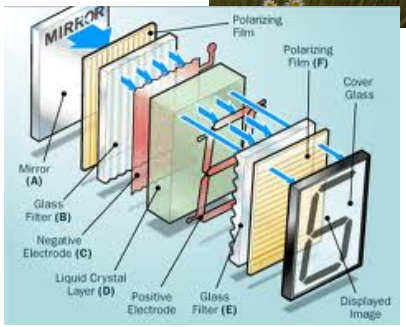
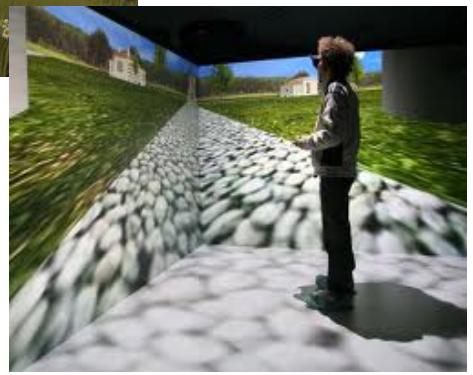
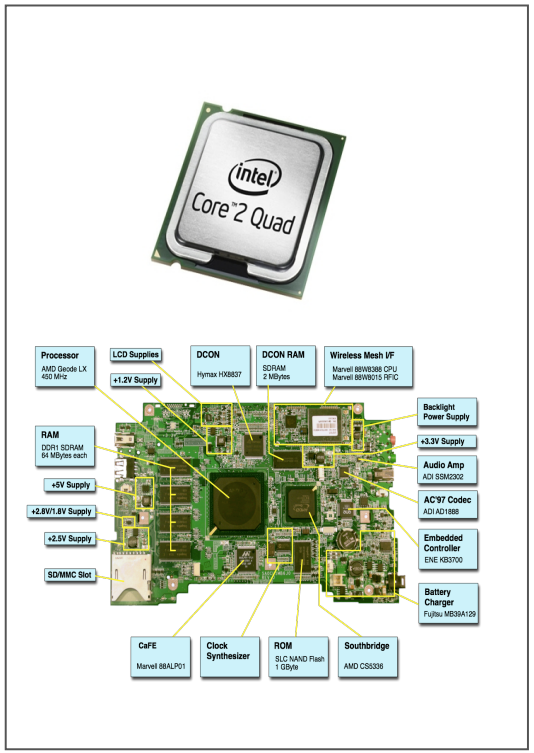
Computer Graphics Hardware
An Overview





Graphics System

Input devices



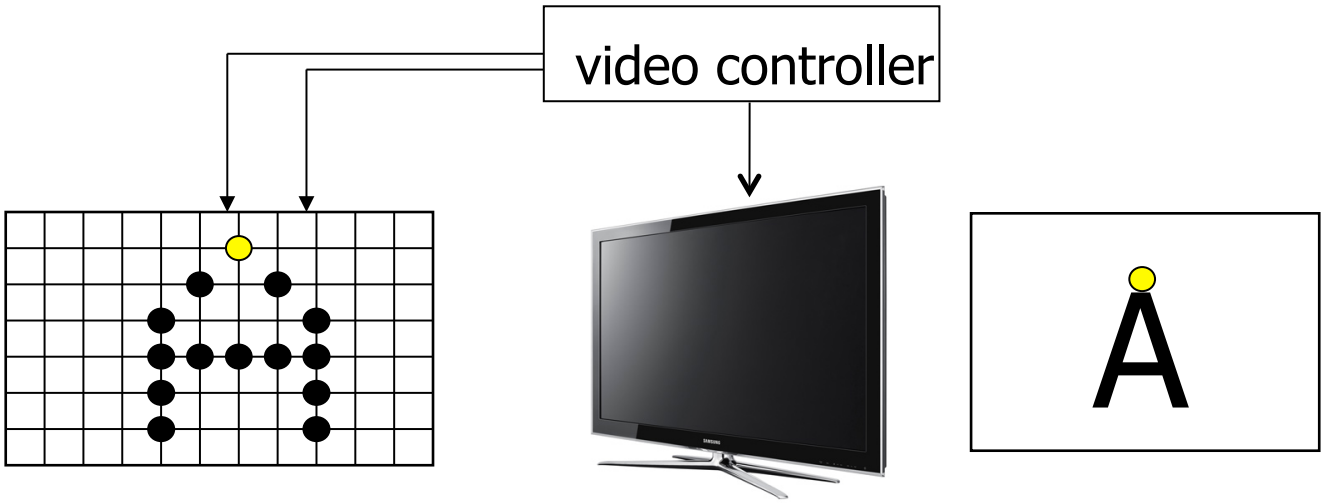
CPU/Memory

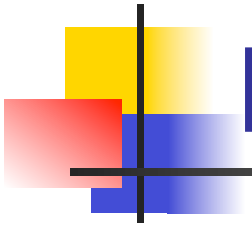
GPU

Monitor

Raster Graphics System

- Raster: An array of picture elements
- Based on raster-scan TV technology
- The screen (and a picture) consists of discrete pixels, and each pixel has a small display area





Frame Buffer

- Frame buffer: the memory to hold the pixel properties (color, alpha, depth, stencil mask, etc)
- Properties of a frame buffer that affect the graphics performance:
  - Size: screen resolution
  - Depth: color level

