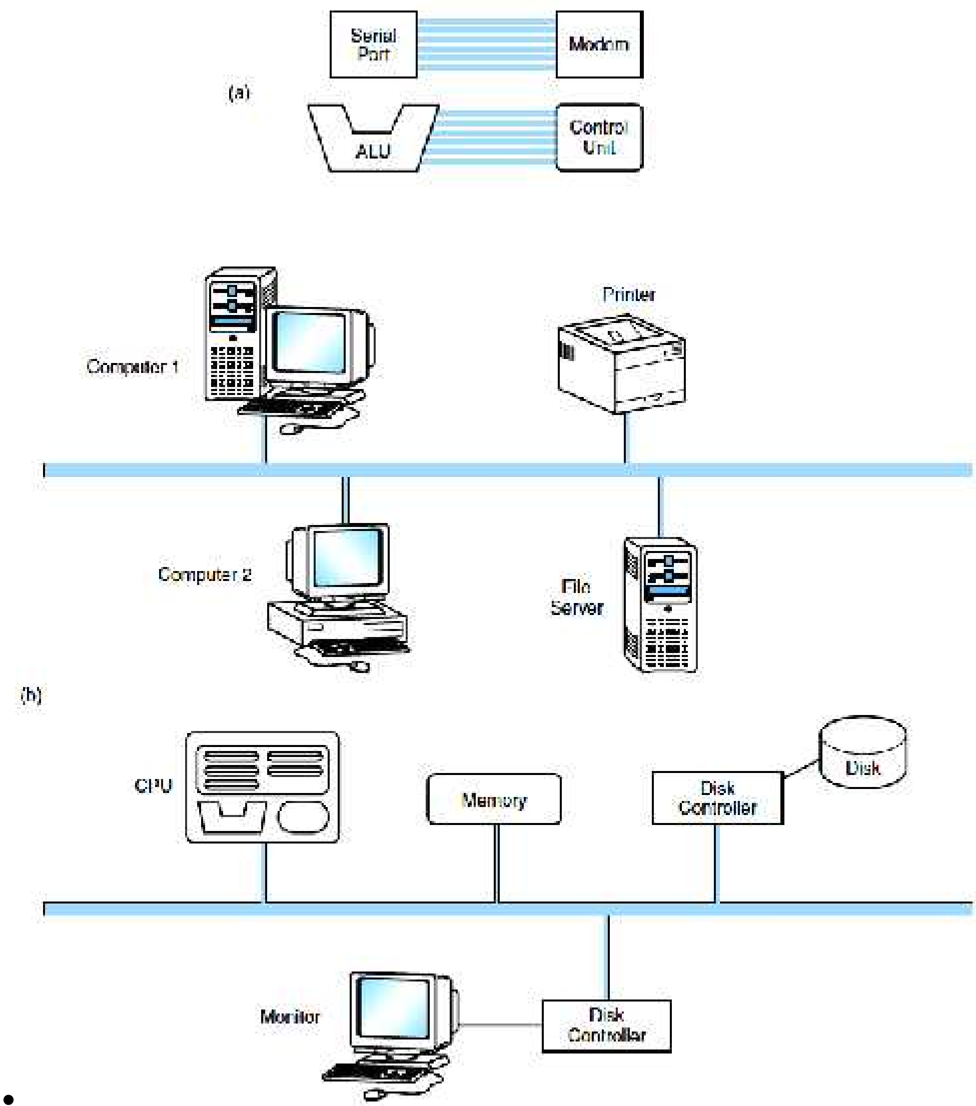
WEEK-10

3-3 The Buses.

The CPU communicates with the other components via a bus. A bus is a set of wires that acts as a shared but common data path to connect multiple subsystems within the system. It consists of multiple lines, allowing the parallel movement of bits. Buses are low cost but very versatile, and they make it easy to connect new devices to each other and to the system. At any one time, only one device (be it a register, the ALU, memory, or some other component) may use the bus. However, this sharing often results in a communications bottleneck. The speed of the bus is affected by its length as well as by the number of devices sharing it. Quite often, devices are divided into master and slave categories, where a master device is one that initiates actions and a slave is one that responds to requests by a master.

A bus can be point-to-point, connecting two specific components (as seen in Figure 3.1a) or it can be a common pathway that connects a number of devices, requiring these devices to share the bus (referred to as a multipoint bus and shown in Figure 3.1b).



