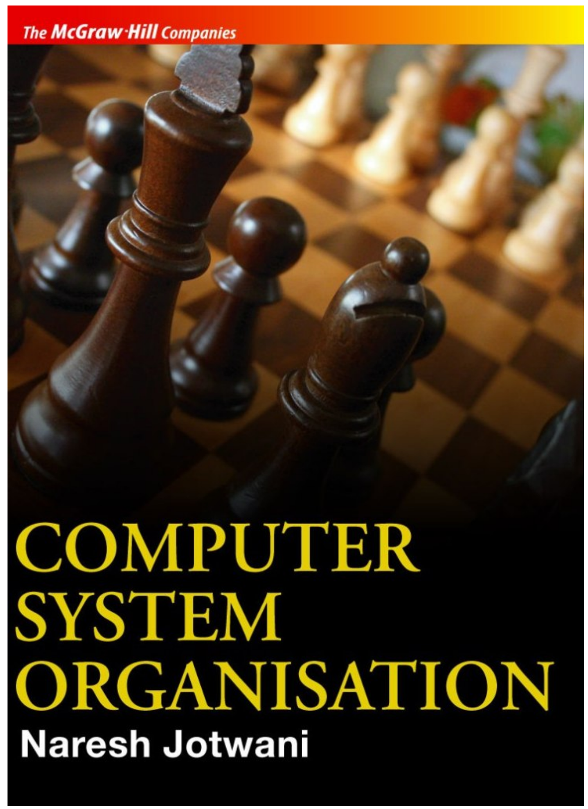
The McGraw·Hill Companies

Mc Graw Hill Education

# COMPUTER SYSTEM ORGANISATION

Naresh Jotwani

## PowerPoint Slides



# CHAPTER 11

# USER INTERACTION

# Introduction

- The earliest computers were extremely difficult to program and operate.

- Advent of personal computer (PC) caused a revolutionary change in the way that computer systems are designed and built.

- Several major advances have contributed to computer technology over the last few decades:
  - advances in VLSI, graphics display, and storage technologies;
  - techniques of software development and computer networking;
  - packaging, mass production, mass marketing, and customer support.

- Quality of user interaction has also gone through a revolutionary change. G*raphical user interface* (GUI) has come into wide-spread use.
- Quality of user interaction depends on how elements of the computer system interact with each other.
- In the design of graphical user interaction, speed of system response to user commands or ‘mouse clicks’ is very important.
- Today *user interaction* is a crucial issue in the design of a computer system.

- Steps involved in user-computer interaction:

  1: user input is sensed by an input device of the system;

  2: program running on the system alters display output in accordance with user input;

  3: eyes perceive change in display;

  4: user determines next input to the system;

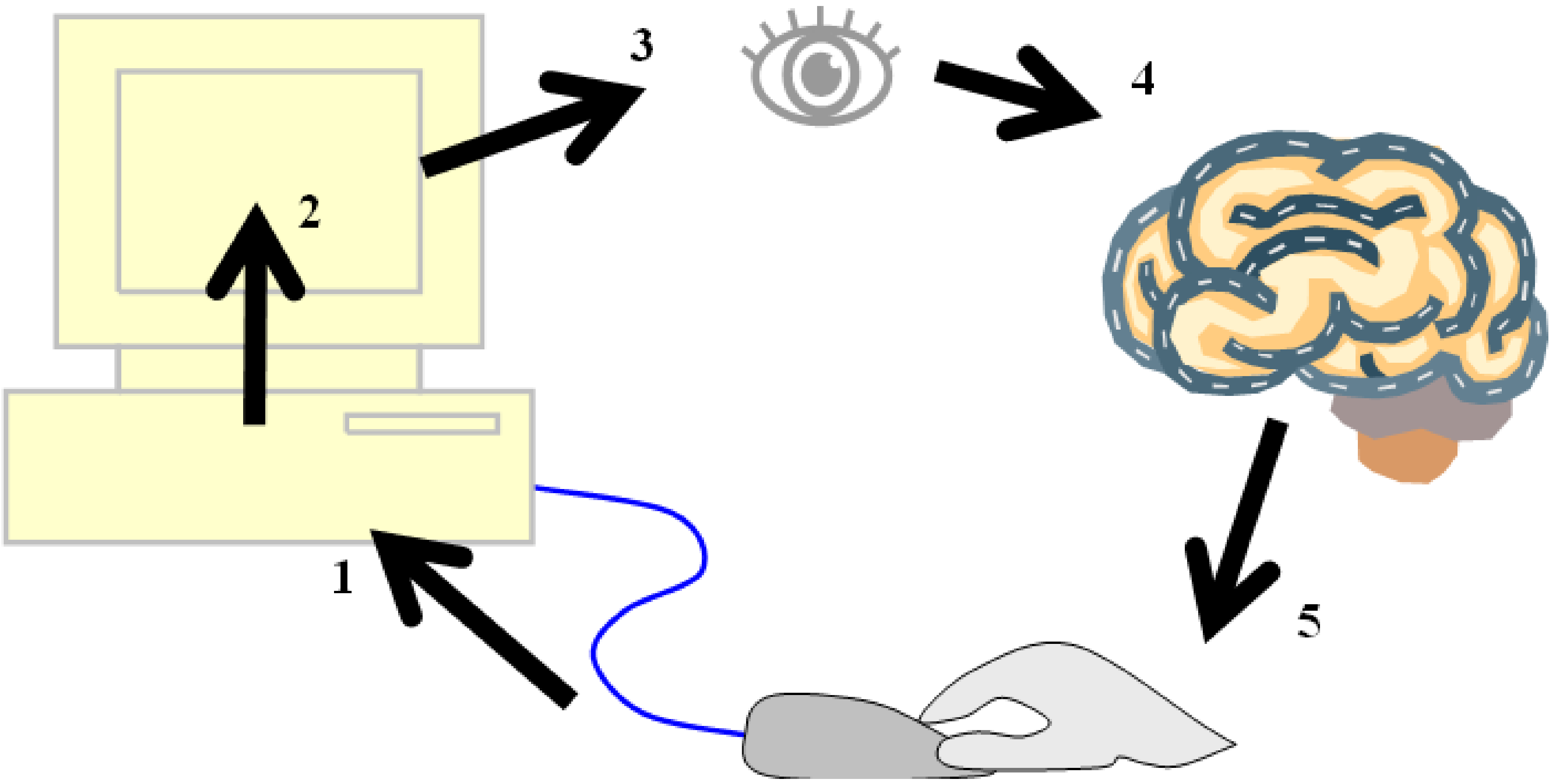
  5: user operates input device accordingly.

