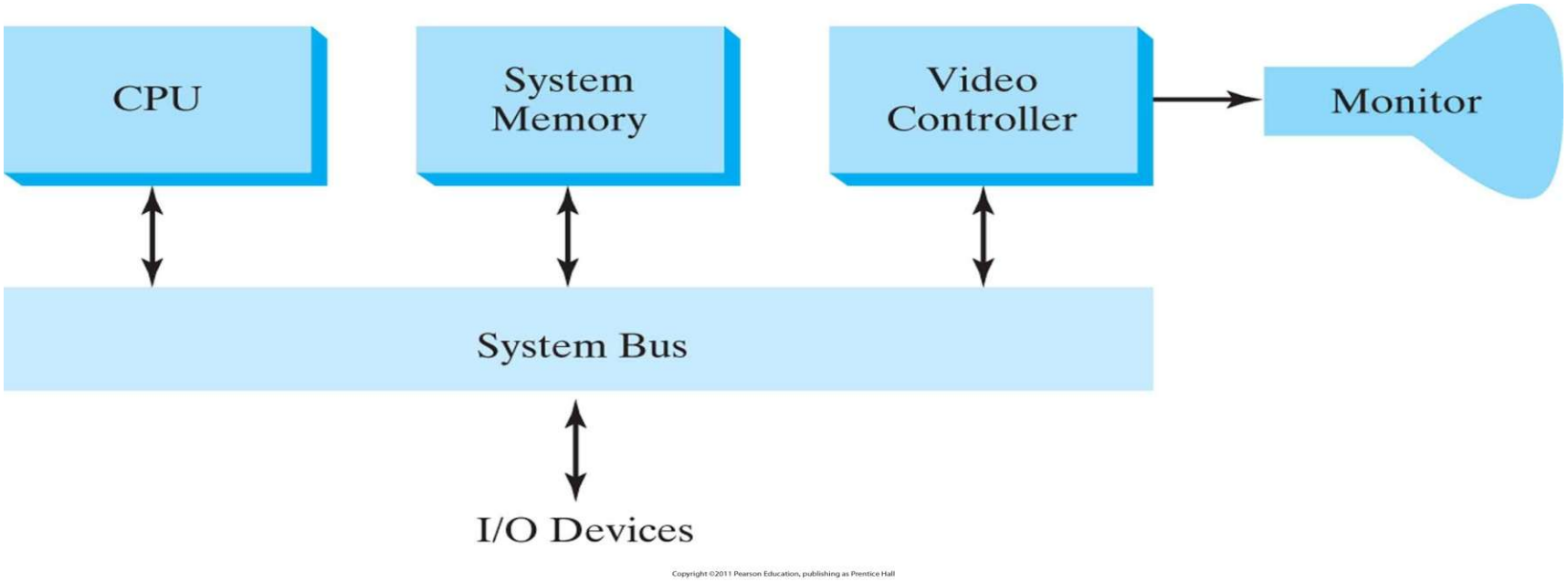
Lecture 1 Computer Graphics Hardware

- Basic graphics hardware components
  1. Display device (monitor)
  2. Video controller
  3. Memory
  4. CPU
  5. System bus



Display Devices

- Variety of display devices (monitors)
  - Cathode Ray Tubes (CRTs)
  - LCD, LED, etc.
  - Projectors

- Fundamentally a display device has the function to present a dot of a certain color at (x, y) position on screen.

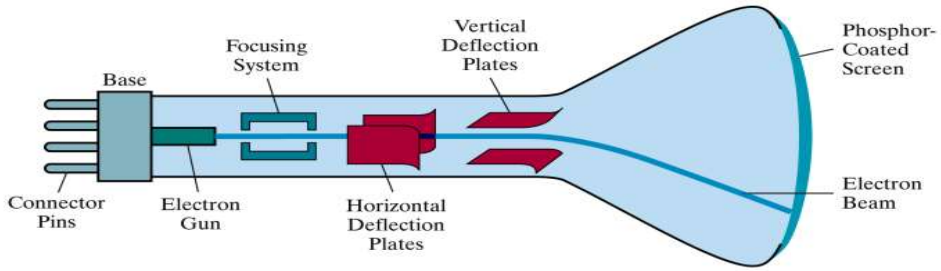


Figure 2-4
Electrostatic deflection of the electron beam in a CRT.

Dots to objects

- A computer image consists the drawing of objects. An object drawing on screen consists of dots of various colors.

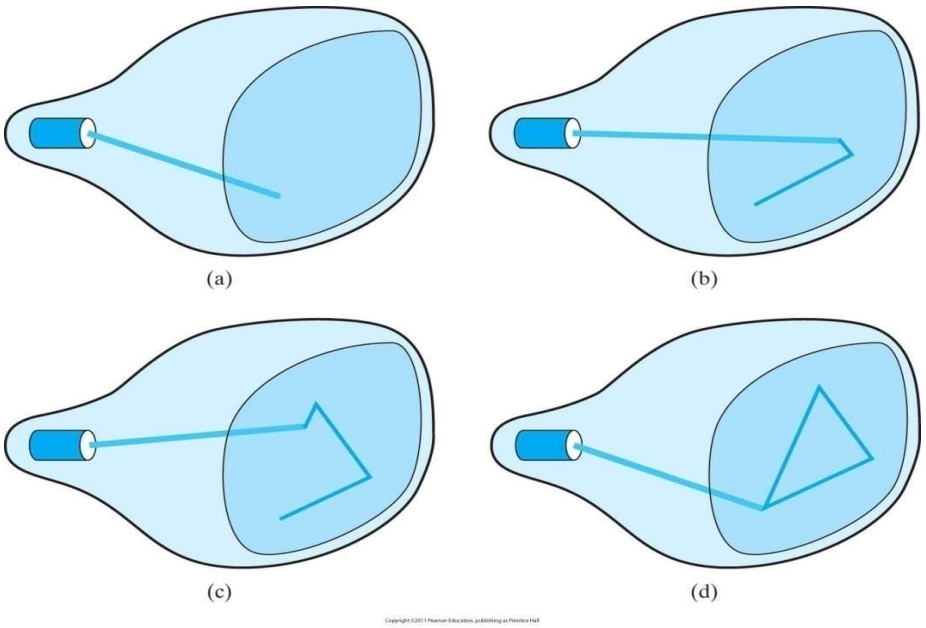
- If we know how to put a dot on a screen, then we can display an object by putting all dots of the object on screen like we do painting.

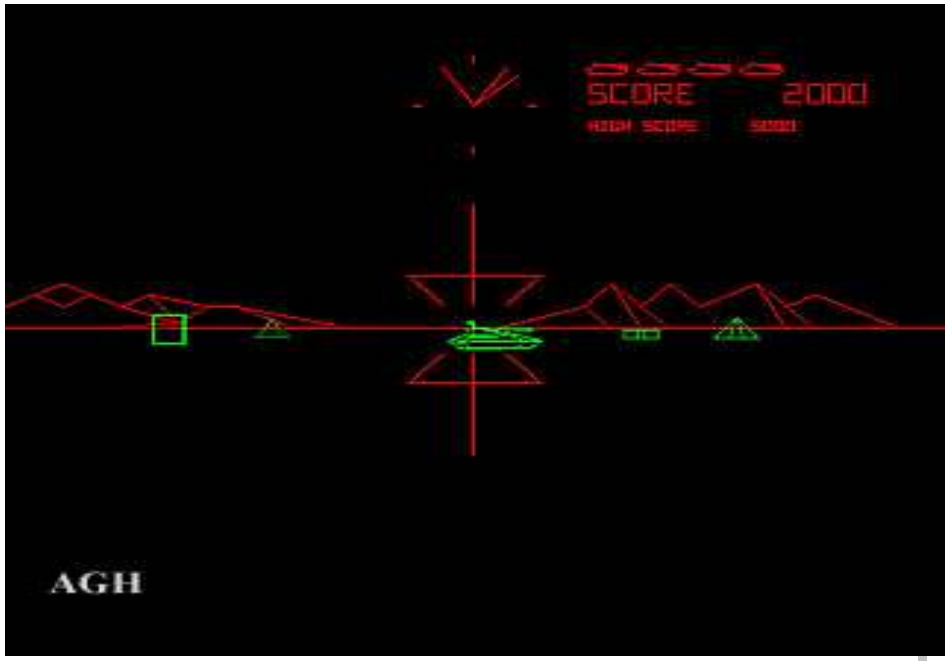
- There are two approaches of displaying images on screen.
  - Vector display: directly put dots of objects onto screen.
  - Raster display: put all dots of object on an image memory, then map the image in the image memory to screen. This is an indirect approach.

