# CS 150
# Digital Design

# Lecture 26 – Graphics Processors

**2012-11-20**

**Professor Kris Pister**

**today's lecture by John Lazzaro**

TAs: Ian Juch, Vincent Lee, Albert Magyar

**www-inst.eecs.berkeley.edu/~cs150/**

Play

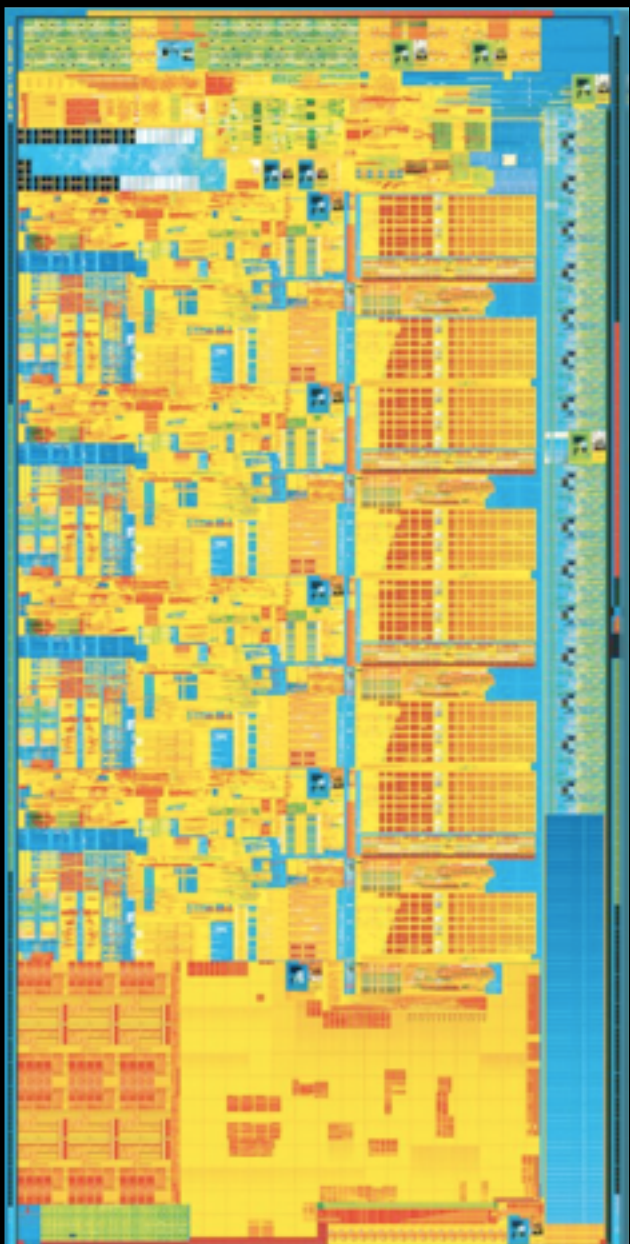
# Today: Graphics Processors

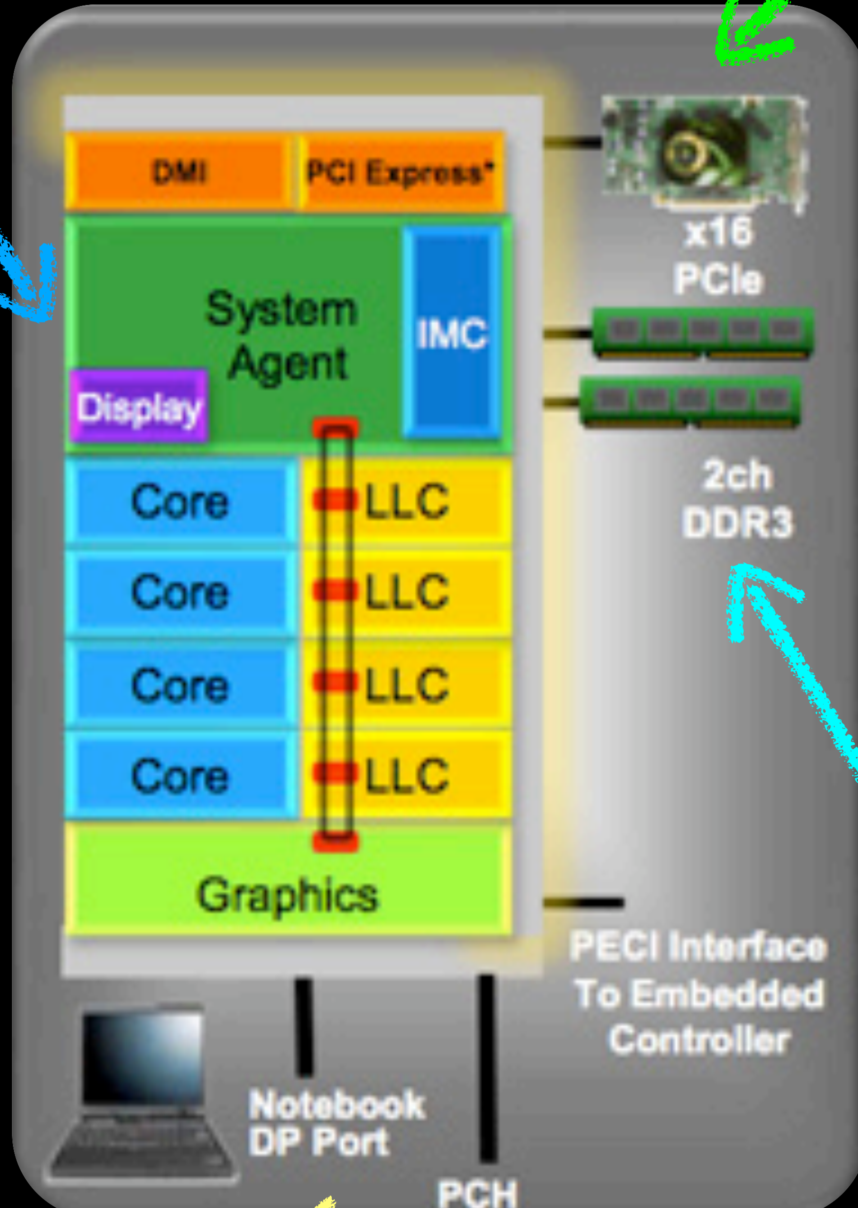
- **Computer Graphics.** A brief introduction to "the pipeline".
- **Stream Processing.** Casting the graphics pipeline into hardware.
- **Unified Pipelines.** GeForce 8800, from Nvidia, introduced in 2006.
- **Kepler.** The latest generation from Nvidia, released earlier this year.

