use of electro mechanical relays, and in subsequent years, diodes and transistor, made it possible to construct more advanced computers, e.g. ENIAC. This is accredited to the fact that these devices could function as on/off switches. On one hand, they create the ability to encode logic into the circuits of the computer. This means that the computer can perform different tasks under different conditions, i.e. the notion of a program. For instance, one could encode the logic *if A OR B then C*. On the other hand, these devices allow the engineers to worry less about the values that could possibly arise in the system: the switch is either on or off. It cannot be anything in between. Therefore, this means that any errors due to fluctuation in voltage levels are greatly reduced. It would be enough to simply distinguish between high voltage and low voltage. This brings us to the question of Analog versus Digital.

In simple terms, a digital system encodes information using a number of devices that have discrete states (e.g. on/off switches). An analog system encodes information using a device that have continuous states (e.g. measurement in an electric circuit).

To build an intuition for digital versus analog, consider the problem of encoding a number using buckets of water. One possibility is to use two kinds of buckets, full and empty. The buckets will be arranged in a special way to encode the number, a strategy that we may agree upon. Another possibility is to use one bucket only, and fill it up to a level proportional to that number. Although the analog system seems to be more accurate, consider what happens if the buckets moved. Any movement of the bucket in the analog system would alter the level of water and hence cause an error in detecting the encoded number. In the digital system, however, a full bucket remains full, and an empty bucket remains empty (even if a small amount of water moves around).

We have encountered an example of digital versus analog before when we examined the abacus and the slide rule. In the following figure, try to identify what is digital and what is analog:

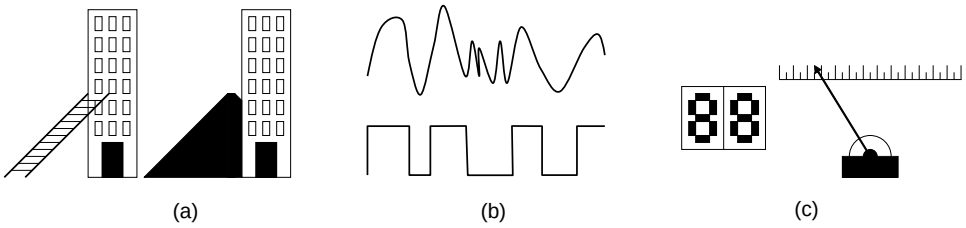


Figure 1: Digital versus Analog

Computers are digital. They store information using *Bits*. A bit is either 1 (ON) or 0 (OFF). These bits actually represent high voltage and low voltage values, respectively. Circuitry inside a computer needs only distinguish between these two. That's why they are called digital circuits. To that matter, computers answer everything using YES NO answers only. A bit of 1 is a YES. A bit of 0 is a NO. They form meaningful answers by putting many bits together, i.e. by combining many YES NO answers. We will see later how to encode numbers using bits for instance.

2 Boolean operators

A convenient way of interpreting bits is by assigning TRUE to 1 (high voltage) and FALSE to 0 (low voltage). This makes it intuitive to talk about logical building blocks such as AND, OR, and NOT. These are also called Boolean operators, after the mathematician George Boole who worked in the field of mathematical logic. For instance, since a statement of the form X AND Y, where X is a statement and Y is a statement, is true only when X and Y are both true, we conclude that 1 AND 1 should be 1.

AND	0	1
0	0	0
1	0	1

Similarly, a statement of the form X OR Y is true if at least one of them is true, for example, 1 OR 0 should be 1.

OR	0	1
0	0	1
1	1	1

The operation NOT differs from AND and OR because it has only one input. The statement NOT X is FALSE if X is TRUE, and TRUE if X is FALSE. Therefore,

$NOT\ 0 = 1$
$NOT\ 1 = 0$

Let's see how we can implement these Boolean operators using diodes. Recall that a diode acts like a switch, so before we start, let's agree on a convention. Referring to the figure below, a diode will conduct if the voltage $V_1 - V_2$ is high enough.