Vector Displays

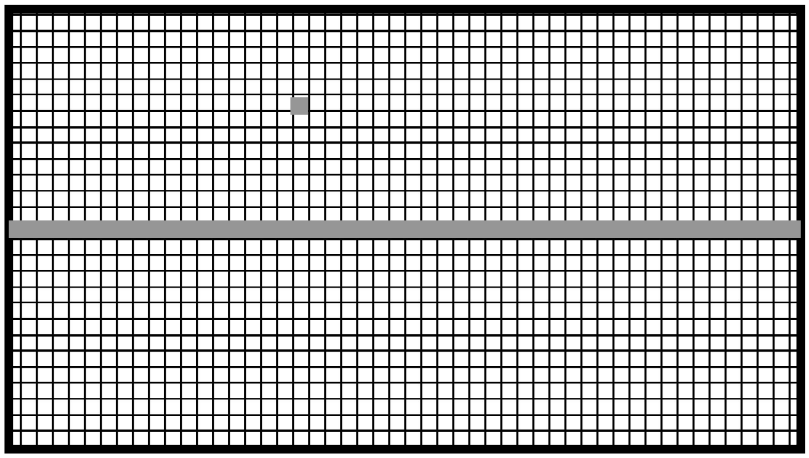
- Vector displays
  - Draw an object dot by dot
  - Control X,Y with vertical/horizontal plate voltage
  - Use intensity as Z
- Advantages
  - Intuitive like using pen to draw image.
  - Fast for simple images.

  E.g. plotters draw diagrams
- Disadvantages
  - Need to compute the position and color of next dot at time of displaying
  - Expensive devices
  - Slow for complex images

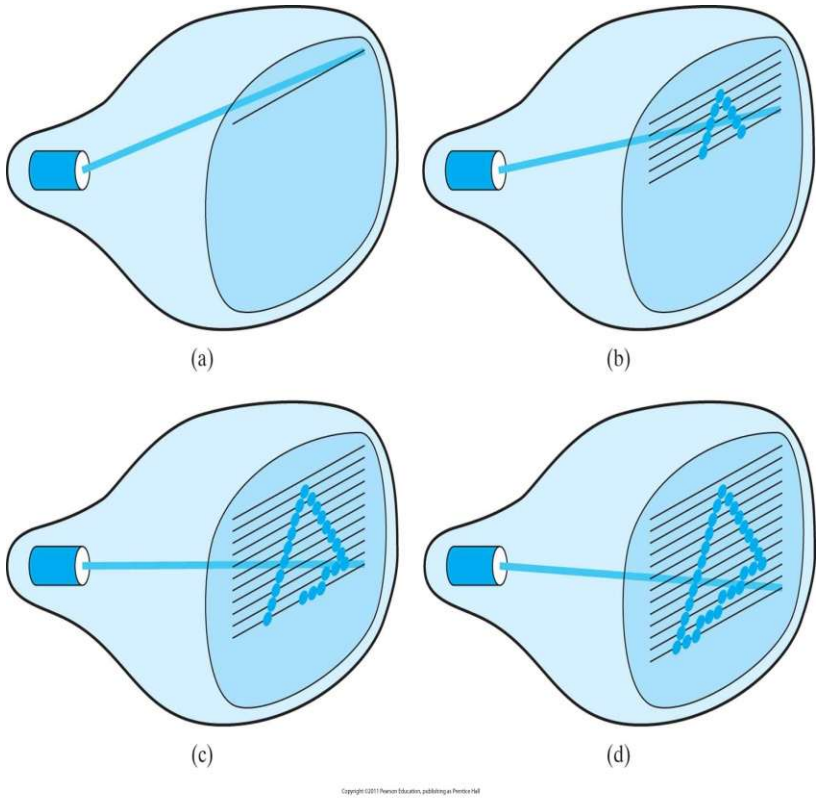




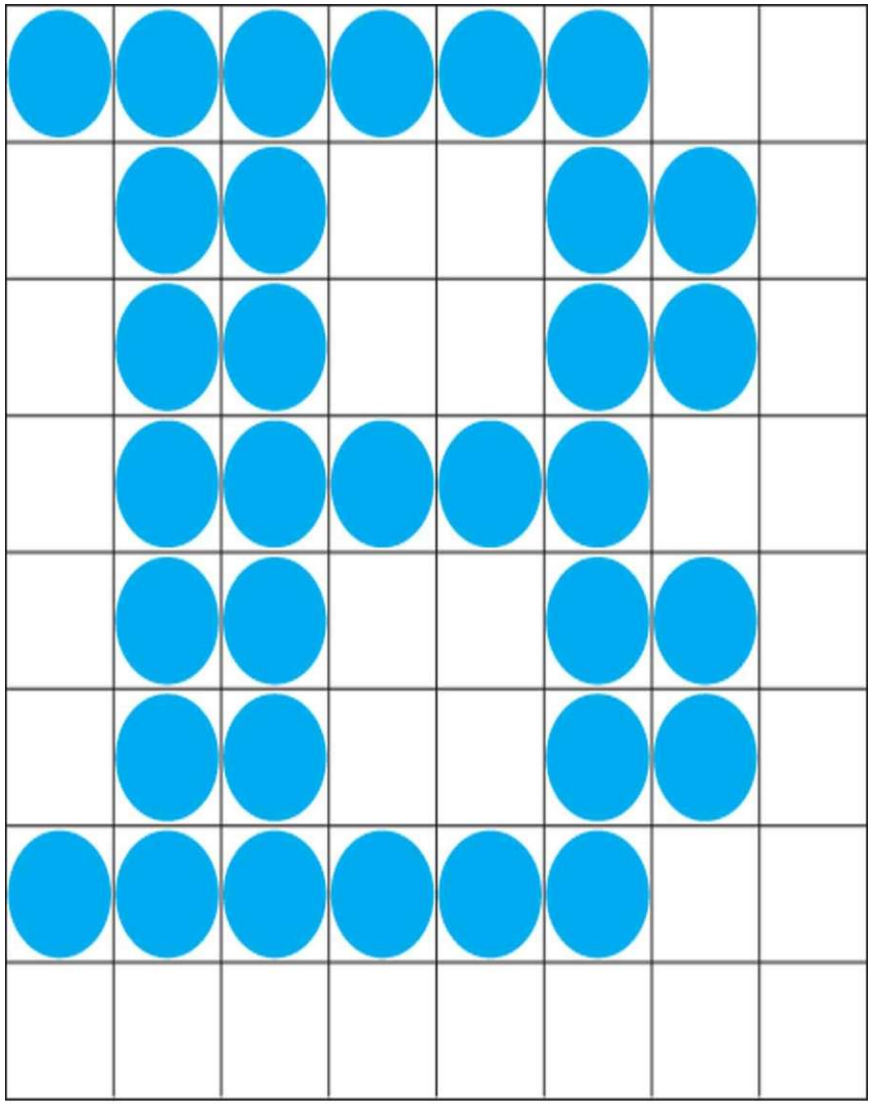
Raster Displays



- Raster (or raster pattern): an image representation by dividing image into lines (rows), and each line into dots.
- Framebuffer: the memory block to store data of all dots of raster pattern line by line. Each dot of an image has a corresponding position in the memory, the value stored in the memory position represents the data, e.g. color, of the dot.
- Video controller : scans each line of raster in framebuffer, set the dot color on screen according to the value stored at the position in the framebuffer.