# PC Graphics, 2012 Edition

# PC Graphics Architecture

Core i5
CPU/IGP

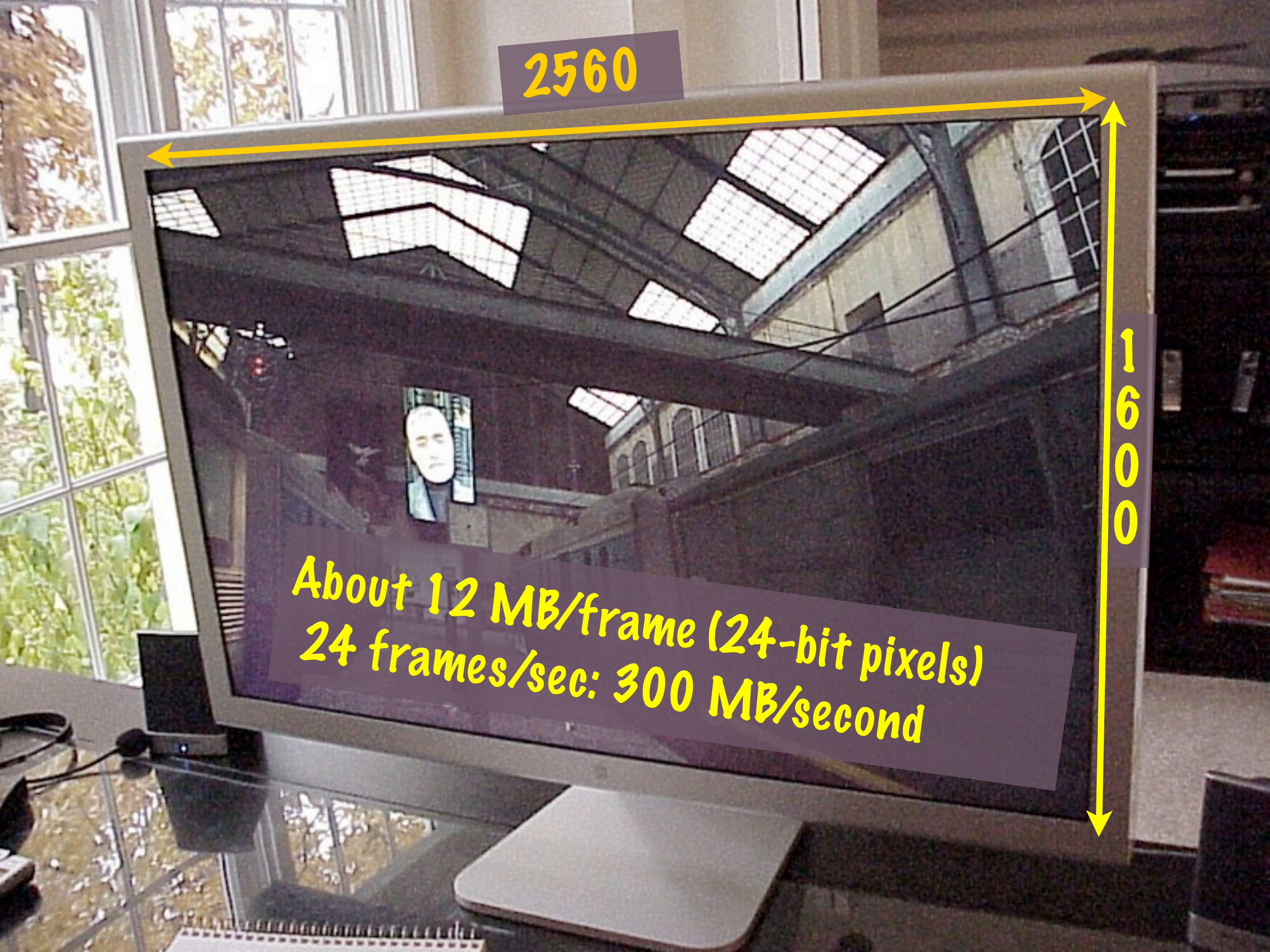





PCIe bus supports discrete GPU, with dedicated RAM and monitor outputs.

