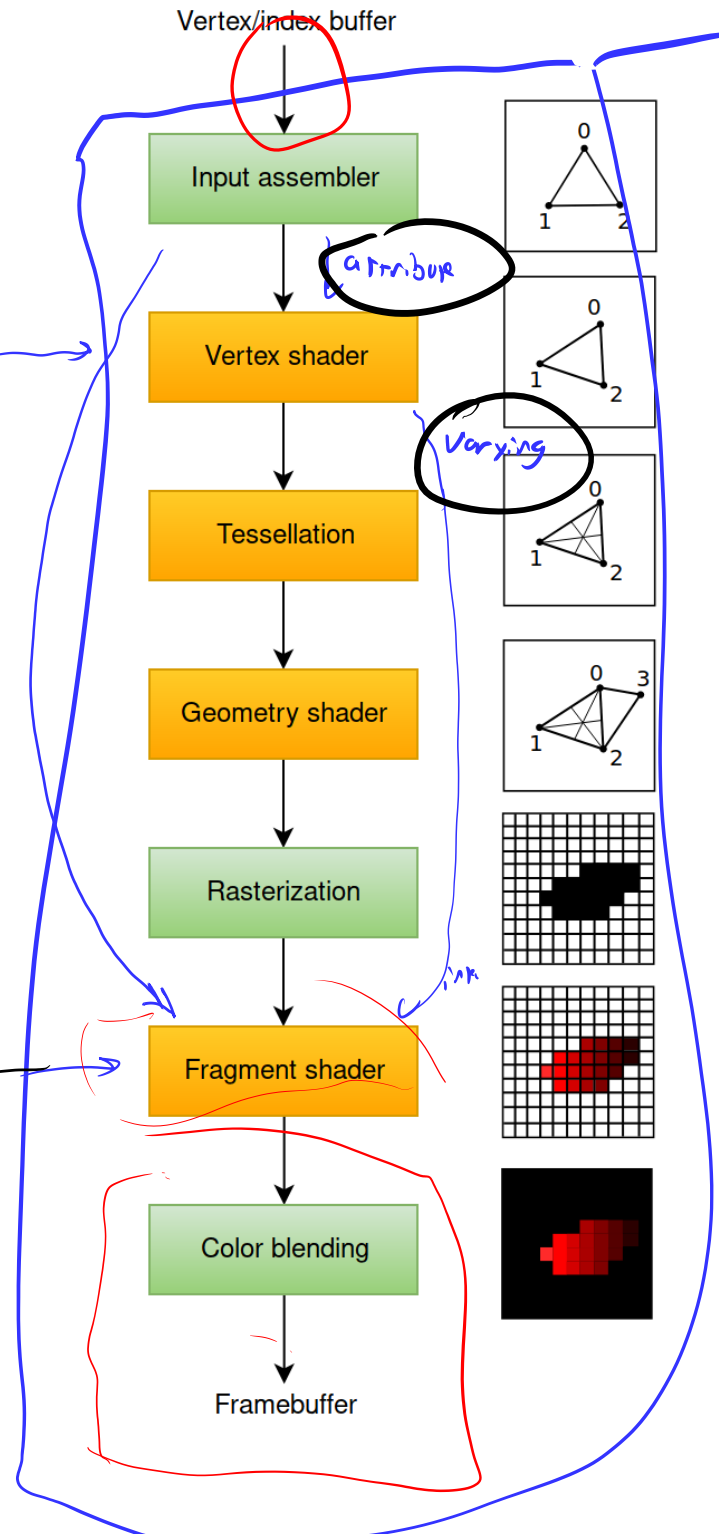


Direct X  $\begin{cases} 1-11 \\ 12 \end{cases}$   
 Open GL  $\begin{cases} 1-2 \\ 3T \end{cases}$

Uniform

Program



Vertex/index buffer

Input assembler

Vertex shader

Tessellation

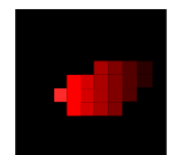
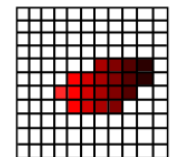
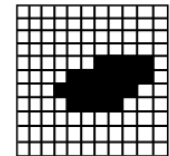
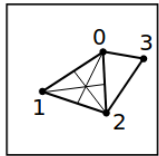
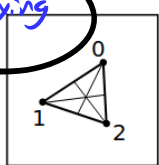
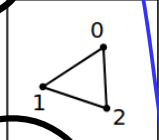
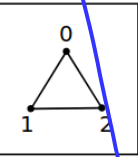
Geometry shader

