Lecture Two
Computer-System Organization

Objectives After reading this chapter, you should understand:

- Hardware components that must be manage by Operating System.
- How hardware was evolved to support OS functions.
- The motion of an application program interface

2.1 **Computer-System Organization**

A modern general-purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access between components and shared memory (Figure 2.1). Each device controller is in charge of a specific type of device (for example, a disk drive, audio device, or graphics display).

Depending on the controller, more than one device may be attached. For instance, one system USB port can connect to a USB hub, to which several devices can connect. A device controller maintains some local buffer storage and a set of special-purpose registers. The device controller is responsible for moving the data between the peripheral devices that it controls and its local buffer storage. Typically, operating systems have a device driver for each device controller.

This device driver understands the device controller and provides the rest of the operating system with a uniform interface to the device. The CPU and the device controllers can execute in parallel, competing for memory cycles. To ensure orderly access to the shared memory, a memory controller synchronizes access to the memory. In the following subsections, we describe some basics of how such a system operates, focusing on three key aspects of the system. We start with interrupts, which alert the CPU to events that require attention. We then discuss storage structure and I/O structure.

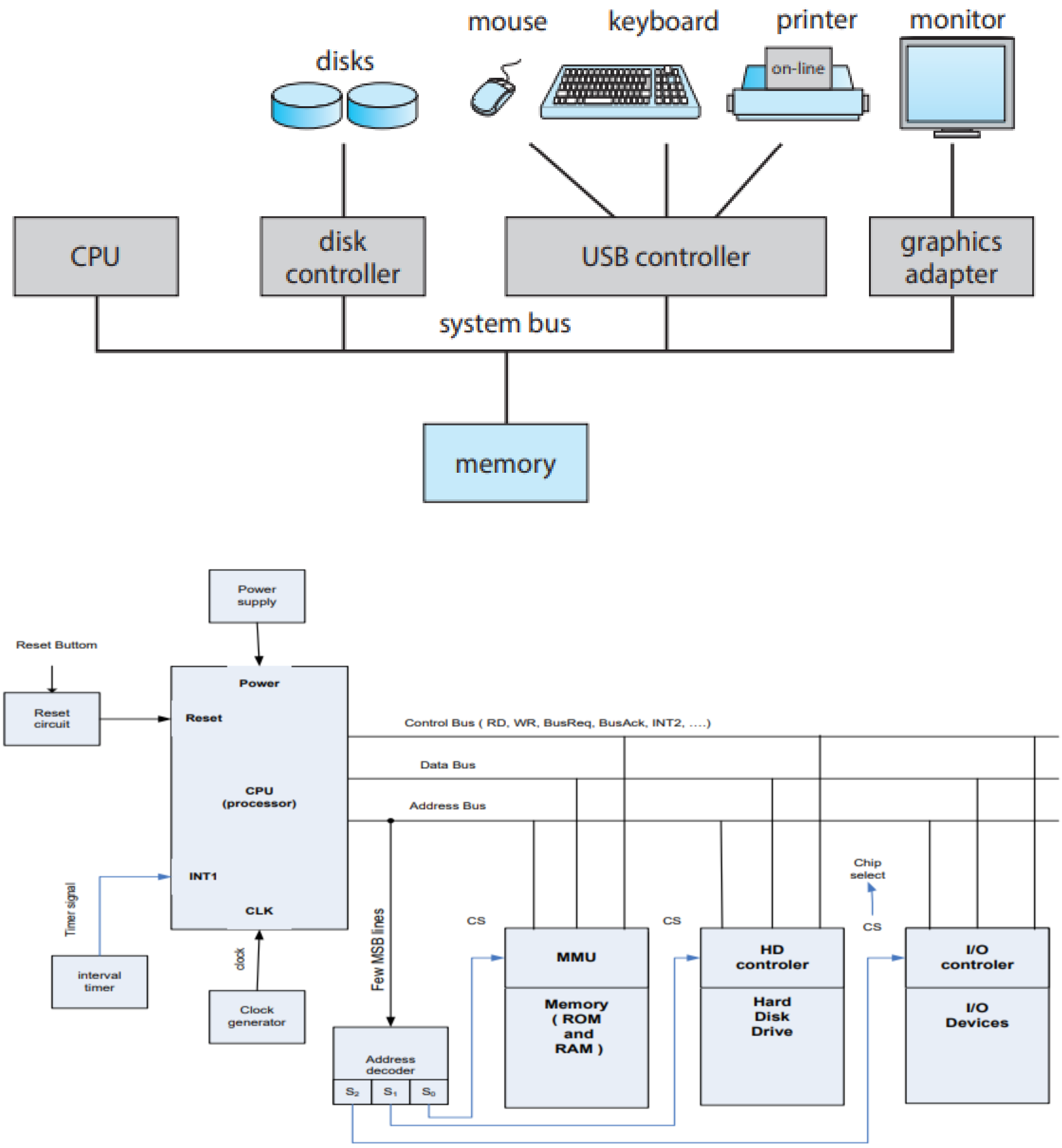

Figure 2.1 a typical PC computer system

a. Interrupts

Consider a typical computer operation: a program performing I/O. To start an I/O operation, the device driver loads the appropriate registers in the device controller. The device controller, in turn, examines the contents of these registers to determine what action to take (such as "read a character from the keyboard"). The controller starts the transfer of data from the device to its local buffer. Once the transfer of data is complete, the device controller informs the device driver that it has finished its operation. The device driver then gives control to other parts of the operating system, possibly returning the data or a pointer to the data if the operation was a read. For other operations, the device driver returns status information such as "write completed successfully" or "device busy". But how does the controller inform the device driver that it has finished its operation? This is accomplished via an interrupt.