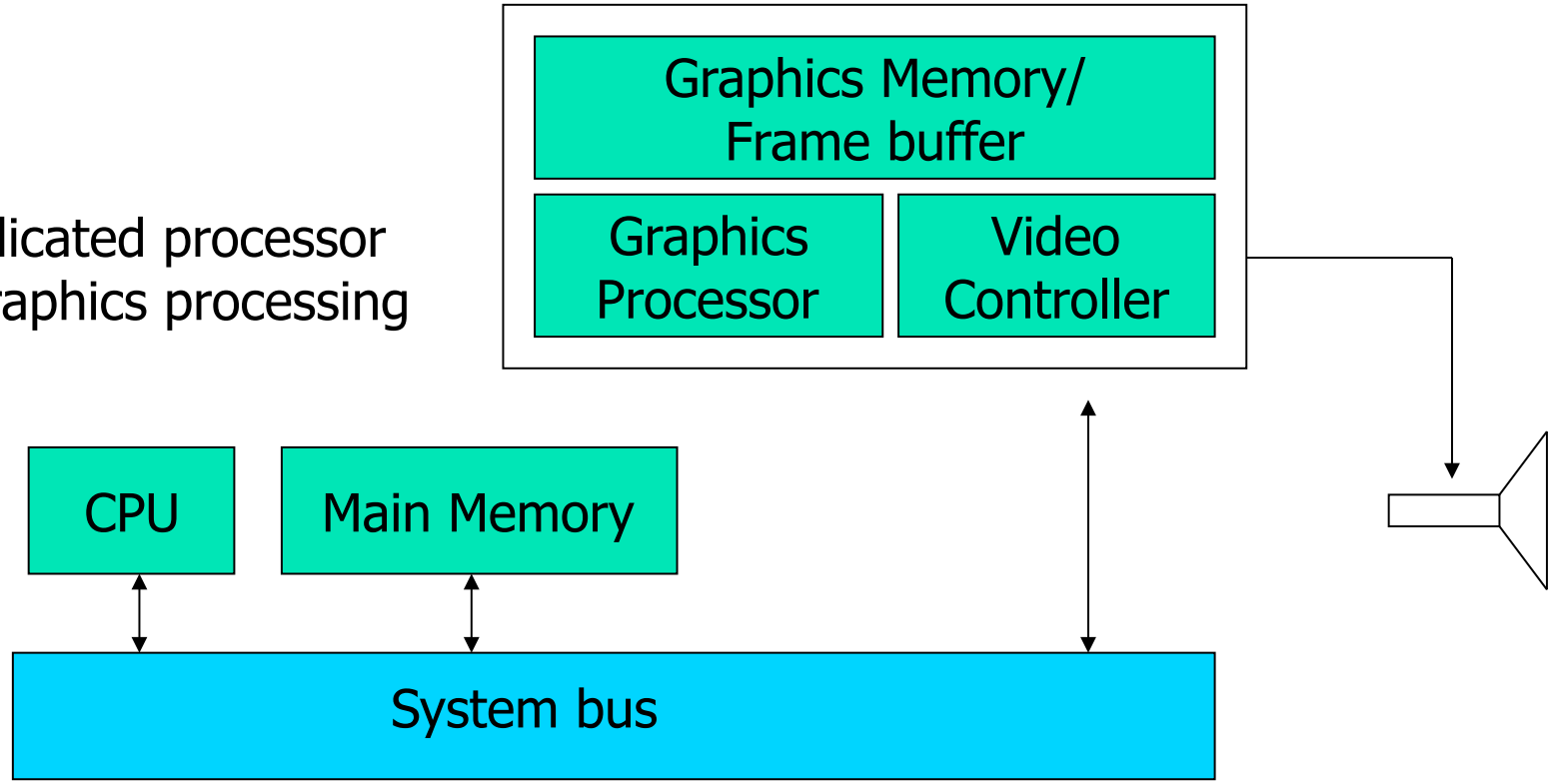
    1 bit/pixel: black and white

    8 bits/pixel: 256 levels of gray or color pallet index

    24 bits/pixel: 16 million colors
  - Speed: refresh speed

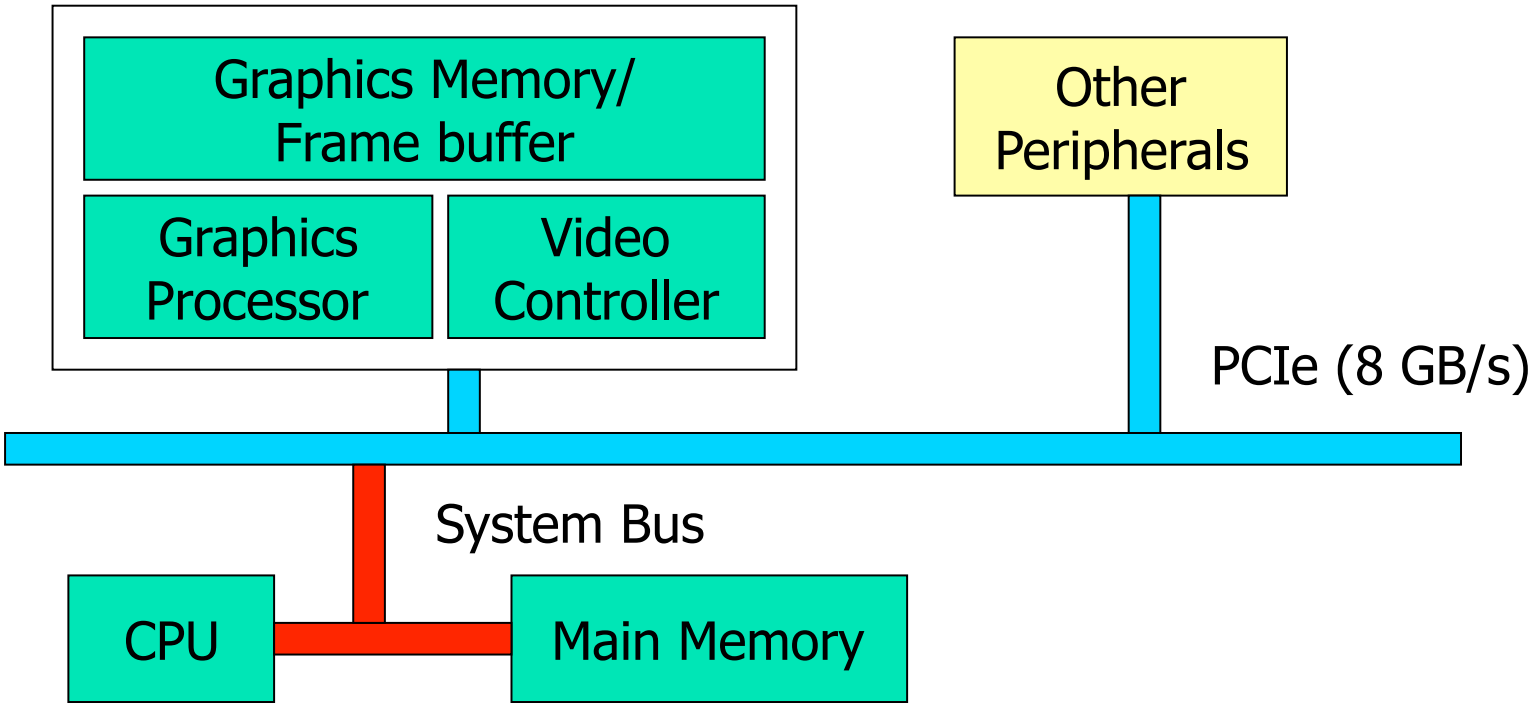
Graphics Accelerator

A dedicated processor
for graphics processing



Graphics Bus Interface

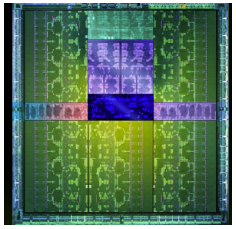
PCI based technology



Graphics Accelerators

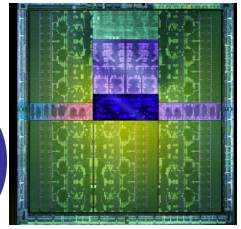
ATI TECHNOLOGIES INC.