Example: raster pattern of characters



Hardware for raster display

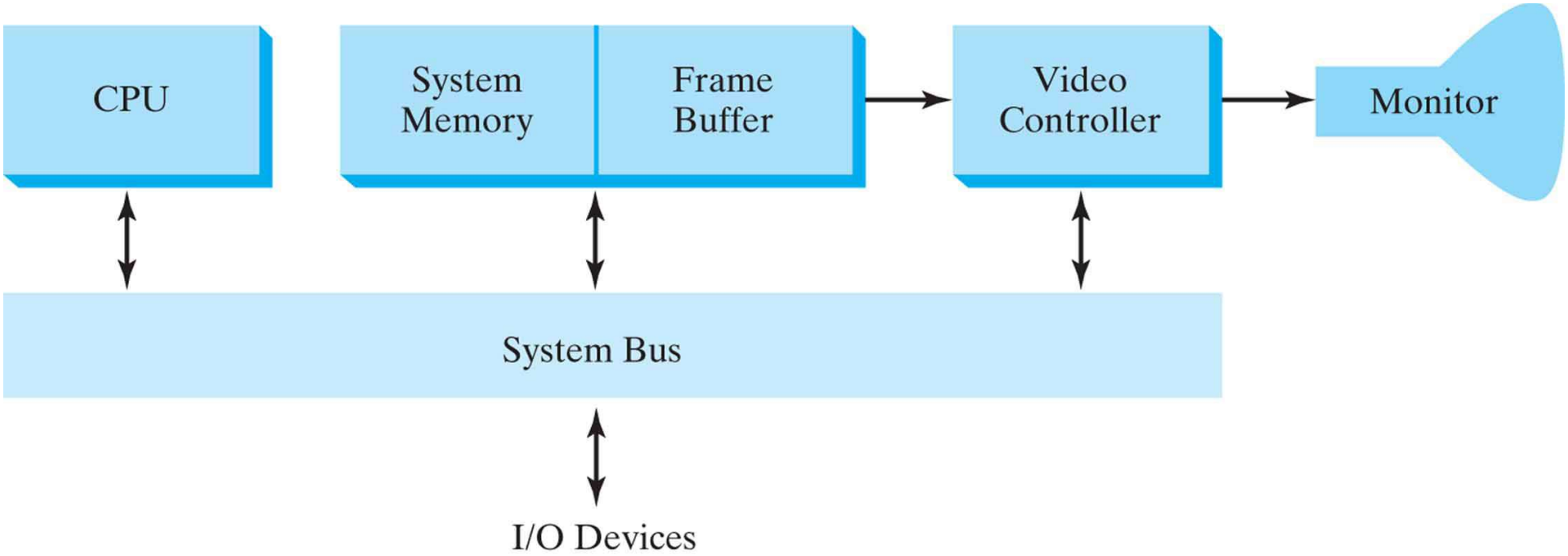
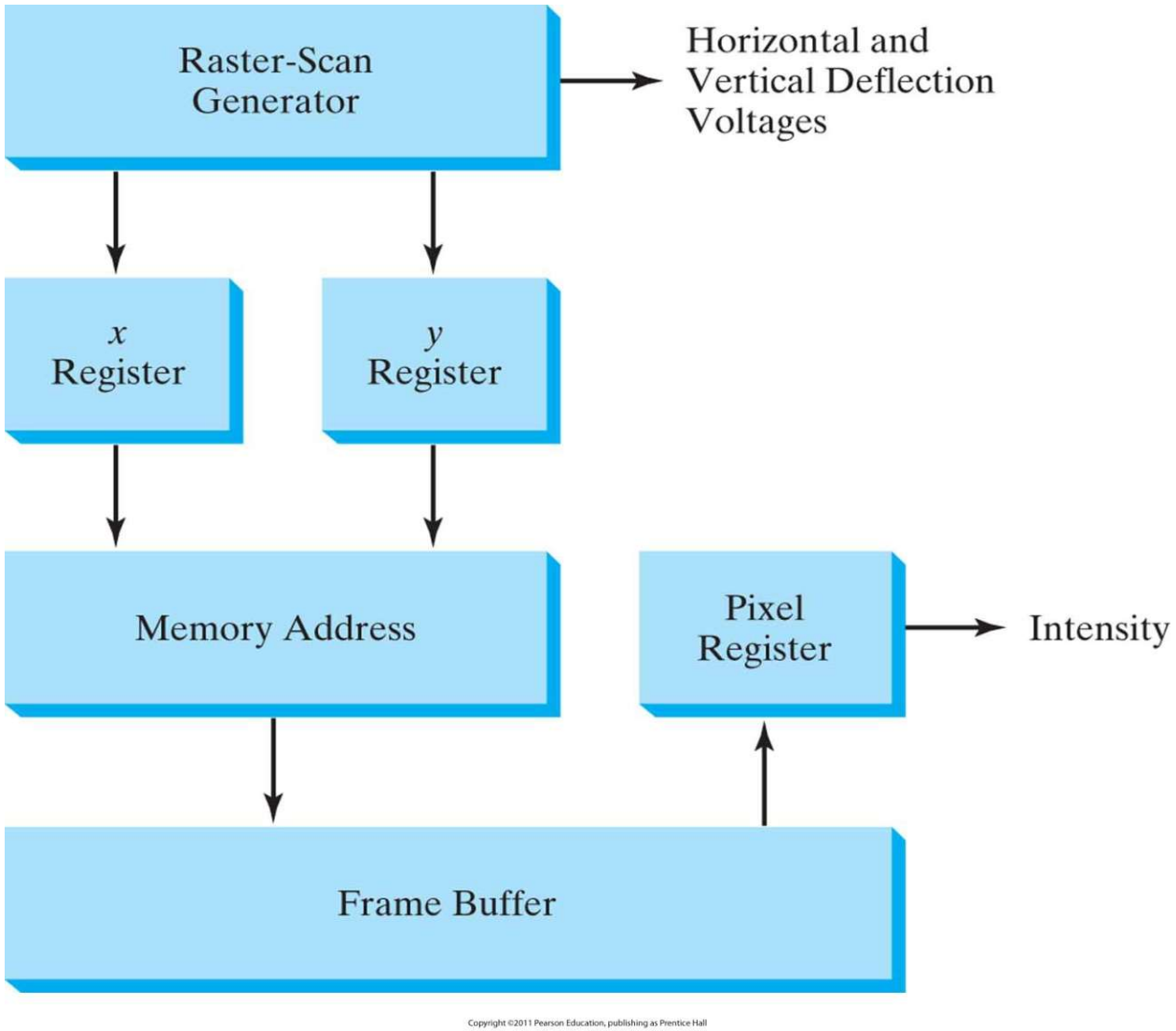


Figure 2-17 Architecture of a raster system with a fixed portion of the system memory reserved for the frame buffer.

Render (draw, rasterize) image to framebuffer

Using CPU



Raster Displays

- Advantages
  - Images are computed/generated/stored in raster a head of displaying. Avoid displaying uncompleted image.
  - Displaying is just mapping the image from raster to screen.
  - **Image generation and displaying are separated. The displaying is independent of the complexity of images.**

- Disadvantages
  - It requires separate memory for raster to store image.
  - It requires high performance video controller to scan raster to display on screen.
  - **It's slow to render complex images, slow for interactive applications and animations.**
    GPU is introduced to accelerate the image generation.

Accelerating image rendering by display processor (e.g. GPU)

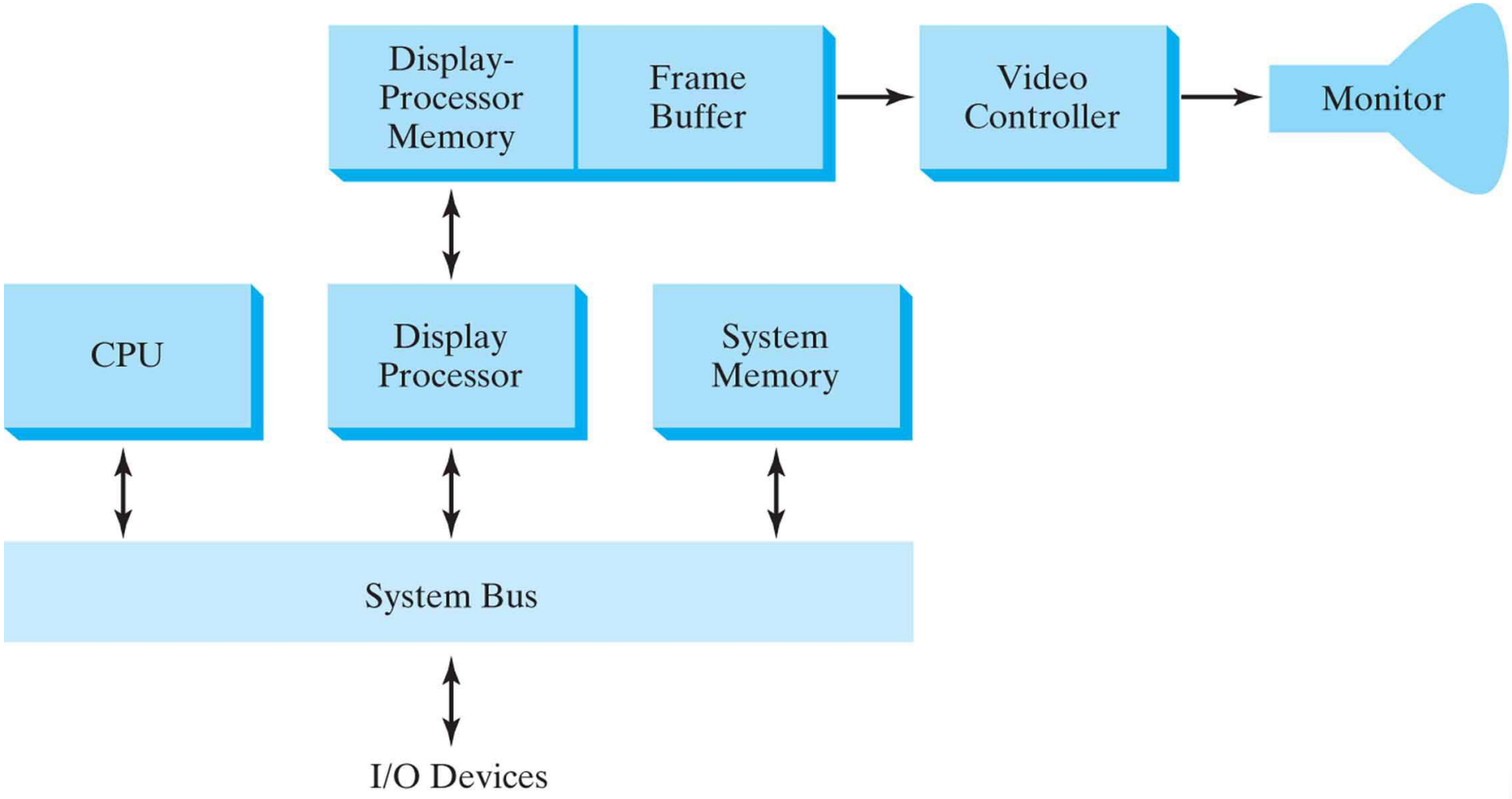


Figure 2-20 Architecture of a raster-graphics system with a display processor.