When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location. The fixed location usually contains the starting address where the service routine for the interrupt is located. The interrupt service routine executes; on completion, the CPU resumes the interrupted computation. A timeline of this operation is shown in Figure 2.2. To run the animation associated with this figure please click here. Interrupts are an important part of a computer architecture. Each computer design has its own interrupt mechanism, but several functions are common. The interrupt must transfer control to the appropriate interrupt service routine. The straightforward method for managing this transfer would be to invoke a generic routine to examine the interrupt information.

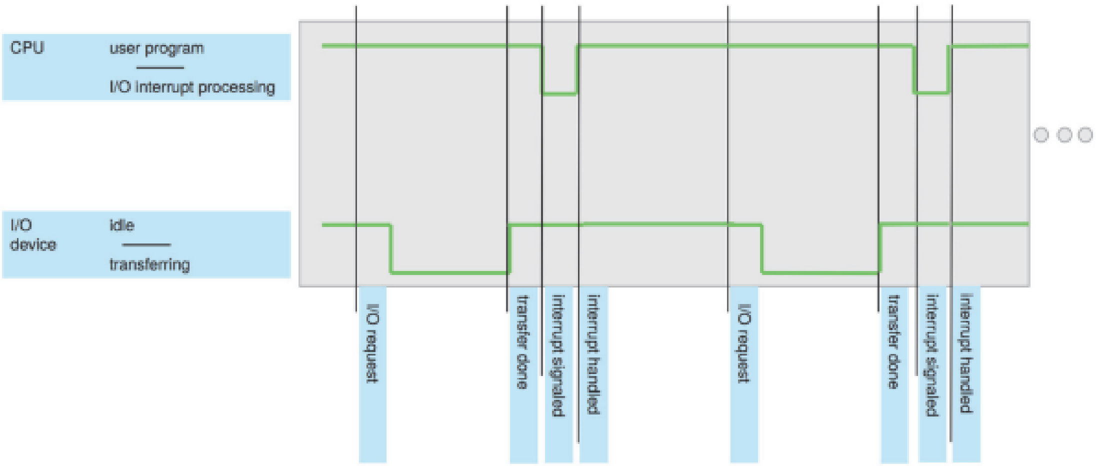

Figure 2.2 Interrupt timeline for a single program doing output

b. Hardware Components

1- **CPU:** Central processing Unit is the main unit of any computer system. The CPU consists of Arithmetic Logic Unit (ALU) and Control Unit (CU).the main registers included in CPU are:

- **Sp :** Stack Pointer Register which holds an address pointing at a location in memory called " Stack Top".
- **PC :** Program Counter Register which holds address of the next instruction to be executed.
- **A :** Accumulator Register which holds the result of any arithmetic or logic operation. Operating System – fourth year – Dr. Dhia A. Alzubaydi 02
- **PSW :** Processor Status Word Register which holds flags showing the status of any arithmetic or logic operation (Zero, overflow, carry, etc).
- **Data Registers (B, C, D,E)** which act as fast storage for temporary data.

2- **Clock Generator** : it is essential electronic circuit generating periodic pulse signal called "clock" as shown in fig 2.2. It is worth remembering that all actions in a computer system are timed precisely and synchronized with clock edges. The clock frequency (1/T) determines the computer speed to a large degree in addition to other factors.