What do GPUs do?



- Graphics processing units (GPUs) are massively parallel processors
  - Process geometry/pixels and produce images to be displayed on the screen
  - Can also be used to perform general purpose computation (via CUDA/OpenGL)
- Evolved from simple video scan controllers, to special purpose processors that implement a simple pipeline with fixed graphics functionality, to complex many-core architectures that contain several deep parallel pipelines
  - Example: nvidia's Kepler GK110 contains 15x192 cores and 7.1 billions transistors
  - A graphics card can easily have more than 2GB of video memory

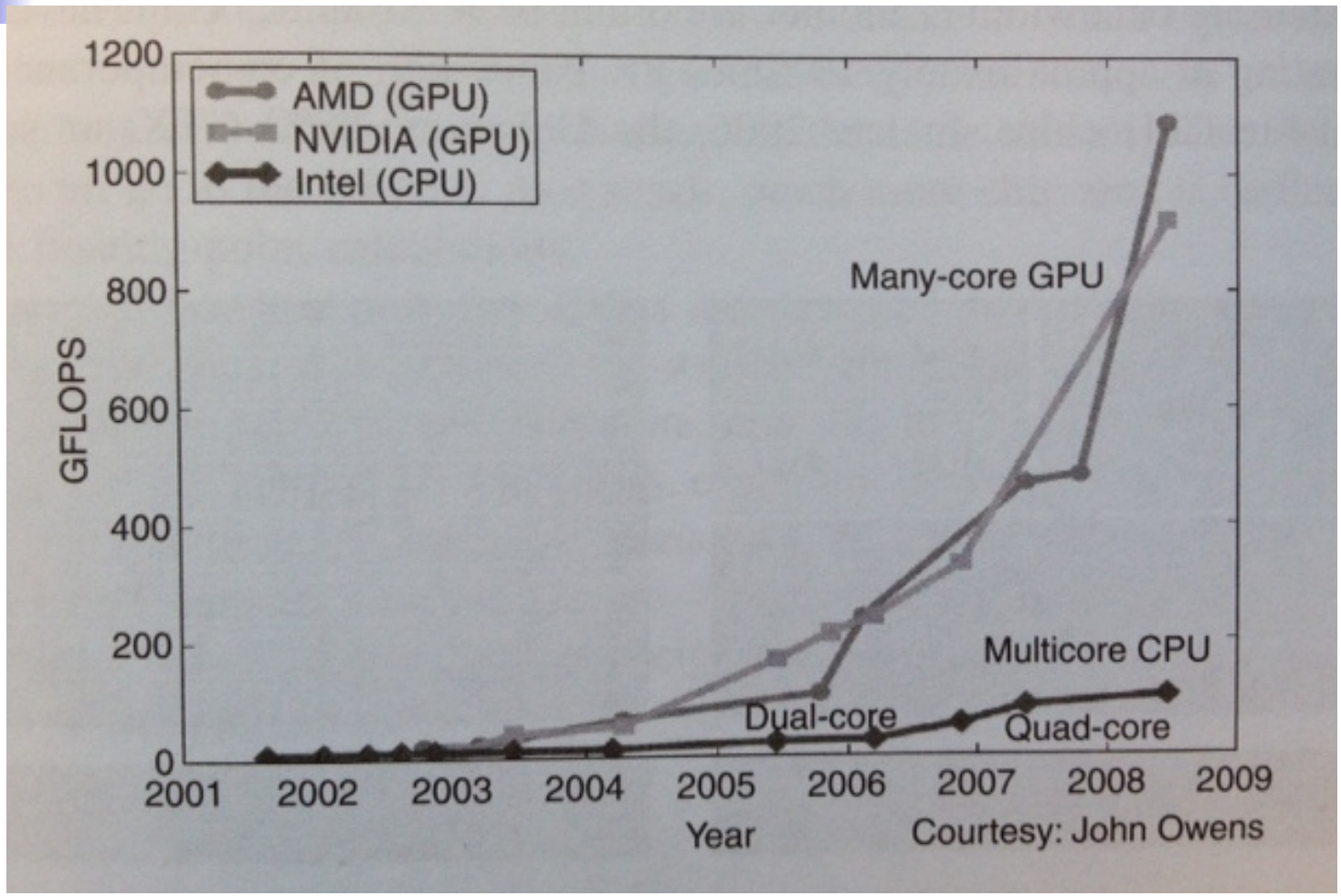
nVidia Kepler GK110 (2012)



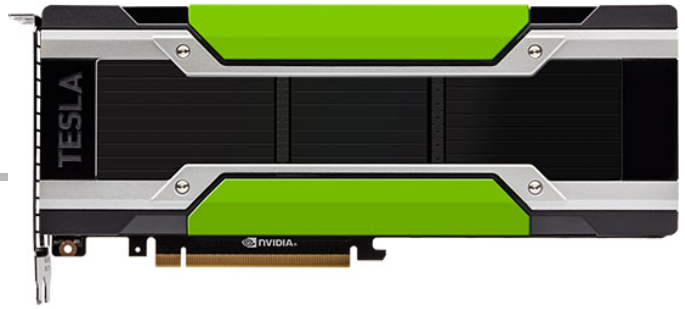
Architecture

- 7.1B Transistors
- 15 SMX units
- > 1 TFLOP FP64
- 1.5 MB L2 Cache
- 384-bit GDDR5
- PCI Express Gen3

