# Johns Hopkins
# Engineering for Professionals

# 605.767 Applied Computer Graphics

Brian Russin

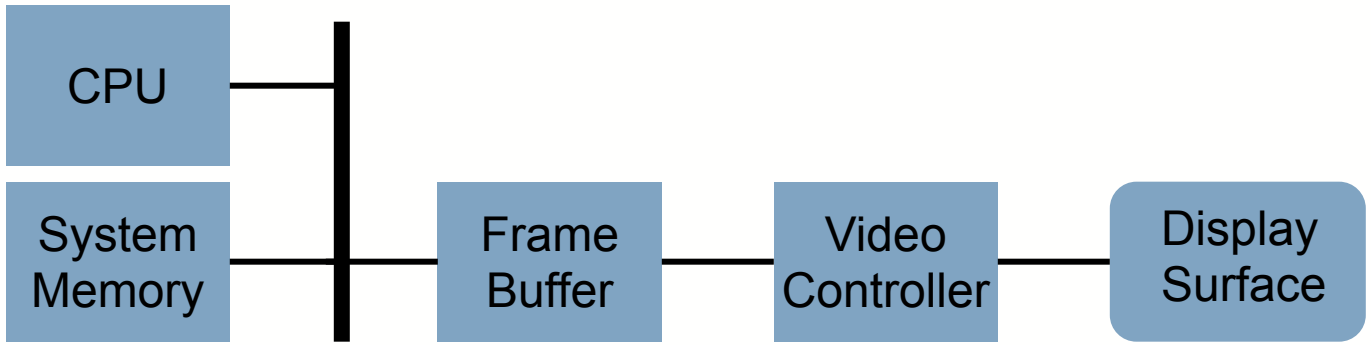
# Module 11E
# Displays and Buffers

# Graphics Architecture

- Graphics hardware overview
  - Display devices
    - Raster displays
  - Buffers and memory usage
- Graphics Architecture
  - Taxonomy
  - Bus and memory bandwidth
  - Case studies

# Simple Raster Graphics System

- Most computer graphics systems today are raster display systems
  - Example: computer with video monitor or flat panel display
- CPU runs programs that generate computer graphics images
  - Image is stored in **framebuffer**
  - Image is presented on a **display surface**
    - E.g., CRT monitor or flat panel display
  - **Video display controller** scans the framebuffer and outputs to display surface
- **Resolution**: number of pixels that can be displayed
  - Horizontal and vertical
    - Common systems today are 1280x1024 or greater




Block Diagram of a Simple Computer Graphics System

# Scanning the Frame Buffer

- **Video display controller (VDC)** reads pixel data from the framebuffer and produces a video signal to control the display device
  - So the proper pixel in the display device is drawn with proper color
  - Runs autonomously – not under program control
    - Synchronized with output device
  - Hardware support for common interfaces (VGA, DVI, HDMI)
    - Analog output
      - Video controller converts digital values in the framebuffer to analog signals sent to the display device
        - Digital to Analog Converters (DAC)
        - Gamma correction should be performed before data reaches DAC
    - Digital output
      - Standard digital interface for flat panel devices recently specified
- Data is read in raster scan order
  - Row by row through the framebuffer
  - Creates output for one **scan line** of pixels from left to right
- VDC may perform other functions

# Refresh Rates

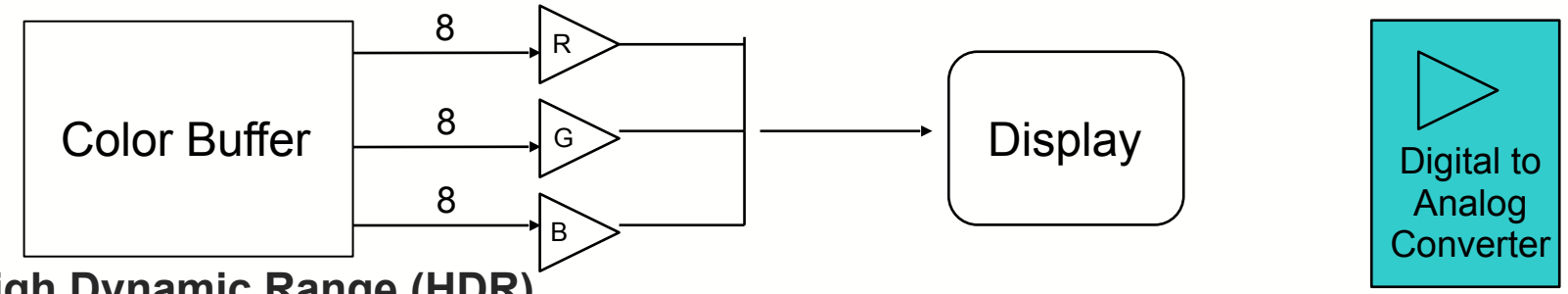
- Image decays quickly on video display devices
  - CRTs and flat panels
  - Phosphors that emit colored light only glow for a fraction of a second
    - Called **persistence**
  - Displays must be constantly **refreshed**
    - At least 60 Hz is standard, but up to 600 Hz can be achieved
- Horizontal / Vertical retrace - movement of electron beam from right to left and bottom to top
  - **Line rate** or **horizontal refresh rate**
    - Time it takes to scan one left-right-left cycle
  - **Vertical refresh rate**
    - Number of times per second the beam goes from bottom to top-left corner
- **Flicker** effects occur if the image is not refreshed often enough
  - Most noticeable in the peripheral vision
    - Can be very distracting
  - Most viewers notice flicker at refresh rates less than 72 Hz