- In the next figure, step *1* shows (typical) *input* provided by the user to the computer system.

- Step *2* shows graphical or textual *output* generated and displayed by the computer system.

- Cycle of five steps shows that the next input given by the user depends on the output seen by the user.

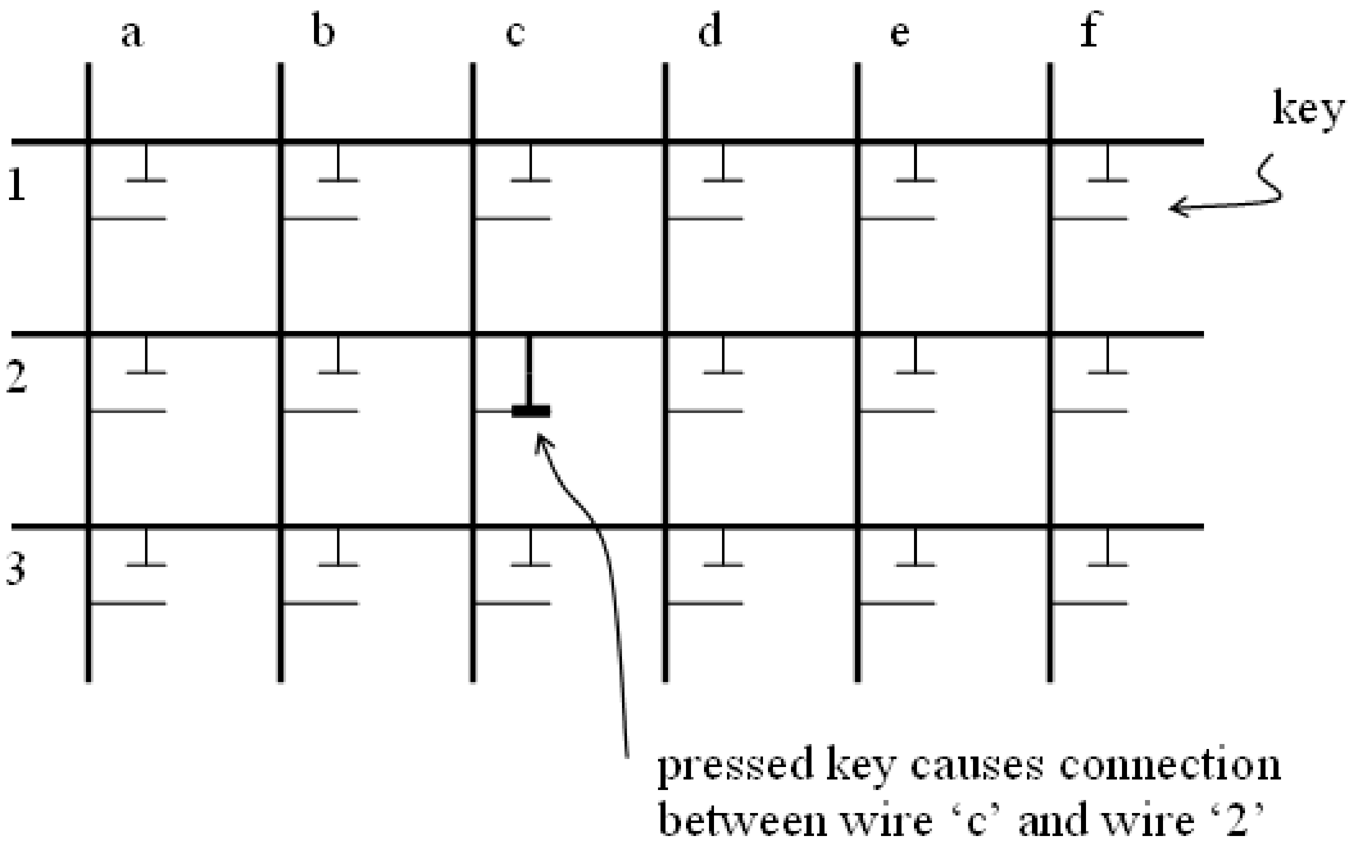
# The cycle of user-computer interaction



