3D Graphics Pipeline & GPU

by 麒麟子@Cocos in 2021

Contents

Part 0. Five Questions

Part 1. 3D Graphics Pipeline

Part 2. A Simple Introduction to GPU

Part 3. Anwsers to the Questions

Five Questions

Q1.How is a 3d scene rendered to a 2d screen?

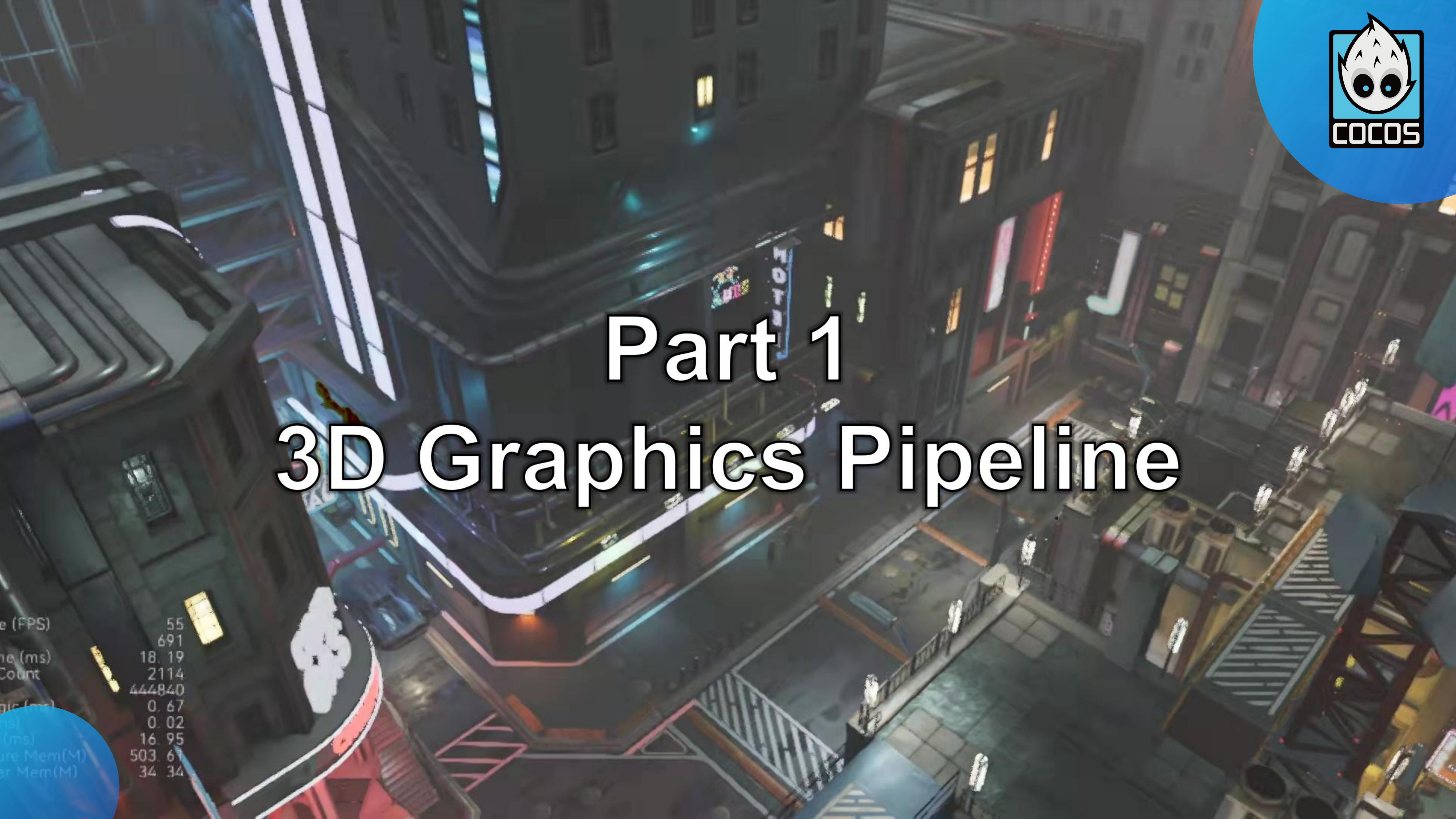
Q2.What are the key differences between GPU and CPU?

Q3.What are bandwidth and fill rate?

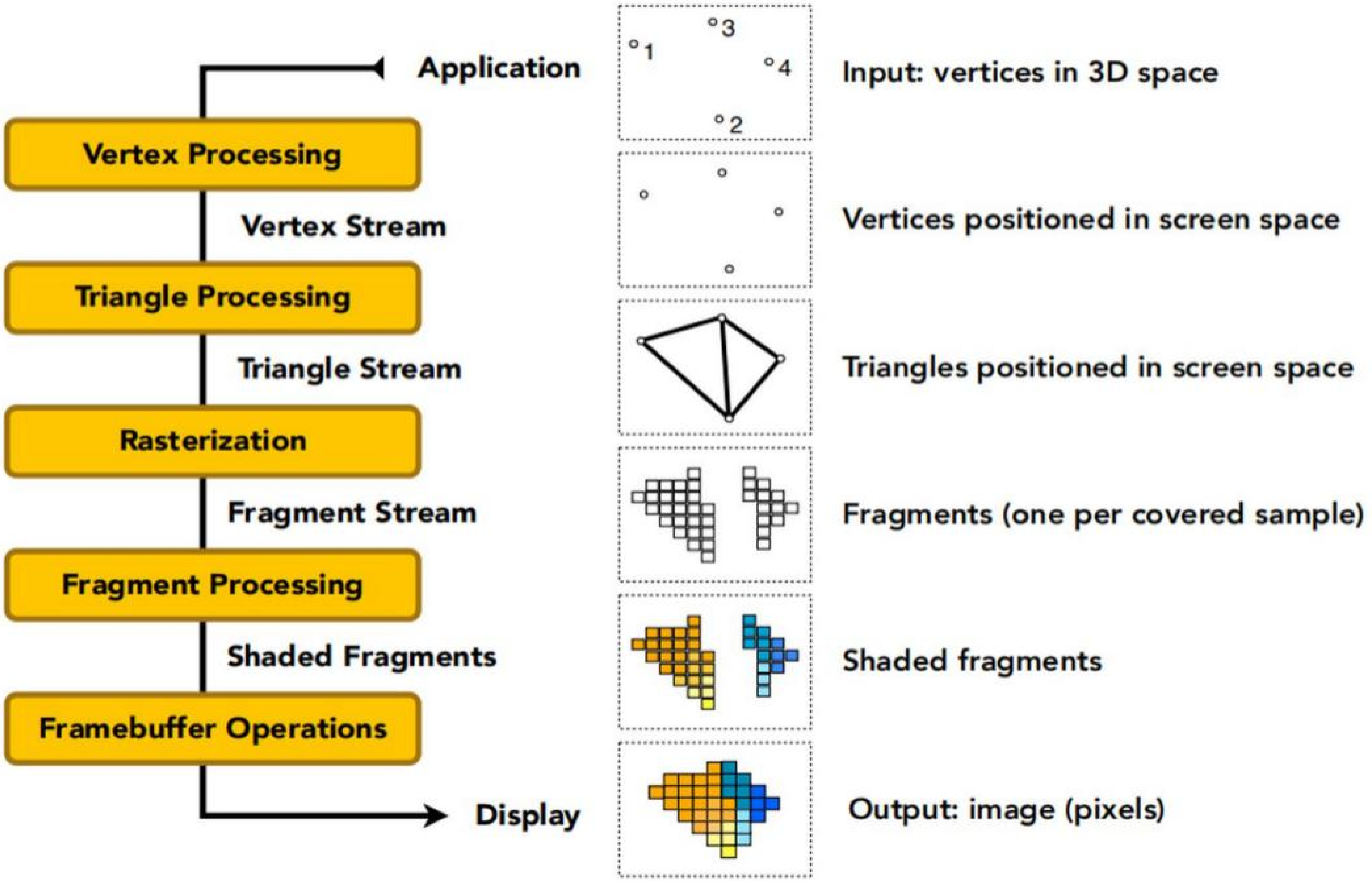
Q4.What is TBDR?

Q5.Why do people say AlphaTest is slower than AlphaBlend?