CPU/GPU Performance Gap



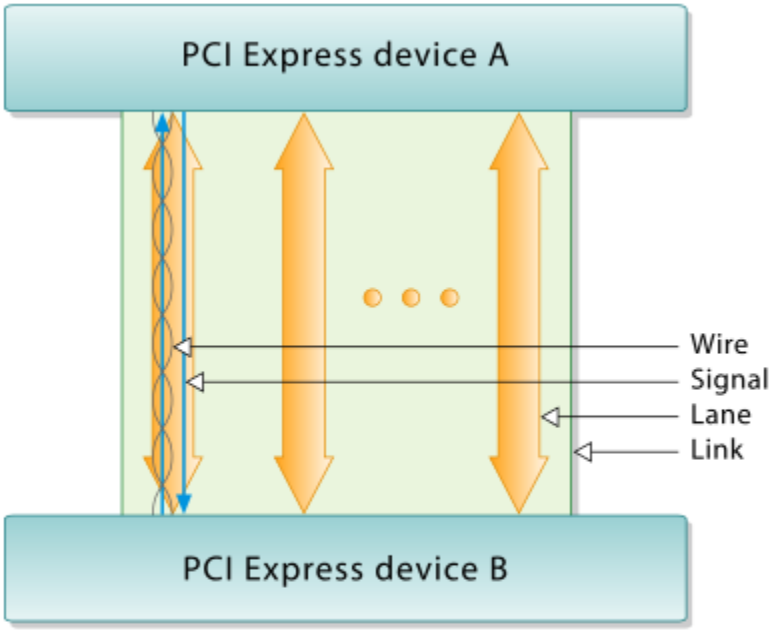
Latest GPU



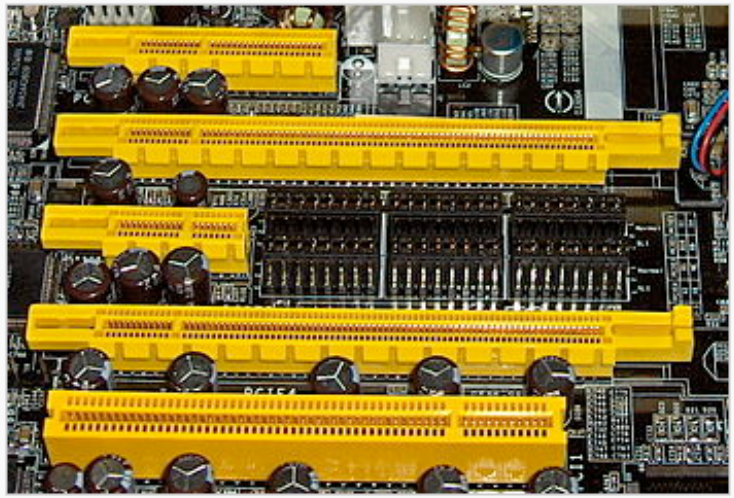
NVIDIA P100 Tesla

PERFORMANCE SPECIFICATION FOR NVIDIA TESLA P100 ACCELERATORS

	P100 for PCIe-Based Servers	P100 for NVLink-Optimized Servers
Double-Precision Performance	4.7 TeraFLOPS	5.3 TeraFLOPS
Single-Precision Performance	9.3 TeraFLOPS	10.6 TeraFLOPS
Half-Precision Performance	18.7 TeraFLOPS	21.2 TeraFLOPS
NVIDIA NVLink™ Interconnect Bandwidth	-	160 GB/s
PCIe x16 Interconnect Bandwidth	32 GB/s	32 GB/s
CoWoS HBM2 Stacked Memory Capacity	16 GB or 12 GB	16 GB
CoWoS HBM2 Stacked Memory Bandwidth	720 GB/s or 540 GB/s	720 GB/s
Enhanced Programmability with Page Migration Engine	✓	✓
ECC Protection for Reliability	✓	✓
Server-Optimized for Data Center Deployment	✓	✓



A PCI Express link between two devices consists of one or more lanes, which are dual simplex channels using two differential signaling pairs.[4]:3



Various slots on a computer motherboard, from top to bottom:

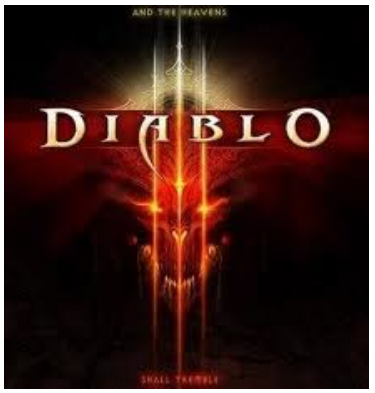
- PCI Express ×4
- PCI Express ×16
- PCI Express ×1
- PCI Express ×16
- Legacy PCI (32-bit, 5 V)

PCI Express link performance[27][29]

PCI Express version	Line code	Transfer rate[a]	Throughput[a]			
			×1	×4	×8	×16
1.0	8b/10b	2.5 GT/s	250 MB/s	1 GB/s	2 GB/s	4 GB/s
2.0	8b/10b	5 GT/s	500 MB/s	2 GB/s	4 GB/s	8 GB/s
3.0	128b/130b	8 GT/s	984.6 MB/s	3.938 GB/s	7.877 GB/s	15.754 GB/s
4.0 (expected in 2017)	*128b/130b*	*16 GT/s*	*1.969 GB/s*	*7.877 GB/s*	*15.754 GB/s*	*31.508 GB/s*
5.0 (far future)[28]	*128b/130b*	*32 / 25 GT/s*	*3.9 / 3.08 GB/s*	*15.8 / 12.3 GB/s*	*31.5 / 24.6 GB/s*	*63.0 / 49.2 GB/s*

Why are GPU's so fast?

- Entertainment Industry has driven the economy of these chips?
  - Males age 15-35 buy $10B in video games / year
- Moore's Law ++
- Simplified design (stream processing)
- Single-chip designs.



Modern GPU has more ALU's

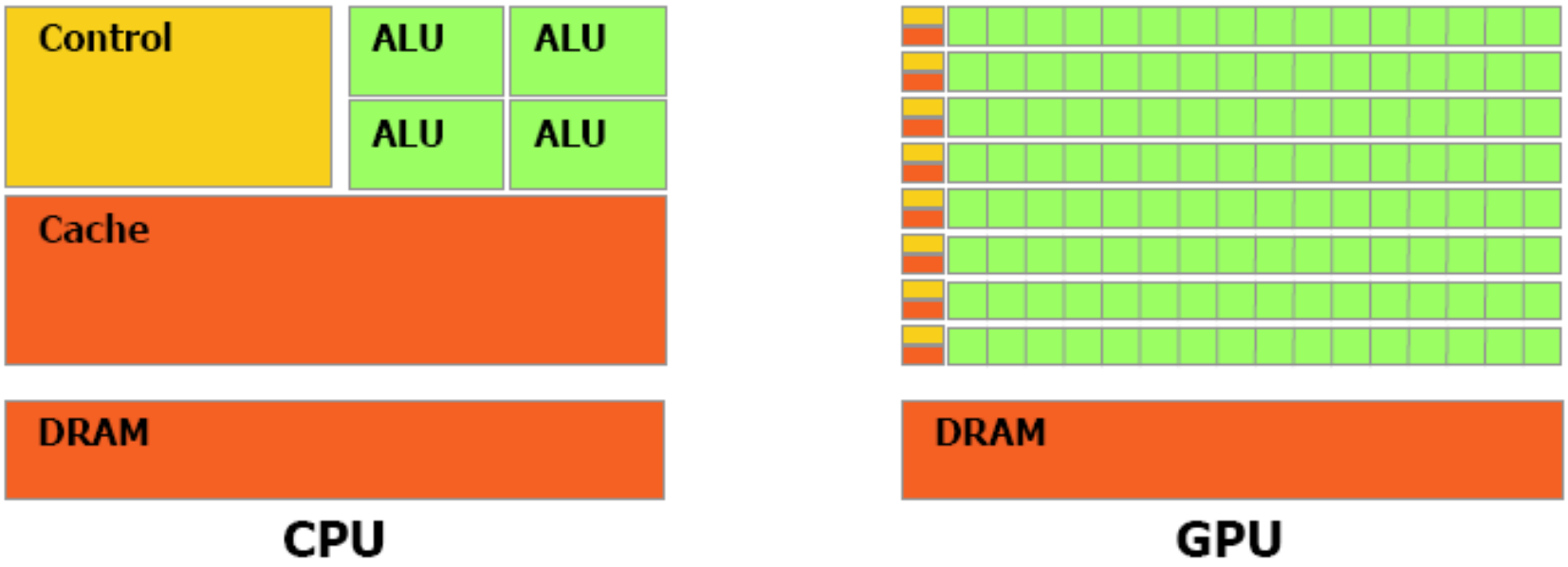
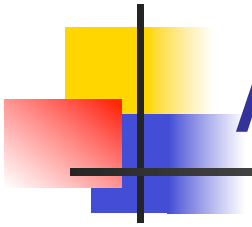
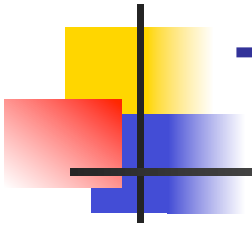