# Keyboard:

- Keyboard or keypad works by electronically detecting the closing of a switch caused by a key depression.
- The next figure represents in a highly simplified way, the inner working of a keyboard.
  - Figure shows six column ‘wires’ and three row ‘wires’ crisscrossing each other in a rectangular pattern.
  - These wires lie underneath the keys that we use, and they do not normally touch each other.
- Column wire ‘c’ and row wire ‘2’ are connected, since the key at that position is assumed to be pressed.

# Simplified diagram of a keyboard:




- At every intersection between column and row wires, a mechanical key is provided. A key press causes column and row wires at that position to become connected for as long as the key remains pressed.

- Careful mechanical design of keys ensures this electrical connection - and also that keys have the right kind of ‘feel’ and ‘bounce’.

- Control electronics of the keyboard constantly scans column and row wires at high speed to detect any signs of key closure.

  – A key closure causes change in the voltage and/or current in exactly one column and one row wire.
  – The change is captured electronically and sent in the form of input data from the keyboard.

- Every key is labeled with a character symbol, and therefore the system can easily correlate pairs of wires with input character of the user.

- Scanning electronics also detects whether, at the time of a key closure, another key is being held down – e.g. *shift*, *ctrl*, *alt*, etc.

- Once the correct key closure is detected, the resulting input data is transferred to the computer using the basic principles of I/O.

- On many systems, the keyboard device generates two 'input events' for each key closure – one when the key is pressed down, and one when it is released.

  – This allows the software inside the computer to provide more user-friendly responses to key presses.

# *Mouse:*

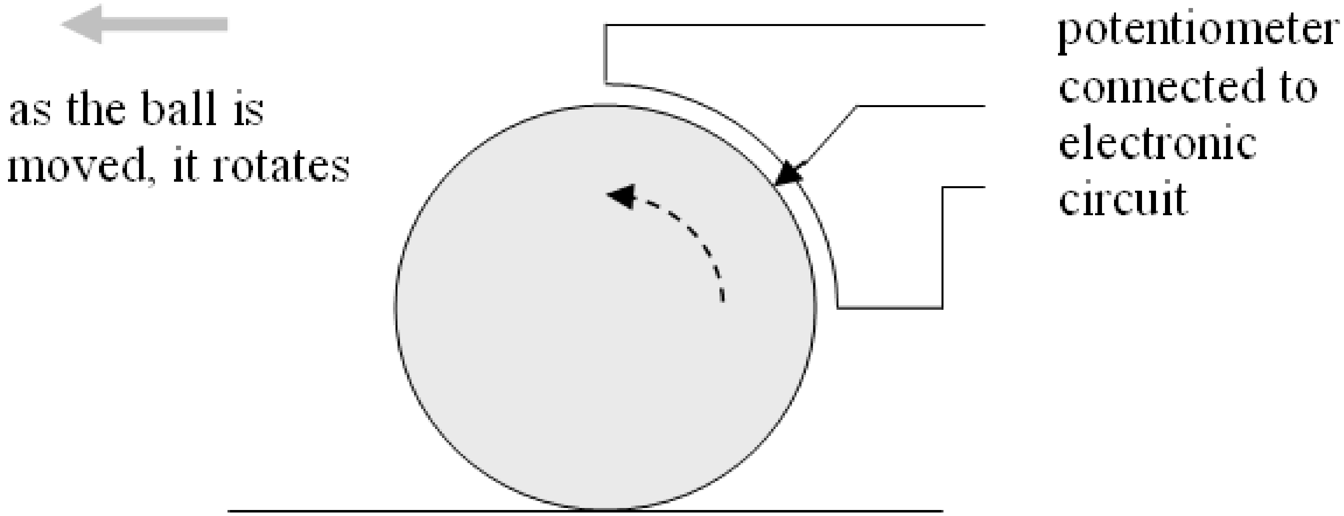
- A simple mechanical mouse houses a ball underneath, which rotates due to friction as we move the mouse on a flat surface.

- Rotation of the ball causes two potentiometers inside the mouse to rotate one way or the other.

- One of these potentiometers is read in by the computer as ‘movement along X direction’, and the other as ‘movement along Y direction’.

- The next figure illustrates this principle (but only one of the two potentiometers is shown).

# Principle of operation of mechanical mouse:




- Outputs of the two potentiometers are converted from *analog* to *digital* form.