Figure 2: Diode conducts if $V_1 - V_2$ is high

Here's a circuit that implements OR.

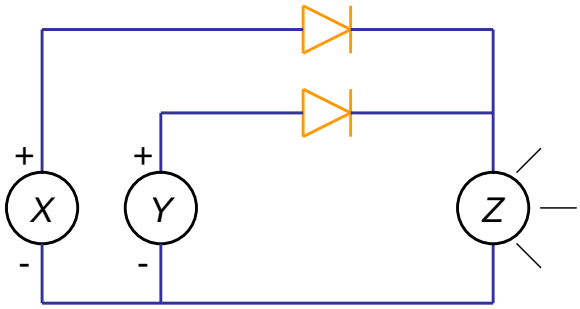


Figure 3: A circuit for OR

- If $X = 0$ (low) and $Y = 0$ (low), then both diodes do not conduct according to the convention of Figure 2. Therefore, no current goes through and Z is OFF. Therefore, if $X = 0$ and $Y = 0$, then $Z = 0$.
- If $X = 0$ (low) and $Y = 1$ (high), then the upper diode will not conduct, but the lower diode will conduct (see Figure 2). The current goes through and Z is ON. Actually, this argument is true if either $X = 1$ or $Y = 1$, because at least one diode will conduct. Therefore, whenever X or Y is 1, $X = 1$.
- if $X = 1$ (high) and $Y = 0$ (low), see above argument.
- If $X = 1$ (high) and $Y = 1$ (low), see above argument.

We conclude that $Z = X\ OR\ Y$.

Now let's construct a circuit that implements AND.

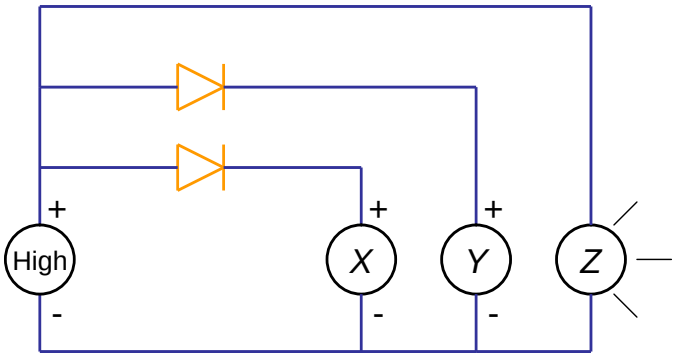


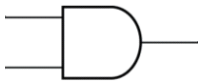
Figure 4: A circuit for AND

- If $X = 0$ (low) and $Y = 0$ (low), then both diodes conduct according to the convention of Figure 2. The makes a short circuit and the current does not go through Z, making Z OFF. Actually, this argument is true if either $X = 0$ or $Y = 0$, because at least one diode will create a short circuit.Therefore, whenever X or Y is 0, $Z = 0$.
- If $X = 0$ (low) and $Y = 1$ (high), see above argument.
- If $X = 1$ (high) and $Y = 0$ (low), see above argument.
- If $X = 1$ (high) and $Y = 1$ (high), then both diodes do not conduct (see Figure 2). Therefore, the current goes through Z, making Z ON. Therefore, if $X = 1$ and $Y = 1$, then $Z = 1$.

We conlude that $Z = X\ AND\ Y$.

What about NOT? Can we construct a circuit for NOT similar to those of AND and OR. The answer is NO. It is not possible to construct the NOT logic using diodes and resistances only. Transistors must be used. Transistors are more complicated than diodes; for instance, they can be formed out of PNP junctions. We are not going to worry about that. In fact, we should not be worrying about these circuits at all. Well, at least not for the purpose of this course. But we can afford to do that because we are going to bring in the power of abstraction.