Display Processor is the Graphics Processing Unit (GPU) in today's computers

Terms of Raster Displays

- Raster pattern is a rectangular array of dots.
  - A dot, also called a picture element, or simply a pixel
  - A scan line is a row of pixels.

- Resolution refers to the number of pixels, typically represented as the number of pixel per scan line times the number of scan lines.

  Examples: 640 X 480, 1024 X 768
  The first number is the number of pixels per rows, the second number is the number of rows.

- *Each pixel corresponds to a memory space in framebuffer, which can be 1 bit, 8 bits, 16 bits, or 24 bits. The number of bits is called color depth.*
  The value of a pixel, i.e., the data stored in the corresponding memory space, represents the color of the pixel according to certain color model and coding scheme, e.g. RGB

- The image represented by raster pattern is called bitmap or pixmap

Framebuffer

- The framebuffer is a part of RAM in a computer allocated to store image in raster pattern. Framebuffer can be a part of main memory, or on separate video card.

- Framebuffer size determines the maximum resolution and color depth of the image.

  Framebuffer size = resolution × color depth,

  Example:

  640X480X8 bit = 2457600 bits = 307200 Bytes

Frame, refreshment, frequency, scanning

Raster display paints the screen by synchronizing scanning the raster pattern. Image in framebuffer is read out by video controller to display on the screen.

- A frame refers to a full scan and display of image on screen.
  A frame must be "refreshed" to draw new images
  - As new pixels are struck by electron beam, others are decaying
  - Electron beam must hit all pixels frequently to eliminate flicker

- Frequency refers the number frames per second the on the screen.
  Example, 60 frames/sec

- Two approaches of scanning to create a frame:

  - Progressive scanning: Scan one line after another

  - Interlaced Scanning: Scan all odd lines one by one and then scan all even lines one by one.

How to render objects to framebuffer

- Suppose we have a list of objects. The list of object instances are represented by data structures and stored in memory. How do we render the objects to framebuffer?
- There are basically two rendering approaches
  - Rasterization: for each object, computer and set the visible pixels of the object to framebuffer.
  - Ray tracing: for each pixel, create a ray from eye to the pixel, compute the color of an object that the ray hits, the set the color to the pixel.
  - **Hybrid approach combines the both.**
  - **Recently an AI approach is introduced to predict pixel colors.**

NVIDIA Keynote at SIGGRAPH 2023, August 9
https://www.youtube.com/live/3qSQjRaseos?feature=share&t=1695

Ray tracing vs Rasterization

- Ray tracing
  - Pro: produces highly realistic lighting, reflections, shadows, global illumination effects. Widely used in film visual effects.
  - Con: computationally intensive, traditionally slower for real-time applications. As GPU technology advances, real-time ray tracing becomes feasible.
- Rasterization
  - Pro: Faster and more efficient for real-time rendering. Widely used in GUI, video game, and interactive graphics applications.
  - Con: Less accurate for complex lighting and reflection effects.

  We will focus on the rasterization approach in this course.
- Hybrid approach takes advantages of both: rasterization for primary visibility and basic shading, ray tracing for specific effects like reflections, shadows, and global illumination.

Unbalance of writing and reading

- CPU and GPU work together to generate (render, write) image into framebuffers.
- Video controller reads framebuffer and displays on screen at a fixed frequency, e.g. 60 Hz.
- Generally, it is slow to render images to framebuffer.
  - Affected by many factors: the complexity of a graphic image, CPU and GPU speed, algorithms, and system status.
- Graphics generating speed matters for real-time animation or virtual reality.
  - Real-time animation needs 24+ frames / ps.
  - Efforts are made to draw images fast .

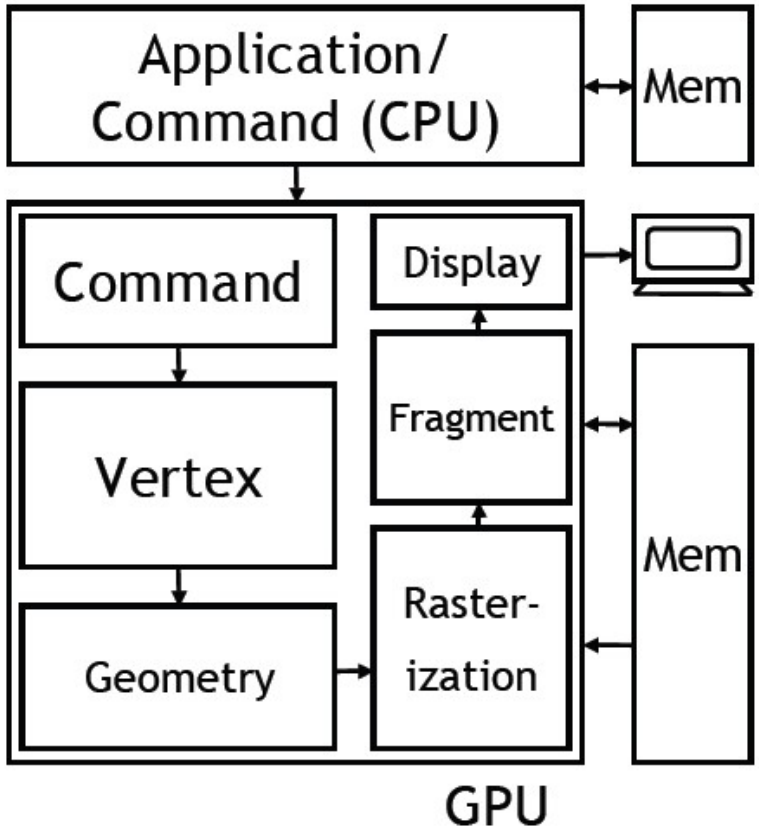
Video card

- A **video card**, also known as a **graphics accelerator card**, **display adapter**, or **graphics card**, is a PC hardware component whose function is to rasterize images in framebuffer and output images to a display.

- **Major components**
  - **Graphics Processing Unit (GPU):** a dedicated graphics microprocessor optimized for floating point calculations which are fundamental to 3D graphics rendering.
  - **Video memory**
    - DDR RAM, 128 MB to 2.0 GB, 400 MHz to 2.4 GHz.
    - Store program, data, and frame buffer

CPU and GPU

- The CPU and GPU communicate over the PCI bus. In CG computing
  - CPU does model computing. It sends graphic object model data (e.g., vertices of a triangle) to video memory and instructions to GPU.
  - GPU processes object data, rasterizes to generate pixel data, and stores the pixel data in framebuffer.
- GPU is a separate, self-contained computing device dedicated for graphics computing. GPU has its own processing units, and its own memory banks.



What GPU does?

- Dedicated to write pixel values into framebuffers of primary graphics primitives like dot, line segment, and triangle with transformations and lighting effects.