Part 1
3D Graphics Pipeline



Overview of Graphics Pipeline



Application

Vertex Processing

Vertex Stream

Triangle Processing

Triangle Stream

Rasterization

Fragment Stream

Fragment Processing

Shaded Fragments

Framebuffer Operations

Display

Input: vertices in 3D space

Vertices positioned in screen space

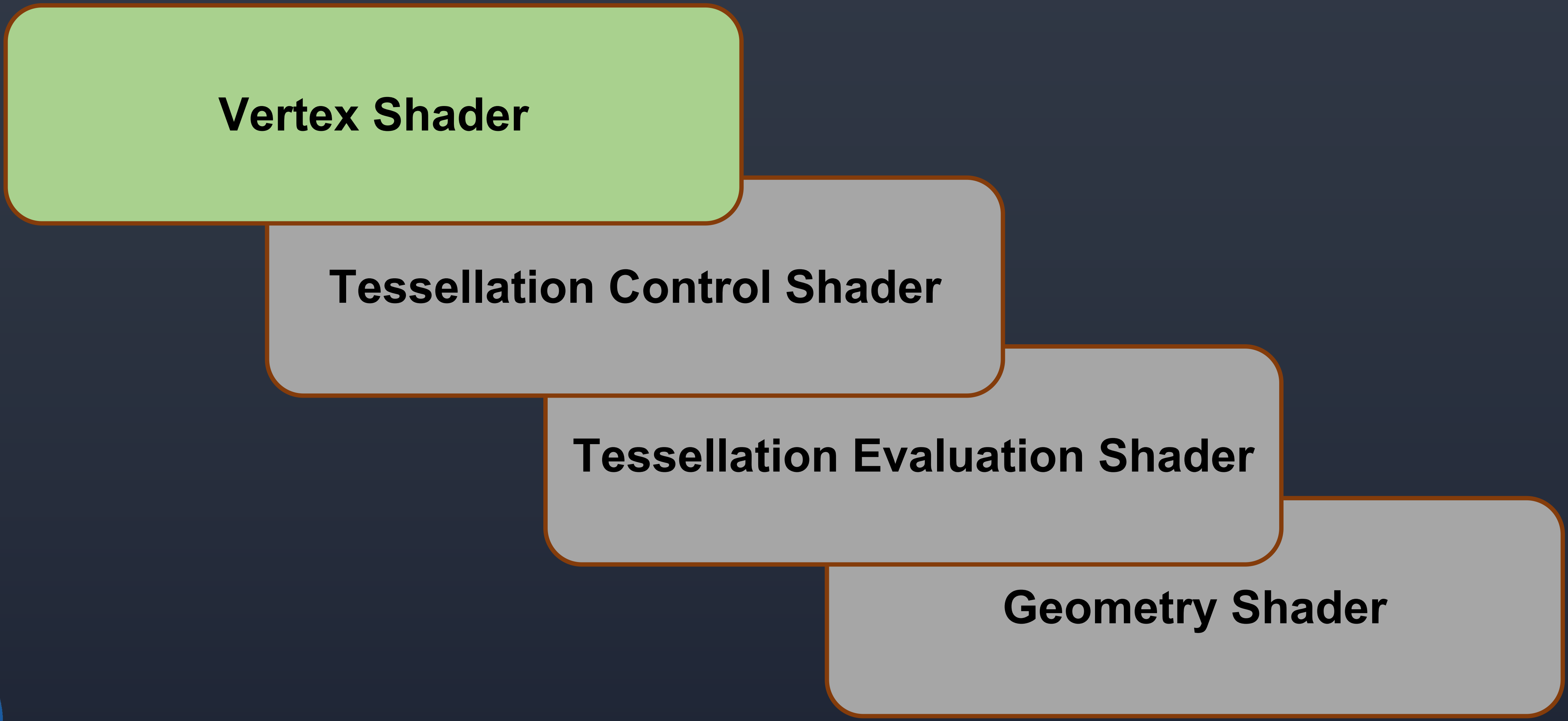
Triangles positioned in screen space

Fragments (one per covered sample)

Shaded fragments

Output: image (pixels)