IGP uses system DRAM as graphics memory.

Display Out


2560
1600
About 12 MB/frame (24-bit pixels)
24 frames/sec: 300 MB/second

# The “unaccelerated” graphics board ...

PCIe Bus Port


300 MB/s easy to sustain.

**Problem:** CPU has to compute a new pixel **every 10 ns.** 10 clock cycles for a 1 GHz CPU clock.

12 MB Frame Buffer A

12 MB Frame Buffer B

Control Logic

DVI Formatter

D/A

Display Out

Double Buffering:

CPU writes A frame in one buffer.

Control logic sends B frame out of other buffer to display.

Q. What kind of graphics are we accelerating?

A. In 2012, interactive entertainment (3-D games). In the 1990s, 2-D acceleration (fast windowing systems, games like Pac-Man).

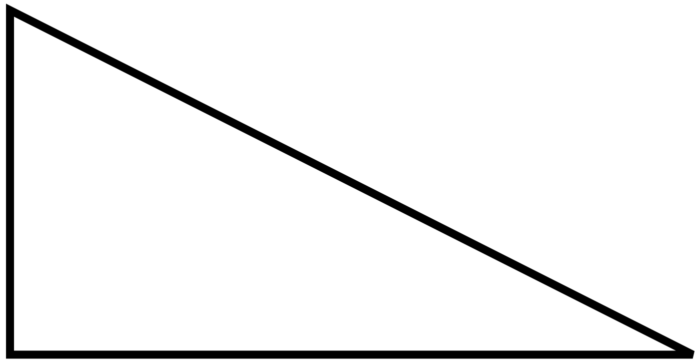
# Graphics Acceleration

Q. In a multi-core world, why should we use a special processor for graphics?

A. Programmers generally use a certain coding style for graphics. We can design a processor to fit the style.

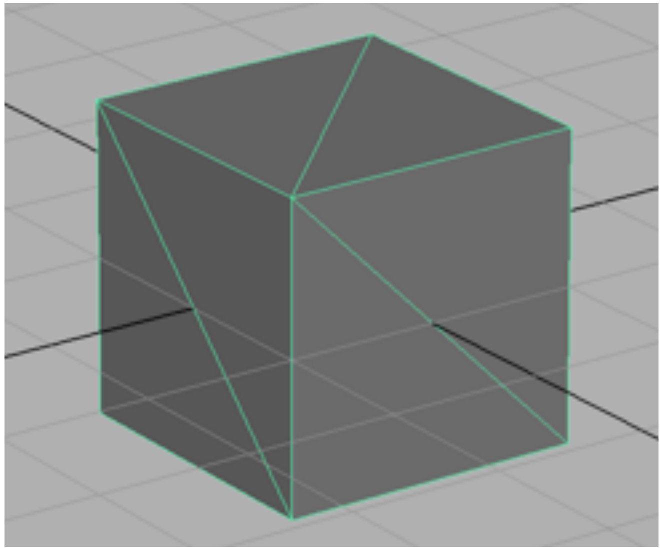
**Next:** An intro to 3-D graphics.

# The Triangle ...



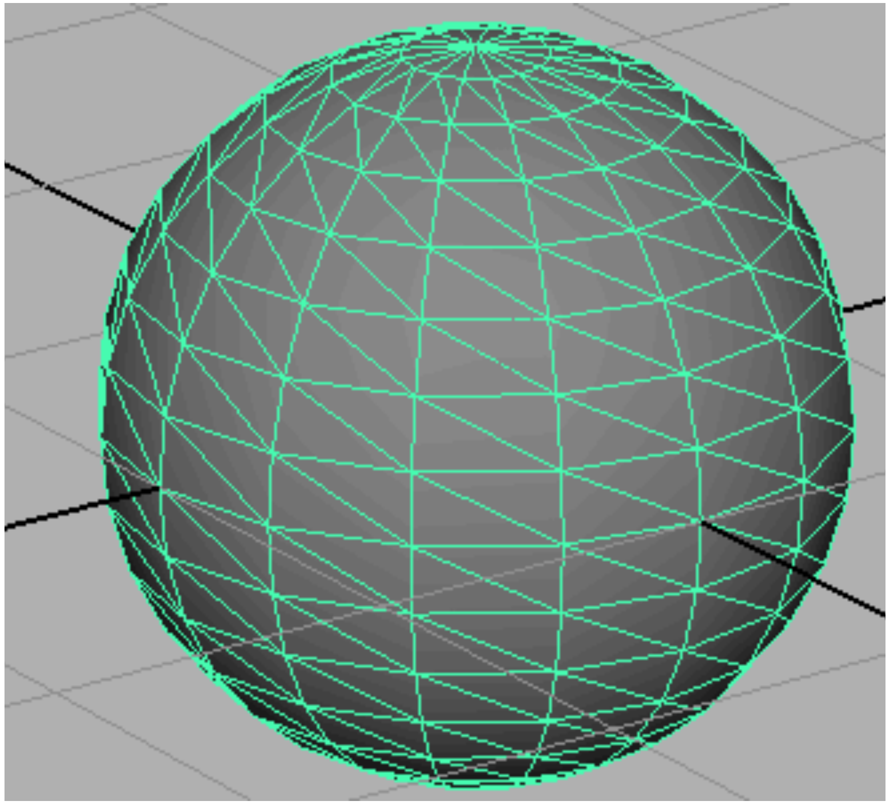
Simplest closed shape that may be defined by straight edges.

