Notes: Computer Hardware and Digital Imaging
CMP 80c, Winter 2001

Jane Wilhelms, wilhelms@cse.ucsc.edu

1 Development of Computers

Digital electronic computers developed from mechanical adding machines used early in the century. During the 1940's, electronic versions were developed partly for wartime coding and computation uses.

Commercial computers were available by the 1960's. These were called *mainframes*, which is a term still used for large central computers. During the 1960's, smaller *minicomputers* became common, and *supercomputers*, that were even more powerful than mainframes and used parallelization (multiple processors) appeared. In the 1970's, *microcomputers* made computing power available on the desktop or in the home. Computer components became available on tiny silicon chips, which greatly reduced the cost and size of computers, while increasing their speed.

Personal computers (PC's) are modern versions of microcomputers. *Workstations* are more powerful computers that have taken the place of minicomputers. Mainframes and supercomputers are still commonly used for certain applications requiring great speed and storage space. Connecting computers together using *networks* has greatly increased their ease of use. The *Internet* and web-based browser's such as *Netscape* are making information available almost instantly throughout the world in ways never before possible.

The text for this class has an excellent description of the history of computers and computer graphics in Chapter 1.

2 Basic Computer Hardware

The basic building blocks of a computer are the *C.P.U.*, the *memory* and other storage, and the *I/O devices.*

The *C.P.U. (central processing unit)* consists of the *A.L.U. (arithmetic and logic unit)* and the *controller.* The CPU does the real work of the computer. Sometimes computers have more than one of them, and some of them may be specialized. The speed of the CPU is some indication of how quickly it can work, yet other things also affect speed. The speed is usually specified in Megahertz (MHz).

Memory generally refers to some fast *RAM - random access memory* where *programs* and *data* (that programs work on) are stored while they are being used. Usually, anything stored in the RAM memory goes away if the power goes off. Memory is measured in *bytes*, which are 8 *bits*, Bits can be thought of on/off switches. A thousand bytes is a *kilobyte* = *K.* A million bytes is a *megabyte* = *Meg* = *Mb.* A billion bytes is a *gigabyte* = *Gig* = *Gb.* A trillion bytes is a *terabyte.*

Computers also store information for the long-term elsewhere, usually on a *disk drive*, where information does not go away if the power goes off. Programs and data are stored on the disk before used in the RAM memory. Disk drives are sometimes called *hard drives* because the disks are hard. *Diskettes* and *zip disks* are more portable forms of storage, that are small and encased in plastic. Information is also stored on *magnetic tapes* using a *tape drive* and on *CDs* (*Compact Disks*).

I/O Devices are for input and output. Typical input devices are the *keyboard* and *mouse* with its mouse buttons. The keyboard is like a typewriter keyboard but with more keys. Other I/O devices are *digitizing tablets, scanners, light pens, joysticks, data gloves, and track balls.*

The most common output device is the *cathode ray tube* (CRT). This is your computer screen. This is also referred to as *terminal*, *console* or *monitor*. Now, monitors are generally *bit-mapped* and can do complex graphics with multiple windows open, and in color. Other output devices are the *printers*, *plotters*, and *head-mounted displays*.

Computer chips or *integrated circuits* are the technology for the above kinds of devices, particularly CPUs and memory.

A *network* is a communication channel that allows computers to talk to each other. Most of the computers on campus can talk to each other. They can usually also talk to computers elsewhere in the world using the *Internet*. A *modem* allows one computer to connect to another computer, usually through telephone lines. A *wireless* modem allows you to connect one computer to another without telephone lines. The rate at which the data is transferred from one computer to another is described by its *baud rate*. At present the availability of information is hampered by the limited speed of modems and networks. You have no doubt noticed how much faster internet access is on campus, as compared to at home. This is because the campus is connected to high-speed networks, whereas your home computer uses telephone lines. Faster *DSL* connections partially remedy this.

3 Computer Screens

CRTs are the basic output device for the computer. While paint reflects light, CRTs *emit* light. An output signal from the computer stimulates the *electron gun* at the back of the CRT, and an electron stream is focused and sent to a certain part of the screen. The screen is coated inside with a *phosphor* of a certain color (or colors) and when electrons hit the phosphor, phosphor electrons are raised to a more excited state and then fall back to their normal energy state emitting light of a certain color. Screens are refreshed many times per second (Hertz), usually between 30 and 120. The brightness of the screen depends on the number of electrons that hit it.