Vertex Processing:Shaders



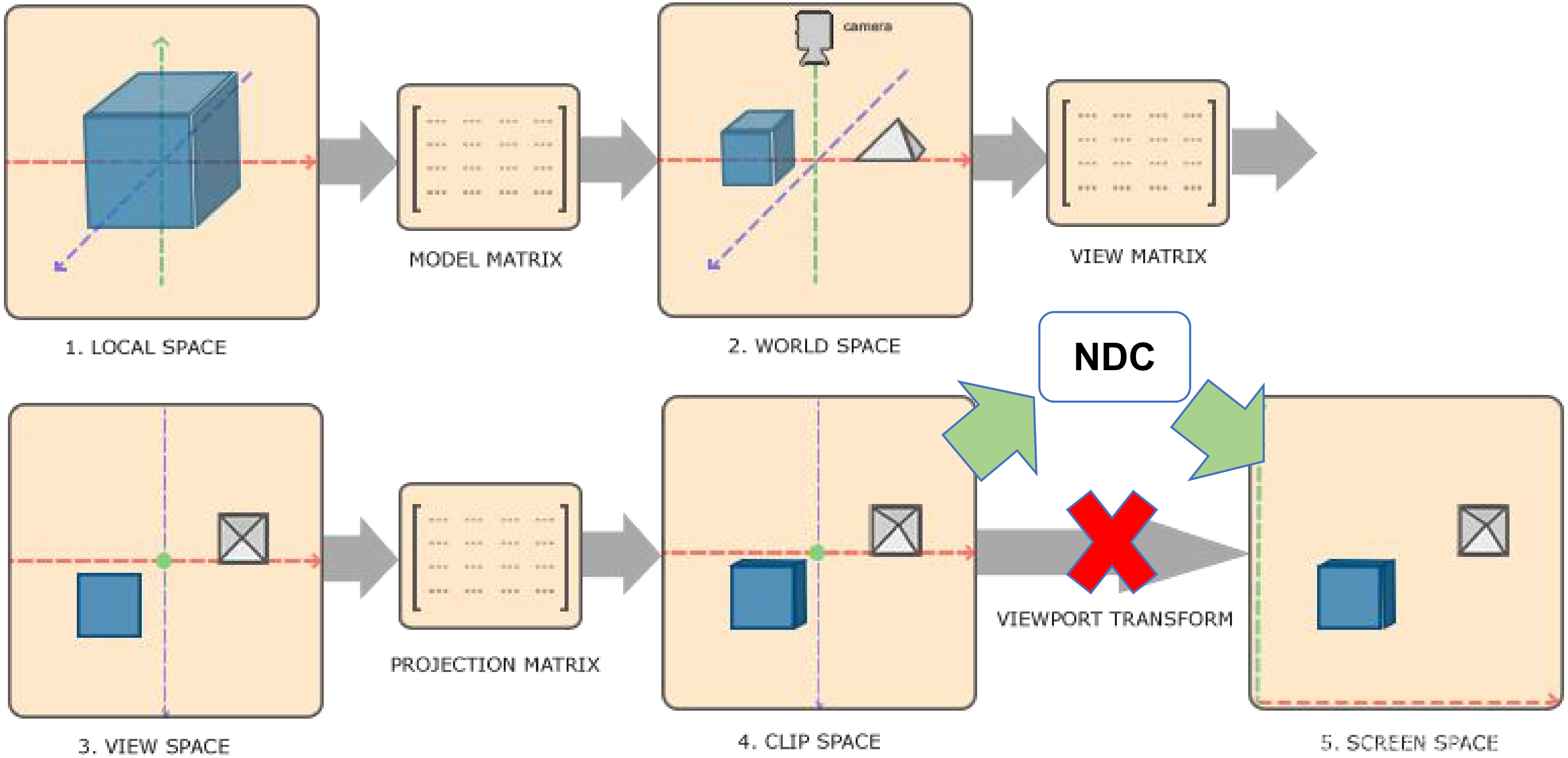
Vertex Shader

Tessellation Control Shader

Tessellation Evaluation Shader

Geometry Shader

Vertex Processing:Vertex Shader

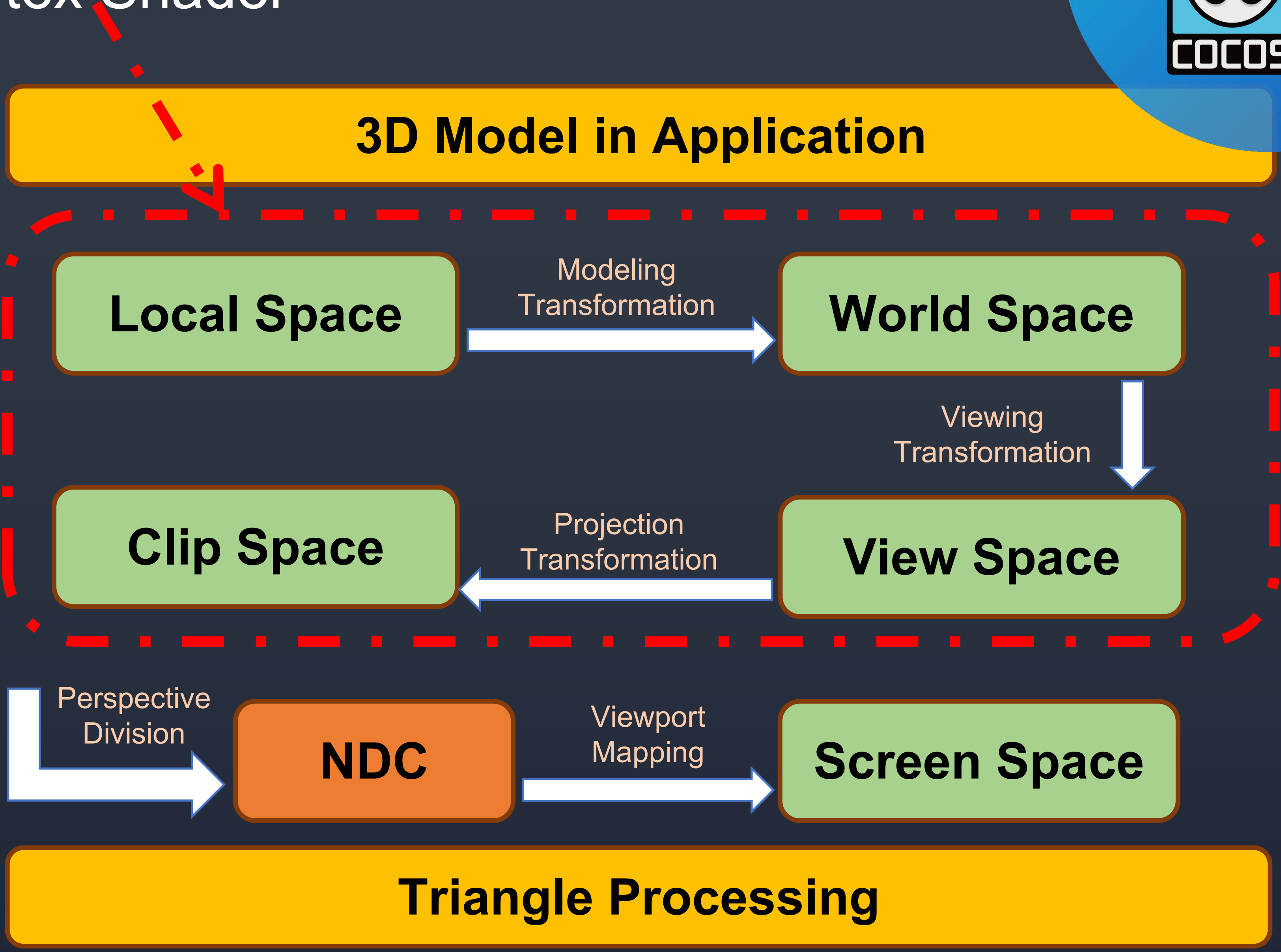


$$V_{clip} = M_{projection} * M_{view} * M_{model} * V_{local}$$

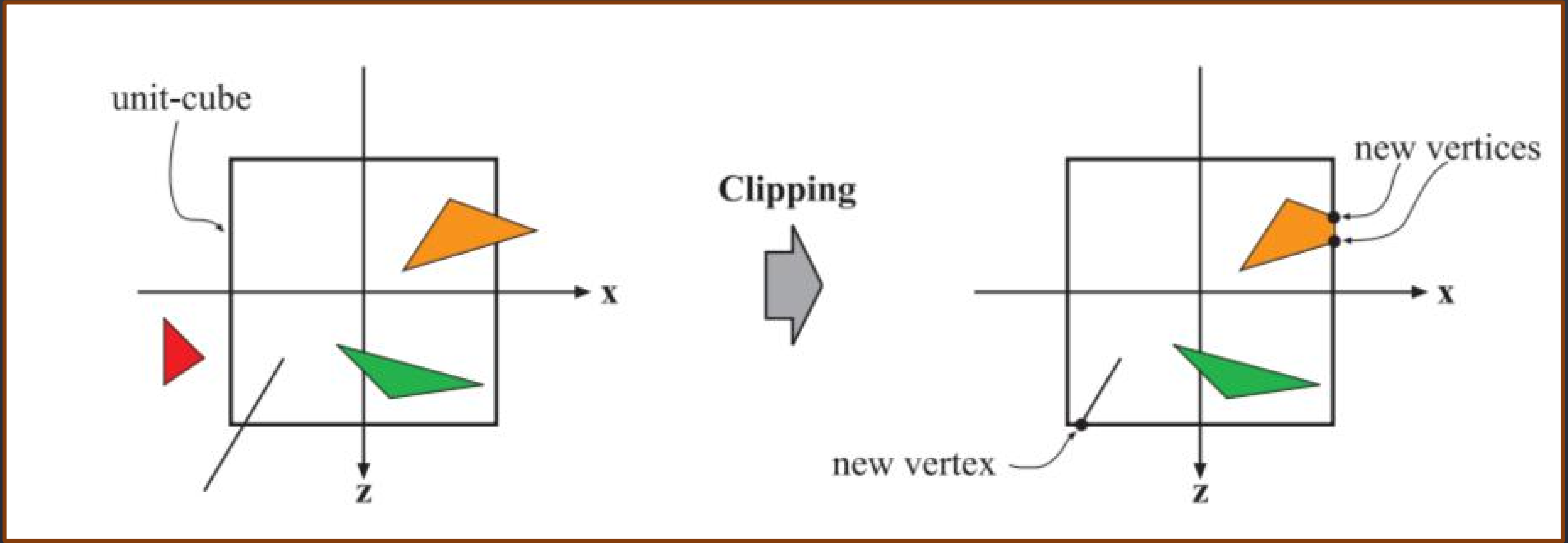
Vertex Processing:Vertex Shader

Vertex Attributes

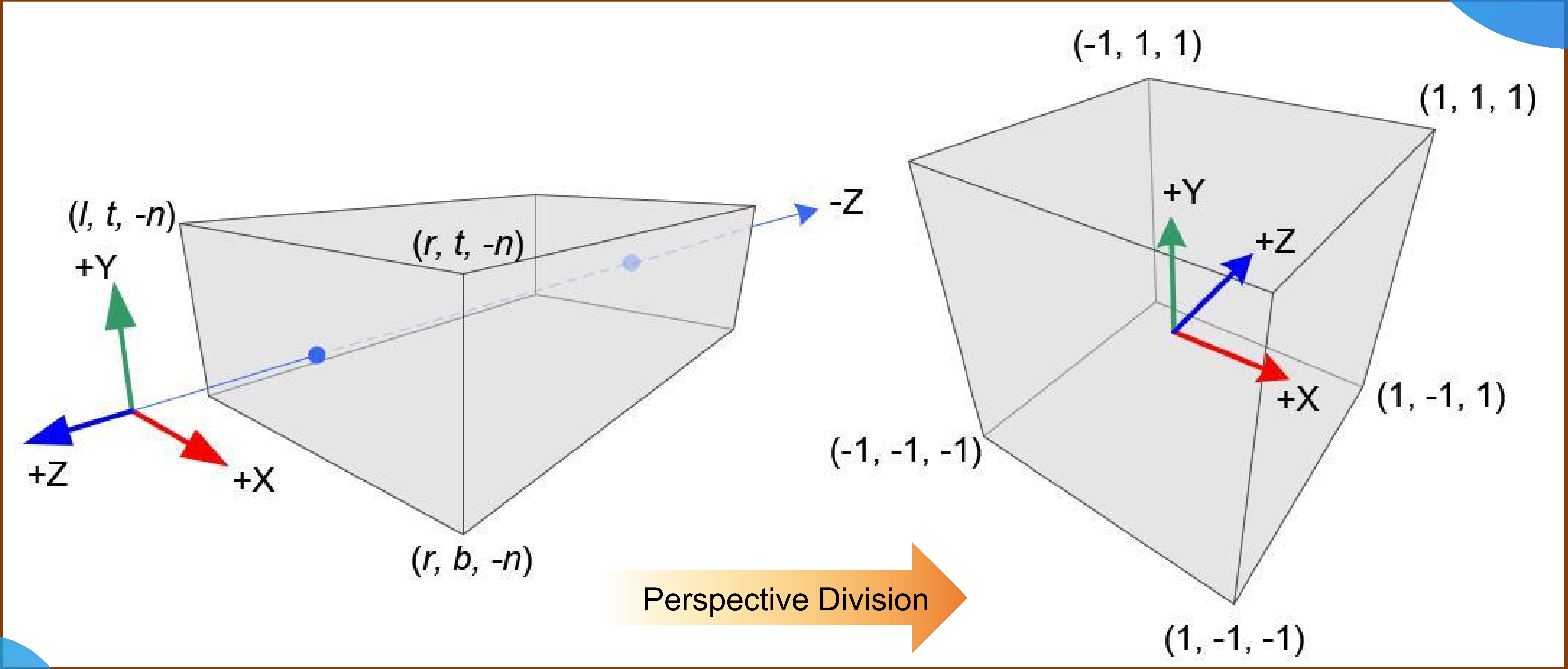
vec4 position
vec3 normal
vec2 texcoord[0..n]
vec3 tangent
vec4 color
vec4 joints
vec4 weights



Clipping in Clip Space

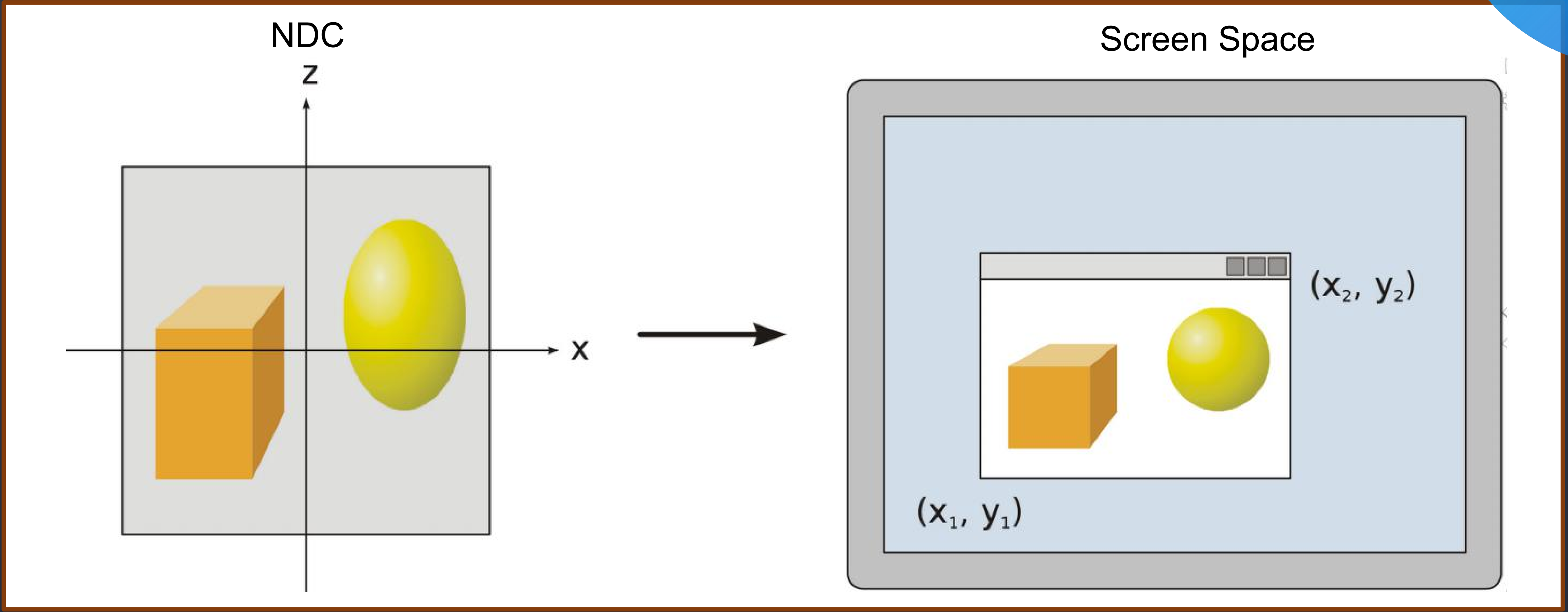


NDC



Normalized Device Coordinates

Viewport Mapping



AS: (X_{screen} - Left) / W_{vp} = (X_{ndc} - (-1))/2 and (Y_{screen} - Bottom) / H_{vp} = (Y_{ndc} - (-1))/2
SO: **(X_{screen},Y_{screen})** = (**(X_{ndc} + 1) / 2 * W_{vp} + Left**,**(Y_{ndc} + 1) / 2 * H_{vp} + Bottom**)

NOTE: the value of Viewport's x,y,w,h in Cocos Creator is in range[0,1];
SO: Left = vp.x * W_{screen}, **W_{vp}** = vp.w * W_{screen}, **Bottom** = vp.y * H_{screen}, **H_{vp}** = vp.h * H_{screen}

