CSCI 120 Introduction to Computation
Ports and I/O devices (draft)

Saad Mneimneh
Visiting Professor
Hunter College of CUNY

1 Ports

As we have seen in previous lectures, communication between the computer (CPU and memory) and I/O devices such as disks, keyboards, mice, printers, monitors, etc... is normally handled through an intermediate device known as a controller. The controller connects to the I/O device by cables either within the computer case, or to a connector on the back (or side) of the computer case, known as **port**. The I/O device is then attached to the port externally by a corresponding connector.

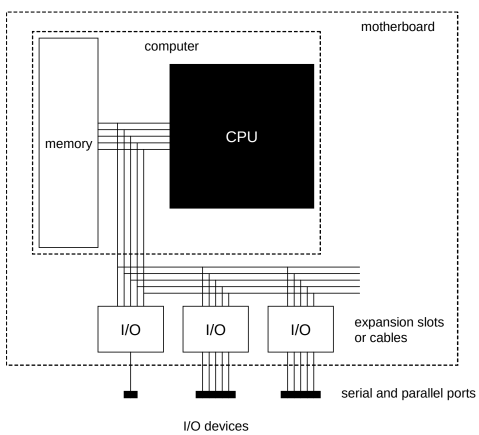

Figure 1: I/O ports

Often, the words "bus", "port", and "connector" are used interchangeably. There are different kinds of ports:

1.1 Serial ports

A serial port connects the I/O device to the I/O controller where transfer of data is performed one bit at a time. Therefore, the original use of serial ports was usually to connect devices that do not require fast data transmission rates, such as mouse, modem (**mo**dulation **demo**dulation device for connecting the computer to the phone line), network by allowing two computers to connect together and exchange data, and printers. The serial port is also referred to as COM port (short for communication) because it was mainly intended to interface with a modem or a similar communication device. Serial ports conform to either the RS-232 standard (25 pins) or the RS-422 standard (9 pins as shown in Figure 2).

Figure 2: Serial port

The 9 pins (signals):

1. carrier detect: determines if the modem is connected to a working phone line
2. receive data: computer receives information sent from the modem
3. transmit data: computer sends information to the modem
4. data terminal ready: computer tells the modem that it is ready to talk
5. signal ground: pin is grounded (low voltage)
6. data set ready: modem tells the computer that it is ready to talk
7. request to send: computer asks the modem if it can send information
8. clear to send: modem tells the computer that it can send information
9. ring indicator: once a call has been placed, computer acknowledges signal (sent from modem) that a ring is detected

RS-422 uses twisted pair wires for each signal allowing more effective communication (faster transmission rate over longer lines). The reason why such communication can be achieved is that twisted pairs reduce the effect of cross-talk and noise:

- Cross-talk (signals leaking between wires in a cable) is minimized: Signals are sent down twisted pair wires such that when one wire in the pair becomes positive the other wire becomes negative by the same amount. Any other wires close to this pair will be affected by cross-talk equal to the sum of the two signals, so if this sum is zero (or nearly zero) then the effect of cross-talk are eliminated.
- Most electrical noise entering into the cable can be eliminated: The receiving electronics are designed to detect only the difference between the two wires in the twisted pair. Since electrical noise affects both wires of a twisted pair equally, the difference eliminates the noise. In the figure below $(A + \epsilon) - (-A + \epsilon) = 2A$.

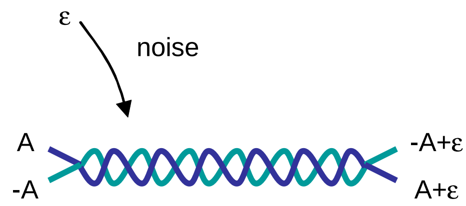

Figure 3: Twisted pair eliminates noise

In recent years, advanced electronics has made economical higher speed serial communication possible, so newer serial communication standards such as USB and FireWire have started to replace RS-232 and RS-422 completely. These make it possible to connect devices that would not have operated feasibly over slower serial connections, such as storage devices, sound devices, and video devices.

1.2 Parallel ports

Unlike a serial port, a parallel port connects the I/O device to the I/O controller where transfer of data is performed with multiple bits *in parallel*, i.e. more than one bit at a time. Originally, parallel ports were developed by IBM as an alternative to the slower speed serial ports and were mainly used for connecting printers. This is why parallel ports are often referred to as printer ports. IBM made the DB 25 connector with 25 pins (see Figure 2) and Centronics made a 36 pin connectors for their printers. IBM engineers later coupled their DB 25 with Centronics 36 in one cable (one connector on each end).