- The roles of CPU and GPU?
  - CPU does model computing
  - GPU does the rendering computing

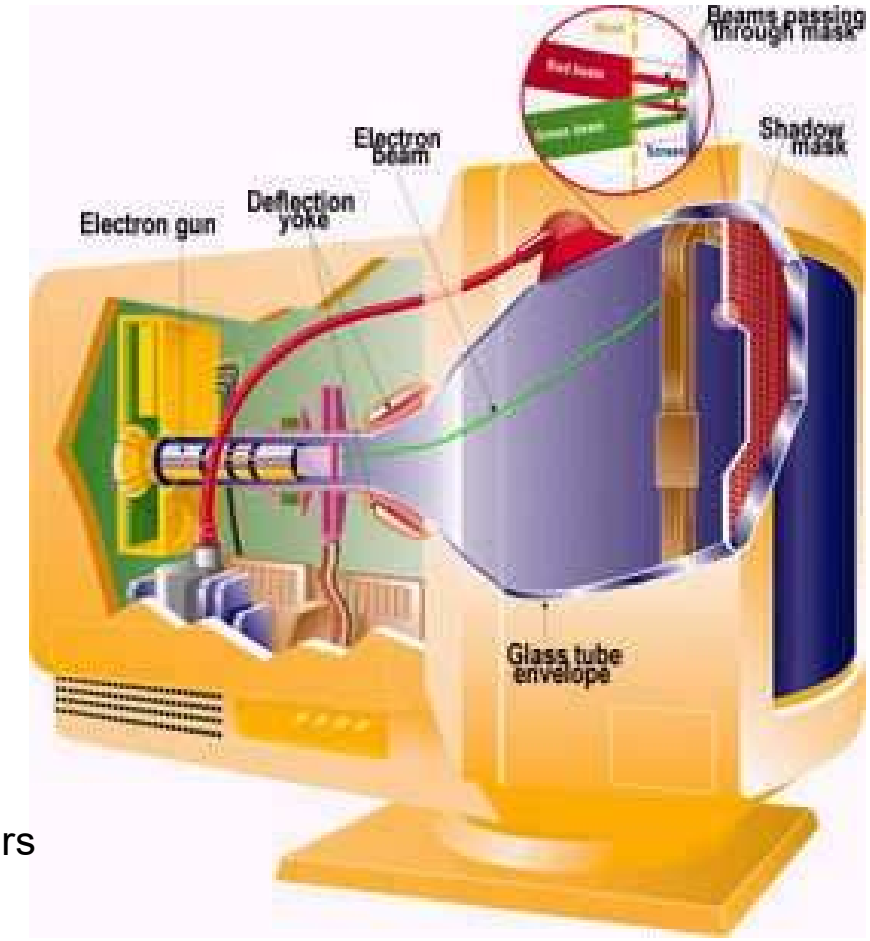
- Major vendors: NVIDIA, AMD, Intel, ARM.

GPU has more computing power

- **GPU has a parallel architecture**
  - GPUs have a large number of arithmetic units and computing is done in parallel.
  - Many computing problems map well to GPU-style computing, e.g. matrix computing.
- **Programmable GPU**
  - New features map to GPGPU, unified shaders.
  - Direct access to computing units through APIs
- **GPGPU has many other applications in accelerated computing:**
  - High Performance Computing
  - ML, AI, automatic cars.

Display Devices

- Cathode Ray Tubes (CRTs)
  - A common display device
  - Evacuated glass bottle
  - Extremely high voltage
  - Heating element (filament)
  - Electrons emit towards anode focusing cylinder
  - Vertical and horizontal deflection plates
  - Beam strikes phosphor coating on front of tube



- CRT Overview
  - CRT technology hasn't changed much in 50 years
  - Early television technology
    - high resolution
    - requires synchronization between video signal and electron beam vertical sync pulse
  - Early computer displays
    - avoided synchronization using 'vector' algorithm
    - flicker and refresh were problematic

Electron Gun

1. Contains a filament that, when heated, emits a stream of electrons

2. Electrons are focused with an electromagnet into a sharp beam and directed to a specific point of the face of the picture tube

3. Deflection of electronic beam can be controlled with either electronic or magnetic fields. E.g. Two pairs of magnetic deflection coils mounted on the outside of the CRT envelope, to control the deflection in horizontal (x) and vertical (y) direction respectively

4. The front surface of the picture tube is coated with small phosphor dots

5. Phosphor emits light for a short period of time when hit by electronic beam.

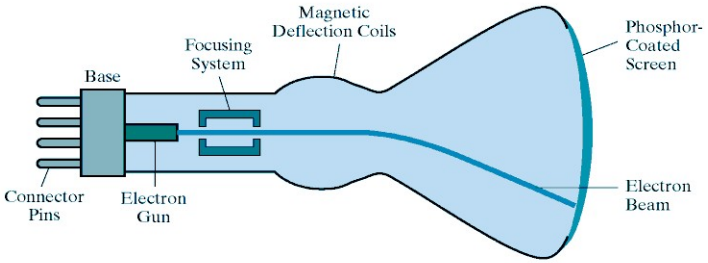


Figure 2-2
Basic design of a magnetic-deflection CRT.

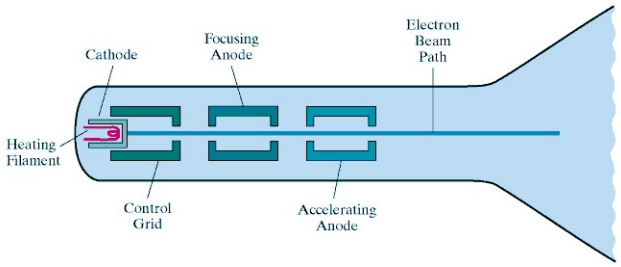


Figure 2-3
Operation of an electron gun with an accelerating anode.

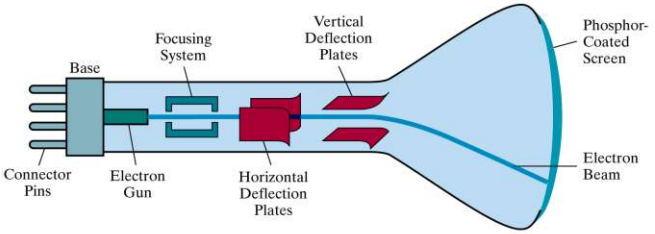
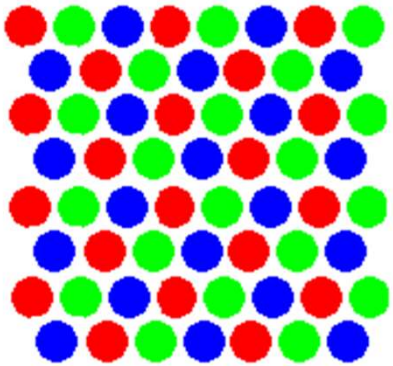


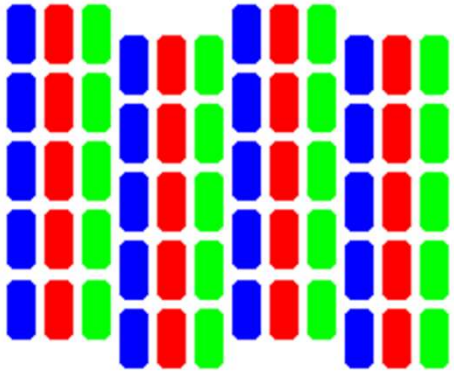
Figure 2-4
Electrostatic deflection of the electron beam in a CRT.

Color CRTs

- Color CRTs are *much* more complicated
  - Requires manufacturing very precise geometry
  - Uses a pattern of color phosphors on the screen:



Delta electron gun arrangement

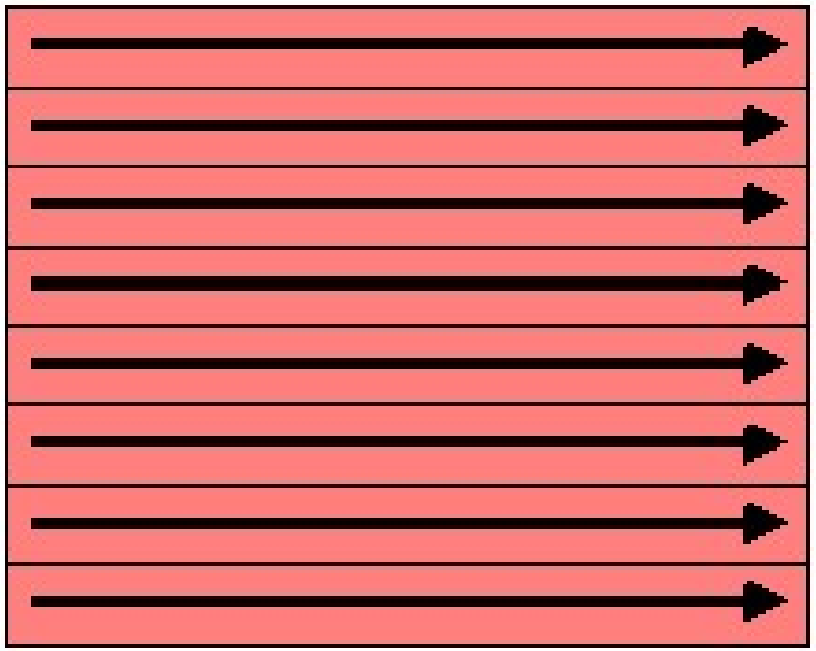


In-line electron gun arrangement

Interlaced Scanning and Progressive scanning

From Computer Desktop Encyclopedia
© 1998 The Computer Language Co. Inc.