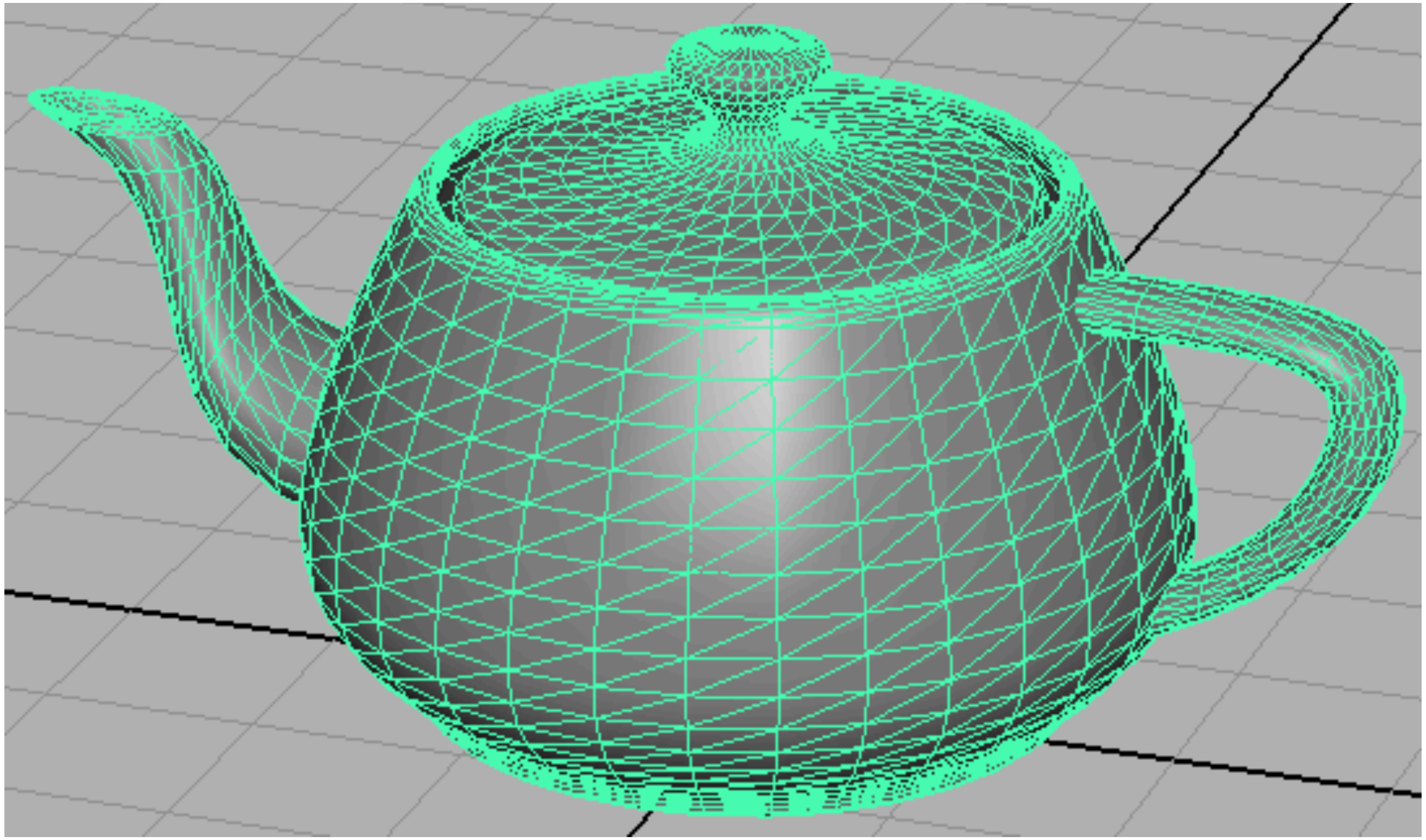
With enough triangles, you can make anything.



A cube whose faces are made up of triangles. This is a 3-D model of a cube -- model includes faces we can't see in this view.

A sphere whose faces are made up of triangles. With enough triangles, the curvature of the sphere can be made arbitrarily smooth.

