

# OPERATING SYSTEM - I/O HARDWARE

[http://www.tutorialspoint.com/operating\\_system/os\\_io\\_hardware.htm](http://www.tutorialspoint.com/operating_system/os_io_hardware.htm)

Copyright © tutorialspoint.com

## Overview

Computers operate on many kinds of devices. General types include *storage devices disks, tapes, transmission devices networkcards, modems*, and *human-interface devices screen, keyboard, mouse*. Other devices are more specialized. A device communicates with a computer system by sending signals over a cable or even through the air.

The device communicates with the machine via a connection point termed a *port forexample, a serialport*. If one or more devices use a common set of wires, the connection is called a *bus*. In other terms, a bus is a set of wires and a rigidly defined protocol that specifies a set of messages that can be sent on the wires.

## Daisy chain

When device A has a cable that plugs into device B, and device B has a cable that plugs into device C, and device C plugs into a port on the computer, this arrangement is called a **daisy chain**. It usually operates as a bus.

## Controller

A controller is a collection of electronics that can operate a port, a bus, or a device. A serial-port controller is an example of a simple device controller. This is a single chip in the computer that controls the signals on the wires of a serial port.

The SCSI bus controller is often implemented as a separate circuit board *ahostadapter* that plugs into the computer. It contains a processor, microcode, and some private memory to enable it to process the SCSI protocol messages. Some devices have their own built-in controllers.

## I/O port

An I/O port typically consists of four registers, called the **status**, **control**, **data-in**, and **data-out** registers.

S.N.	Register & Description
1	<b>Status Register</b> The status register contains bits that can be read by the host. These bits indicate states such as whether the current command has completed, whether a byte is available to be read from the data-in register, and whether there has been a device error.
2	<b>Control register</b> The control register can be written by the host to start a command or to change the mode of a device. For instance, a certain bit in the control register of a serial port chooses between full-

	duplex and half-duplex communication, another enables parity checking, a third bit sets the word length to 7 or 8 bits, and other bits select one of the speeds supported by the serial port.
3	<b>Data-in register</b> The data-in register is read by the host to get input.
4	<b>Data-out register</b> The data out register is written by the host to send output.

## Polling

Polling is a process by which a host waits for controller response. It is a looping process, reading the status register over and over until the busy bit of status register becomes clear. The controller uses/sets the busy bit when it is busy working on a command, and clears the busy bit when it is ready to accept the next command. The host signals its wish via the command-ready bit in the command register. The host sets the command-ready bit when a command is available for the controller to execute.

In the following example, the host writes output through a port, coordinating with the controller by handshaking

- The host repeatedly reads the busy bit until that bit becomes clear.
- The host sets the write bit in the command register and writes a byte into the data-out register.
- The host sets the command-ready bit.
- When the controller notices that the command-ready bit is set, it sets the busy bit.
- The controller reads the command register and sees the write command.
- It reads the data-out register to get the byte, and does the I/O to the device.
- The controller clears the command-ready bit, clears the error bit in the status register to indicate that the device I/O succeeded, and clears the busy bit to indicate that it is finished.

## I/O devices

I/O Devices can be categorized into following category.

S.N.	Category & Description
1	<b>Human readable</b> Human Readable devices are suitable for communicating with the computer user. Examples are printers, video display terminals, keyboard etc.
2	<b>Machine readable</b>

	Machine Readable devices are suitable for communicating with electronic equipment. Examples are disk and tape drives, sensors, controllers and actuators.
2	<b>Communication</b> Communication devices are suitable for communicating with remote devices. Examples are digital line drivers and modems.