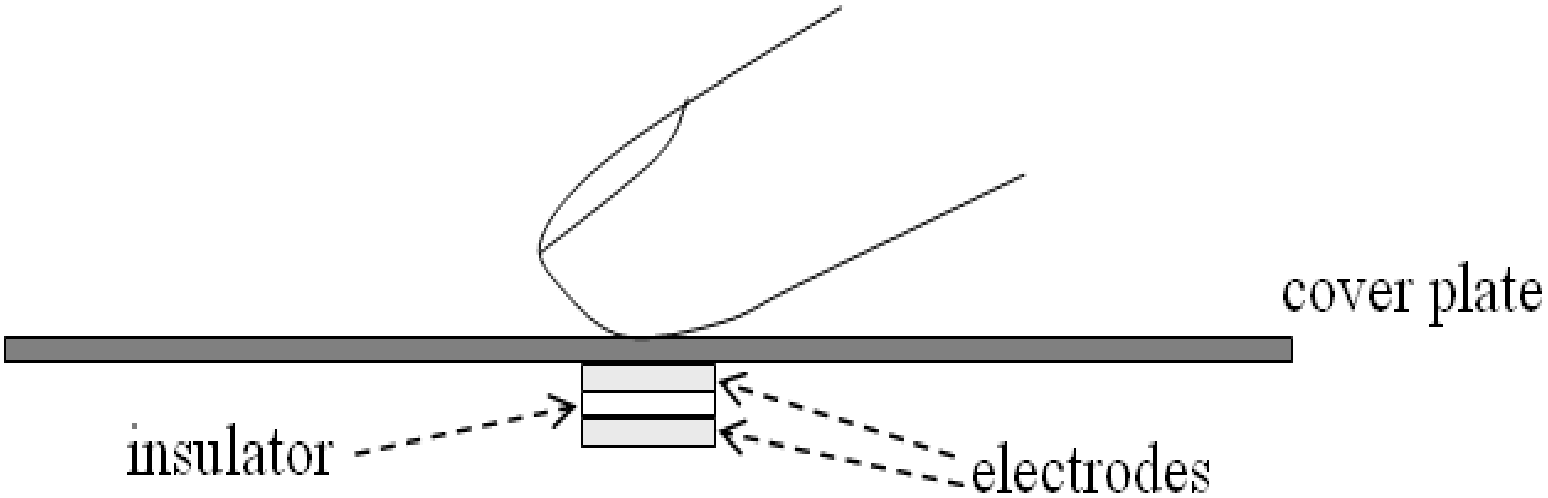
- Device interface then sends this ‘mouse input’ to the processor. It also senses the status of left and right mouse buttons. This device interface makes use of programmed I/O with interrupt.

# *Touch-sensitive input devices:*

- Basic principle of touch-sensitive input devices is illustrated in the next figure.

- Underneath the protective cover plate are two electrodes separated by a thin layer of insulating material (also known as *dielectric*).

- When human finger touches the cover plate just above the two electrodes, the effective *capacitance* between the two electrodes changes.




- This change is sensed by electronic circuit connected to the electrodes, and in this way the presence or absence of touch is sensed.

- A *touchpad* is created by providing a rectangular grid of such electrodes under the cover plate.

# Graphics Display

## Graphics display:

- Two types of devices are used – *liquid crystal display* (LCD) and *cathode ray tube* (CRT).
- Users nowadays expect much from a graphics display in terms of its resolution, range of colours, speed, etc.
- An image is made up of a large number of *picture elements* – known as *pixels* – arranged in a regular grid of closely-spaced rows and columns.

- The next figure – in part (a) – shows an image of a hand holding a pen.

- Pixels are arranged with a high density. Therefore from normal reading distance our eye does not see individual pixels, but only what appears to us as a ‘photographic’ image of the object.

- Part (b) shows *an enlarged view of the image*, in which individual square-shaped pixels are seen.

- The enlargement is *four times along each of the two dimensions* – which means that each pixel in (b) occupies sixteen times the area it occupies in the original image (a).

# An image and its enlarged view:

(a)

(b)