Figure 1-2. The GPU Devotes More Transistors to Data Processing



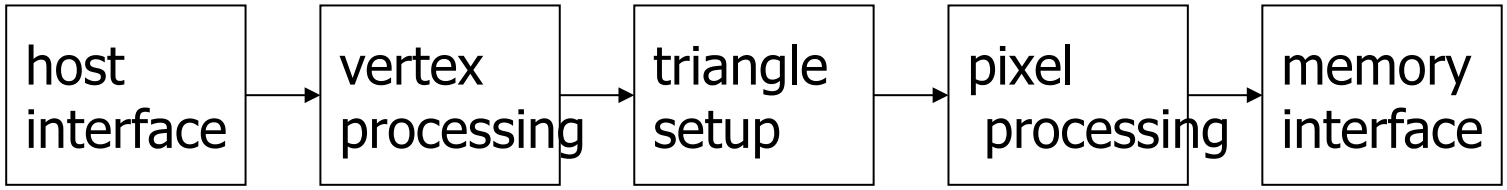
A Specialized Processor

- Very Efficient For
  - Fast Parallel Floating Point Processing
  - Single Instruction Multiple Data Operations
  - High Computation per Memory Access
- Not As Efficient For
  - Double Precision
  - Logical Operations on Integer Data
  - Branching-Intensive Operations
  - Random Access, Memory-Intensive Operations



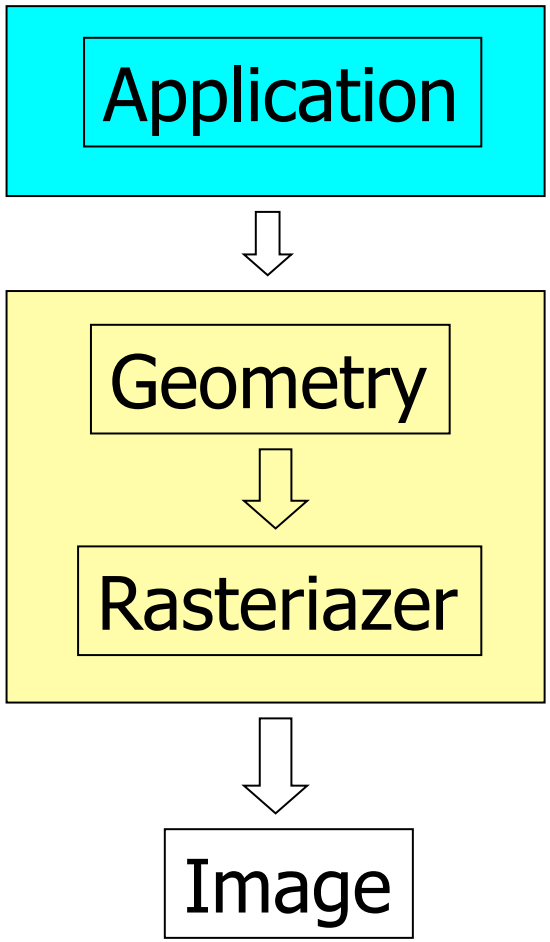
The Rendering Pipeline

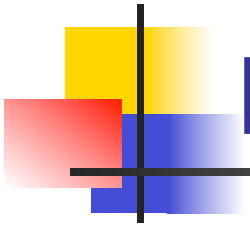
- The process to generate two-dimensional images from given virtual cameras and 3D objects
- The pipeline stages implement various core graphics rendering algorithms
- Why should you know the pipeline?
  - Necessary for programming GPUs
  - Understand various graphics algorithms
  - Analyze performance bottleneck



The Rendering Pipeline

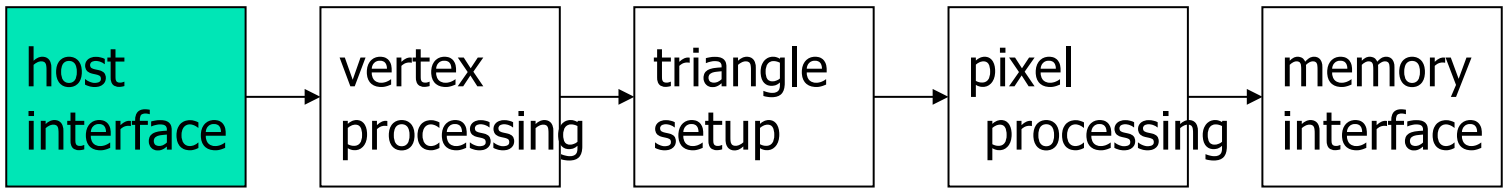
- The basic construction – three conceptual stages
- Each stage is a pipeline and runs in parallel
- Graphics performance is determined by the slowest stage
- Modern graphics systems:
  - Software
  - hardware