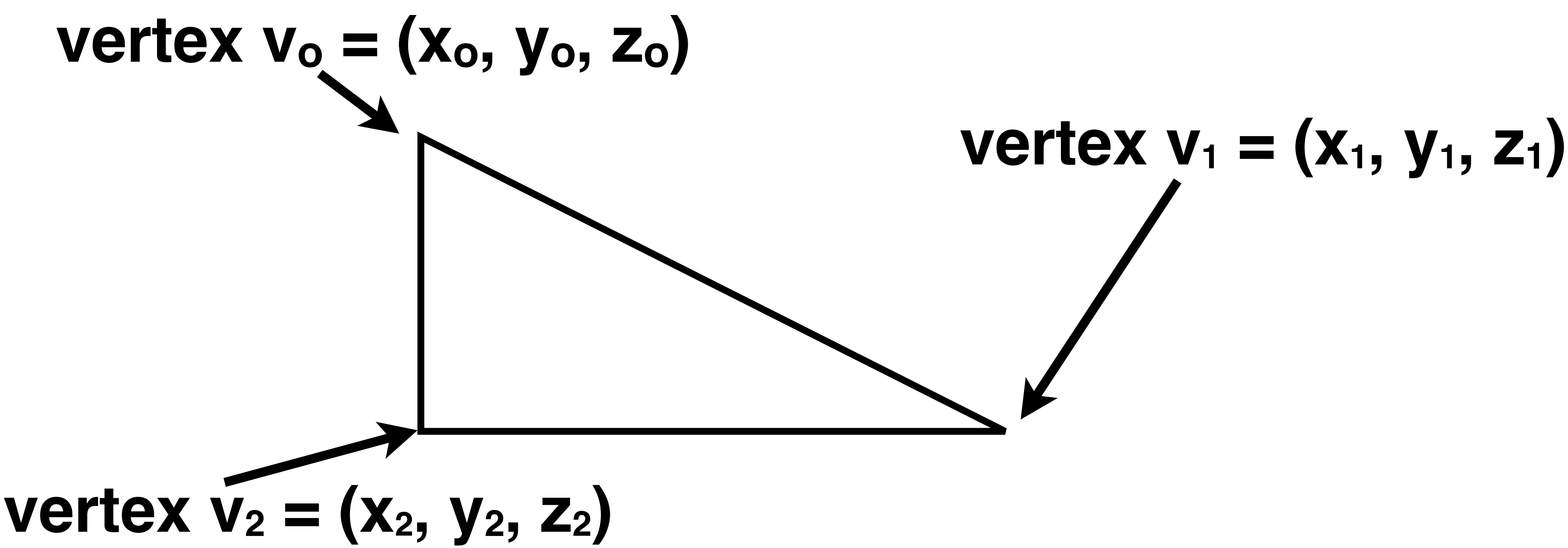




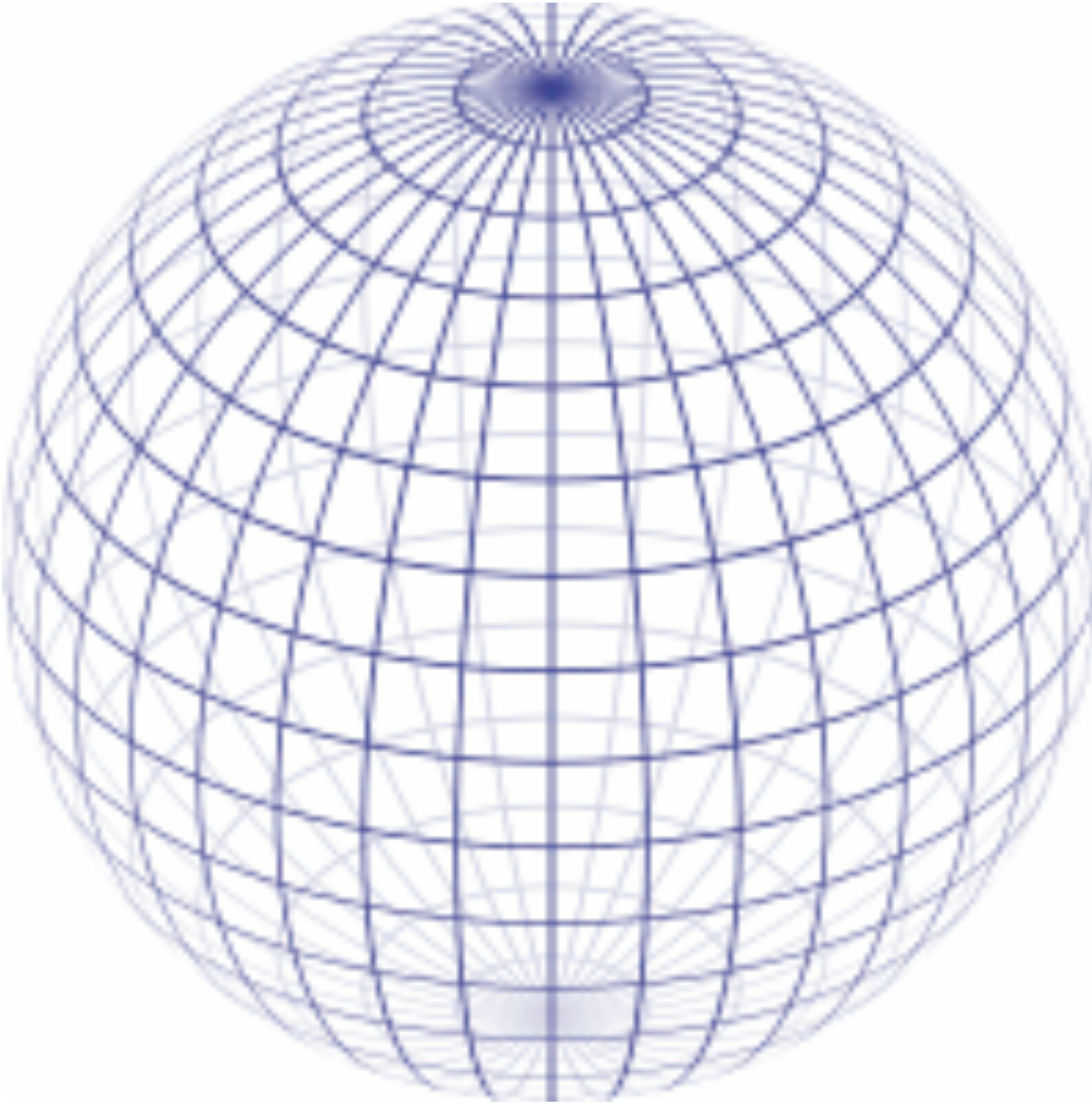
A teapot (famous object in computer graphics history). A "wire-frame" of triangles can capture the 3-D shape of complex, man-made objects.