# Interlaced and Progressive Scans

- Computer monitors are usually **non-interlaced**
  - Also called **progressive scan**
- Televisions are generally **interlaced**
  - Alternates between even and odd horizontal lines each refresh

# Color Buffer

- **Framebuffer** memory stores pixel display values
  - Most commonly part of the graphics card
- True color – stores the color directly in the pixel value
  - Usually 3-4 bytes per pixel with 8 bits per color component

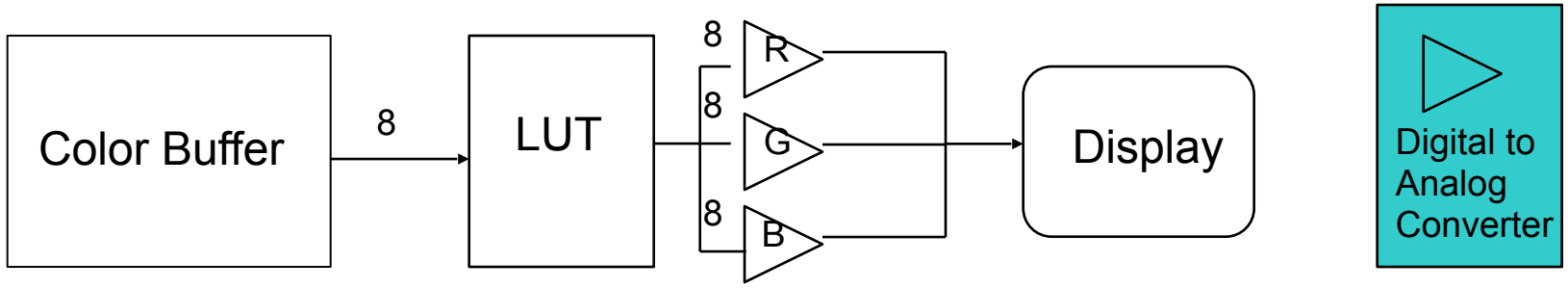



- **High Dynamic Range (HDR)**
  - 10 bits per channel
  - 10 times the brightness and 100 times the contrast of a typical monitor
- High color
  - 16 bits to represent color per pixel
  - Generally 5 bits R, 6 bits G, 5 bits B
    - More bits for green since it has largest luminance impact on visual system
    - Can lead to **banding** or **contouring**

# Indexed Color and Color Lookup Tables

- Indexed color – pixel value stores an index into a color table
  - Uses less memory – usually 8 bits per pixel
    - Number of **bit planes** is number of bits per pixel in the framebuffer
  - **Color Lookup Tables** associate pixel values and displayed colors
    - Programmable color selection
- **Palette** - total set of colors the system is capable of displaying
  - Number of bits in each entry in the lookup table – LUT width
    - Typically 24 bits: 8 bits red, 8 bits green, 8 bits blue - 16.7M possible colors
- Color capability in indexed color systems

A system with b bit planes and an LUT width of w can display $2^b$ simultaneous colors from a palette of $2^w$



# True Color

- True color often uses 4 bytes per pixel
  - RGBA – where A is for alpha
    - 32 bit representation also can improve performance
    - Some architectures optimized for groups of 4 bytes
  - 24 bit representation (RGB) is called **packed pixel format**
    - Saves memory
  - 24 bits is usually acceptable for real-time rendering
    - Banding effects much less noticeable
    - Gamma correction can reduce effective number of color levels slightly
  - Some systems use higher than 24 bits for internal color representation
    - For multi-pass methods
      - Uses higher precision to improve fidelity of color computations
- OpenGL supports color as float values in range [0.0f,1.0f]
  - Essentially 24 bits per color
    - Mantissa

# Z Buffer and Stencil Buffer

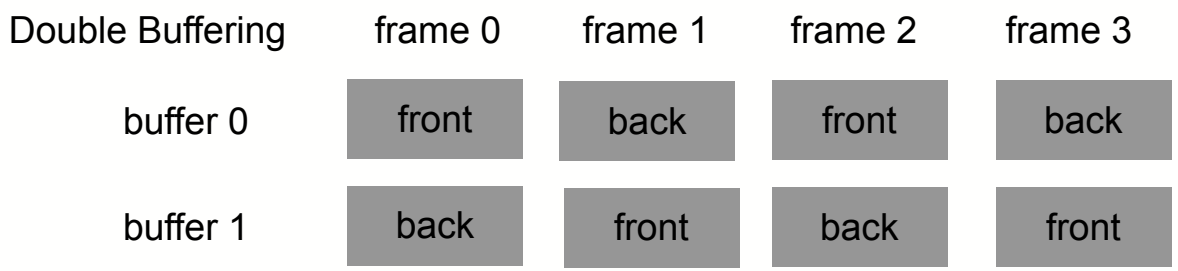
- Z Buffer or depth buffer is used for visible surface detection
  - Generally 16, 24 or 32 bits per pixel
  - 8 bit stencil buffer stand-alone, or if 24 bits dedicated to depth
- Orthographic projection
  - Distances between z values are constant
- Perspective projection
  - Non-uniform depth distribution
  - More precision near the near clipping plane
    - Depths are compressed near the far plane
  - Higher depth precision can help avoid z buffer aliasing issues
    - Called z fighting
    - Text gives example of sheet of paper on desk
  - Keeping ratio of near over far plane high helps