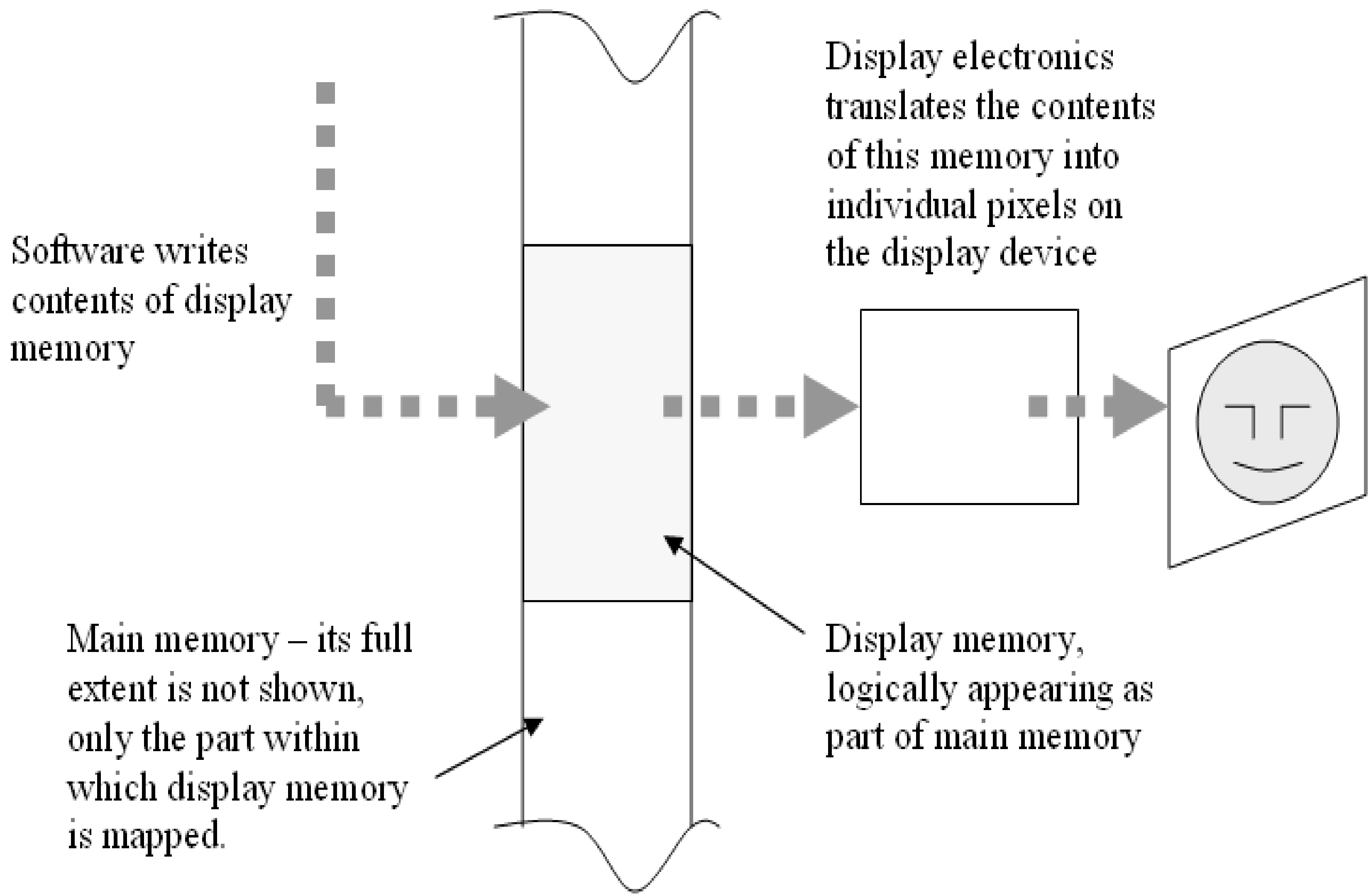
- Each pixel in this monochrome figure is white, black, or an intermediate shade of grey.

- Suppose this display provides 256 shades of grey, including white and black.
  - Shades can be numbered from 0 to 255, with 0 representing black and 255 representing white. So we need 8 bits of storage per pixel to represent its possible shades of grey.

- *Display memory* in the computer system stores the pixel-by-pixel binary encoding of on-screen display.

- Often, this display memory is provided within the physical memory address space of the processor, as shown schematically in the next figure.

# Graphical display sub-system:



- Display memory is shared by (a) running program which is using the display, and (b) display electronics.
- The program writes pixel values into display memory, while the display electronics transfers and ‘translates’ the display memory contents for the display device.
- To display animated images, sequence of thirty images per second is adequate.
  - Display device should be capable of changing the displayed image thirty times in a second.
  - Data paths in the system should have the capacity to handle such data rates.

- For animations stored on disk, the disk-to-memory data path should be capable of sustaining the high data transfer speeds needed.
  – Software must de-compress images at this rate and place them in display memory

- For images created ‘on the fly’ - as in a video game - software tricks are used to speed up the generation and display of images.

- Images being downloaded over a network also need to be de-compressed at the required rate.
  – If the network cannot deliver data at the required speeds, the animated image seen will be jerky rather than smooth.

# Display technology:

- Special-purpose *display electronics* transfers the binary-encoded pixels from display memory to the actual display device provided on the system.

- Then, suitable display technology makes each pixel visible to the eye.