Recall the concept of abstraction? Once I know how something works, I can use it without worrying much about the detail. So we are going to treat those circuits as building blocks, or black boxes. We call them gates. Therefore, we have the AND gate, the OR gate, and the NOT gate. We can even form combinations of these gates and give them new names. For instance an AND gate followed by a NOT gate is known as a $NAND$ gate (short for $NOT\ AND$, i.e the result of the AND operation inverted). Standard gates like AND, OR, NOT, and others, have special symbols so that they can be easily expressed in the design of digital circuits.



inputs		output
0	0	0
0	1	0
1	0	0
1	1	1

Figure 5: AND gate

inputs		output
0	0	0
0	1	1
1	0	1
1	1	1

Figure 6: OR gate

input	output
0	1
1	0

Figure 7: NOT gate

Now we can use those gates to build larger digital circuits, such as a flip-flop shown in Figure 8, with inputs X and Y, and output Z.

A flip-flop is a circuit that produces an output value of 0 or 1, which remains constant until a temporary pulse from another circuit (its inputs) causes it to shift to the other value. The output will flip or flop between 0 and 1 under the control of external signals. For the flip-flop of Figure 8, as long as both inputs $X = 0$ and $Y = 0$, Z (whether 0 or 1) will not change. By temporarily making $X = 1$, we force $Z = 1$. By temporarily making $Y = 1$, we force $Z = 0$. You can trace the behavior of this circuit yourself, but I will go over it in detail in class.

Now that you know how to construct a flip-flop, you can use it as a black box. Indeed, this is so true because Figure 8 is not the only way of constructing a flip-flop. So your flip-flop may look nothing like Figure 8. Fortunately, as long as you can abstract the function of a flip-flop, you can use it in your design without worrying too much about the detail. Someone else will build the flip-flop for you.

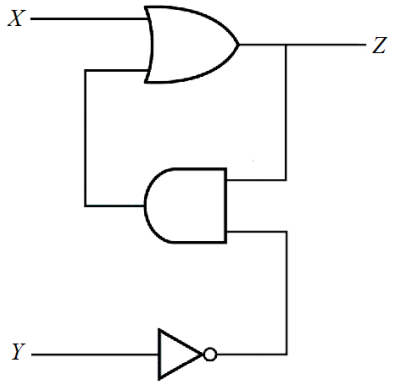


Figure 8: A flip-flop

Now the question is: why is a flip-flop useful? Well, what can you say about someone who, given a value, will remember it forever, until instructed to change it? Must have good memory! And computers need memory. So the flip-flop is one means of storing a **single bit** within a computer (recall that computers speak bits). All the bits constitute what is known as the machine's main memory.

3 Main memory

A computer's main memory is organized in untis of 8 bits each, called bytes. A number of bytes constitute a word. This number depends on the machine.

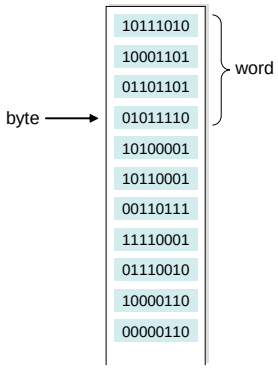


Figure 9: Main memory

For instance, in a 32 bit machine, each word consists of 4 bytes. Recall that the control unit of the CPU fetches instructions and data from the main memory. Although a byte is the standard unit, instructions and data are fetched from memory at the word level. Therefore, in a 32 bit machine, a number is represented by 32 bits (we will explore the bit representation of numbers in more detail later). Generally, a larger word size implies faster and more accurate computation, since more bytes can be fetched simultaneously from the memory and larger numbers can be represented. In the last figure of the previous lecture, the width of the bus connecting the memory, the CPU, and to the input/output

controller is equal to the word size. Nevertheless, bytes, and even bits, can still be accessed if needed depending on the type of instructions executed by the control unit of the CPU.

Each byte has a unique address. The byte can be accessed by presenting the address to the memory module (we will probably not cover the circuitry that enables this). An important feature is that the addresses of bytes are consecutive. For instance, the address of the first byte is 0, the second is 1, the third is 2, etc... This allows to store data that spans over more than one byte while being able to retrieve it easily (we don't have to scatter it). To reflect the ability to access bytes individually in any order, the computer's main memory is often called Random Access Memory (RAM).