# Single Buffering

- Single buffering not generally acceptable for real time graphics
  - If it takes longer than one refresh to generate a frame we will see partial images as they are being drawn
  - If frame can be drawn within refresh rate can see **tearing**
    - When clearing buffer of drawing large polygons
    - See partial images as beam passes areas being drawn
    - Older architectures allowed queries for where the refresh / beam was
      - Could synchronize drawing to avoid tearing
  - Single buffering can be useful for pipeline optimization

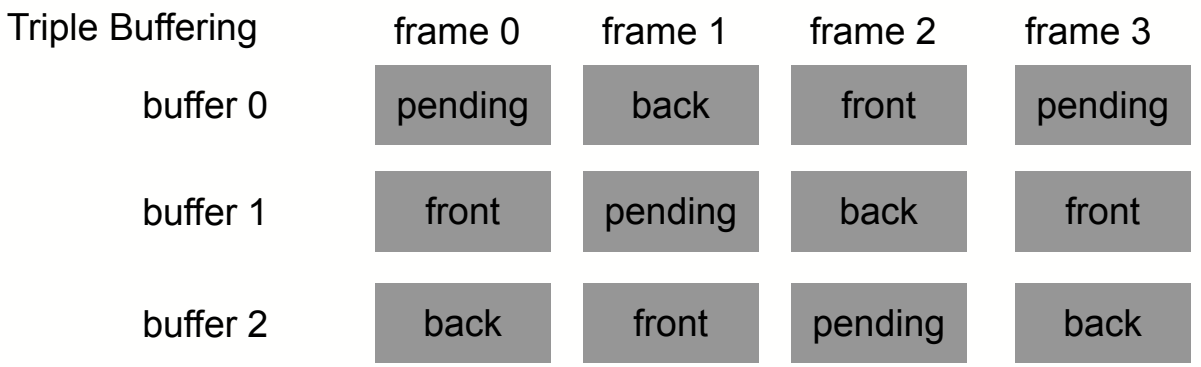
# Double Buffering

- Most systems provide double buffering
  - Ability to draw image to background buffer (called **backbuffer**) and then **swap** it to the screen
  - Essential for smooth animation and scene redraw
  - Memory required is at least as large as the framebuffer
- Applications that control the whole screen
  - Use color buffer flipping or page flipping
    - Change address of buffer memory areas
    - Swaps role of the buffers
- Windowed applications copy back buffer to front buffer
  - Hardware called the BitBLT engine performs **BLT swapping** or **blitting**



# Triple Buffering

- Can augment the double buffer with a 2nd double buffer
  - Called the **pending buffer**
    - Technique called **triple buffering**
- Advantage – better load balancing
  - Can start drawing new scene more quickly
    - Clear pending buffer and start redrawing while back buffer waits for swap
- Disadvantage
  - Latency can increase – delayed response to user events
- Supported in DirectX, OpenGL extensions, Vulkan, Metal



# Stereo Buffers

- Some hardware supports stereo rendering and viewing
  - Two images are used to make objects look three dimensional
    - **Stereo vision** or **stereopsis**
  - Section 21.3.1 (18.1.4 in 3rd Edition)
  - One view is rendered for left eye and one for the right eye
    - Images are called a stereo pair
  - Several hardware options to display these images
    - 2 small screens in front of each eye – head mounted display
    - Shutter glasses in which each eye is only allowed to view one image at a time
      - Synchronized with monitor which alternates images

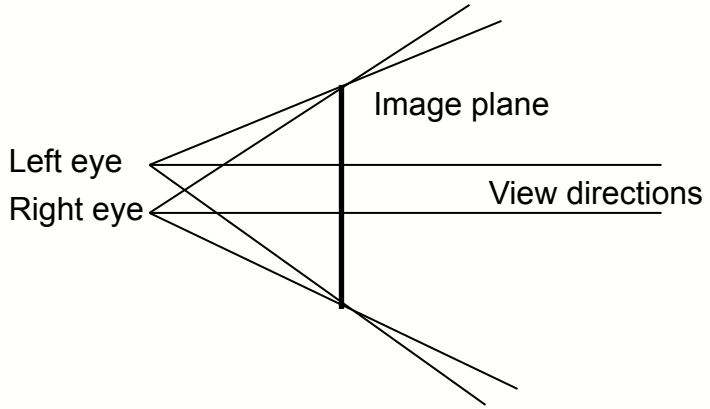



Image plane is shared between 2 frustums
Frustums are asymmetric