Bits and Bytes

Here is a sort of glossary of computer buzzwords you will encounter in computer use:

Bit

Computer processors can only tell if a wire is on or off. Luckily, they can look at lots of wires at a time (see buss), and react to a complex pattern of ons and offs in pretty sophisticated ways. To translate these patterns into something that makes sense to humans, we consider a wire that is on to be a "1" and a wire that is off to be a "0". Then we can look at the wires leading into a computer and read something like 00110111 00010000. We don't know what that represents to the processor, it's just a pattern. Each place in the pattern is a bit, which may be 1 or 0. If it means a number to the processor, the bits make up a binary number.

Binary Numbers

Most of us count by tens these days. Ancient cultures used to count by 5s or 12s or 24s, but for the last thousand years, counting by tens has been the norm. when you see the number 145, you just know it includes one group of ten tens, plus four groups of ten, and five more. Ten tens is a hundred or ten squared. Ten hundreds is a thousand, or ten to the third. There's a pattern here. Each digit represents the number of tens raised to the power of the position of the digit, provide you start counting with zero and count right to left.

If you do the same thing with bits that can only be 1 or 0, each position in the list of bits represents some power of two. 1001 means one eight plus no fours plus no twos plus one extra. This is called binary notation. You can convert numbers from binary notation to decimal notation, but you seldom have to.

Bytes

Numbers like 00110111 10110000 are a lot easier to read if you put spaces every 8 bits. In decimal notation, we use commas every three digits for the same reason. There's nothing special about 8 bits, it just kind of got started that way. Hardware is easier to build if you group the wires consistently from one piece to another. Some older hardware used to group wires in 10s, but in the 70s the idea of working in groups of 8 really took over, especially in the design of integrated circuits. Somebody made a joke about a group carrying a byte of the data, and the term stuck. Sometimes you hear a group of four bits called a nibble.

The largest number you can represent with 8 bits is 11111111, or 255 in decimal notation. Since 00000000 is the smallest, you can represent 256 things with a byte. (Remember, a bite is just a pattern. It can represent a letter or a shade of green.) The bits in a byte have numbers. The rightmost bit is bit 0, and the left hand one is bit 7. Those two bits also have names. The rightmost is the least significant bit or lsb. It is least significant, because changing it has the smallest effect on the value. Which is the msb? (Bytes in larger numbers can also be called least significant and most significant.)

Hexadecimal Numbers

Even with the space, 00110111 10110000 is pretty hard to read. Software writers often use a code called hexadecimal to represent binary patterns. Hexadecimal was created by taking the decimal to binary idea and going the other way. Someone added six digits to the normal 0-9 so a number up to 15 can be represented by a single symbol. Since they had to be typed on a normal keyboard, the letters A-F were used. One of these can represent four bits worth, so a byte is written as two hexadecimal digits. 00110111 10110000 becomes 37B0.

Here's a handy table:

Hex	binary	decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

With three different schemes running around, it's easy to confuse numbers. 1000 can translate to a thousand, eight, or four thousand and ninety six. You have to indicate which system you are using. The fact that you still sometimes see an obsolete system called octal (digits 0-7. You can work it out) adds to the potential for confusion. Hexadecimal numbers can be indicated by writing them 1000hex 1000h or 0x1000. Binary numbers can be written 1000bin . Octal numbers were just written with an extra leading 0. Decimal numbers are not indicated, unless there's some possibility of confusion, such as one in a page of hex numbers.

Buss

In electrical systems, a wire that connects to more than two devices is called a buss. Typically you have a power buss that supplies current to all of the parts that need it, and a ground buss that takes the current back to the power supply. (All current paths must be a round trip.)

In computer engineering, the concept of a buss has been expanded to mean a group of wires that carries data around the system. There's usually enough wires to handle one to four bytes. The size of these busses has a big effect on the efficiency of the system. A 32 bit buss can handle numbers twice as long (meaning 2 to the 16th bigger) than a 16 bit buss.

Serial Data

You can send big numbers down a narrow buss if you send it in chunks. If you have an eight bit buss, you can send bytes one after another, and the processor can put the bytes together. This can be down with a single wire buss. Then the bits come one at a time -- this is called serial data transmission.

Memory

A computer wouldn't be much use if it couldn't store data. There have been many schemes for storing data over the years, but the way it's done today involves wiring transistors so they stay on when turned on and stay off when turned off. A transistor can then store a bit. The transistors are organized in groups of 8, so each group can store a byte. A single integrated circuit may have millions of these groups.