Although we have mentioned that the flip-flop is a means of storing bits (of each individual byte), the RAM in modern computers is constructed using other technologies. Many of these technologies store bits as tiny electric charges that dissipate quickly (forgetful memory!). Therefore, these devices require additional circuitry that repeatedly *refreshes* the charges many times a second. Such memory is called dynamic random access memory (DRAM). We also refer to Synchronous DRAM (SDRAM) when additional techniques are used to decrease the time needed to retrieve the content from memory.

As you may suspect by now, memory capacity is measured in terms of bytes. You may have heard the terms kilobytes KB, megabytes MB, gigabytes GB, TB terabytes, etc... 1 KB = 1024 bytes (not 1000 as the term kilo would normally mean). This is due to the fact that the number system used in computers is the binary system. Therefore, $2^{10} = 1024$ is more naturally represented than 1000. Similarly, 1 MB = $1024 \times 1024 = 2^{10} \times 2^{10} = 2^{20} = 1048576$ bytes (close to but not exactly 1000000). 1 GB = $2^{30} = 1073741824$ bytes. It is common to have a personal computer with 512 MB of memory these days. Mass storage devices, like a hard disk for instance, can have up to 100 GB, or even more.

4 Mass storage

Main memory is volatile. When the computer is switched off, everything is forgotten (after all it is electric charges). Therefore, we rely on additional memory devices called mass storage, or secondary storage devices. These include magnetic disks, optical disks, and flash memory. Secondary storage devices have two main characteristics:

- They have much larger capacities than main memory, so they provide a sort of permanent storage for information that can be retrieved any time.
- They typically require mechanical motion and, therefore, require significantly more time to store and retrieve data than main memory (which is purely electronic).

4.1 Magnetic disks (hard disk)

Most computer systems have secondary storage based on magnetic disks (hard disks). The amount of such secondary storage often exceeds the amount of main memory by at least a factor of 100. But there must be an economic equilibrium. Therefore, main memory is 100 times more expensive per bit than magnetic memory. In other words, the electronic bit is 100 times more expensive than the magnetic bit.

The figure below shows a typical hard disk. The disk consists of several platters, which rotate **at constant speed** around a common spindle. The surface of each platter is covered with a magnetizable material. Each platter is read or written by the head at the end of an arm. The arms are physically attached together, and they can move their heads toward or away from the spindle. When a given head is stationary, the surface that passes underneath it is called a track.