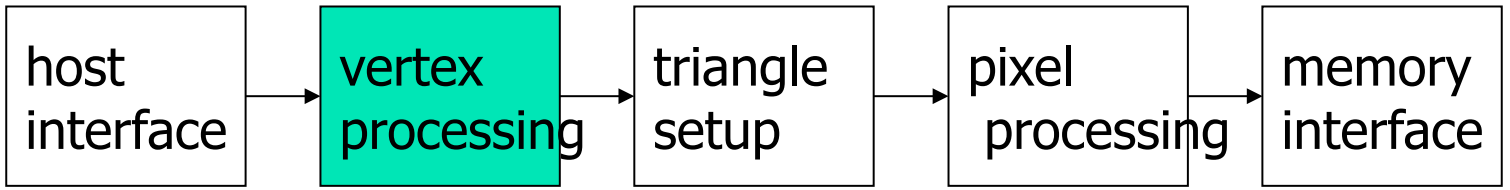
Host Interface

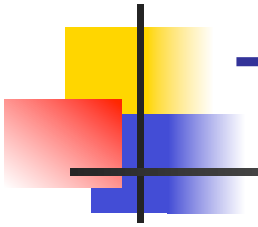
- The host interface is the communication bridge between the CPU and the GPU
- It receives commands from the CPU and also pulls geometry information from system memory
- It outputs a *stream* of vertices in object space with all their associated information (normals, texture coordinates, per vertex color etc)



Vertex Processing

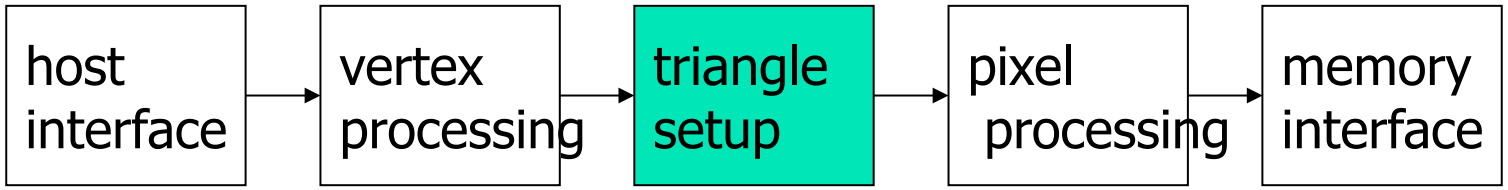
- The vertex processing stage receives vertices from the host interface in object space and outputs them in screen space
- This may be a simple linear transformation, or a complex operation involving morphing effects
- Normals, texcoords etc are also transformed
- No new vertices are created in this stage, and no vertices are discarded (input/output has 1:1 mapping)





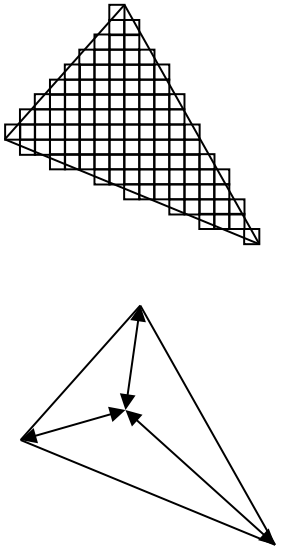
Triangle setup

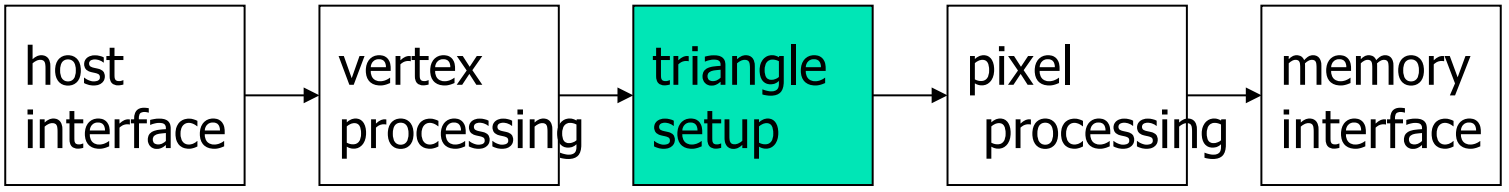
- In this stage geometry information becomes raster information (screen space geometry is the input, pixels are the output)
- Prior to rasterization, triangles that are backfacing or are located outside the viewing frustrum are rejected
- Some GPUs also do some hidden surface removal at this stage

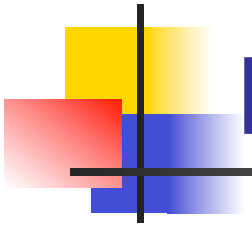


Triangle Setup (cont)

- A fragment is generated if and only if its center is inside the triangle
- Every fragment generated has its attributes computed to be the perspective correct interpolation of the three vertices that make up the triangle

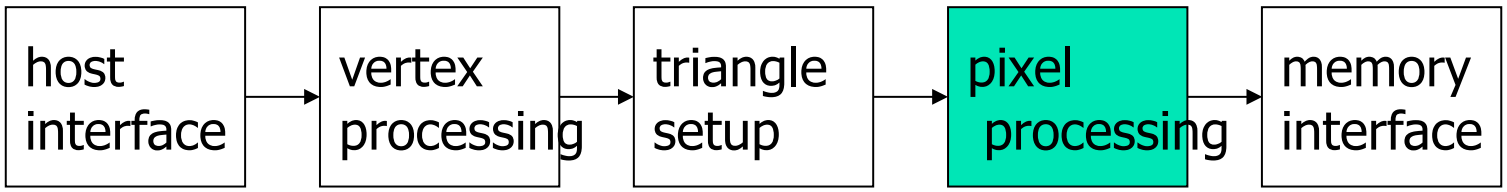






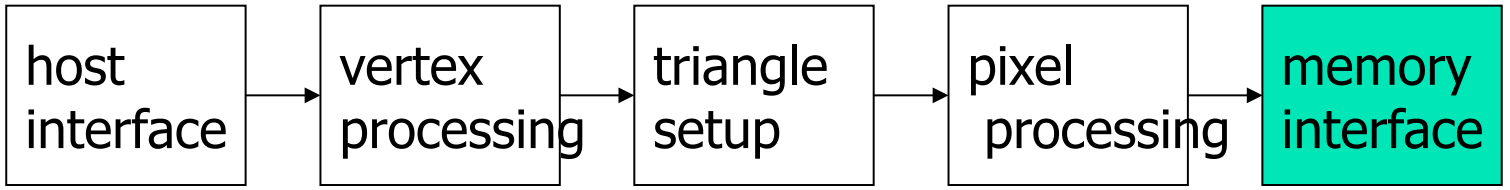
Fragment Processing

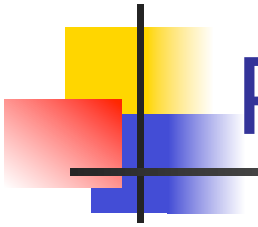
- Each fragment provided by triangle setup is fed into fragment processing as a set of attributes (position, normal, texcoord etc), which are used to compute the final color for this pixel
- The computations taking place here include texture mapping and math operations
- Typically the bottleneck in modern applications