Rasterization

Fragment shader

Color blending

Framebuffer



Attribute

Varying

Link

Matrix

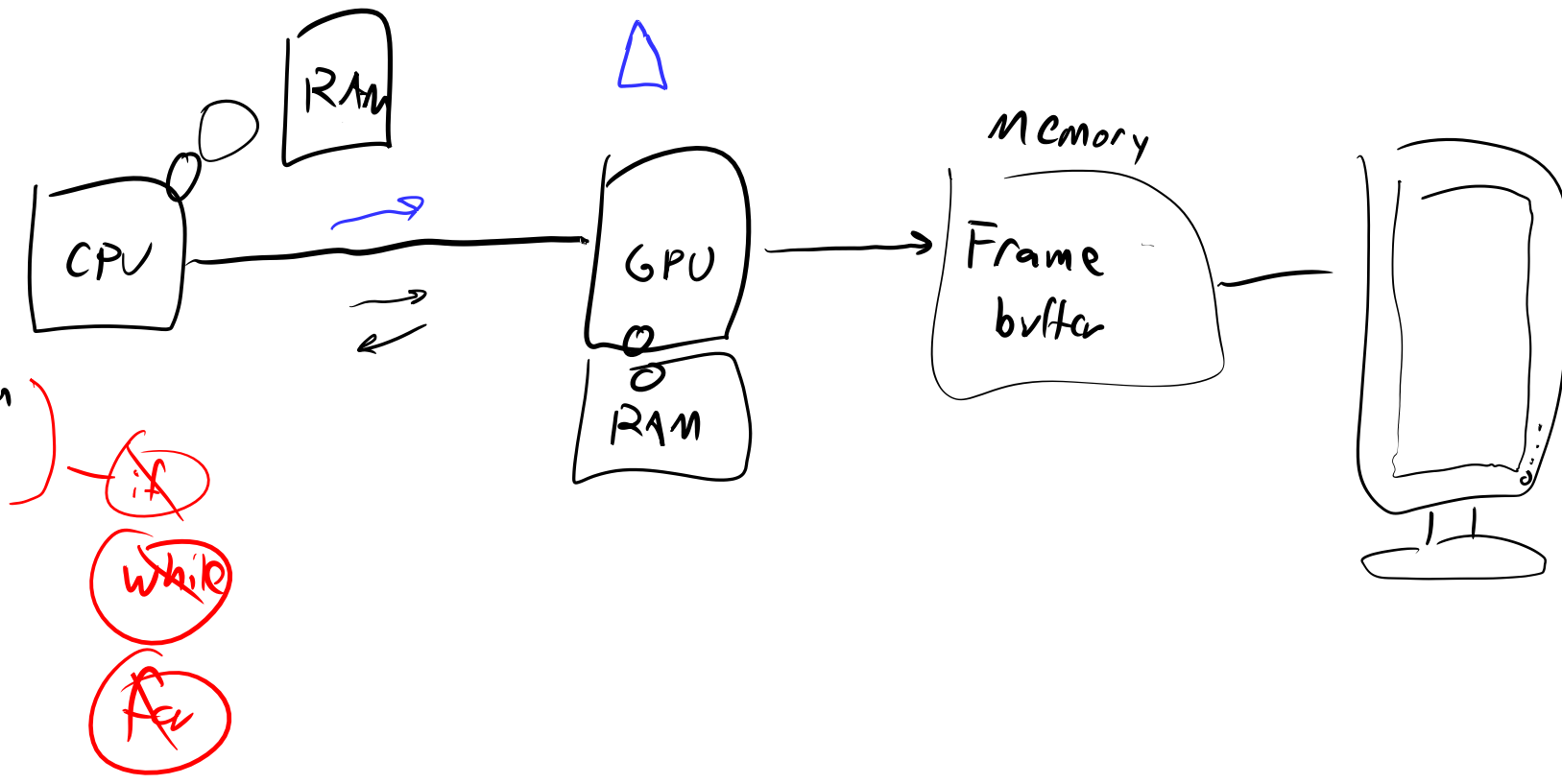
# GPU

massively parallel

SIMD

Single Instruction

Multiple Data

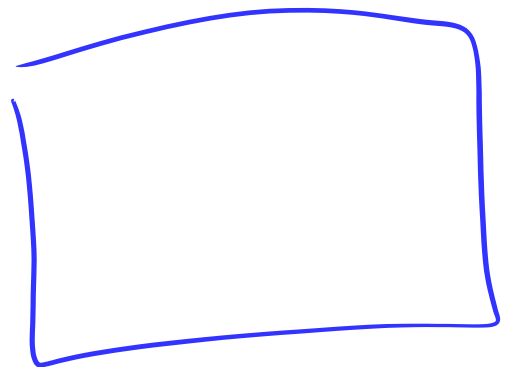


(2,3) (7,11) (21,-3)  
c c c  
T T F  
c=2 c=7

ret 2 ret 7 ret -3  
c = -3

```
int min ( int a, int b ) {  
    int c;  
    if ( a < b )  
        c = a;  
    else  
        c = b;  
    return c;  
}
```

tex



Canvas

↓

CONTEXT

# Vulkan Explicit GPU Control



Complex drivers lead to driver overhead and cross vendor unpredictability

Error management is always active

Driver processes full shading language source

Separate APIs for desktop and mobile markets

Application

Traditional graphics drivers include significant context, memory and error management

GPU

Application responsible for memory allocation and thread management to generate command buffers

Direct GPU Control

GPU

Simpler drivers for low-overhead efficiency and cross vendor portability

Layered architecture so validation and debug layers can be unloaded when not needed

Run-time only has to ingest SPIR-V intermediate language

Unified API for mobile, desktop, console and embedded platforms

Vulkan delivers the maximized performance and cross platform portability needed by sophisticated engines, middleware and apps

Wob GZ

CPU

GPU

HTML / JS

GLSL source

FS  
VS

OpenGL