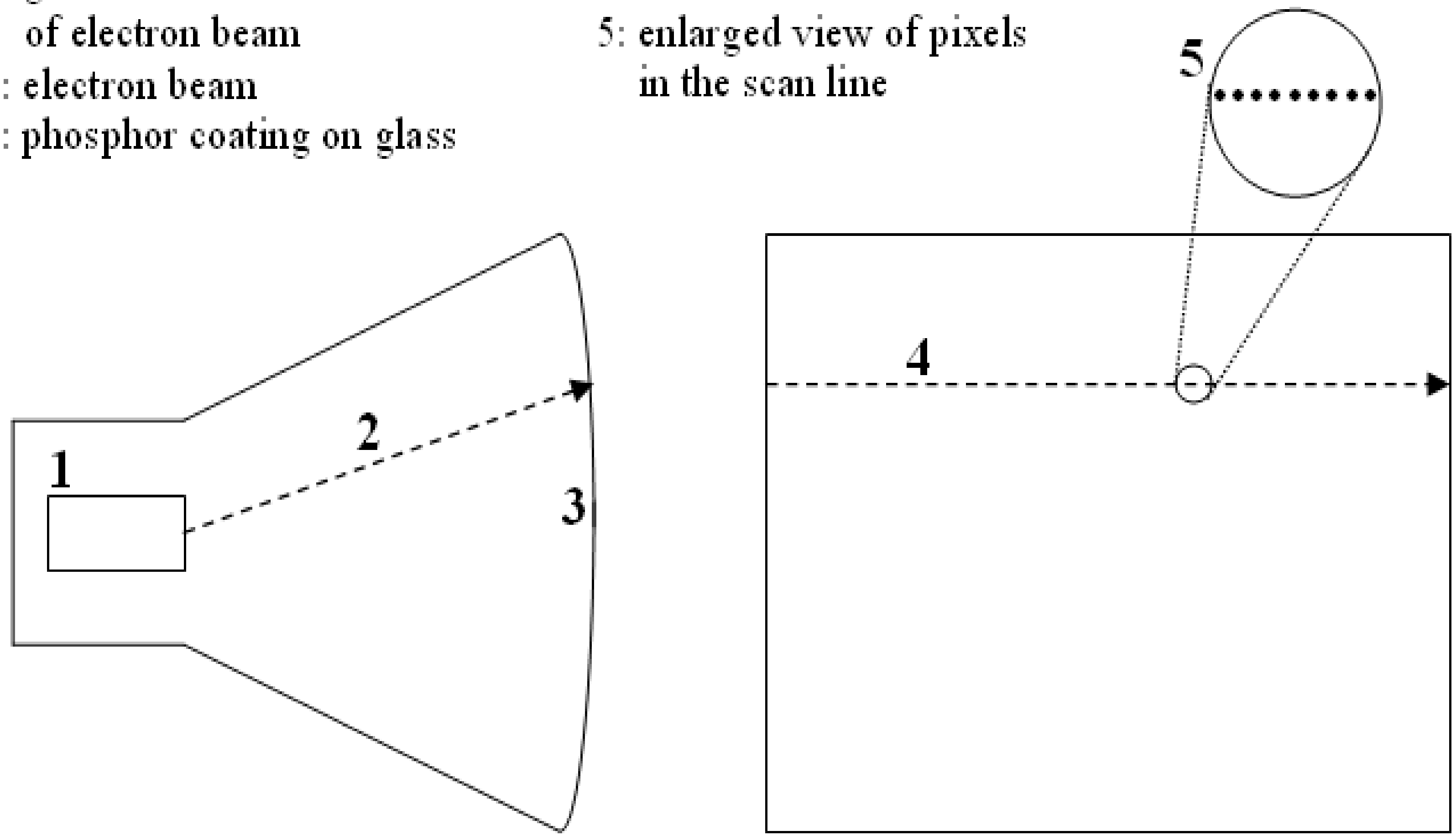
- Two main technologies are used today for displaying images:
  - one based on *cathode ray tubes* (CRT), and the other based on *liquid crystal displays* (LCD).

# *CRT Displays:*

- The term *cathode ray* refers to a beam of electrons – ‘cathode’ is the element which emits the electrons.
- Beam of emitted electrons is accelerated and focused using electric and magnetic fields. It becomes a ‘high-speed electronic pen’ which produces the image pixel-by-pixel on a glass surface coated with a phosphorescent substance.
- Phosphorescence at a pixel position lasts for only a few milliseconds. Therefore the displayed image must be *refreshed* many times a second.
- Static image also needs to be refreshed; animated images are created by changing the displayed image in every ‘frame’. [See the next figure.]

# Principle of operation of CRT display:

1: generation and deflection of electron beam
2: electron beam
3: phosphor coating on glass
4: one scan line
5: enlarged view of pixels in the scan line




(a) (b)

- Part (a) of the figure (side view):
  - Electron beam originates in the small rectangular element shown on the left, travels to the right, and hits the phosphor-coated inside front surface.
- Part (b) of the figure (front view):
  - Display screen has rectangular shape.
  - Electronics inside the tube deflects the beam such that it traces out horizontal scan lines on the front screen, one after the other, from top to bottom.
- To create pixels in each scan line, the beam is modulated in time. Colour & intensity of light at each pixel is determined by display memory.

- Pixel-by-pixel information in display memory must be translated into an appropriate modulation pattern.

- Thus the beam creates an image on the phosphor coating on the inside surface. Since the front surface is transparent, we see the image in front of the screen.

- Part (b) shows one scan line going from left to right, with a tiny part of the scan line enlarged to show the pixels.

- Pixels along one scan line, and also scan lines from top to bottom, are placed so close together that the user perceives a continuous image on the screen.

- A pixel in a colour display is made up of three distinct sub-pixels, one of each primary colour (R,G,B).