•

Fig.(3-1) a) Point to Point Buses b) A Multi-Point Buses

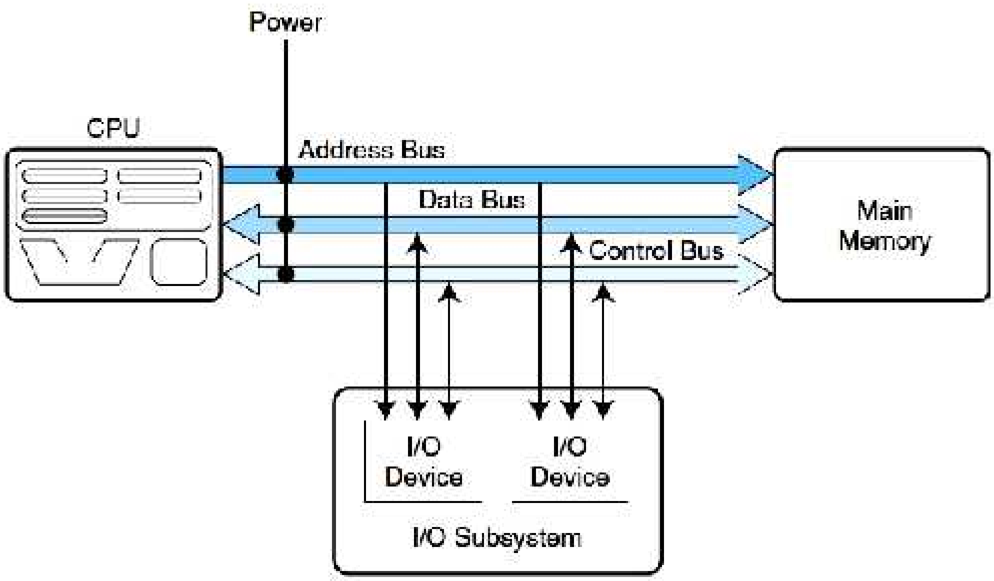


FIGURE 3.2 The Components of a Typical Bus

Because of this sharing, the *bus protocol* (set of usage rules) is very important. Figure 3.2 shows a typical bus consisting of data lines, address lines, control lines, and power lines. Often the lines of a bus dedicated to moving data are called the *data bus*.

- ***Data lines***: Contain the actual information that must be moved from one location to another.

- ***Control lines****:* Indicate which device has permission to use the bus and for what purpose (reading or writing from memory or from an I/O device, for example). Control lines also transfer acknowledgments for bus requests, clock synchronization signals, and interrupts.

- ***Address lines***: Indicate the location (in memory, for example) that the data should be either read from or written to. The *power lines* provide the electrical power necessary.

Typical bus transactions include sending an address (for a read or write), transferring data from memory to a register (a memory read), and transferring data to the memory from a register (a memory write). In addition, buses are used for I/O reads and writes from peripheral devices. Each type of transfer occurs within a *bus cycle*, the time between two ticks of the bus clock.

3.4 Clocks.

Every computer contains an internal clock that regulates how quickly instructions can be executed. The clock also synchronizes all of the components in the system.

As the clock ticks, it sets the pace for everything that happens in the system. The CPU uses this clock to regulate its progress, checking the otherwise unpredictable speed of the digital logic gates. The CPU requires a fixed number of clock ticks to execute each instruction. Therefore, instruction performance is often measured in *clock cycles*—the time between clock ticks—instead of seconds. The *clock frequency* (sometimes called the clock rate or clock speed) is measured in MHz. The *clock cycle time* (or clock period) is simply the reciprocal of the clock frequency. For

example, an 800MHz machine has a clock cycle time of 1/800,000,000 or 1.25ns. If a machine has a 2ns cycle time, then it is a 500MHz machine.

3.5 The Input / Output Subsystem.

Input and output (I/O) devices allow us to communicate with the computer system. I/O is the transfer of data between primary memory and various I/O peripherals.

Input devices such as keyboards, mice, card readers, scanners, voice recognition systems, and touch screens allow us to enter data into the computer.

Output devices such as monitors, printers, plotters, and speakers allow us to get information from the computer.

These devices are not connected directly to the CPU. Instead, there is an interface that handles the data transfers. This interface converts the system bus signals to and from a format that is acceptable to the given device. The CPU communicates to these external devices via input/output registers. This exchange of data is performed in two ways. In memory-mapped I/O, the registers in the interface appear in the computer's memory map and there is no real difference between accessing memory and accessing an I/O device. Clearly, this is advantageous from the perspective of speed, but it uses up memory space in the system.

With instruction-based I/O, the CPU has specialized instructions that perform the input and output. Although this does not use memory space, it requires specific I/O instructions, which implies it can be used only by CPUs that can execute these specific instructions. Interrupts play a very important part in I/O, because they are an efficient way to notify the CPU that input or output is available for use.