

# **Frame Buffer & Video Controller**

## **Unit 1 – Lecture 3**

# Frame Buffer

Each screen pixel corresponds to a particular entry in a 2D array residing in memory. This memory is called a **frame buffer** or a **bit map**.

The number of **rows** in the frame buffer equals to the number of **raster lines** on the display screen.

The number of **columns** in this array equals to the number of **pixels** on each raster line.

The term pixel is also used to describe the row and the column location in the frame buffer array that corresponds to the screen location. A 512x512 display screen requires 262144 pixel memory locations.

# Frame Buffer

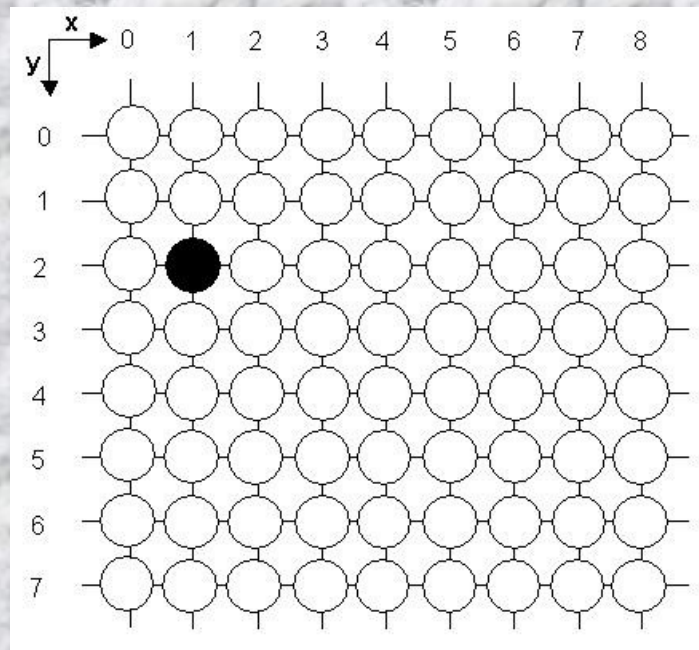
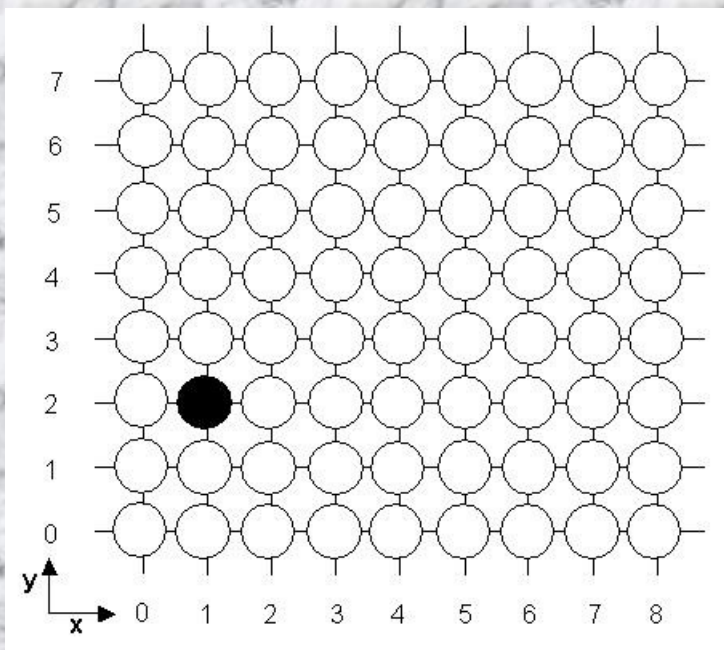
Whenever we wish to display a pixel on the screen, a specific value is placed into the corresponding memory location in the frame buffer array.

Each screen pixel's location and corresponding memory's location in the frame buffer is accessed by nonnegative integer coordinate pair  $(x, y)$ .

The **x** value refers to the **column**, the **y** value to the **row** position.