Triangle Processing

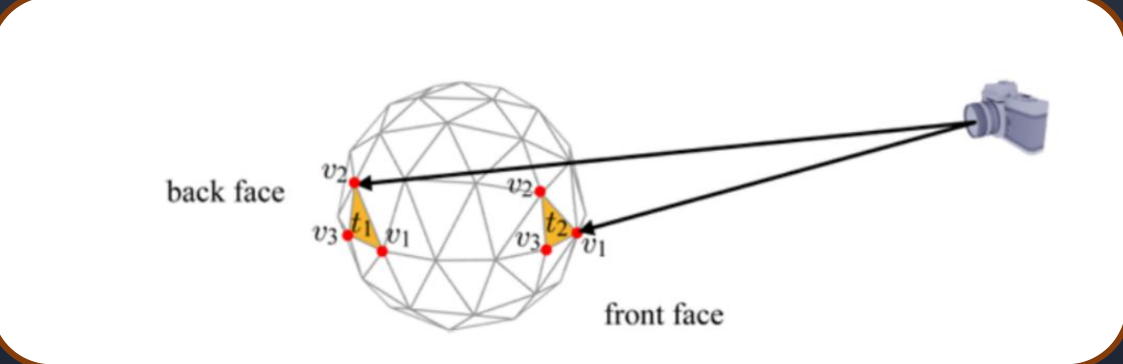
Primitive Assembly

Vertex Stream

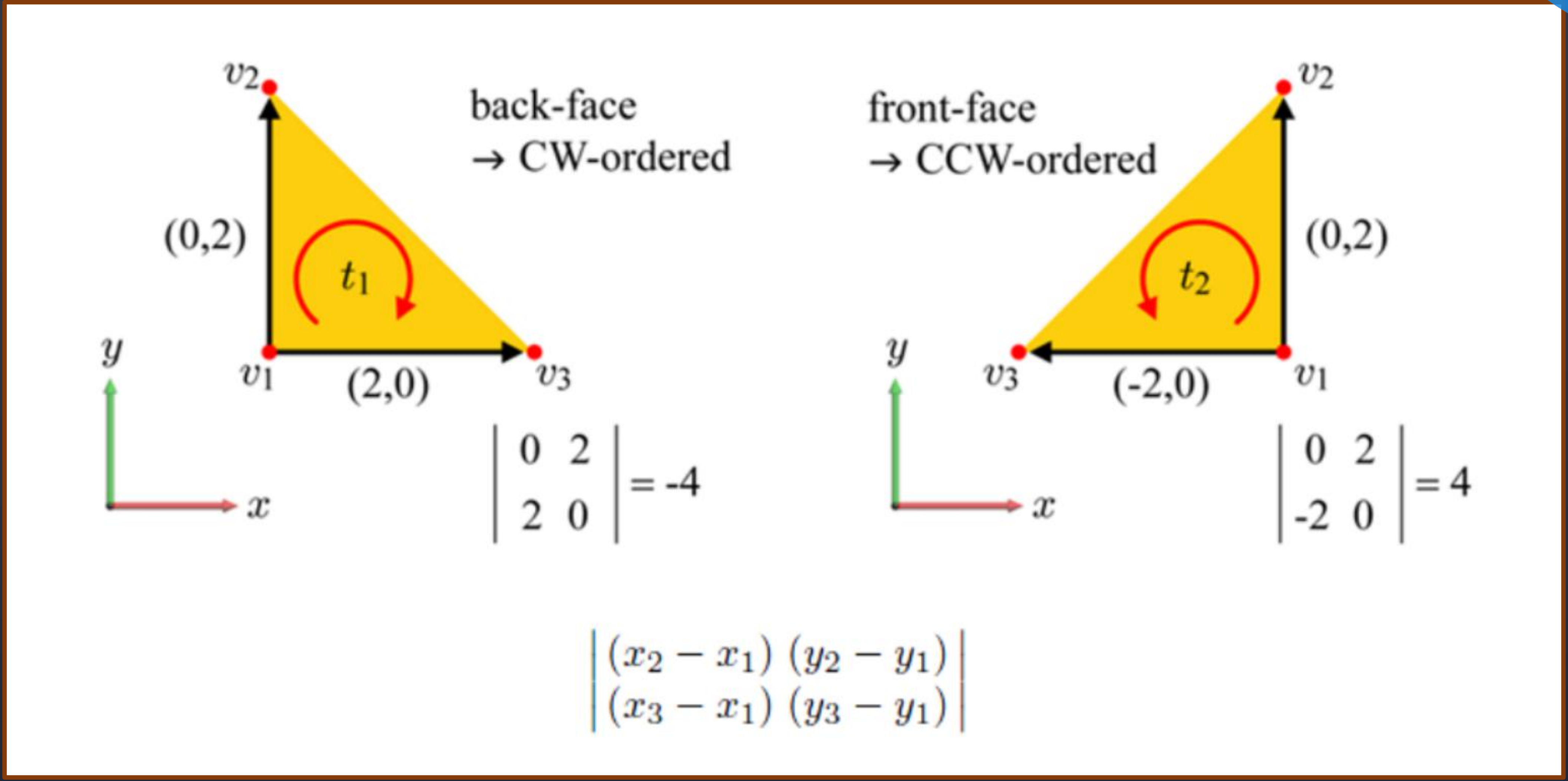
LINES\LINE_STRIP\LINE_LOOP
TRIANGLE_LIST\TRIANGLE_STRIP\TRIANGLE_FAN

Face Culling

Clockwise(CW)
Counter-Clockwise(CCW)



Face Culling

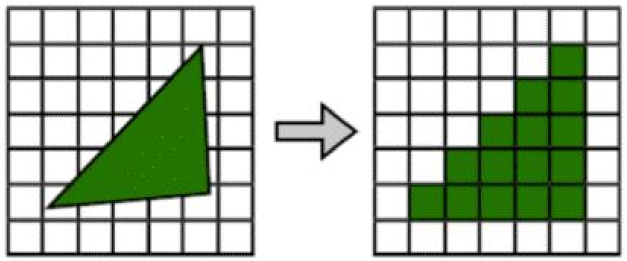


$$\begin{vmatrix} (x_2 - x_1) & (y_2 - y_1) \\ (x_3 - x_1) & (y_3 - y_1) \end{vmatrix}$$