Interlaced



Non-interlaced

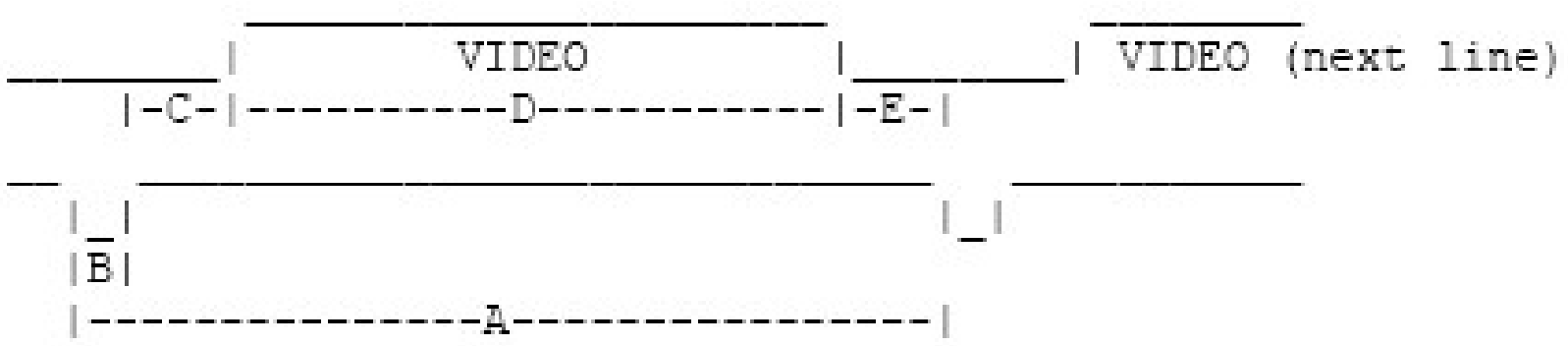
Video Graphics Arrays (VGA)

- Graphics hardware standard introduced by IBM
- Specifications:
  - 256 KB Video RAM
  - 16-color and 256-color modes
  - 262,144-value color palette (6 bits each for red, green, and blue)
  - 25.175 MHz or 28.322 MHz master clock
  - Maximum of 720 horizontal pixels
  - Maximum of 480 lines
  - Refresh rates at up to 70 Hz
  - Vertical Blanking interrupt

VGA timing information

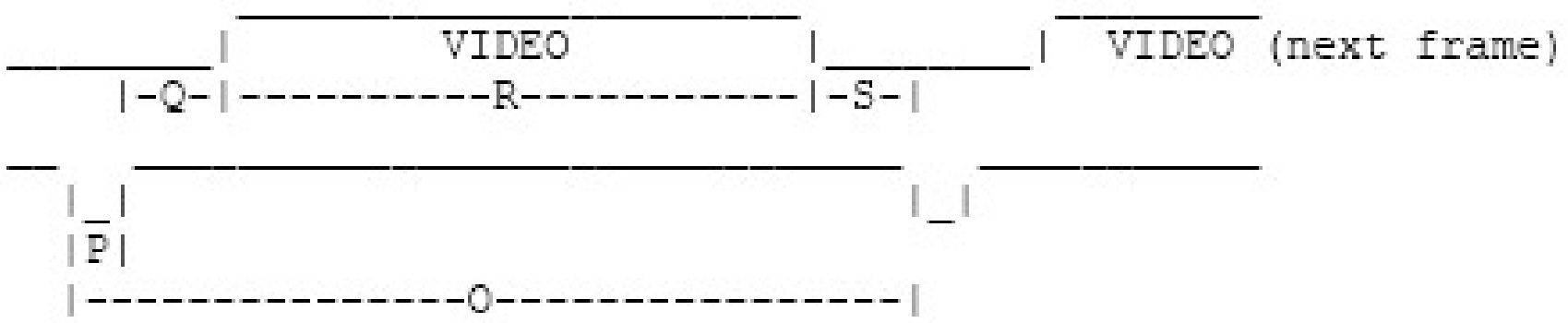
- Horizontal timing

Horizonal Dots	640	640	640	
Vertical Scan Lines	350	400	480	
Horiz. Sync Polarity	POS	NEG	NEG	
A (us)	31.77	31.77	31.77	Scanline time
B (us)	3.77	3.77	3.77	Sync pulse lenght
C (us)	1.89	1.89	1.89	Back porch
D (us)	25.17	25.17	25.17	Active video time
E (us)	0.94	0.94	0.94	Front porch



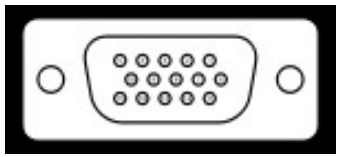
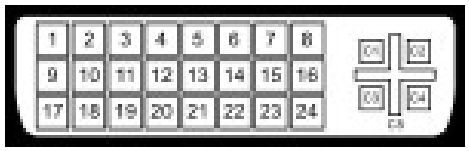
Vertical Timing

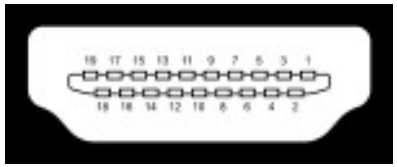
Horizonal Dots	640	640	640	
Vertical Scan Lines	350	400	480	
Vert. Sync Polarity	NEG	POS	NEG	
Vertical Frequency	70Hz	70Hz	60Hz	
O (ms)	14.27	14.27	16.68	Total frame time
P (ms)	0.06	0.06	0.06	Sync length
Q (ms)	1.88	1.08	1.02	Back porch
R (ms)	11.13	12.72	15.25	Active video time
S (ms)	1.2	0.41	0.35	Front porch