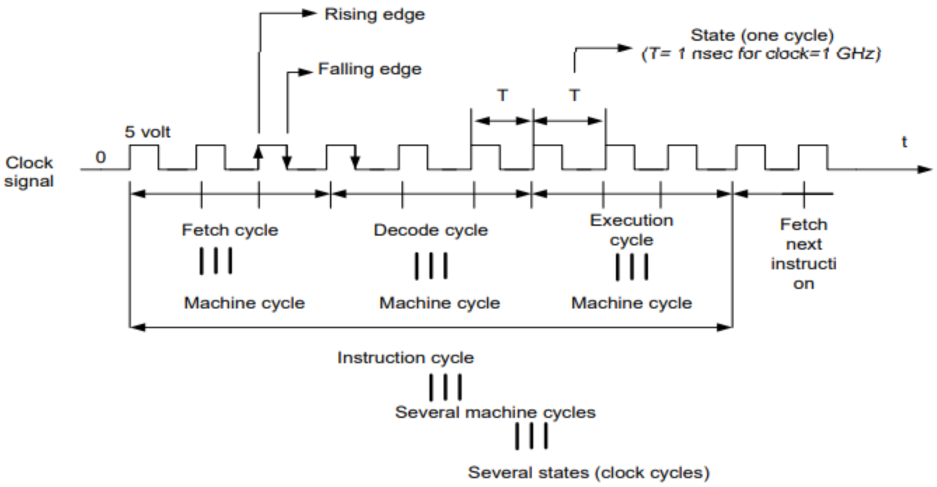

Fig 2.2 Instruction Cycle

3- **Reset Circuit** : when a reset button is pressed, a pulse signal is issued to CPU which forces a value of Zeros to PC and an instruction fetch cycle is started from a location in memory pointed at by pc. This means that " Reset" forces computer to start execution of a program loaded at address zero (Usually BIOS program stored in Read Only Memory ROM).

4- **Interval Timer** : It generates a periodic pulse but much slower than clock generator. Each Timer period is called “Time Slice” and equivalent to millions of clock periods (states). This means that during a Time Slice, it is possible to execute part of a program. The interval timer, as will be shown later, has a big role in multiprogramming system.

5- **Power Supply** : It is necessary for CPU and all other components.

6- **Memory** : It is the second important unit in any computer system (after CPU). It is very fast addressable storage. It is used to hold programs when they are being executed in addition to other functions (e.g stack,…) there are two types of memory:

   a. Random Access Memory (RAM) used for storing programs and data temporarily.

   b. Read Only Memory (ROM) used to hold programs and data permanently (e.g BIOS).

7- **Hard Disk Drive** : It is a large volume of permanent storage for programs and data but it is not addressable and of slow access as well.

8- **I/O Devices** :the computer is not useful unless interfaced to external world such as printers, keyboards, displays, scanners, mouse, camera, etc.

9- **Controllers** : To connect any device to CPU, it is necessary to use a proper interface circuit (card) called " controller". The controller is usually driven by a proper program called " Driver". This means that " Controller" is a hardware (H/W) while "driver" is a software (S/W). Here, it is worth noting that a "Drive" is different from "Driver" as it represents the instrument itself e.g disk drive which consists of disk, motor, electronic circuits, heads, etc. Also it is worth noting that a memory controller is usually called as Memory Management Unit (MMU).

10- **Address Decoder** : Few of most significant address signals are decoded to enable selection of one device that should respond to CPU.

11- **Address Bus** : Set of copper lines carrying electrical signals (voltages) which are used to select one location only in the whole computer system whether for " Read" or "Write" cycles

12- **Data Bus** : several lines carrying data to or from addressed location.

13- **Control Bus** : Several Lines having several functions. One line, for example, is used to carry "Read" signal to declare that a cycle is a "Read" one.

14- **Interrupt (INT)** : It is input signal to CPU which when "active" forces certain value (address) to PC and initiates instruction cycle and hence the CPU starts executing " Interrupt Subroutine"

2.2 **I/O Data Transfer**

There are several methods for transferring data between I/O devices and memory as follows: •Programmed I/O. •DMA I/O.

2.3 Programmed I/O There are two types:

a. Polling I/O
b. Interrupt I/O,

In polling, CPU executes a program to scan I/O device periodically to check its need for service. In interrupt I/O, there is no scan at all, however, when I/O device requires service, it activates an interrupt signal to CPU and hence interrupt subroutine will be executed which should provide the requested service.