Rasterization

Rasterizing

Primitives ❯ **Pixels** =

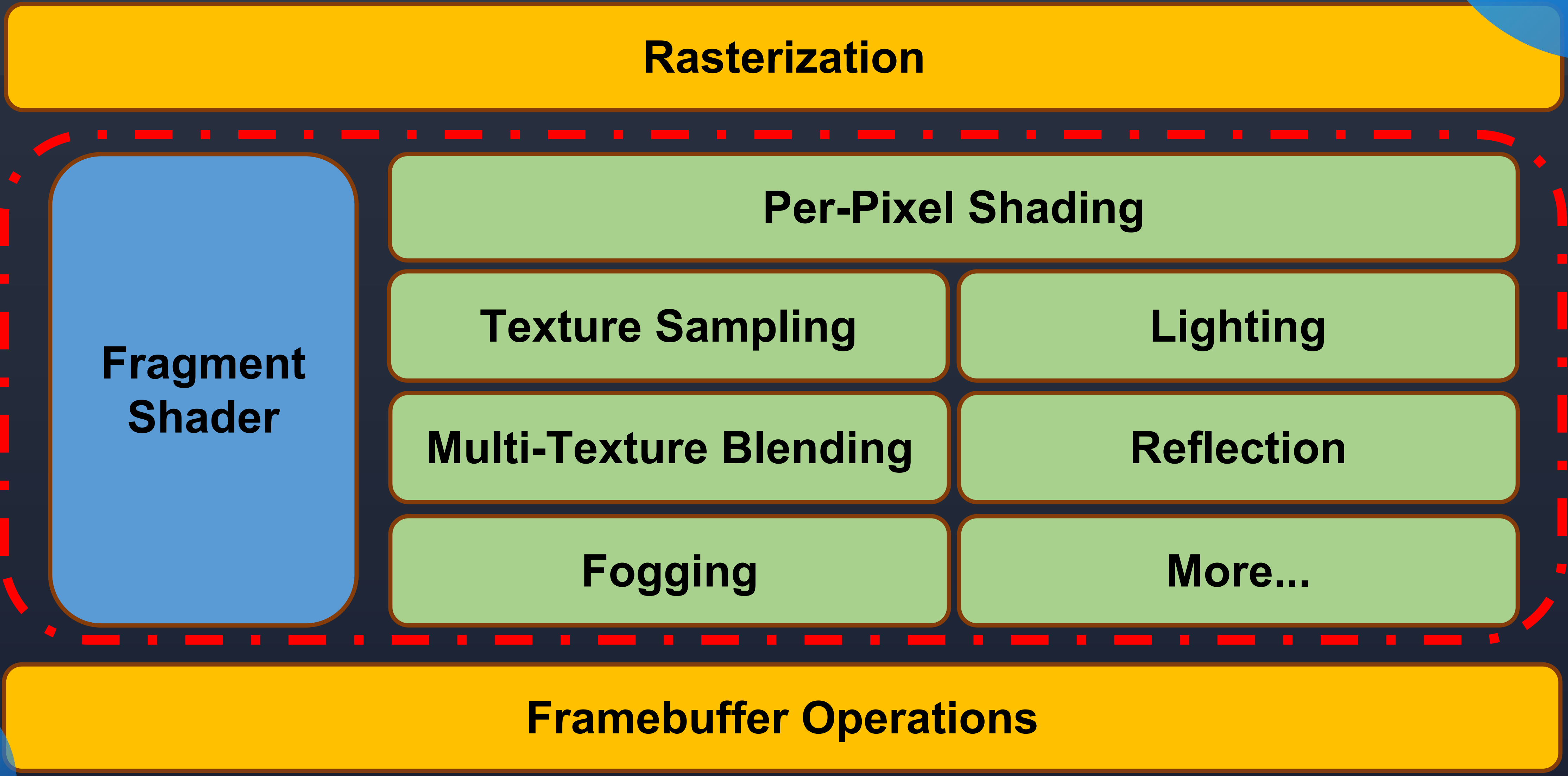


+

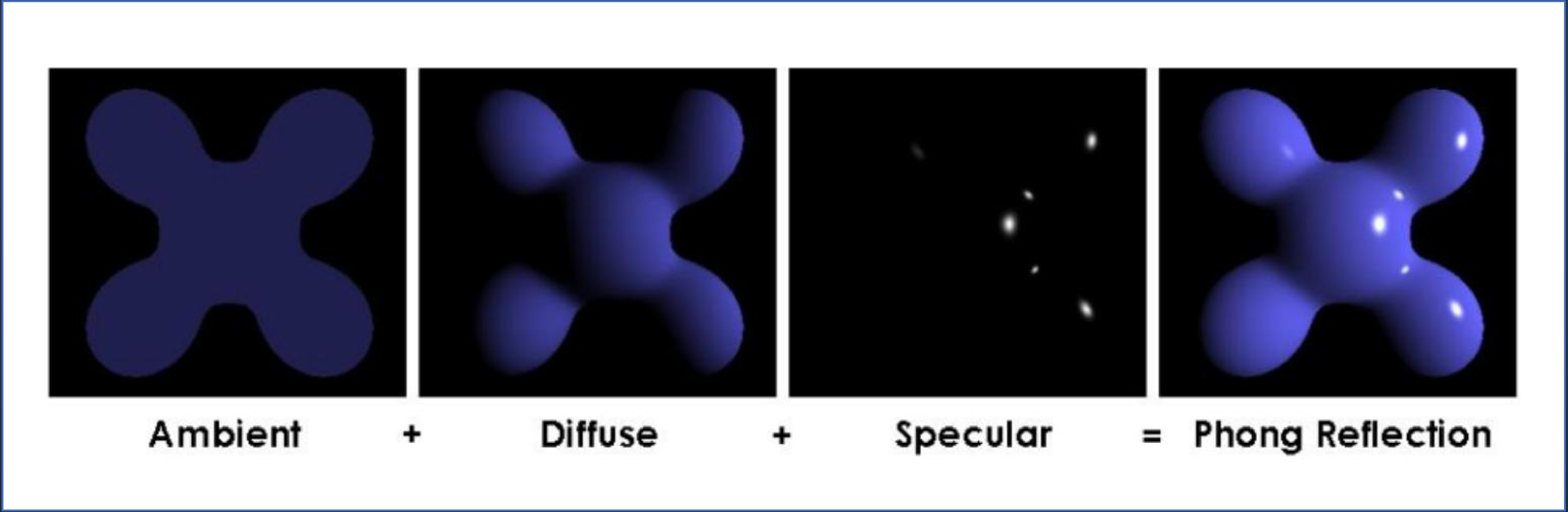
Interpolation

All of Vertex Shader Outputs(Position\Color\Normal\Texcoord etc.)