Memory Interface

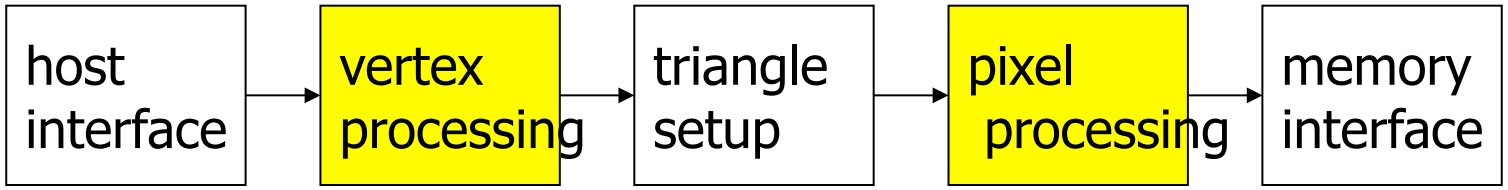
- Fragment colors provided by the previous stage are written to the framebuffer
- Used to be the biggest bottleneck before fragment processing took over
- Before the final write occurs, some fragments are rejected by the zbuffer, stencil and alpha tests
- On modern GPUs, z and color are compressed to reduce framebuffer bandwidth (but not size)





Programmability in the GPU

- Vertex and fragment processing, and now triangle set-up, are programmable
- The programmer can write programs that are executed for every vertex as well as for every fragment
- This allows fully customizable geometry and shading effects that go well beyond the generic look and feel of older 3D applications