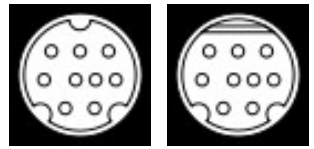


Other components

- Video BIOS (firmware)
- RAMDAC
- Output connectors

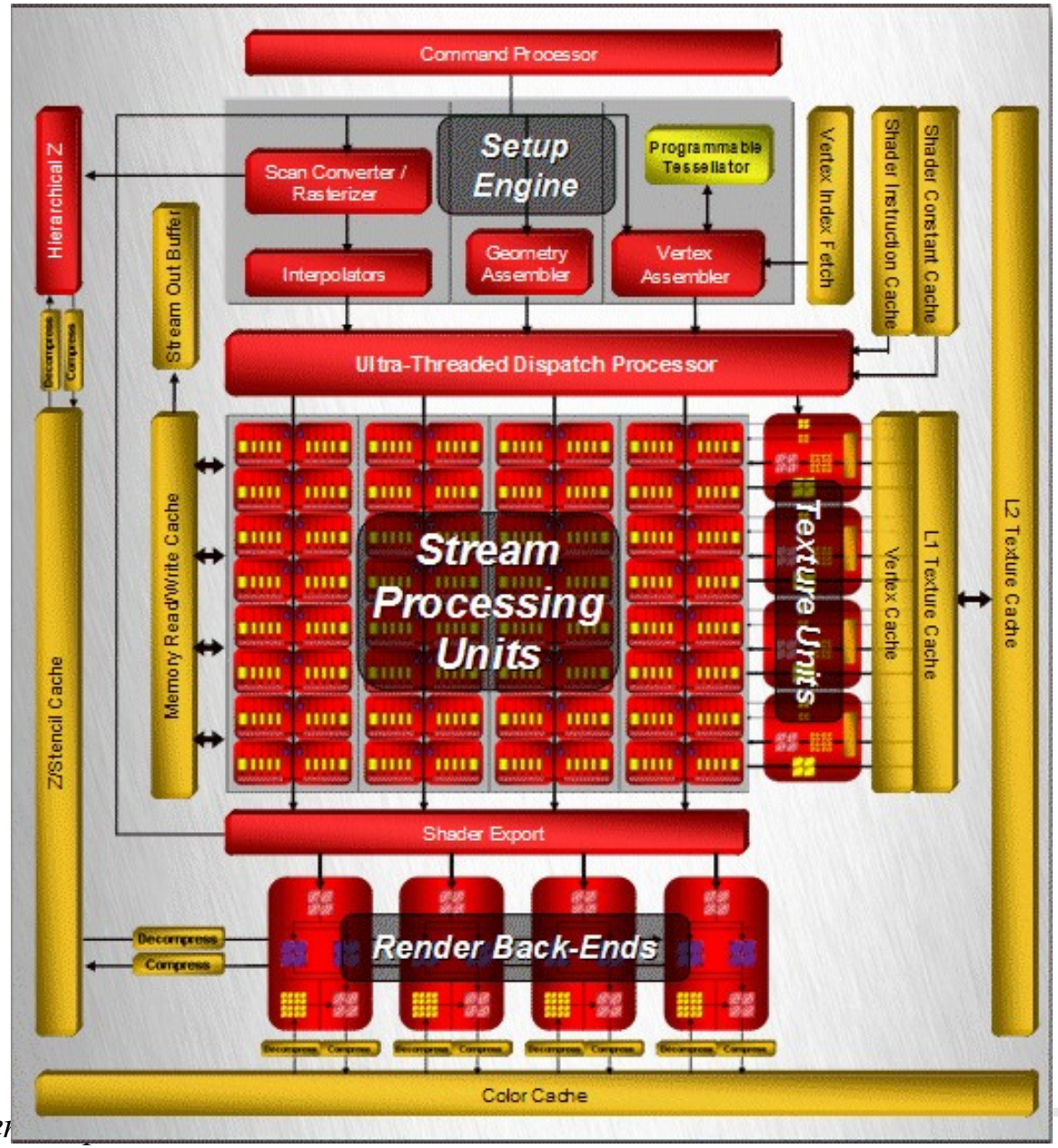





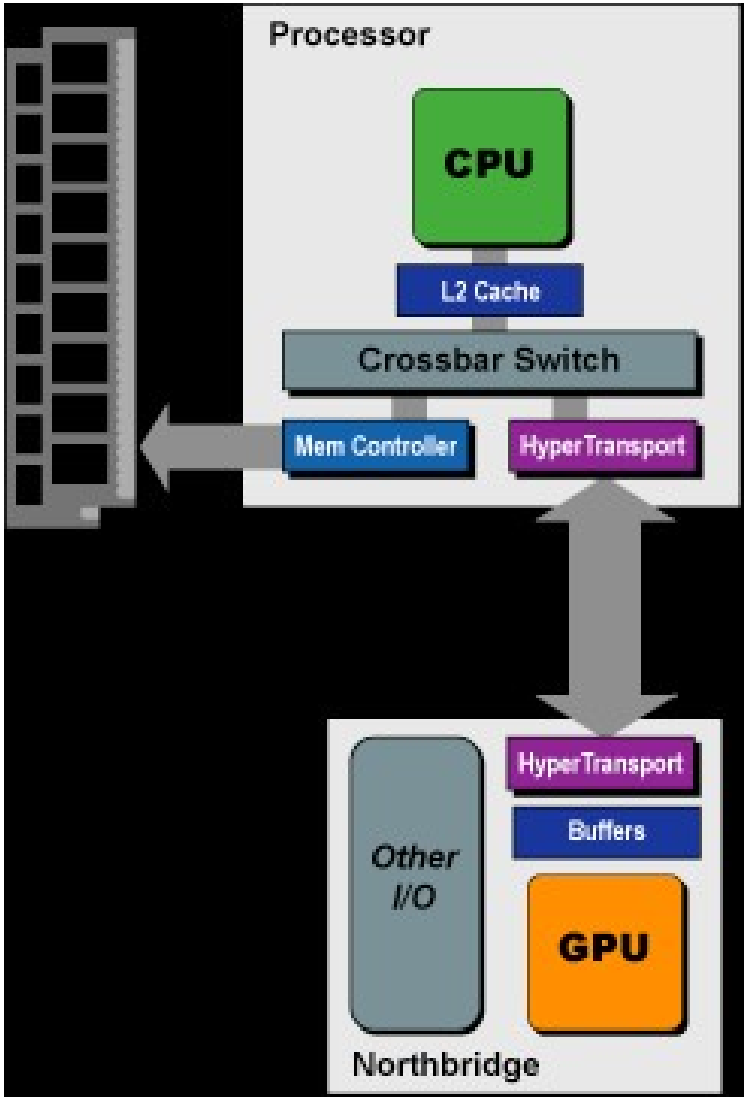



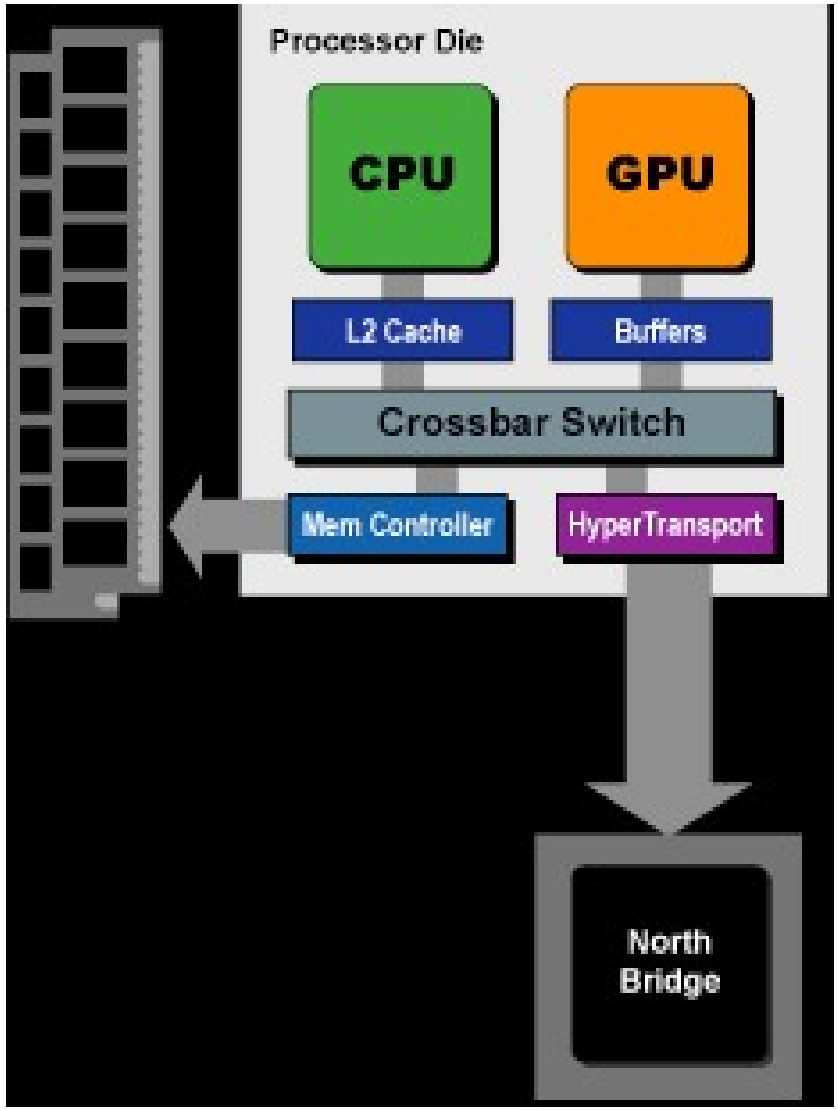
- Motherboard interface
  - PCI: 32 bit, 33 MHz. Replaced the previous buses from 1993. PCI allowed dynamic connectivity between devices, avoiding the jumpers manual adjustments
  - AGP: First used in 1997. Dedicated to graphics bus, 32 bits, 66 MHz.
  - PCI-Express: Point to point interface, released in 2004. In 2006 provided double data transfer rate of AGP, 32 bits, 133 MHz

A logical block diagram of the R600 GPU, 2007. Source: AMD.