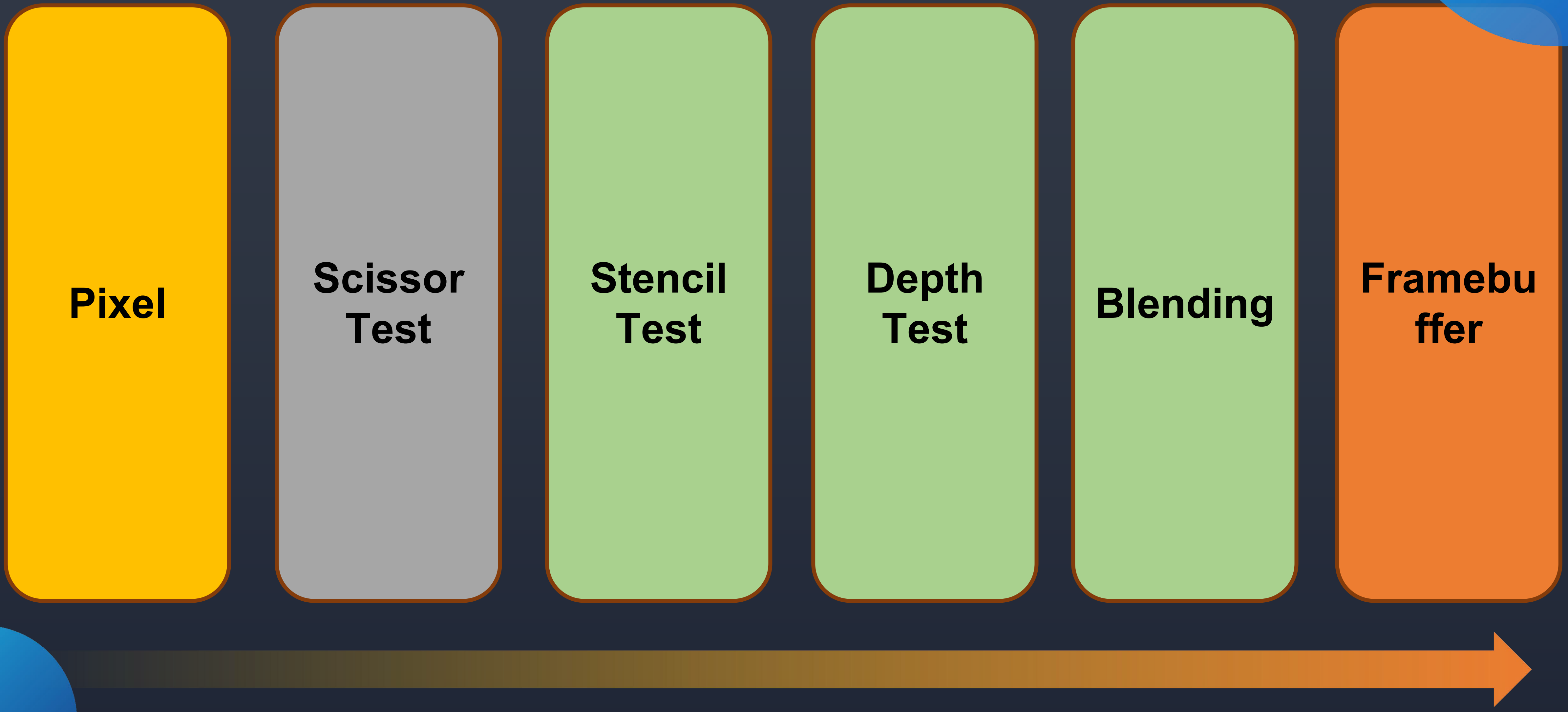
Fragment Processing



Fragment Processing: Fragment Shader



Framebuffer Operations



Pixel

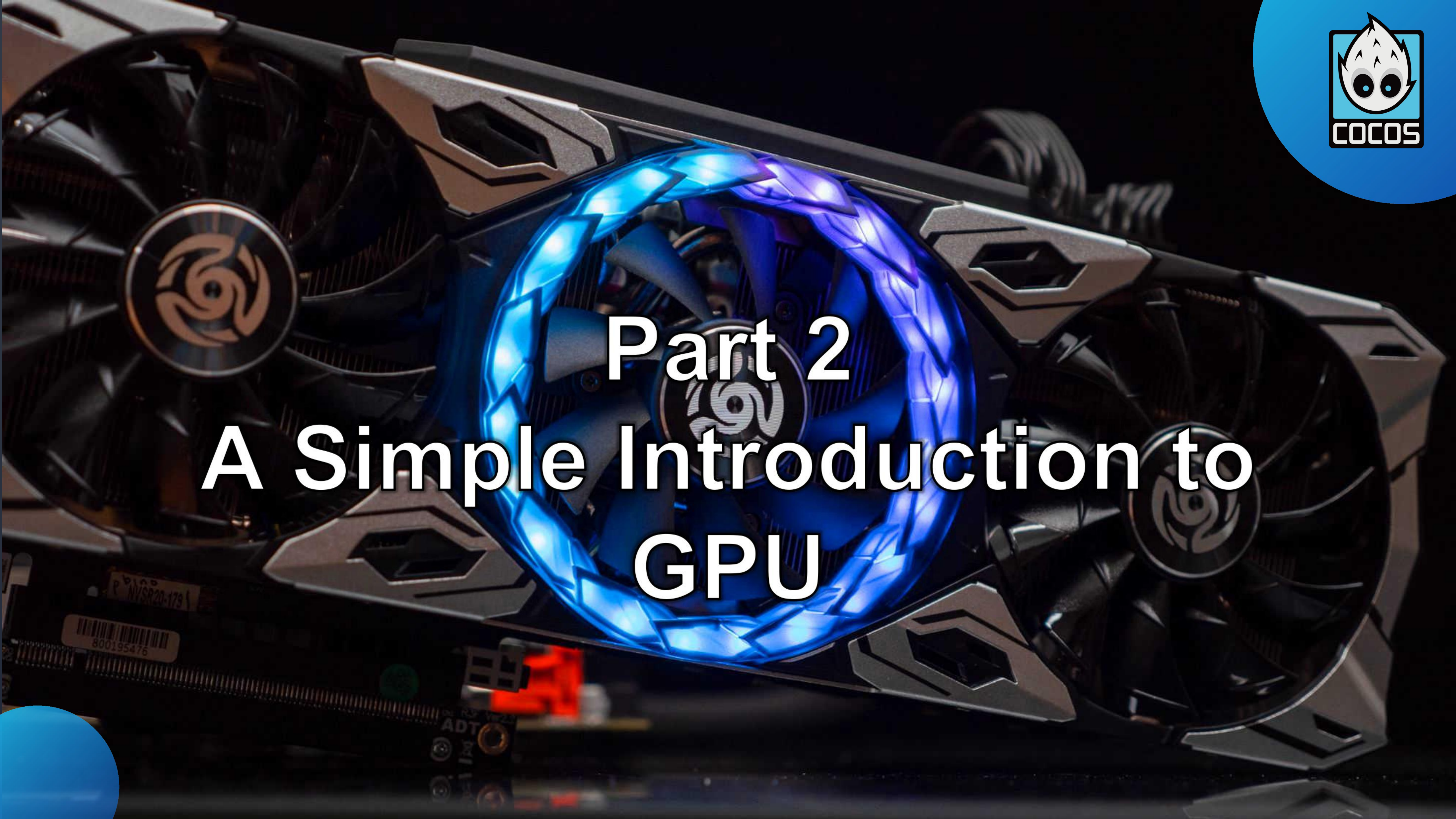
Scissor Test

Stencil Test

Depth Test

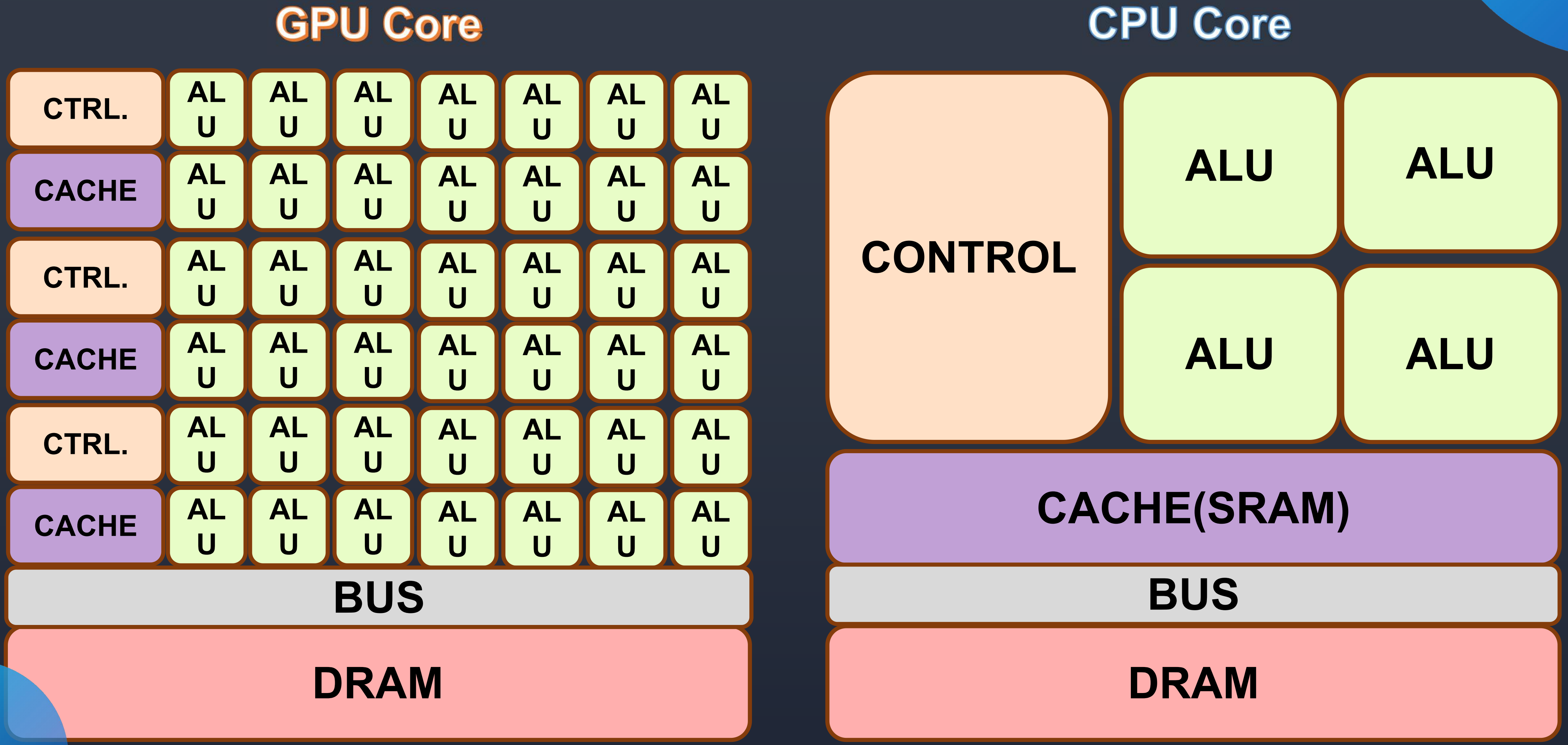
Blending

Framebuffer

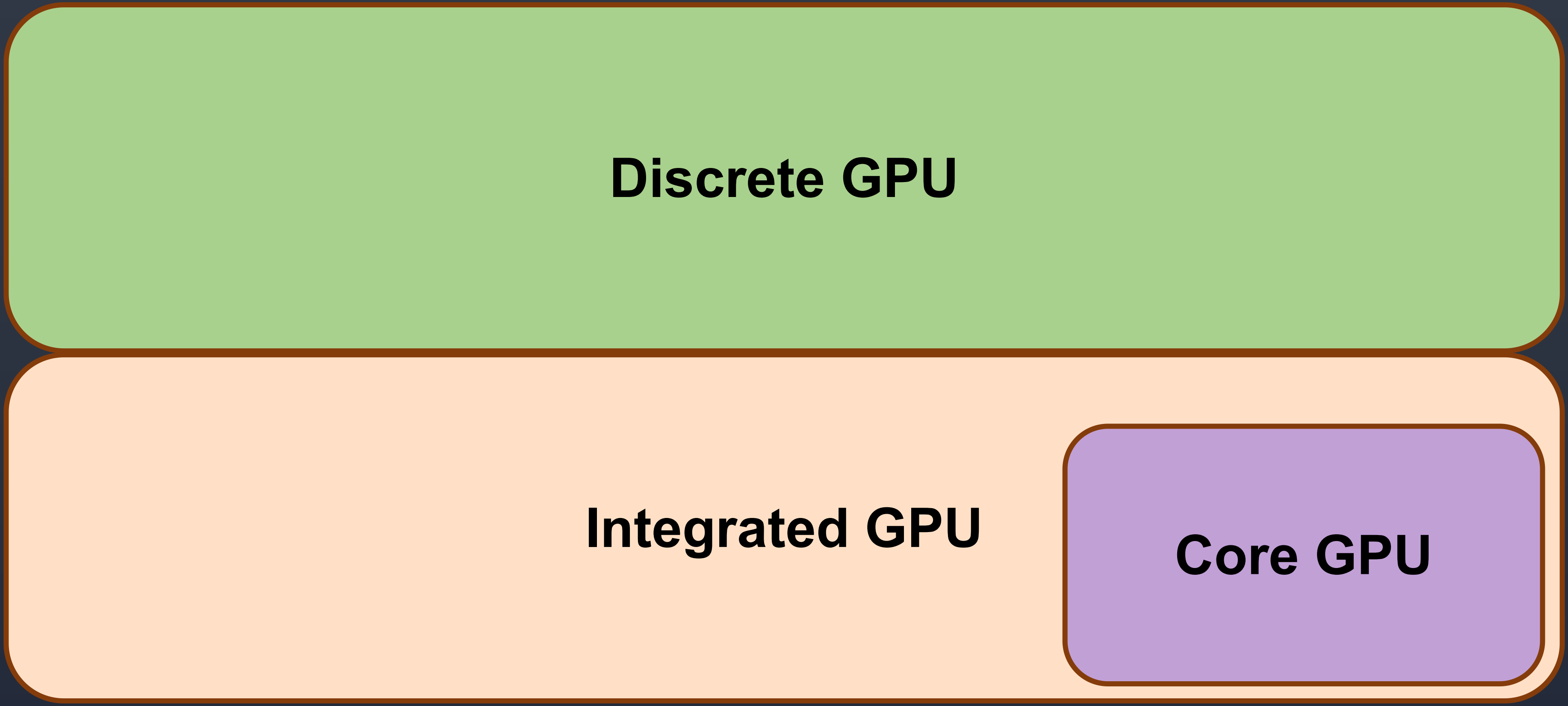


Part 2
A Simple Introduction to GPU

GPU Core vs. CPU Core



Types of GPU

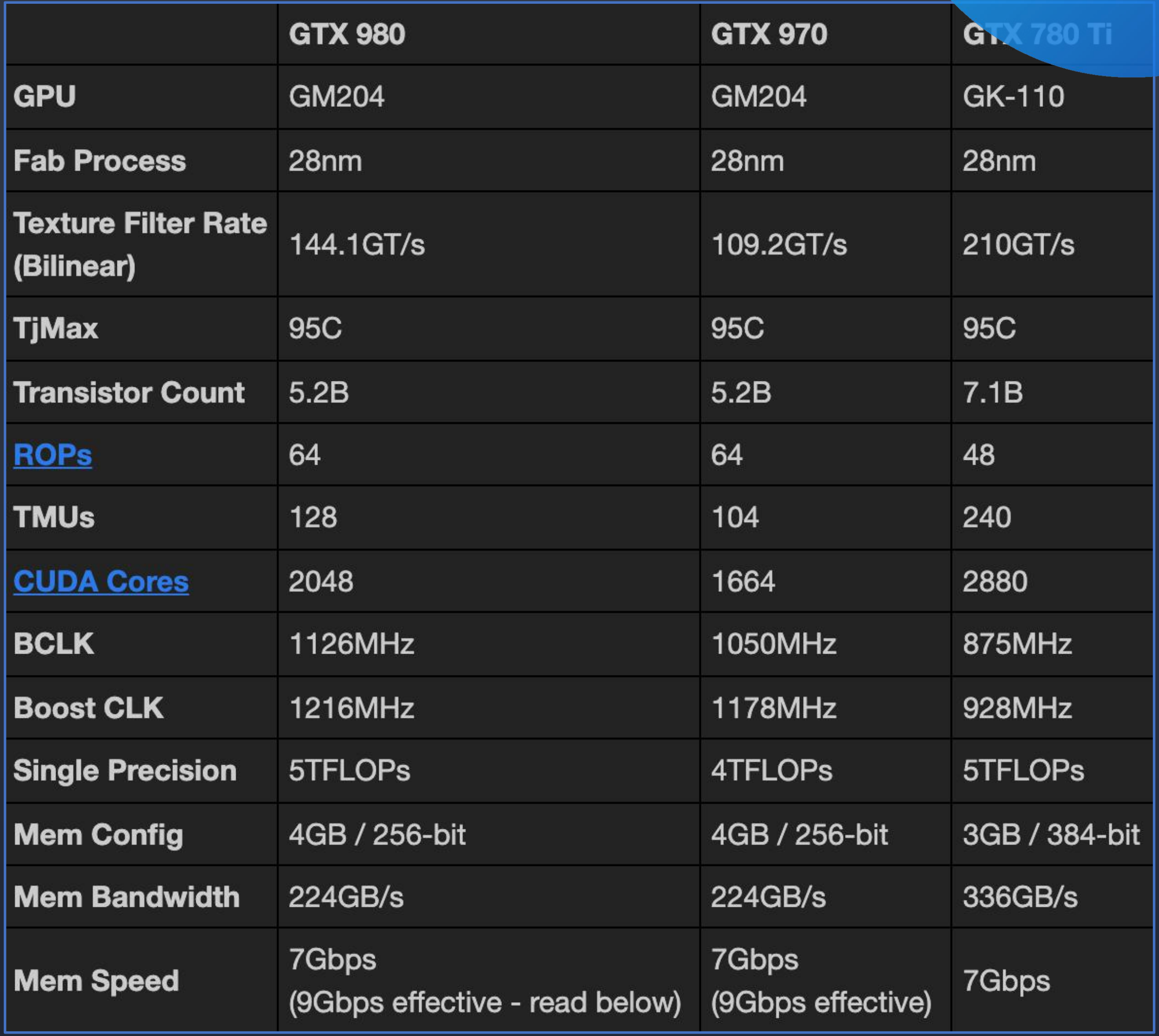


Main Parameters of GPU

Core Block (MHz): Base Clock / Boost Clock
Memory Clock (Gbps)
Memory Size (GB): Standard Memory Config
Memory Interface: DDR1、DDR2、... DDRn
Memory Bus Width (bit): Memory Interface Width
Memory Bandwidth (GB/S)
ROPs: the number of Raster Operations Processors
TMUs: the number of Texture Mapping Units

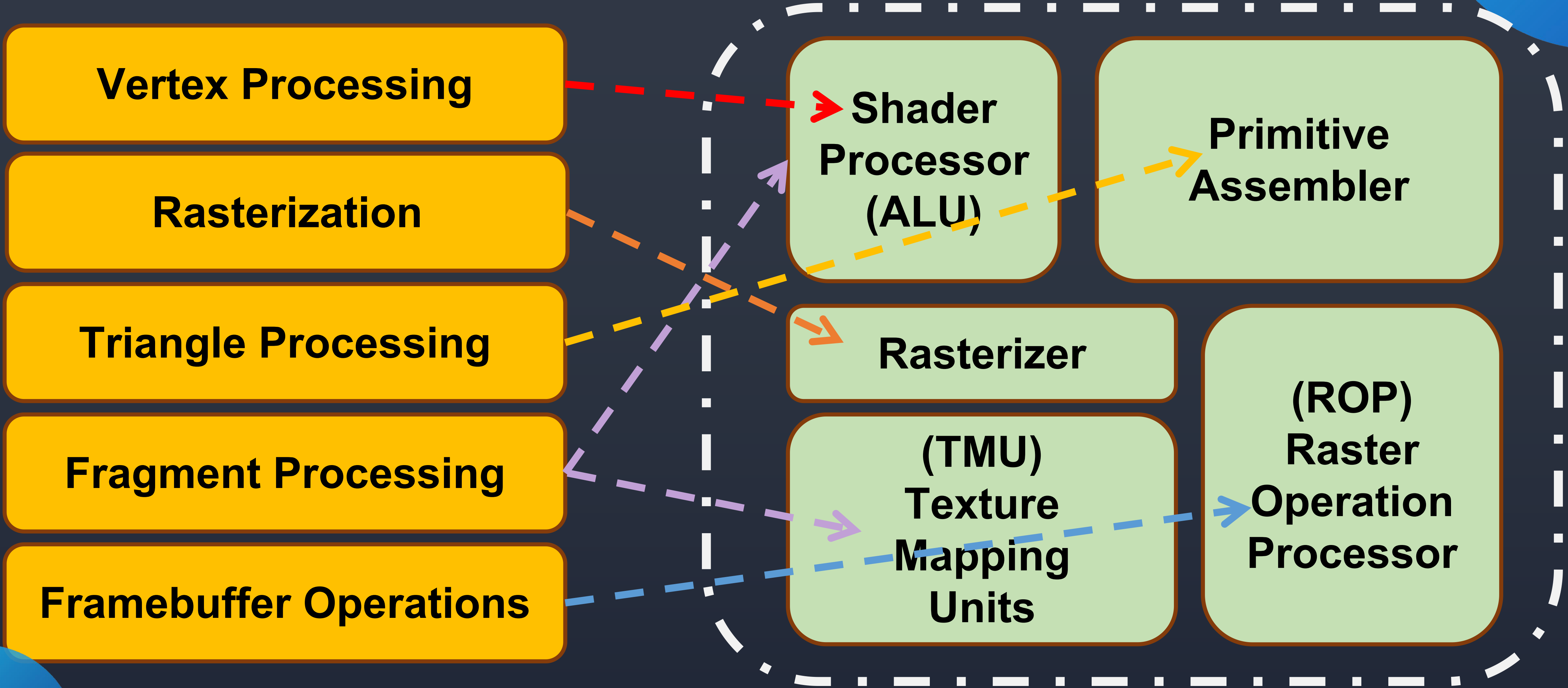