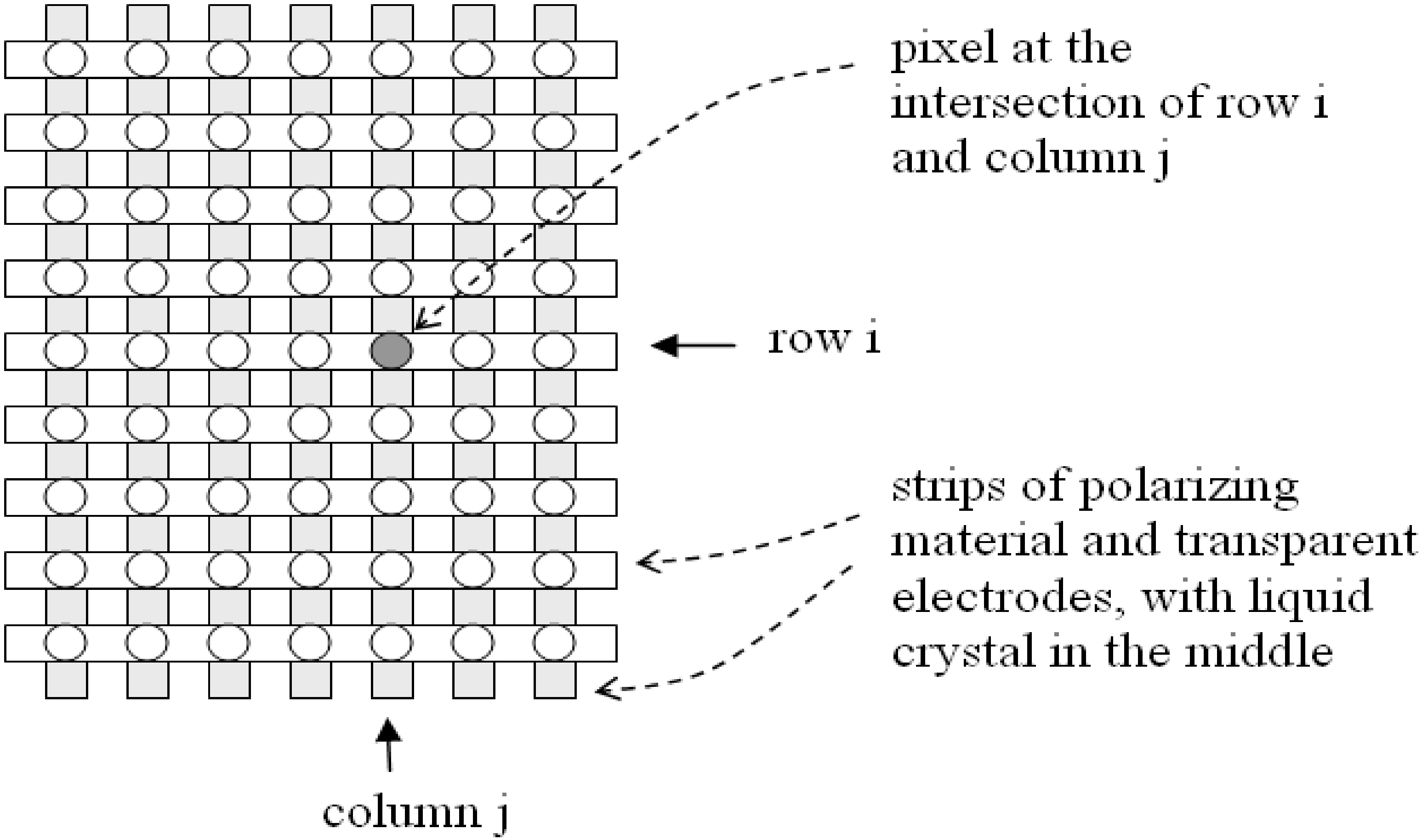
# *Liquid crystal displays:*

- Materials known as *liquid crystals* have the property that their transparency to light can be varied by applying a voltage along the path of light.

- To achieve such controlled transparency to light in a liquid crystal, *polarized* light is needed – for which a thin plate of polarizing material is used.

- See the next figure. Rectangular grid of pixels is made of thin horizontal and vertical strips of polarizer, and transparent electrodes for applying voltage.

- Liquid crystal forms a thin layer at each pixel position in the middle of this sandwich of strips.




- Horizontal and vertical strips of transparent electrodes allow voltage to be applied at each pixel position, according to the shade required.

- *Display electronics* translates contents of display memory into voltage pulses which are applied to the row and column electrodes.
  - At each pixel position, voltage pulses control the transparency of liquid crystal - and thus its observed intensity.

- In a colour liquid crystal display, transparency of liquid crystal to the three primary colours is independently controlled at each pixel position.