The Graphics Pipeline

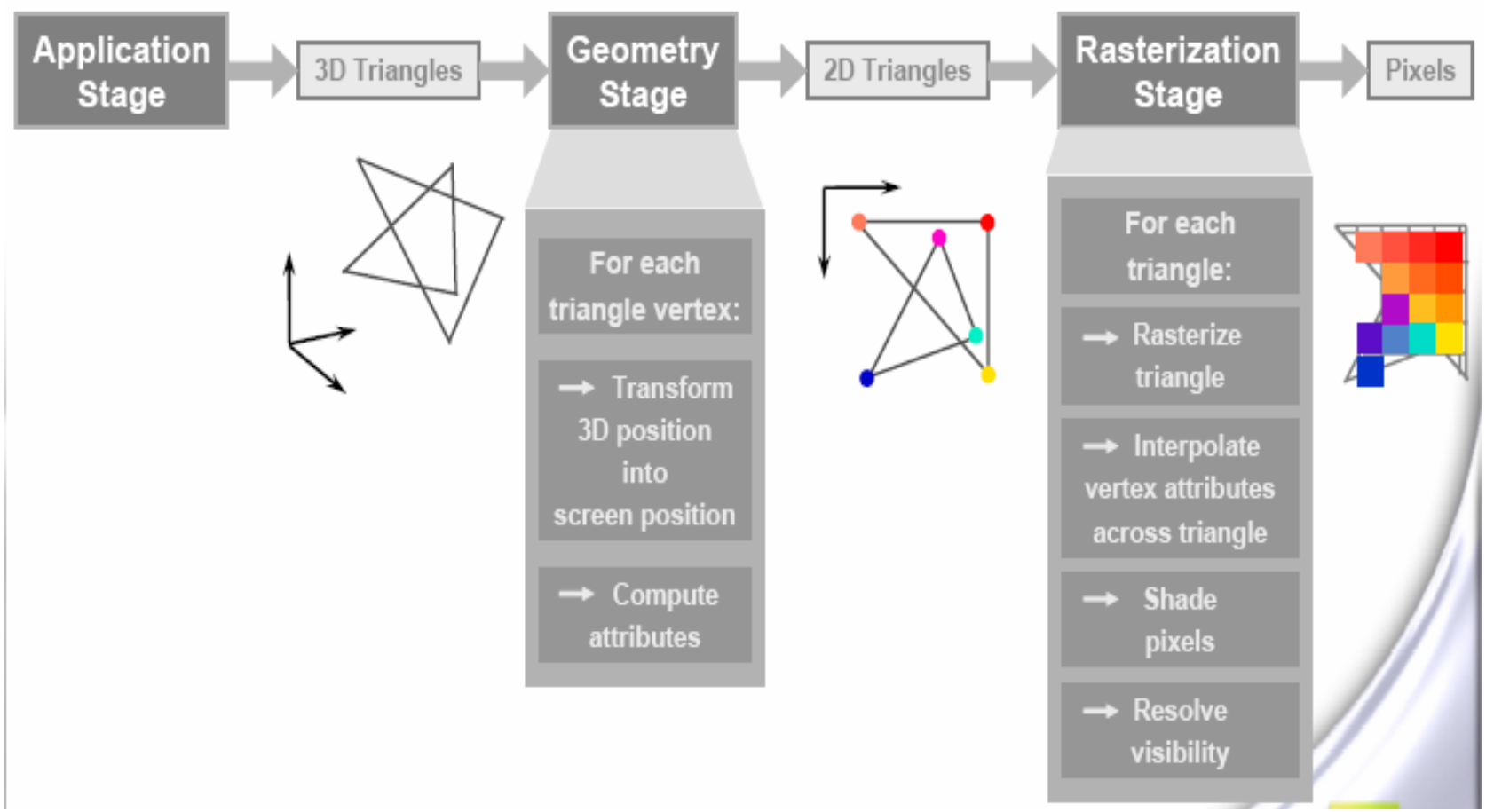
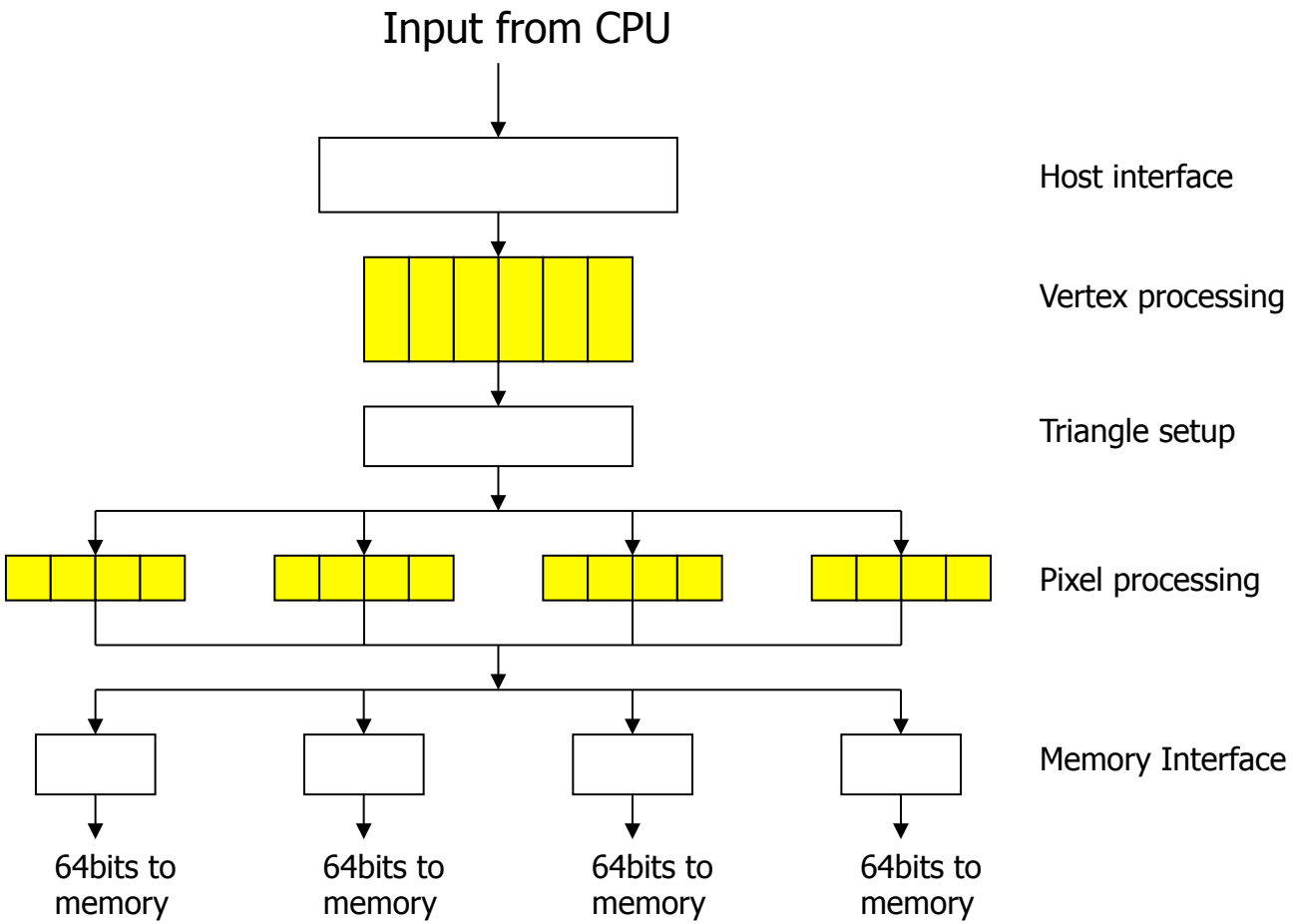
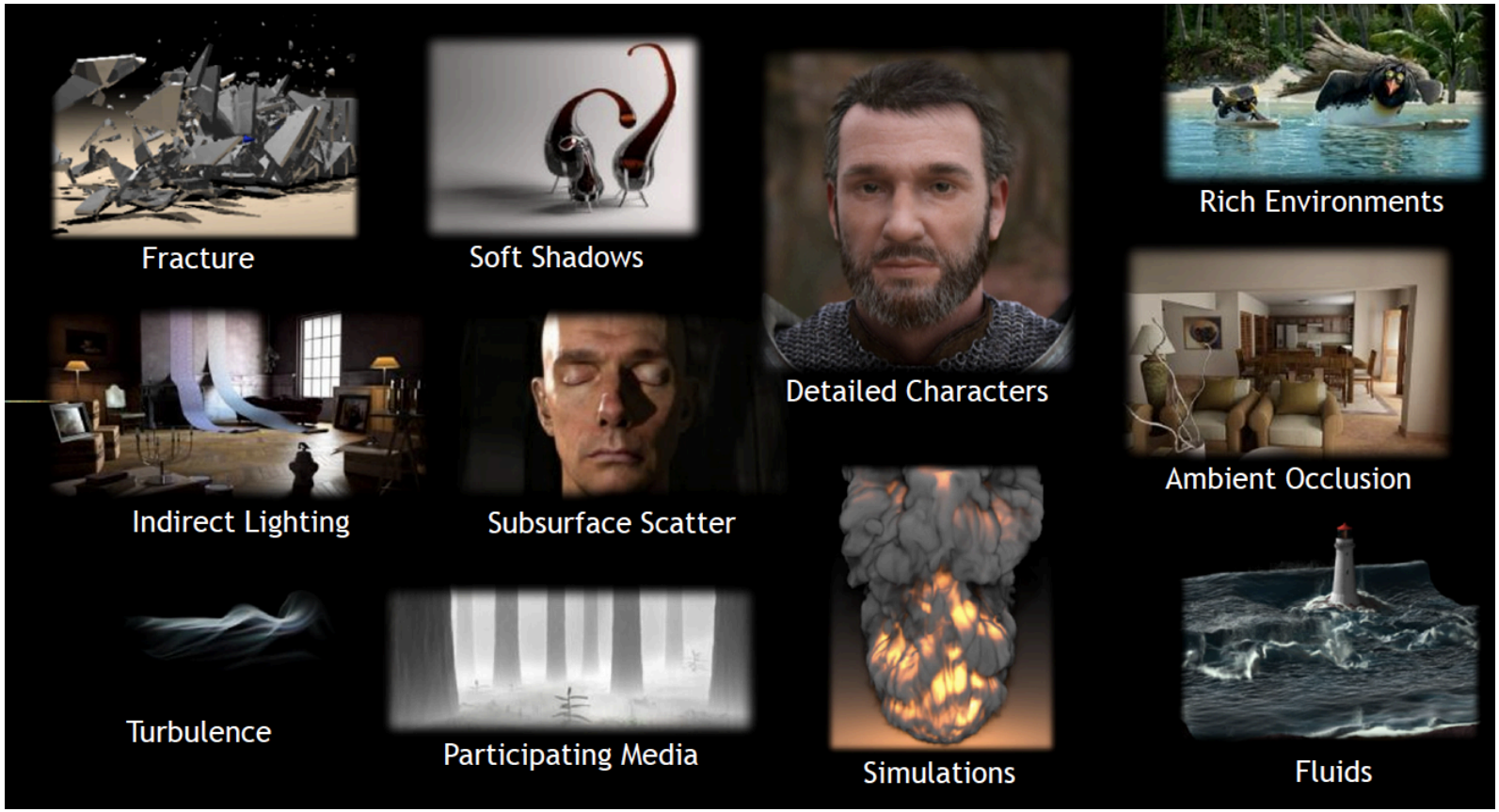


Diagram of a modern GPU



The Quest for Realism



(courtesy: nvidia)