Figure 4: Parallel port

The 25 pins (signals):

- 1: tells printer that data is being sent
- 2-9: used to carry data (8 bits in parallel)
- 10: acknowledge signal from printer to computer that data was received
- 11: printer busy
- 12: out of paper
- 13: printer is online

- 14: computer sends auto feed signal to printer
- 15: printer tells computer there is an error
- 16: new print job ready, initialize printer
- 17: computer remotely takes printer offline
- 18-25: grounded (low voltage)

For the most part, the USB has also replaced the parallel port. As of 2006, most modern printers are connected through a USB connection, and often don't even have a parallel port connection. On many modern computers, the parallel port is omitted for cost savings, and is considered to be a legacy port.

However, disk drives can still connect by other kinds of parallel ports such as SCSI and ATA/IDE.

1.2.1 SCSI, pronounced "skuzzy"

SCSI stands for "Small Computer System Interface". With SCSI, one can daisy chain up to 15 devices together.

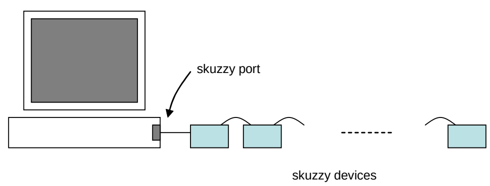

Figure 5: Daisy chaining skuzzy devices (up to 15)

Since its standardization in 1986, SCSI has been commonly used in the Apple Macintosh and Sun Microsystems computer lines. It has never been popular in the IBM PC world, due to the lower cost and adequate performance of its ATA hard disk standard. The introduction of USB, FireWire, and ATA made SCSI a relatively unattractive proposition on PC due to its high cost and rising complexity.

At this time, SCSI is popular on high-performance workstations, servers, and high-end peripherals. Desktop computers and notebooks typically use the ATA/IDE or the newer SATA interfaces for hard disks, and USB or FireWire connections for external devices.

1.2.2 ATA

ATA stands for "Advanced Technology Attachment" and is used for connecting storage devices such as hard disks and CD drives inside a computer. Although the standard has always had the official name "ATA", marketing dictates caused an early version to be dubbed Integrated Drive Electronics (IDE). This is due to the drive controller being contained on the drive itself as opposed to a separate controller connected to the motherboard. The standard following IDE, Enhanced IDE (EIDE) allowed the support of drives having more storage capacity.

The interface at first only worked with hard disks, but eventually an extended standard came to work with a variety of other devices, generally those using removable media. These devices include CD and DVD drives, tape drives, and large-capacity floppy drives such as the Zip drive. The extension bears the name Advanced Technology Attachment Packet Interface (ATAPI), with the full standard now known as ATA/ATAPI.

The original ATA specification used a 28-bit addressing mode. This allowed for the addressing of 2^{28} (268435456) sectors of 512 bytes each, resulting in a maximum capacity of 137 gigabytes. The cable transfers data 16 bits at a time.

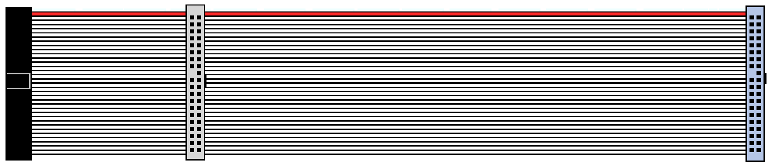

Figure 6: Standard ATA cable with 40 wires

The newer specification (80 wires) allows a 48-bit addressing, for a maximum capacity of 2^{48} sectors of 512 bytes each, i.e. 144 Petabytes (144115188075855872 bytes).

1.3 Serial ATA, SATA

With the market introduction of Serial ATA in 2003, the original ATA was retroactively renamed Parallel ATA (PATA). The first generation of SATA supported a speed of 1.5 Gbps (Giga bits per second). Soon after, enhancements were made to the standard and a speed of 3 Gbps was possible. The Serial ATA International Organization SATA-IO plans to further increase the maximum throughput of Serial ATA to 6 Gbps around the year 2007.

To ease their transition to SATA, many manufacturers have produced drives which use controllers largely identical to those on their PATA drives and include a bridge chip on the logic board. Bridged drives have a SATA connector, may include either or both kinds of power connectors, and generally perform identically to native drives. They may, however, lack support for some SATA-specific features. As of 2004, all major hard drive manufacturers produce either bridged or native SATA drives.