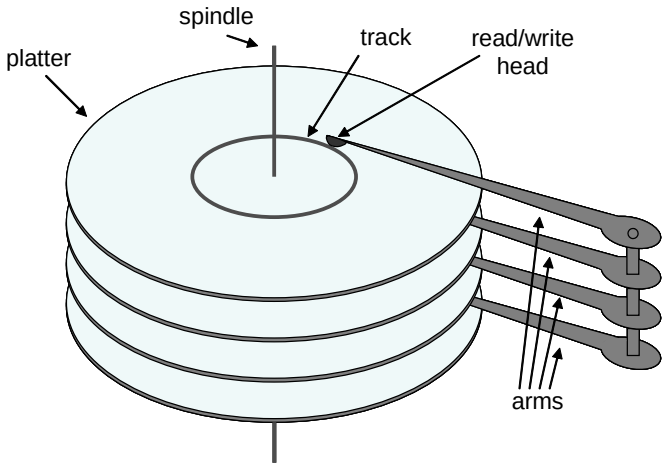


Figure 10: Hard disk

The read/write heads are vertically aligned at all times, and therefore, the set of tracks underneath them are accessed simultaneously. Such set of tracks is known as cylinder. This construction should justify why hard disks are much, much slower than main memory. There are two components to the mechanical motion: platter rotation and arm movement. As of today, disks rotate at speeds of 5400-15000 revolutions per minute (RPM), with 7200 RPM being the most common. Although 7200 RPM may seem fast, one rotation takes 8.33 milliseconds (this is known as the **rotation delay**), which is almost 100000 times longer than the typical 100 nanosecond access times of main memory. So imagine this: if we have to wait for a full rotation for a particular item to come under the read/write head, we could access main memory 100000 times during that span! On average, we only have to wait for half a rotation, but this is still significant. Moving the arms also takes some time, known as the **seek time**. As of today, average seek times for hard disks are in the range of 3 to 9 milliseconds. In order to benefit from the time spent waiting for the mechanical movements, disks access not just one item but several at a time. Information is divided into a number of equal sized chunks, called *pages* or *sectors*, of bits that appear consecutively within the cylinders, and each disk read or write is of one or more entire pages. For a typical disk, a page might be 10KB to 16KB in length. Once the read/write head is positioned correctly and the disk has rotated to the beginning of the desired page, reading or writing a magnetic disk is entirely electronic (except for the disk rotation), and large amount of data can be read or written quickly.

4.2 Optical disks (CD, DVD)

Another class of mass storage is based on optical technology. A famous example is the compact disk CD. CDs are 12 centimeters in diameter and consist of reflective material covered with a clear protective coating. Information is created on them by creating variations in their reflective surfaces (that's why we say burning a CD). This information can then be retrieved by means of a laser beam that monitors irregularities on the reflective surface of the CD as it spins. CDs were originally used for audio recordings. CDs used today for computer data storage use essentially the same format. In particular, information on a CD is stored on a single physical track which spirals from the inside out.

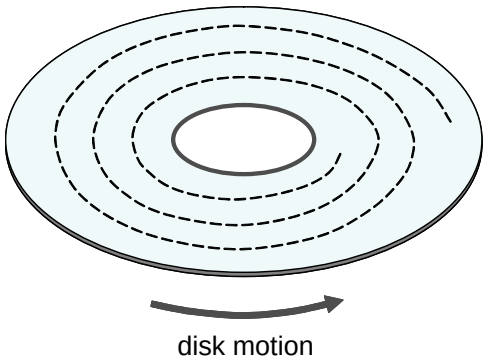


Figure 11: CD

The track is divided into units (called sectors) of 2KB of data, which is equivalent to 1/75 of a second of music in case of audio recordings. Unlike magnetic disks where all tracks are circular, the distance on the track is greater towards the edge of the CD than to its center. This means that more information can be stored toward the edges. For audio CDs, players must be able to rotate the CD at different speeds in order to play music at a constant rate. However, most CD systems used for computer data storage spin at a faster, constant speed, and thus must accommodate variations in data transfer rates.

As a consequence, CD storage systems perform best when dealing with long continuous strings of data, such as music. Traditional CDs have capacities in the range of 600 to 700 MB. However, newer DVDs (Digital Versatile Disks) provide capacities of up to several GB. These use multiple semi-transparent layers that serve as distinct surfaces when viewed under a precisely focused laser. These disks are usually used to store multimedia presentations, including entire movies. CDs and DVDs are now readable and writable. Here's a table that summarizes the storage capacity of different technologies:

CD	600-700 MB
DVD SL (single layer)	4.7 GB
DVD DL (dual layer)	8.5 GB

Here's a table that summarizes the read/write capabilities of different formats:

CD-ROM	non-recordable
CD-R	recordable once (burning)
CD-RW	rewritable, whole CD must be erased before rewriting
DVD-ROM	non-recordable
DVD-R	recordable once (burning)
DVD+R	recordable once (burning)
DVD-RW	rewritable, whole DVD must be erased before rewriting
DVD+RW	rewritable without erase (like hard drive)