# Triangle defined by 3 vertices

By transforming ($v' = f(v)$) all vertices in a 3-D object (like the teapot), you can move it in the 3-D world, change it's size, rotate it, etc.

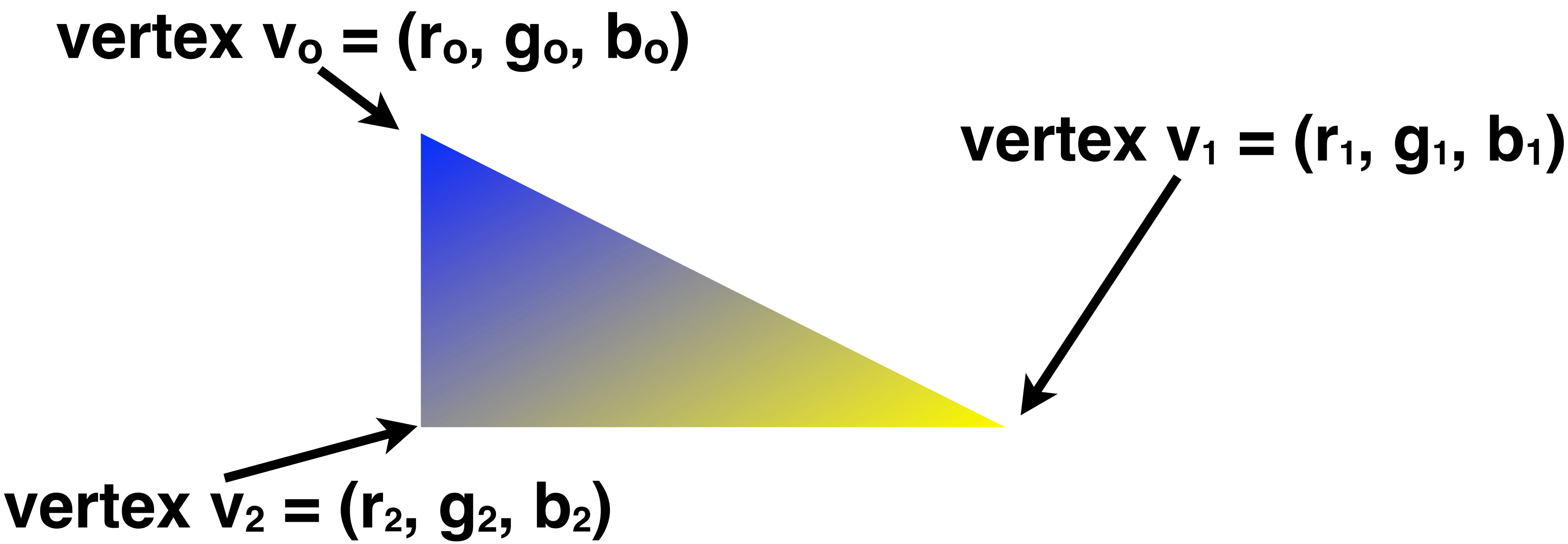



If a teapot has 10,000 triangles, need to transform 30,000 vertices to move it in a 3-D scene ... per frame!