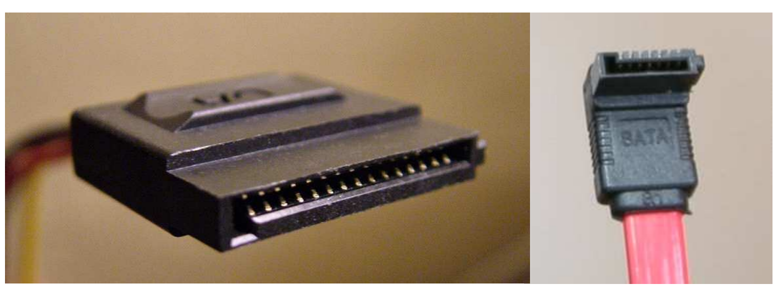

Figure 7: SATA 15 pin and 7 pin connectors

1.4 USB port

A USB port, short for Universal Serial Bus, is used today to connect almost all kinds of devices such as mouse, printer, digital camera, scanner, speaker, MP3 player, CD, DVD, removable hard disk, and flash drives. Desktop computers typically have 6 to 8 USB ports, laptops usually have 2.

Figure 8: A USB connector

The latest version of USB, called USB 2.0, is a more advances and faster USB, with speeds at 480 Mb/s (40 times more than that of its predecessor USB 1.1). With USB, one can attach multiple devices to the same port (up to 127) using daisy chaining. Daisy chain means that first USB device connects to the USB port, the second USB device connects to the first USB device, the third USB device connects to the second USB device, and so on. An alternative to daisy chaining is to use a **USB hub**. This is a device that plugs into the USB port and provides multiple USB ports.

When the computer powers up, it queries all of the devices connected to the bus and assigns each one an address. This process is called enumeration, devices are also enumerated when they connect to the bus. The computer also finds out from each device what type of data transfer it wishes to perform:

- Interrupt: A device like a mouse or a keyboard, which will be sending very little data, would choose the interrupt mode.
- Bulk: A device like a printer, which receives data in one big packet, uses the bulk transfer mode. A block of data is sent to the printer (e.g. in 64-byte chunks) and verified to make sure it is correct.
- Isochronous: A streaming device (such as speakers or camcorder) uses the isochronous mode. Data streams between the device and the host in real-time at a guaranteed rate, and there is no error correction.

As devices are enumerated, the computer keeps track of the total bandwidth that all of the isochronous and interrupt devices are requesting. They can consume up to 90% of the 480 Mb/s of bandwidth that is available. After 90 percent is used up, the host denies access to any other isochronous or interrupt devices. Bulk transfers use any bandwidth left over (at least 10 percent).

Here are some USB features:

- Up to 127 devices can connect to the port, either using daisy chaining or by way of USB hubs.
- With USB 2.0, the bus has a maximum data rate of 480 Mb/s.
- Individual USB cables can run as long as 5 meters; with hubs, devices can be up to 30 meters (six cables' worth) away from the host.
- A USB cable has two wires for power (+5 volts and ground) and a twisted pair of wires to carry the data.

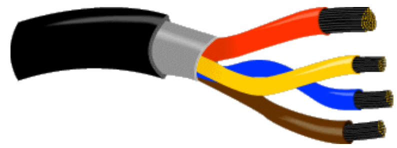

Figure 9: USB cable

- Low power devices (such as mice) can draw their power directly from the bus. High-power devices (such as printers) have their own power supplies and draw minimal power from the bus. Hubs can have their own power supplies to provide power to devices connected to the hub.
- USB devices are hot-swappable, meaning you can plug them into the bus and unplug them any time.

1.5 FireWire port

FireWire was originally created by Apple and later standardized by IEEE (Institute of Electrical and Electronics Engineers) in 1995 as the **IEEE 1394 High Performance Serial Bus**. The designers of FireWire had several particular goals in mind when they created this standard: fast transfer of data, ability to put lots of devices on the bus, ease of use, hot-swap ability, power through the cable, low cabling cost, and low implementation cost. Therefore, FireWire is very similar to USB with only few differences as illustrated in the following table:

	USB 2.0	FireWire 400	FireWire 800
speed	480 Mb/s	400 Mb/s	800 Mb/s
number of devices	127	63	63
hot-swappable	yes	yes	yes
bus power	yes	yes	yes
cable type	4 wires	6 wires	8 wires
communication	host-based	peer-to-peer	peer-to-peer

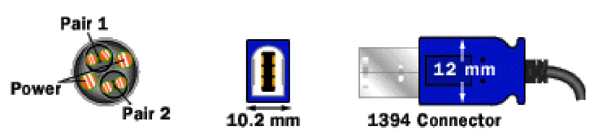

Figure 10: FireWire cables and connector

Speed aside, the big difference between FireWire and USB 2.0 is that USB 2.0 is host-based, meaning that devices must connect to a computer in order to communicate. FireWire is peer-to-peer, meaning that two FireWire cameras can talk to each other without going through a computer.