DVD-R is a format developed by the DVD Forum in 1997. DVD+R (plus R) is developed by a coalition of companies in 2002 (Dell, HP, Mitsubishi Chemical Corporation, Philips Electronics, Ricoh Company, Ltd., Sony Corporation, Thomson multimedia, Yamaha Corporation). As a consequence, DVD-R is retroactively called DVD minus R. The same coalition produced the DVD+RW format. Although DVD-R was developed five years earlier, the market shows little sign of settling down in favor of DVD-R or DVD+R. The main reason is the availability of DVD readers that can handle all the formats. A DVD reader can usually handle CDs as well. The "random" rewriting capability of DVD+RW makes this format very popular. They do not have to be erased before rewriting (as in CD-RW or DVD-RW), and rewriting can start anywhere (not necessarily at the end of the recording). Rewirtable disks use a phase-changing metal alloy film. This film can be melted by the laser's heat to level out the marks made by the laser and then lasered again to record new data. In theory, rewritable disks can be rewritten few thousands times before their optical material is damaged. See http://www.jegsworks.com/Lessons/lesson6/lesson6-9.htm for additional information.

4.3 Flash memory

Flash memory derives from a type of memory originally known as **ROM** (Read Only Memory). ROM is basically a memory that can be read but not written. Each 1 bit is represented by a diode. Each 0 bit is represented by the absence of a diode. Therefore, the ROM is manufactured with the predetermined bits of 1's (diodes) and 0's (no diodes). When a high enough voltage is applied to the ROM, a diode conducts indicating 1. Where there is no diode, it is therefore 0.

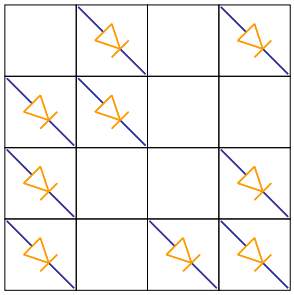


Figure 12: ROM

Another type of ROM is PROM which stands for programmable ROM. In simple terms, this ROM is manufactured with a diode for every bit. Therefore, all bits are originally set to 1. By applying a very high voltage on a particular diode (bit), the diode will burn and stop conducting. This now becomes a ROM. However, the PROM is programmable once. A mistake cannot be undone.

Therefore, working with ROM and PROM can be time consuming. EPROM (Erasable Programmable ROM) is developed to address this issue. Without going into the detail of how it works, EPROM uses transistors to represent bits. When a certain voltage is applied, a transistor acts like an electron gun, and emits electrons which are then trapped into a chamber, giving it a negative charge. A sensor monitors the amount of charge passing through the chamber. A high negative charge is interpreted as a 0; otherwise, it's a 1. To rewrite an EPROM, we must erase it first. To erase it, we must supply a level of energy strong enough to bring back the electrons to their normal level. In a standard EPROM, this is best accomplished with UV light. Therefore, EPROM erasure is not selective. It will erase the entire EPROM to obtain a new EPROM will all 1's again.

Although EPROM is a big step up from PROM in terms of reusability, it still requires dedicated equipment to program. Also, changes to the EPROM cannot be made incrementally; the whole EPROM must be erased. Electrically Erasable Programmable ROM (EEPROM) removes the biggest drawbacks of EPROM. In EEPROM, instead of using UV light, we can return the electrons to normal with the localized application of an electric field to each chamber. This erases the targeted bits of the EEPROM (setting them back to 1), which can now be rewritten as before (reset to 0). The EEPROM is changed one byte at a time, which makes it versatile but slow. In fact, EEPROM is too slow to use in many products that make quick changes to the stored data.

Flash memory is the response to this limitation. Flash memory is a type of EEPROM that uses in-circuit wiring to erase by applying an electrical field to the entire memory, or to predetermined sections of the memory called blocks. Flash memory works much faster than the traditional EEPROM because it writes data in chunks, usually 512 bytes in size, instead of one byte at a time. Today flash drives can hold up to few GB. So why aren't they replacing magnetic or optical disks? The answer is that repeated erasing slowly damages the electron chambers. Therefore, they are not suitable if the content is to be altered many times a second. Moreover, after a certain number of erase operations, chambers can be damaged (don't worry because you are likely to get rid of it before you reach that number).