Pixel Fill Rate (MPixels/S)
Texture Fill Rate (MTexels/S)
GFLOPS: Giga (10 billion) Float-point Operations Per Second

--Formulas--
*Pixel Fill Rate = Core Block * ROPs*
*Texture Fill Rate = Core Block * TMUs*
*TFLOPS = 1024 * GFLOPS*

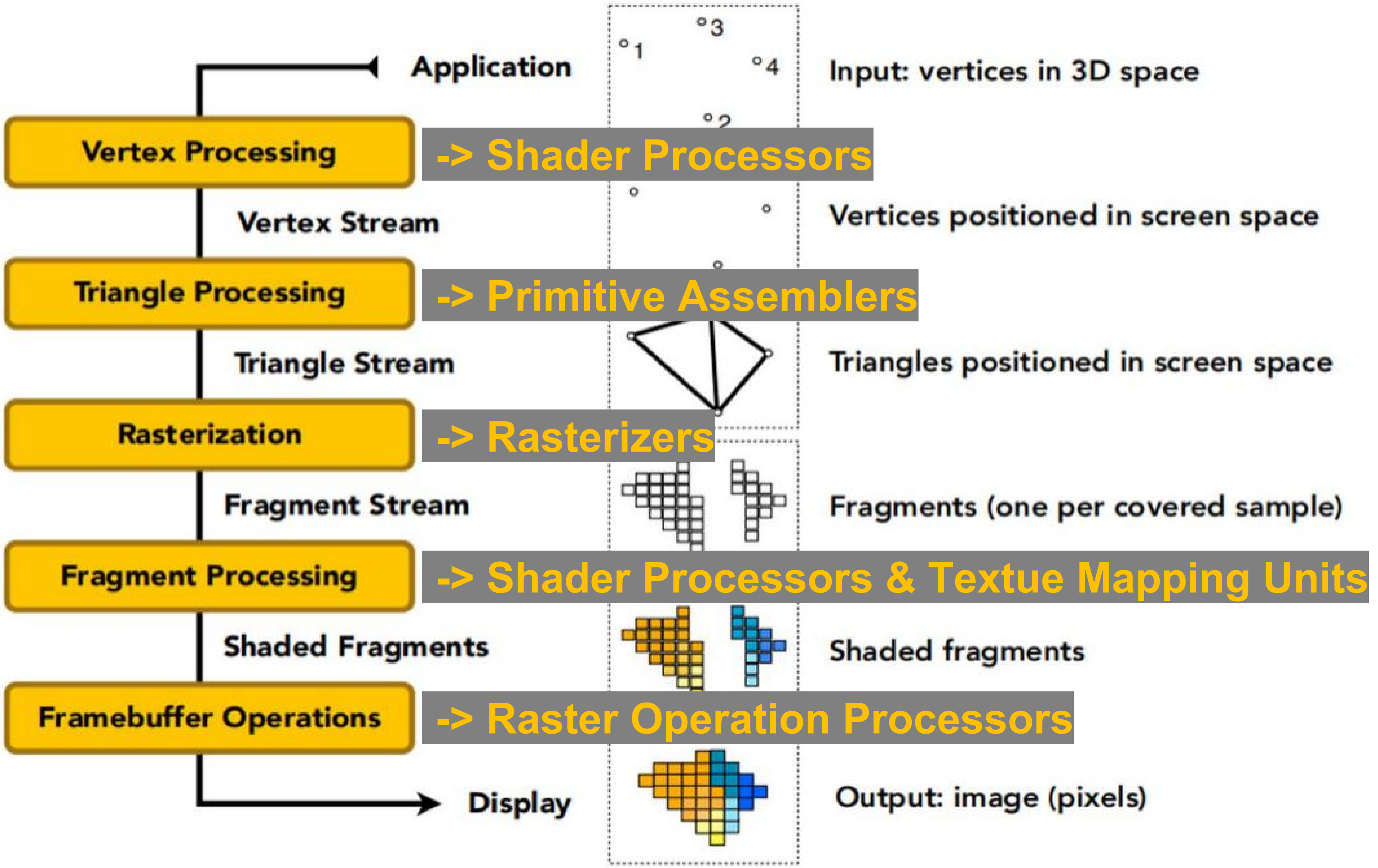


	GTX 980	GTX 970	GTX 780 Ti
GPU	GM204	GM204	GK-110
Fab Process	28nm	28nm	28nm
Texture Filter Rate (Bilinear)	144.1GT/s	109.2GT/s	210GT/s
TjMax	95C	95C	95C
Transistor Count	5.2B	5.2B	7.1B
ROPs	64	64	48
TMUs	128	104	240
CUDA Cores	2048	1664	2880
BCLK	1126MHz	1050MHz	875MHz
Boost CLK	1216MHz	1178MHz	928MHz
Single Precision	5TFLOPs	4TFLOPs	5TFLOPs
Mem Config	4GB / 256-bit	4GB / 256-bit	3GB / 384-bit
Mem Bandwidth	224GB/s	224GB/s	336GB/s
Mem Speed	7Gbps (9Gbps effective - read below)	7Gbps (9Gbps effective)	7Gbps