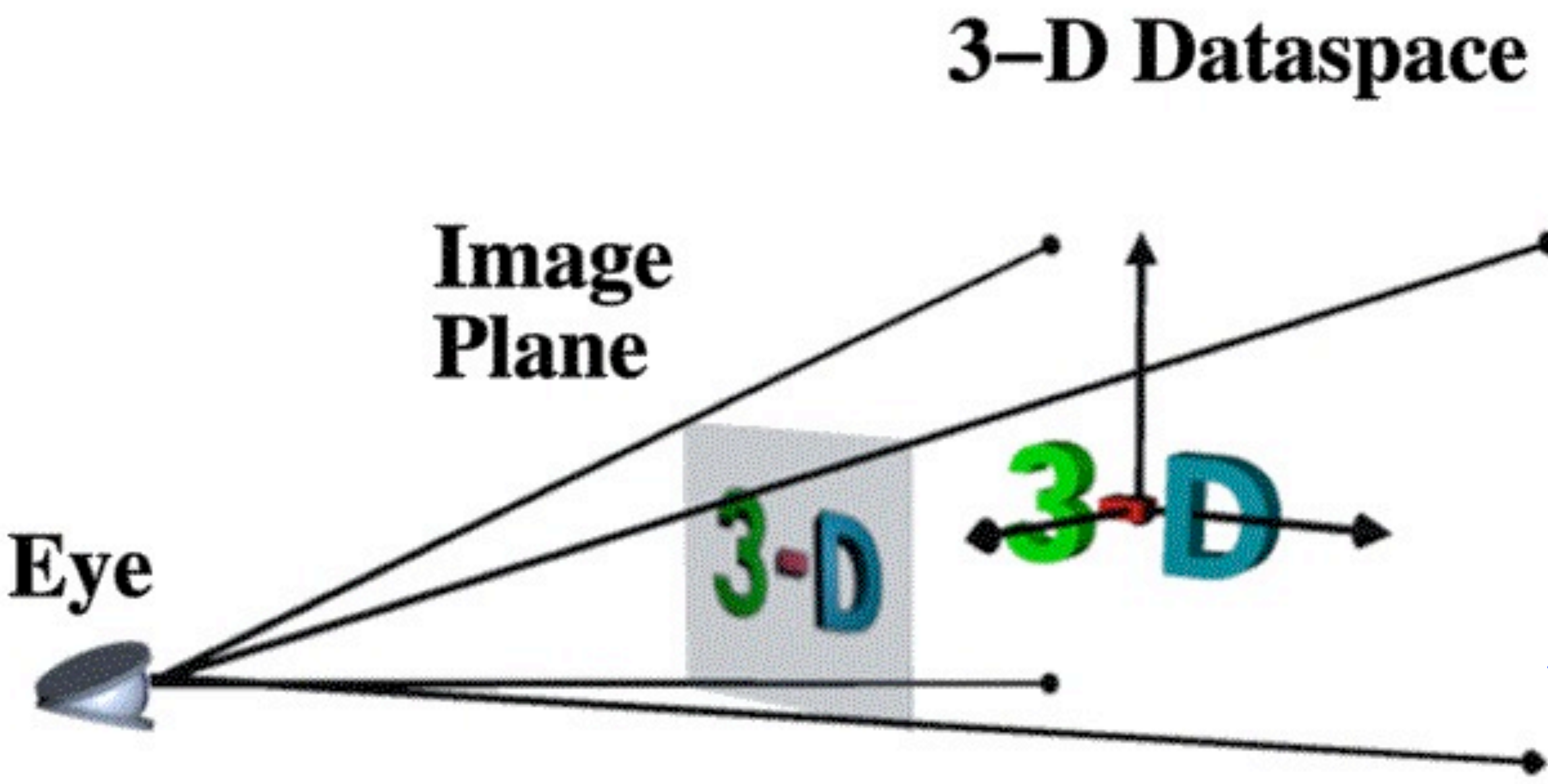
# Vertex can have color, lighting info ...

If vertices colors are different, this means that a smooth gradient of color washes across triangle.

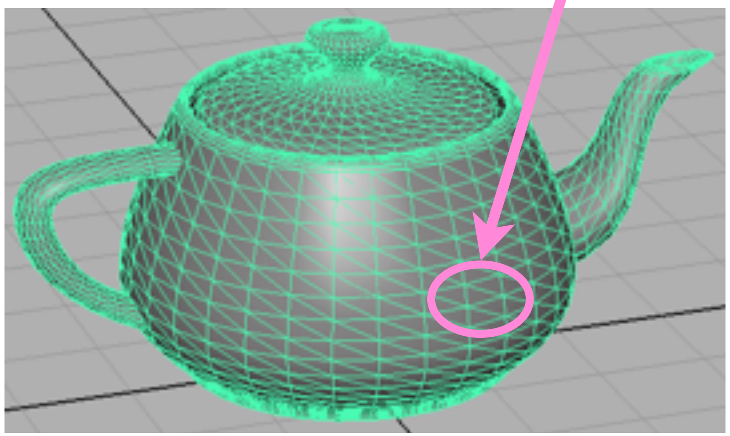



More realistic graphics models include light sources in the scene. Per-vertex information can carry information about how light hits the vertex.