Following are the differences between I/O Devices

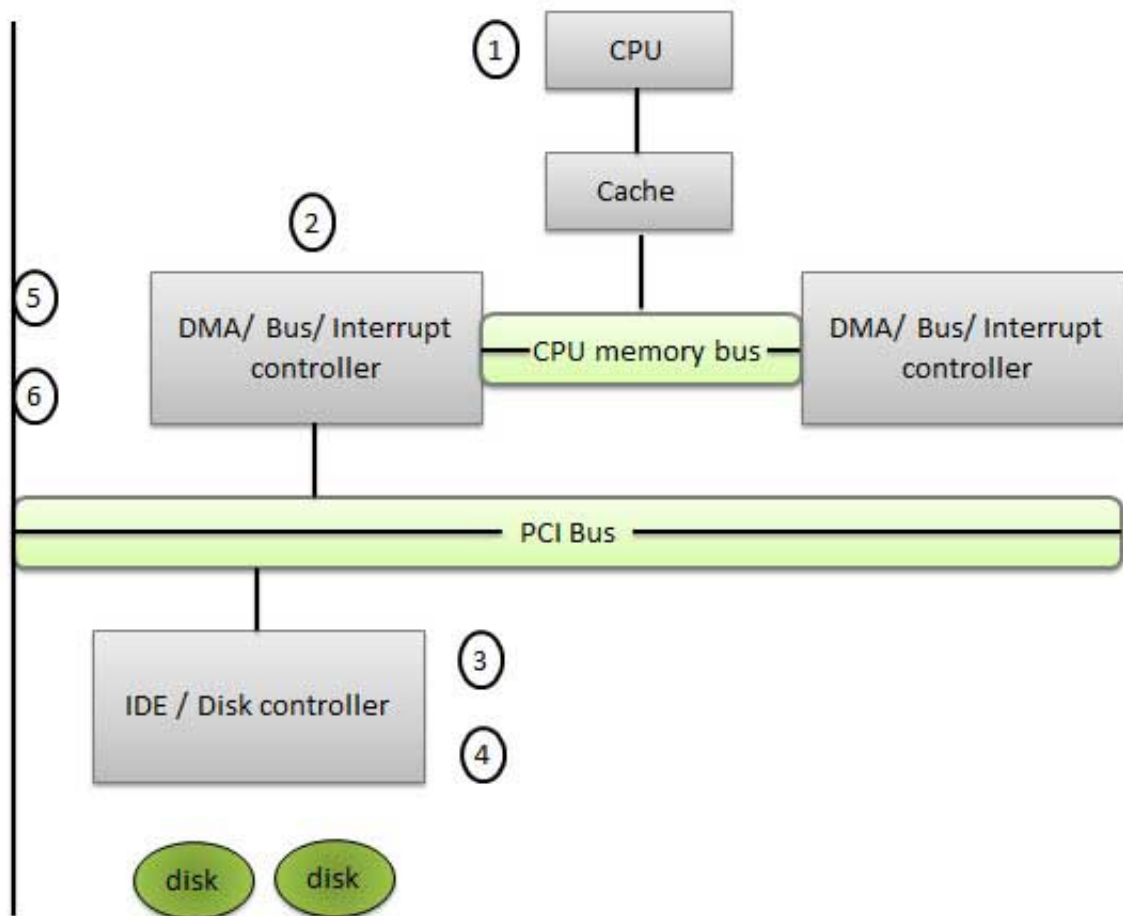
S.N.	Criteria & Description
1	<b>Data rate</b> There may be differences of several orders of magnitude between the data transfer rates.
2	<b>Application</b> Different devices have different use in the system.
3	<b>Complexity of Control</b> A disk is much more complex whereas printer requires simple control interface.
4	<b>Unit of transfer</b> Data may be transferred as a stream of bytes or characters or in larger blocks.
5	<b>Data representation</b> Different data encoding schemes are used for different devices.
6	<b>Error Conditions</b> The nature of errors differs widely from one device to another.

## Direct Memory Access *DMA*

Many computers avoid burdening the main CPU with programmed I/O by offloading some of this work to a special purpose processor. This type of processor is called, a Direct Memory Access *DMA* controller. A special control unit is used to transfer block of data directly between an external device and the main memory, without intervention by the processor. This approach is called Direct Memory Access *DMA*.

*DMA* can be used with either polling or interrupt software. *DMA* is particularly useful on devices like disks, where many bytes of information can be transferred in single I/O operations. When used with an interrupt, the CPU is notified only after the entire block of data has been transferred. For each byte or word transferred, it must provide the memory address and all the bus signals controlling the data transfer. Interaction with a device controller is managed through a device driver.

Handshaking is a process between the DMA controller and the device controller. It is performed via wires using terms DMA request and DMA acknowledge.



Step	Description
1	Device driver is instructed to transfer disk data to a buffer address X.
2	Device driver then instruct disk controller to transfer data to buffer.
3	Disk controller starts DMA transfer.
4	Disk controller sends each byte to DMA controller.
5	DMA controller transfers bytes to buffer, increases the memory address, decreases the counter C until C becomes zero.
6	When C becomes zero, DMA interrupts CPU to signal transfer completion.

## Device Controllers

A computer system contains a many types of I/O devices and their respective controllers

- network card
  - graphics adapter
  - disk controller
  - DVD-ROM controller
  - serial port
  - USB
  - sound card
-