Each member of the group is connected to one wire of the data buss. A group can be instructed by some other wires to copy the state of the buss, or to connect their outputs to the buss, so the buss reflects what's in this group. These other wires are in fact a second buss called the address buss. By manipulating the address buss, the central processor can choose which particular group of transistors (or memory location) to read or modify. The number of wires in the address buss determines how many memory locations it could possibly address.

This kind of memory is called RAM for random access memory. Since it depends on transistors to stay on, all data goes away when the power is turned off. Some computers can keep the memory by never really turning off. They have a battery that keeps enough power to the memory transistors that they don't forget.

Another kind of memory is called ROM, for read only memory. There are various types of this, but the most common is like an array of fuses. Any that are blown represent a 0. Nothing can change what's in read only memory, so any program or data in there is available as soon as the computer is turned on.

Drives

Since the memory is cleared when the power goes off, there needs to be some mechanical system for keeping data between jobs. The medium used for storing the data can vary from magnetic tape to optical discs, and some devices allow the media to be easily removed and replaced. Most of these storage systems involve some kind of spinning disc. There is an elaborate scheme for keeping track of the data on a disk – the bytes are grouped into blocks, the blocks into files, the files into directories (or folders), and directories into partitions (or volumes). The user generally only sees files and above.

The Central Processing Unit

The central processing unit, or CPU is the heart of the computer. The CPU reads an instruction from memory (Instructions are bit patterns, just like anything else.), carries it out, and looks for the next instruction. The instructions are simple things like copy a value from memory. The CPU has its own memory locations called registers. Special hardware makes it possible to add or subtract the registers from each other. To add two numbers, the CPU must fetch the first number and put it in a register, fetch the other number and put it in another register, add the two registers, and put the result back into memory. Each of these operations requires an instruction.

Clock

Luckily the CPU can do all of this very quickly. The whole operation is controlled by an oscillator circuit called the system clock, which runs at millions of hertz (cycles per second). It would be simple to think one clock cycle means one instruction, but instructions vary in complexity, and take anywhere from 4 to 20 cycles to complete. Operations are further slowed down by the memory, which has trouble keeping up. Some CPUs have super high speed memory called cache where numbers that are needed a lot can be stored and retrieved more quickly.

Peripheral devices

The CPU communicates with memory via the address and data buss. To communicate with the rest of the world, other buses are used. (Places where external devices can be connected are sometimes called ports.) These busses may be shared or connected to a single device. They may serial or the multi wire type called parallel. Devices connected to the system are called peripherals; this includes keyboards, monitors, mice, graphics tablets, printers, MIDI systems and a lot more. Each has its own kind of data and electrical characteristics, but the connection at the port has to be standardized enough to allow interchange of similar devices. The following are the kinds of connections fond in various systems.

Parallel Port

This an old standard, originally designer for printers, so it's often called the printer port, although other things can be connected here and printers can be connected in other ways. As data ports go, this one is pretty slow.

IDE/ATA

This is a parallel buss designed for bulk data storage devices. This is usually hidden inside the box, since the connectors used aren't very strong. There are wires in the IDE buss that select which device is active, so the logical location of a device (drive A, B and so on) depends on which connector its on.

SCSI

This is another type of parallel buss for bulk storage. It's a lot stronger mechanically than IDE, so it's often used between boxes. SCSI is an evolving standard that is periodically adapted to work at faster speeds. SCSI accommodates seven devices on a buss, and each must have a unique ID number set on its back panel.

SVGA

This is a type of video connector. It's one of many, but the most common right now.

Comm Port

This is a type of serial port that has been around for decades. Another name for it is RS-232, which is the name of a technical document that describes how it should work. It's the slowest port of all. Only very simple devices are connected here.

Modem

One thing often found connected to a serial port is a modem, which is a box that converts data into tones that can be transmitted over the telephone. In many cases a modem is built into the computer, so the modem connection goes right to a phone line.

Ethernet

There are many systems designed to connect computers to each other. Ethernet is one of the most popular because it is very fast and relatively cheap to build. Computers don't connect directly to each other with Ethernet-- they go by way of a box called a hub or switch that allows several computers to talk on a party line. If there are only two, or to use Ethernet to connect a computer to a printer a special cable can be used without a hub.

USB

USB is a new high speed serial system. It's supposed to accommodate up to 128 devices, and allows the devices to be connected without turning the power off. (Fussing with IDE or SCSI with the power on can damage things.)

Firewire

Firewire, also known as IEEE 1394, is an even faster serial system. It's also more reliable than USB for a variety of reasons. There is a contest going on between firewire and SCSI to see which is faster. Firewire is definitely more convenient.

MIDI

MIDI is a communications system designed for musical instruments. It is used to control other things, but music is the main thing. MIDI is discussed at great length elsewhere on this site.

Source: http://www.co-bw.com/Audio_bits_bytes.htm