2.4 Direct Memory Access (DMA)

There are two useful control signals for this operation which are: BusReq and BusAck. BusReq is an input signal to CPU and when activated, it forces CPU to separate itself from its external "Buses" at the end of machine cycle. When separation occurs, CPU activates BusAck signal to inform I/O devices that they can use all the Buses. Now, it is possible to explain DMA as follows:-

1. I/O device activates BusReq line and wait till BusAck is activated.
2. When BusAck is activated, I/O device can master the Buses and hence use them to address memory and make data transfer between Memory and I/O device. From the above, it is clear that programmed I/O is implemented by making CPU executes a proper program while in DMA, the CPU does not interfere as it separates it self from its Buses. Operating System – fourth year –

2.5 CPU Modes of Operation (States) There are two main modes:

- Supervisor Mode : In this mode, the CPU can execute all instructions including " user "and" privileged" instructions.
- User Mode : In this mode, the CPU can execute " user" instructions only and can not execute privileged instructions. The CPU modes are useful for programs protection in multi programming environment.

2.6 Bootstrapping When a computer is powered up , or Reset activated, BIOS is started and loads " Boot Sector" from disk (Hard, CD, floppy). The Boot Sector contains a program which enables loading of main OS components into memory which then started. the above operations are called "Bootstrapping" which aims at loading OS into memory and running it.

2.7 Application Programming Interfaces (APIs) APIs are set of subroutines that control computer resources and considered as part of OS. As shown in Figure (2.3), a user program (application) can use these APIs by calling them via special instructions called "System Calls". A system call acts as a " software interrupt" and hence changes CPU mode from " User" to " Supervisor" which in turn prevents an " address violation exception" from occurring. Here, it should be noted that if user program tries to use normal subroutine call instruction then an address violation exception will occur as the called address is outside user program " address space" as will be explained later.

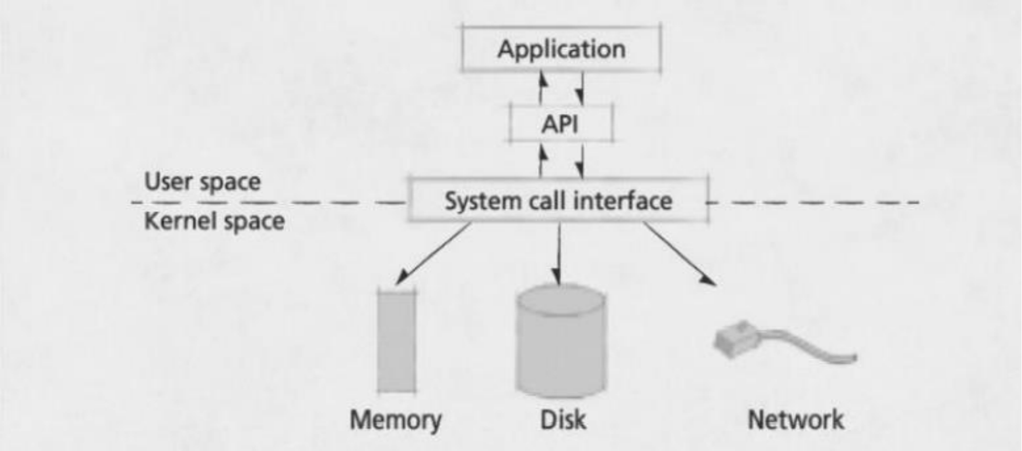

Figure 2.3 Application program interface

2.8 Multiprocessing It is equivalent to " multiprocessors" and not to " multi processes". Multiprocessing means that CPU consists of several processors sharing memory and controlled by single OS for the purpose of speeding up computer operation.

2.9Buffering A " Buffer" is an area of memory for holding data temporarily during data transfer between running program and I/O device. This technique speeds up the computer operation. Examples of these buffers are:

a. Hard Disk Buffer : In this case, OS writes data quickly to disk buffer (i.e to memory) and later on data will be transferred to the slow disk drive. When reading, the reverse occurs.

b. Keyboard buffer: It is used to hold characters typed by a slow user and later the running program will process these characters.

2.10 Spooling It is , also, a technique for speeding up computer operation. In this case, the data to be written to a very slow device (e.g printer) are written, first, to intermediate medium speed device (such as disk) and later on transferred to the slow device