On most all computers we use now, the electron gun scans the whole screen in regular patterns (*the raster*), often across each line (*scanline*) and down to the next line. (This depends on the machines, they can scan differently.) It can control the amount of electrons hitting each individual addressable spot on the screen, called *pixels*. The resolution of the screen is the number of pixels across x number of pixels up and down. Usually, the resolution is about 1000 by 1000.

On old *ascii* or *dumb* terminals, usually the gun is set up only to draw characters (letters, numbers and punctuation). The word *ascii* comes from a code set up to represent these characters as numbers in the computer. *Bit-mapped* or *graphics* terminals let you turn on and off each pixel, and draw complex images.

Bit-mapped terminals now run a *user-interface* of some kind that lets you have several *windows* open on the screen, so that you can do different things in different windows. Some of the windows may show pictures.

A computer image is really a value of brightness (if *grey-scale*), or color (if colored) for each pixel on the screen. Images thus have a resolution right-to-left (in x) and top-to-bottom (in y). They are *digital images*, meaning that they can be represented as numbers indicating the brightness or color at each pixel.

The digital image shown on the screen is stored as numbers for each pixel in a part of the memory of the computer called the *frame buffer*. Numbers in the frame buffer are sent to the CRT where they are mapped to amounts of electrons sent to each pixel on the screen. The number of *bits-per-pixel* there are for each pixel represented in the frame buffer indicates how many different intensities can be shown on the screen.

1 bit-per-pixel can only show black (if 0=off) or white (if 1=on). 2 bits-per-pixel can store the numbers 0,1,2, and 3. If we count in *binary*, rather than *decimal*, you only use 0 and 1 before carrying, so these numbers in binary are 00,01,10, and 11. This is four choices. If 0 is black, and 3 is white, then 1 and 2 can be two shades of grey. How many numbers can you represent with 3 bits-per-pixel?

With 8 bits-per-pixel you can represent 256 values. You can always find the number of values you can represent by multiplying 2 by itself that number of times (raising 2 to that power). So, for 1 bit-per-pixel ($2^1 = 2$, or off and on). 8 bits-per-pixel is (2^8) or 256. The machines in the graphlab have 24 bits-per-pixel. Fancier, *high-end* machines may have more.

It is helpful to have twice the number of bits per pixel you would like for a good image, when doing computer animation. Then you can have two frame buffers: one that is shown, and one where the next image is drawn. When the next image is ready, that frame buffer is shown, and the other frame buffer is used for the next image after that. This way, you see a smooth transition between images. This is called *double-buffering*, as opposed to *single-buffering*. If you double-buffer with just 24 bits-per-pixel, each image only has 12 bits, and there is some degradation in image quality.

Color displays have 3 phosphors, in red, green and blue. These are sometimes called *RGB monitors*. Usually the phosphors are arranged as three dots for each pixel, close together. (Look closely at a color TV.) Mixing these three *primaries* makes many more colors, (e.g., R+G= yellow; R+B = magenta; B+G = cyan).

To represent an image in color, usually some bits-per-pixel are used for red, some for green, and some for blue. If you have 8 bits-per-pixel, 3 are red, 3 are green, and 2 are blue. How many choices of blue does that give you? If you have 24 bits-per-pixel (*full-color*), you have 8 bits for each. This is many colors but not nearly all the colors visible in nature. Not all colors seen in nature can be seen on a computer screen, not only because of the limited number of bits, but because the phosphors cannot represent all colors.

Color images can also be represented using a *color table*. Then each number in the frame buffer is an index into a table, where each table entry represents a color as a combination of RGB. This is a good method for less than *full-color* (24 bit-per-pixel) displays.

One problem with digital images is *aliasing*. Each pixel can only be on and off, so diagonal lines appear jagged (this is sometimes called *jaggies*). Solutions to this are called *anti-aliasing*. They help, usually, by blurring the edges, but the problem is fundamentally there and cannot be truly removed.