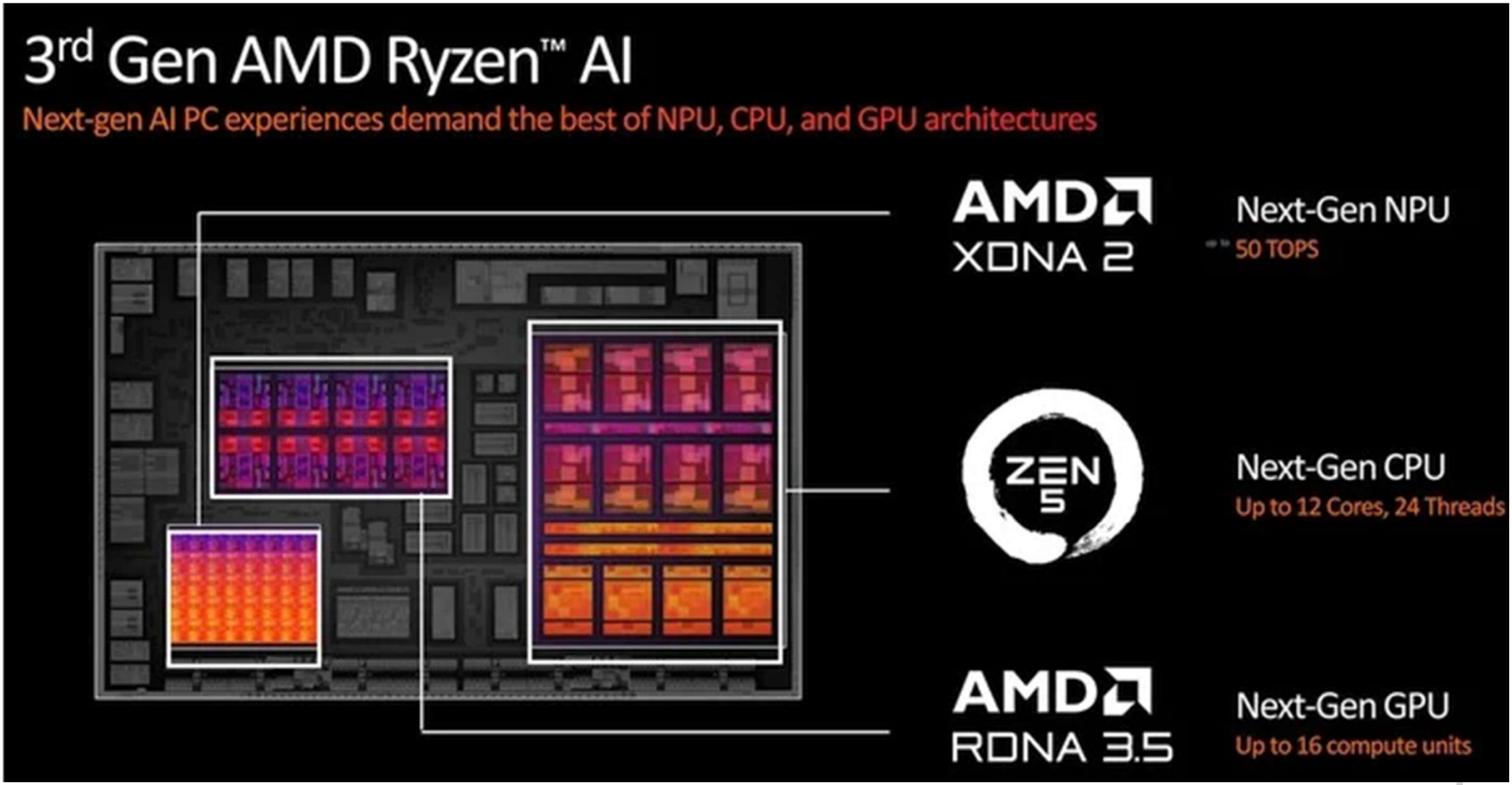


A closer look at AMD's CPU/GPU Fusion



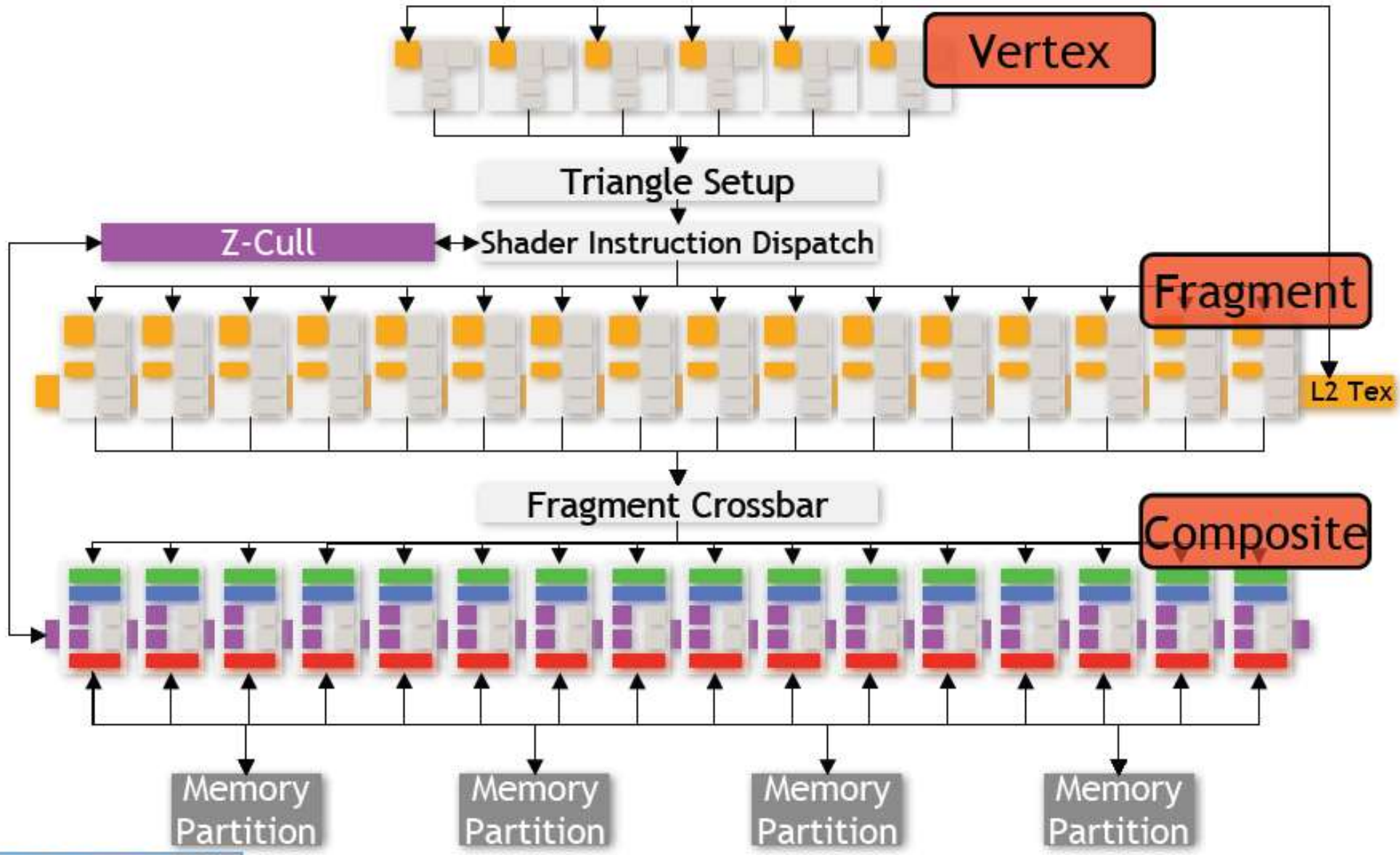


CPU,GPU and NPU in chip package



NVIDIA GeForce 6800 3D Pipeline

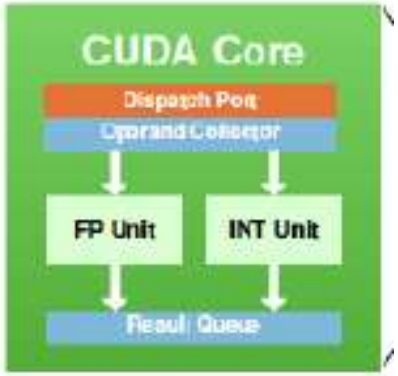
SIGGRAPH2007



GPGPU

Courtesy Nick Triantos, NVIDIA

NVIDIA GPGPU
Kepler GK110 (May 2012)



- 28nm chip, 7.1B transistors
- 15 Streaming Multiprocessors
  - —192 CUDA cores
    - fully pipelined FP and INT units
    - IEEE 754-2008; fused multiply add
  - —four warp schedulers
    - 32-thread groups (warps)
    - 4 warps issue and execute concurrently
    - 2 inst/warp/cycle
  - —64 DP FP units
  - —32 SFU
  - —32 LD/ST units
- 6 64-bit memory controllers

	KEPLER GK110
Compute Capability	3.5
Threads / Warp	32
Max Warps / Multiprocessor	64
Max Threads / Multiprocessor	2048
Max Thread Blocks / Multiprocessor	16
32-bit Registers / Multiprocessor	65536
Max Registers / Thread	255
Max Threads / Thread Block	1024
Shared Memory Size Configurations (bytes)	16K 32K 48K
Max X Grid Dimension	2^32-1
Hyper-Q	Yes
Dynamic Parallelism	Yes

http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf

NVIDIA GPU architecture evolution

NVIDIA GPU architectures, capabilities, and GPU product examples

GPU Architecture	Compute Capability	Example GPUs
Kepler	3.0 – 3.7	GTX 780, K80
Maxwell	5.0 – 5.2	GTX 980, Titan X
Pascal	6.0 – 6.2	GTX 1080, P100
Volta	7.0	V100
Turing	7.5	RTX 2080, T4
Ampere	8.0 – 8.6	A100, RTX 30XX
Hopper	9.0	H100
Blackwell	9.0+	B100, GB200 (TBD)

GPU product example of Blackwell architecture

GPU Variant	CUDA Cores	Tensor Cores
B200 Accelerator	~16,896	~528
GB200 Superchip	~18,432	~576

GPUs with more types of cores for higher capability

Types of cores in NVIDIA GPUs, usage, and computing precision support

Core Type	Primary Use	Precision Support
CUDA Cores	General compute	FP32, INT, etc.
Tensor Cores	Matrix math / AI ops	FP16, BF16, FP8, INT8
RT Cores	Ray tracing	-
Transformer Engine	Optimized transformer training	Dynamic FP8

Example:
RTX 5090 (GB202-300 Die, 170 SMs enabled)
Streaming Multiprocessors (SMs): 170
CUDA Cores: 21,760 (128 per SM)
Tensor Cores (5th Gen): 680 (4 per SM)
RT Cores (4th Gen): 170 (1 per SM)