DEVICE CONTROLLERS, MEMORY MAPPED I/O AND PORT MAPPED I/O

Device Controllers:

A device controller is a hardware unit which is attached with the input/output bus of the computer and provides a hardware interface between the computer and the input/output devices. On one side it knows how to communicate with input/output devices and on the other side it knows how to communicate with the computer system though input/output bus. A device controller usually can control several input/output devices.

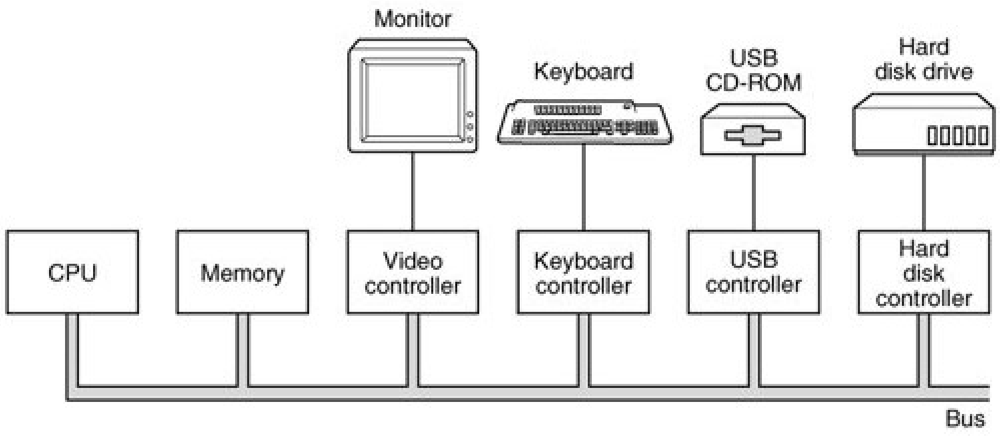

Fig: A model for connecting the CPU, memory, controllers, and I/O devices

Typically the controller is on a card (eg. LAN card, USB card etc). Device Controller play an important role in order to operate that device. It's just like a bridge between device and operating system.

Most controllers have DMA(Direct Memory Access) capability, that means they can directly read/write memory in the system. A controller without DMA capability provide or accept the data, one byte or word at a time; and the processor takes care of storing it, in memory or reading it from the memory. DMA controllers can transfer data much faster than non-DMA controllers. Presently all controllers have DMA capability.

DMA is a memory-to-device communication method that by passes the CPU.

Memory-mapped Input/Output:

Each controller has a few registers that are used for communicating with the CPU. By writing into these registers, the operating system can command the device to deliver data, accept data, switch itself on or off, or otherwise perform some action. By reading from these registers, the operating system can learn what the device's state is, whether it is prepared to accept a new command, and so on.
In addition to the control registers, many devices have a data buffer that the operating system can read and write. For example, a common way for computers to display pixels on the screen is to have a video RAM, which is basically just a data buffer, available for programs or the operating system to write into.

There are two alternatives that the CPU communicates with the control registers and the device data buffers.

Port-mapped I/O :

each control register is assigned an I/O port number, an 8- or 16-bit integer. Using a special I/O instruction such as

IN REG,PORT

the CPU can read in control register PORT and store the result in CPU register REG. Similarly, using

OUT PORT,REG

the CPU can write the contents of REG to a control register. Most early computers, including nearly all mainframes, such as the IBM 360 and all of its successors, worked this way.

In this scheme, the address spaces for memory and I/O are different, as shown in Fig. (a).Port-mapped I/O uses a special class of CPU instructions specifically for performing I/O.

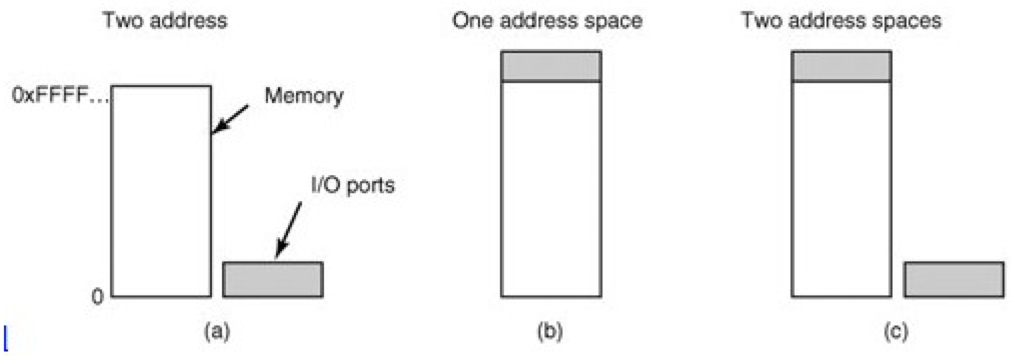

Fig:(a) Separate I/O and memory space. (b) Memory-mapped I/O. (c) Hybrid.

On other computers, I/O registers are part of the regular memory address space, as shown in Fig.(b). This scheme is called memory-mapped I/O, and was introduced with the PDP-11 minicomputer.Memory-mapped I/O (not to be confused with memory-mapped file I/O) uses the same address bus to address both memory and I/O devices, and the CPU instructions used to access the memory are also used for accessing devices. In order to accommodate the I/O devices, areas of the CPU's addressable space must be reserved for I/O.