Responsibilities of Processors



Responsibilities of Processors



-> Shader Processors

-> Primitive Assemblers

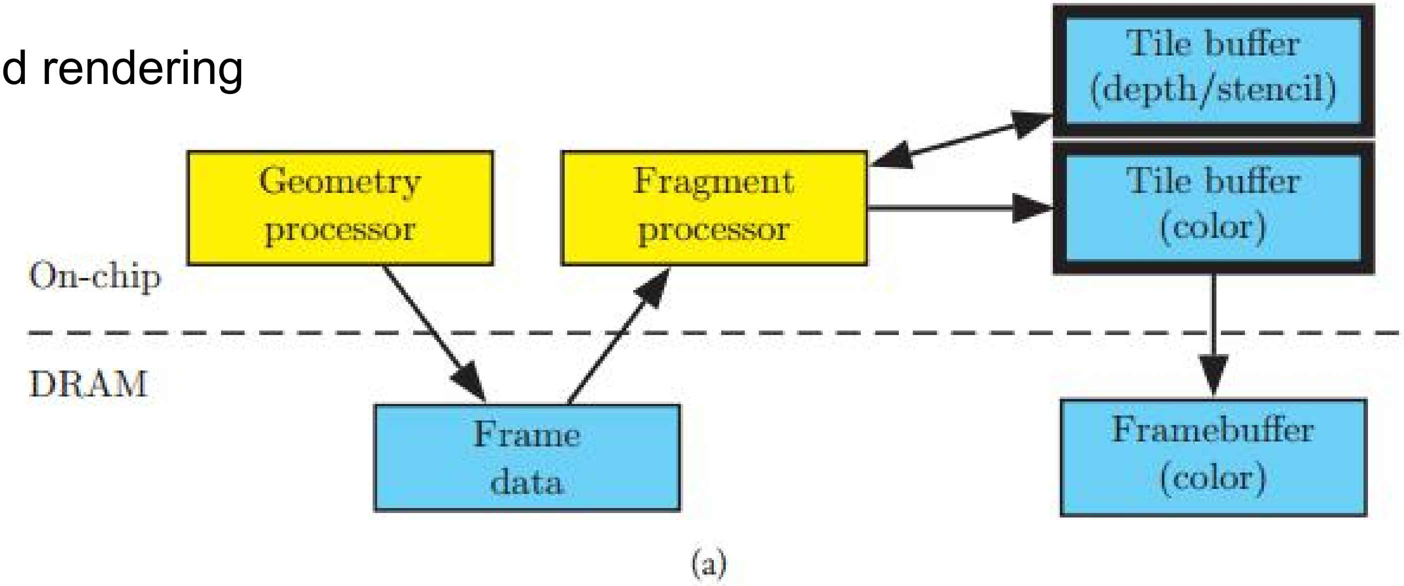
-> Rasterizers

-> Shader Processors & Textue Mapping Units

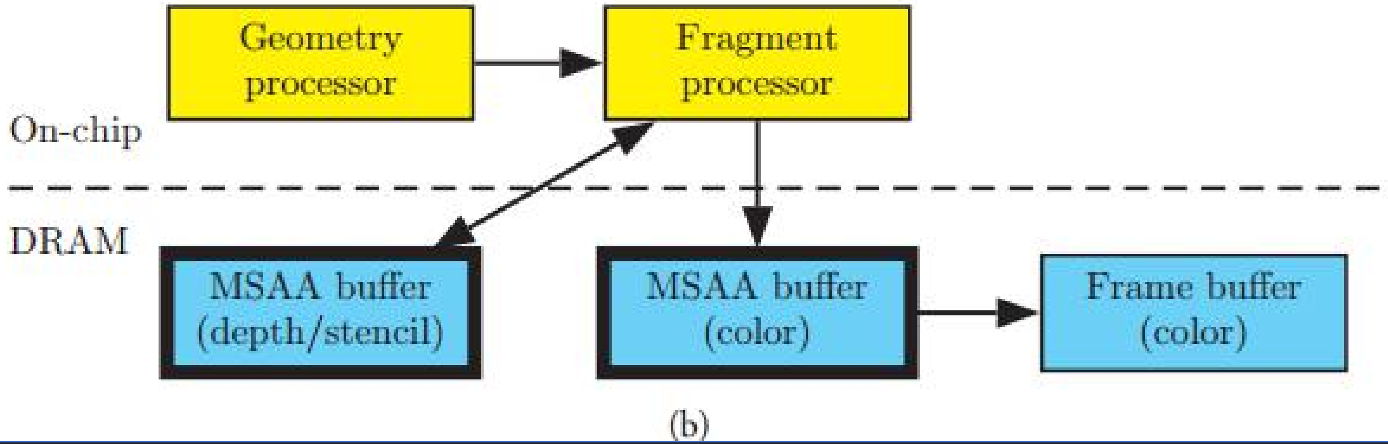
-> Raster Operation Processors

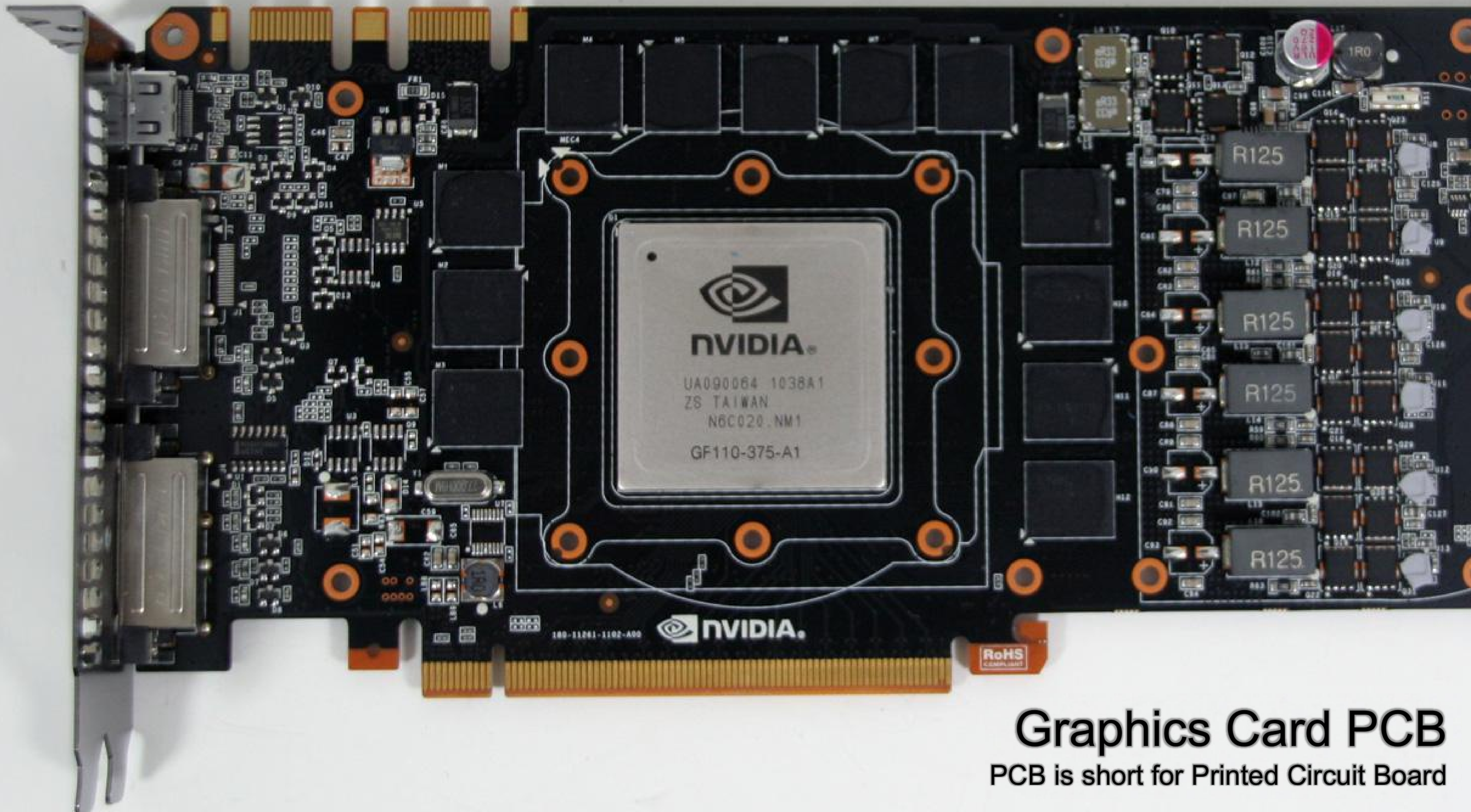
TBDR vs. IR

TBDR tile-based-deferred rendering



IR immediate rendering





Graphics Card PCB

PCB is short for Printed Circuit Board

Part 3
Answers to the Questions

Answers

Q1.How is a 3d scene rendered to a 2d screen?

See slides 5 to 11.

Q2.What are the key differences between GPU and CPU?

All of GPUs on mobile are integrated on the CPU, and often based on TBDR architecture.

Q3.What are bandwidth and fillrate?

Bandwidth is the theoretical maximum amount of data that the bus can handle at any given time.

pixel fill rate is the number of pixels a video card can render to screen and write to video memory in a second.

texture fill rate is the number of textured pixels that a graphics card can render on the screen every second.

Answers

Q4.What is TBDR?

Answer: an architecture of GPU,which apply the ability to do z-testing between rasterization and fragment processing phase.

Q5.Why do people say AlphaTest is slower than AlphaBlend?

Answer: Because the AlphaTest will interrupt the workflow of TBDR,then brings extra work to be done.

THANKS