FireWire really shines when it comes to digital video applications. Most digital video cameras or camcorders now have a FireWire plug. When you attach a camcorder to a computer using FireWire, the connection is amazing. Like USB, this is due to the support of of isochronous devices where data streams between the device and the computer in real-time with guaranteed bandwidth and no error correction. Essentially, this means that a device like a digital camcorder can request that the computer allocate enough bandwidth for the camcorder to send uncompressed video in real-time to the computer. Therefore, when the computer-to-camera FireWire connection enters isochronous mode, the camera can send the video in a steady flow to the computer without anything disrupting the process.

The following table provides a comparison among SATA, PATA, USB, and FireWire:

	SATA	PATA	FireWire	USB 2.0
speed	3 Gbps	133 Mbps	800 Mbps	480 Mbps
cable length	2 m	46 cm	4.5 m	5 m
power cable needed	yes	yes	no	no

In general, USB ports have replaced mouse, keyboard, audio, serial, parallel, and SCSI ports, and FireWire ports have replaced audio, parallel, and SCSI ports.

1.6 Bluetooth

Bluetooth was collectively developed by Ericsson, Intel, IBM, Nokia, and Toshiba. Bluetooth transmits data via low power radio waves (wireless). It communicates in the frequency band 2.402 GHz and 2.480 GHz. This frequency band has been set aside by international agreement for the use of industrial, scientific, and medical devices (ISM).

A number of devices make use of the ISM band, e.g. baby monitors, garage doors, new generation of cordless phones. Making sure that Bluetooth and these other devices don't interfere with one another has been a crucial part of the design process. One way that Bluetooth devices avoid interfering with other systems is by sending a very weak signal of about 1 milliwatt (compare this to the 3 watt signal of a cell phone for instance). Although the low power limits the range of a Bluetooth device to about 10 meters (32 feet), it ensures that battery consumption is very low. Unlike infra-red (e.g. TV remote control), Bluetooth does not require "line-of-sight" between communicating devices. For instance, walls will not stop a Bluetooth signal.

Bluetooth can connect up to eight devices simultaneously. With all of those devices in the same 10 meter (32-foot) radius, one might think they would interfere with one another, but it's unlikely. Bluetooth uses a technique called **frequency hopping** that makes it rare for more than one device to be transmitting on the same frequency at the same time. In this technique, a device will use 79 individual, randomly chosen frequencies within a designated range, changing from one to another on a regular basis. In the case of Bluetooth, the transmitters change frequencies 1600 times every second. This same technique minimizes the risk that portable phones or baby monitors will disrupt Bluetooth devices, since any interference on a particular frequency will last only a tiny fraction of a second. Bluetooth 1.0 has a maximum transfer rate of 1 Mbps. Bluetooth 2.0 can manage up to 3 Mbps.

1.7 Direct Memory Access (DMA)

We have seen that I/O controllers communicate with the CPU using the memory-mapped I/O technique. This means that any communication between a peripheral device and the memory (e.g. retrieving data from a hard disk) has to go through the CPU (using Load and Store instructions). Such approach will slow down the system considerably because the CPU will have to execute every data transfer. Modern bus designs allow for an I/O controller to communicate directly with memory without the CPU being involved. Such an I/O controller is called a DMA controller, which stands for Direct Memory Access. A DMA transfer essentially copies a block of memory from one device (I/O controller or memory) to another. While the CPU still initiates the transfer, it does not execute the transfer itself. Advanced bus designs, such as PCI (Peripheral Components Interconnect), typically allow a device to take control of the bus and perform the transfer itself.

For instance, to retrieve data from a sector on a hard disk, the CPU puts the request on the bus, including any necessary signaling and addressing to access the desired sector, and informs the hard disk controller to place the data in a specified area in memory. The CPU can then continue with other tasks while the controller performs the read operation and deposits the data in main memory.

The use of DMA also complicates the communication taking place over a computer's internal bus. Data must move between the CPU and main memory (e.g. fetching instructions), between the CPU and each controller, and between each controller and main memory (DMA). Coordination of all this activity on the bus is a major design issue. Even with excellent designs, the internal bus can become an impediment as the CPU and I/O controllers complete for bus access. This is known as the **von Neumann bottleneck** because it is a consequence of the von Neumann architecture.