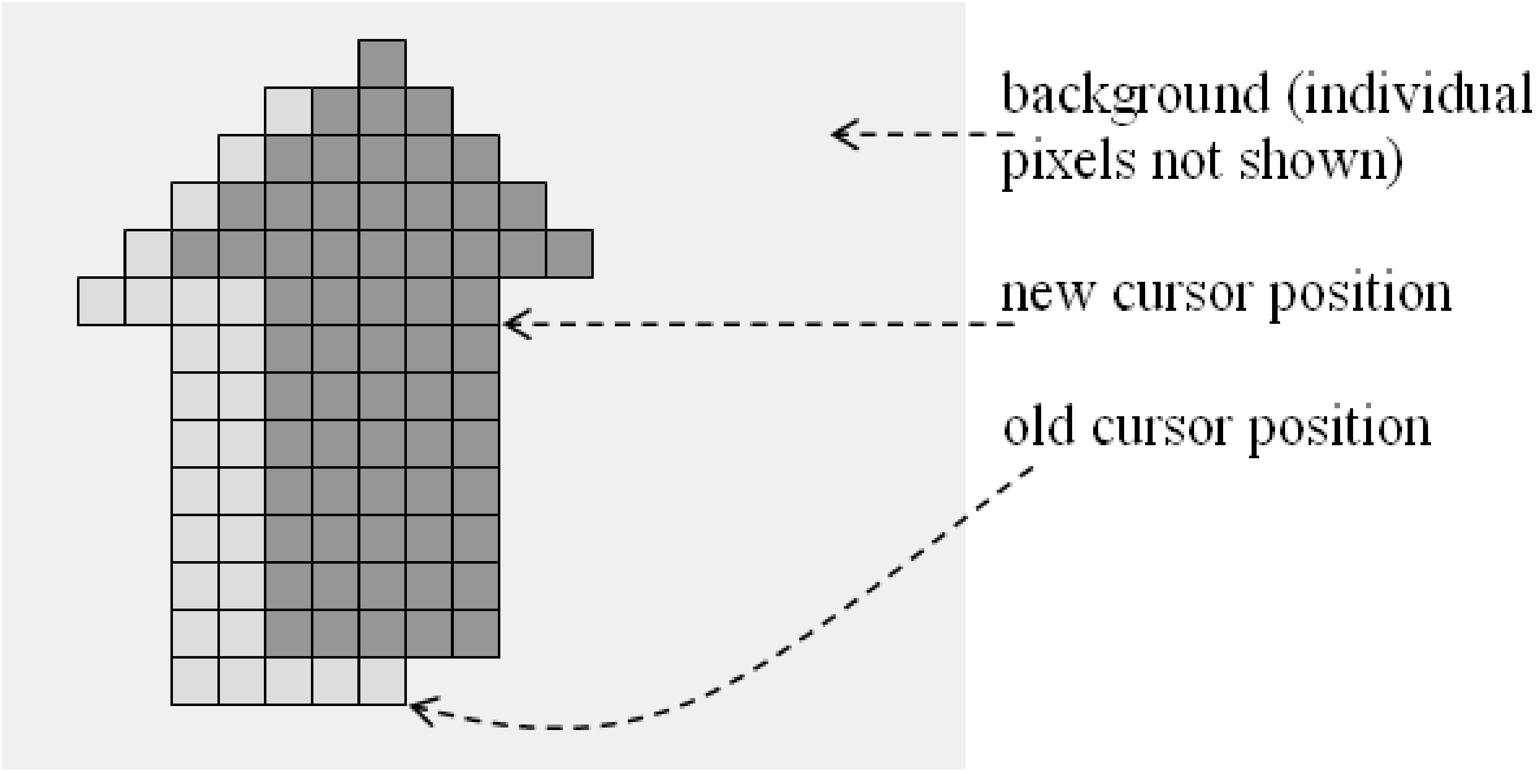
- Liquid crystal displays are flat in shape, since they do not use an electron beam, and they consume less power than CRT displays.

# Software for user interaction:

- Graphics displays can provide extremely responsive and effective *graphical user interfaces*.
- Example: Mouse and display screen are two distinct devices – one for input and one for output. Software can 'tie together' mouse movement and cursor movement on the screen. The following algorithm outlines how this is achieved:

```
repeat
{
   Read change in mouse position, say Δm
   Translate Δm into change in cursor position,
      say Δc
   Change display memory for new cursor position
}
```

- Many times in a second, the program reads in the change in mouse position, say Δm.
- Program translates Δm into a corresponding change in the on-screen cursor position, say Δc.
- When the 'program loop' is sufficiently fast, any discontinuity between hand movement and on-screen cursor movement is not perceivable.
- The next figure shows an enlarged view of one discrete step of on-screen cursor movement.




Background to the cursor is shown in a very light shade; the cursor in its old position is shown slightly darker, while in its new – i.e. changed – position, it is shown in a still darker shade.

- To achieve perceived movement of the cursor, the program should:
  - (i) replace background pixels by cursor pixels in the new cursor position, and
  - (ii) replace ‘exposed’ pixels of the cursor in its old position by background pixels.

- These operations are carried out by the program in display memory – and then (as seen earlier) display electronics does the rest.

- When the keyboard is used for input, there is one crucial difference:
  - Keys are necessarily pressed one at a time, so it is not always possible to create on the screen the effect of continuous and arbitrary movement.

- Graphical user interface uses complex entities such as *cursor*, *icons*, *dialog boxes*, and so on.

- Programmer thinks of an icon as an entity in itself, on which appropriate operations can be carried out.

- Therefore, libraries of functions are available to facilitate programming with ‘higher level’ entities such as *cursor*, *icons*, *dialog boxes*, and so on.

- This makes it possible, for example, to create a new icon with a certain shape, to define some of its properties, and to associate it with another entity or operation on the system.

# Summary

Elements of user interface:

- Keyboard, touchpad, mouse
- Graphics displays
- Display technology (CRT, LCD)
- Software for user interaction