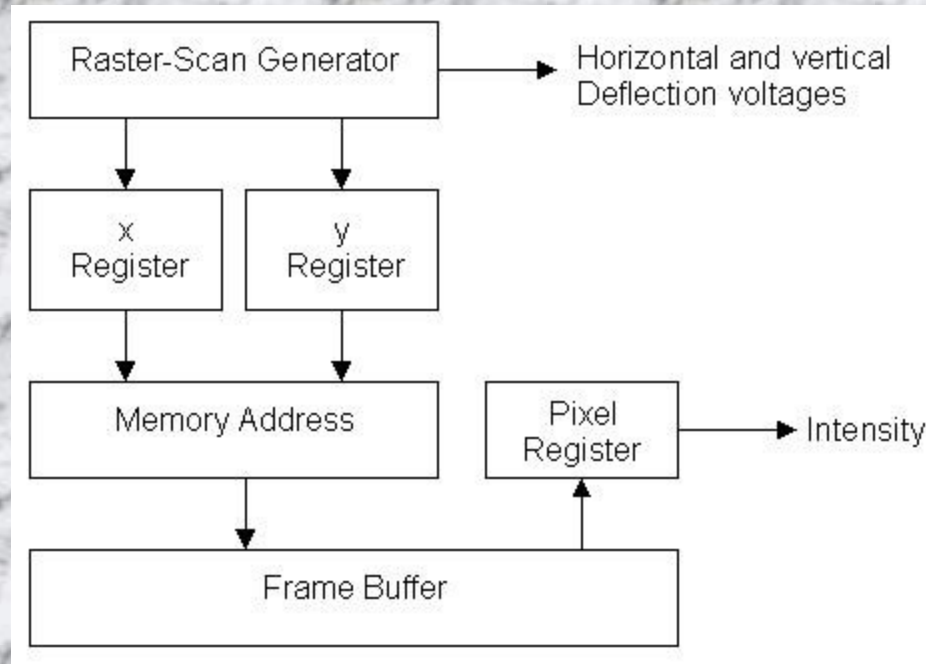
# Frame Buffer

The **origin** of this coordinate system is positioned at the **bottom-left corner** of the screen or it is positioned at the **upper-left corner** of the screen.



# Video controller

**Video controller** is used to control the operation of the display device (Monitor/Screen). Video controller accesses the frame buffer to refresh the screen. In figure, the basic refresh operations of the video-controller are shown.



## Video controller

Two registers are used to store the coordinates of the screen pixels.

Initially, the **x register** is set to **0** and the **y register** is set to the value for the top scan line.

The contents of the frame buffer at this pixel position are then retrieved and used to set the **intensity** of the CRT beam.

Then the **x register** is **incremented** by **1**, and the process is repeated for the next pixel on the top scan line.

## Video controller

This procedure is continued for each pixel along the top scan line.

After the **last pixel** on the **top** scan line has been processed, the **x register** is **reset** to **0** and the **y register** is set to the value for the **next scan line** down from the top of the screen.

Pixels along this scan line are then processed in turn, and the procedure is repeated for each successive scan line.

## Video controller

After cycling through all pixels along the bottom scan line ( $y=0$ ), the video controller resets the registers to the first pixels position on the top scan line and the refresh process starts over.

The screen must be **refreshed** at a **rate** of at least **60 frames per second**.



# Input and Output Devices

The user of an interactive graphics system communicates with the graphics program by means of **input devices** such as **keyboard, mouse, joystick, light pen, graphics tablet (digitizer), touch panels, voice systems,** and **scanners.**

Typically, the primary **output device** in a graphics system is a video monitor such as **Cathode Ray Tube (CRT)** and **Liquid Crystal Display (LCD)**. We can obtain hard-copy output for our images in several formats for presentations or archiving. Hard copy devices include **slides film, printers,** and **plotters.**

# Computer display standard

Various computer display standards or **display modes** have been used in the history of the personal computer.

They are often a combination of

**Display resolution:** specified as the width and height in pixels,

**Color depth:** measured in bits, and

**Refresh rate:** expressed in hertz.

# Computer display standard

A computer image is usually represented as a discrete grid of pixels.

The number of pixels determines the resolution of the image. Typical **resolutions range** from 320x200 to 2000x1500

**The color depth:** is the number of distinct colors that can be represented by a pixel depends on the number of bits per pixel (bpp).

A **1 bpp** image uses 1 bit for each pixel, so each pixel can be either on or off.

# Computer display standard

Each additional bit doubles the number of colors available, so a 2 bpp image can have 4 colors, and a 3 bpp image can have 8 colors:

- 1 bpp,  $2^1 = 2$  colors (monochrome)
- 2 bpp,  $2^2 = 4$  colors
- 3 bpp,  $2^3 = 8$  colors
- ...
- 8 bpp,  $2^8 = 256$  colors
- 16 bpp,  $2^{16} = 65,536$  colors (**Highcolor**)
- 24 bpp,  $2^{24} \approx 16.7$  million colors (**Truecolor**)

## Computer display standard

For **color depths of 15 or more** bits per pixel, the depth is normally the sum of the bits allocated to each of the red, green, and blue components(**RGB**).

**Highcolor**, usually meaning 16 bpp, normally has **five bits** for **red** and **blue**, and **six bits** for **green**, as the human eye is more sensitive to errors in green than in the other two primary colors.

# Computer display standard

For applications involving **transparency**, the **16 bits** may be divided into five bits each of red, green, and blue, with one bit left for transparency.

A **24-bit** depth allows 8 bits per component.

On some systems, **32-bit** depth is available: this means that each 24-bit pixel has an extra 8 bits to describe its **opacity** (for purposes of combining with another image).

# Applications of Computer Graphics

- ✓ Computer Aided Design (CAD)
- ✓ Computer Aided Geometric Design (CAGD)
- ✓ Entertainment (animation, games, etc.)
- ✓ Computer Art
- ✓ Presentation Graphics
- ✓ Education and Training
- ✓ Geographic Information Systems (GIS)
- ✓ Visualization (Scientific Vis., Inform. Vis.)
- ✓ Medical Visualization
- ✓ Image Processing
- ✓ Graphical User Interfaces