# We see a 2-D window into the 3-D world



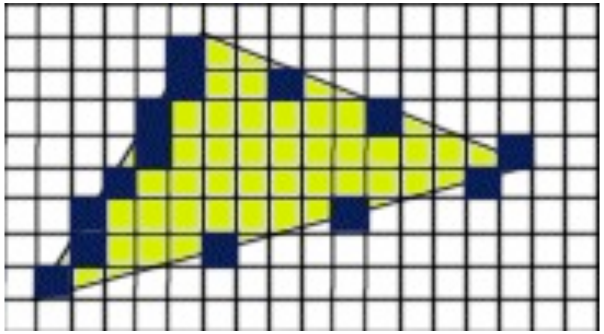

Let's follow one 3-D triangle.



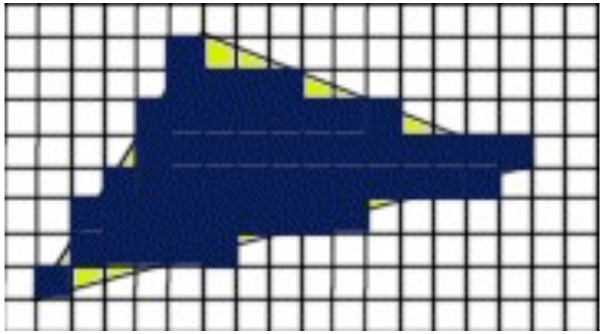
# From 3-d triangles to screen pixels

**First, project each 3-D triangle that might "face" the "eye" onto the image plane.**

Then, create "pixel fragments" on the boundary of the image plane triangle



Then, create "pixel fragments" to fill in the triangle (rasterization).



Why "pixel fragments"? A screen pixel color might depend on many triangles (example: a glass teapot).

# Process pixel fragment to "shade" it.

Algorithmic approach: Per-pixel computational model of metal and how light reflects off of it. Move teapot and what reflects off it changes.

# Process each fragment to "shade" it.

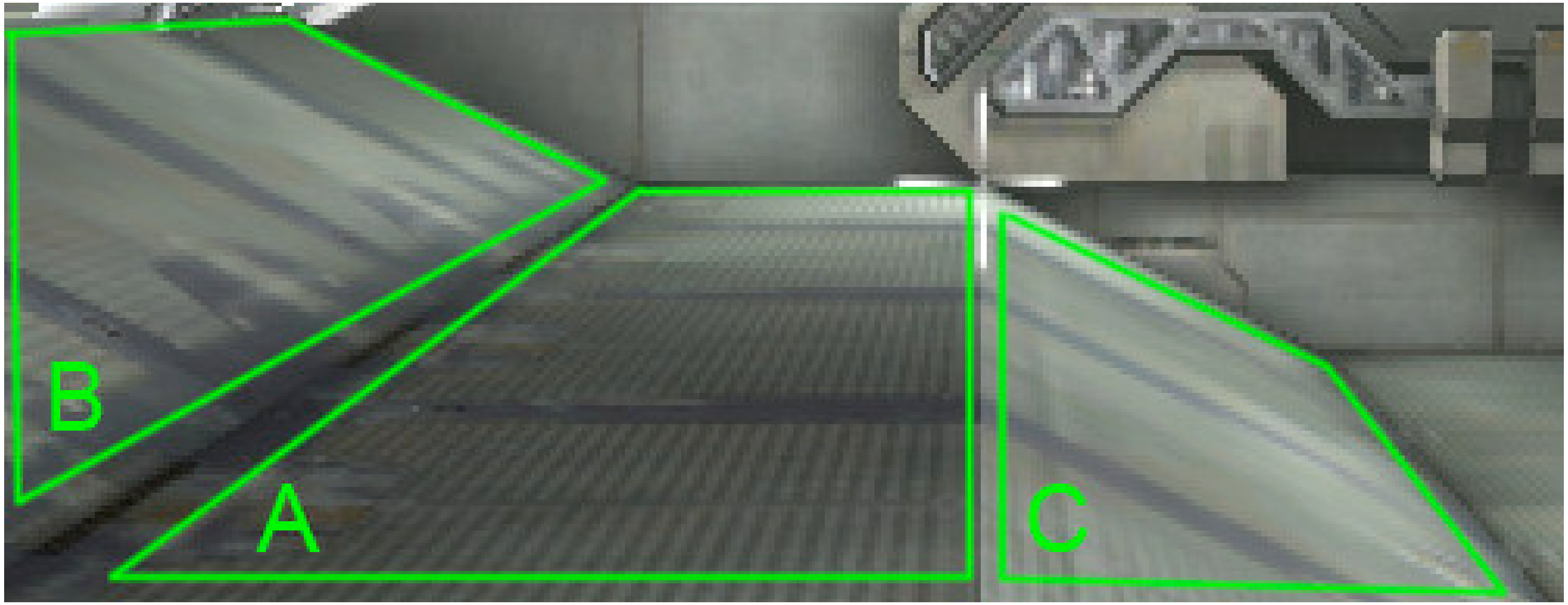
Artistic approach: Artist paints surface of teapot in Photoshop. We "map" this "texture" onto each pixel fragment during shading.

Final step: Output Merge. Assemble pixel fragments to make final 2-d image pixels.

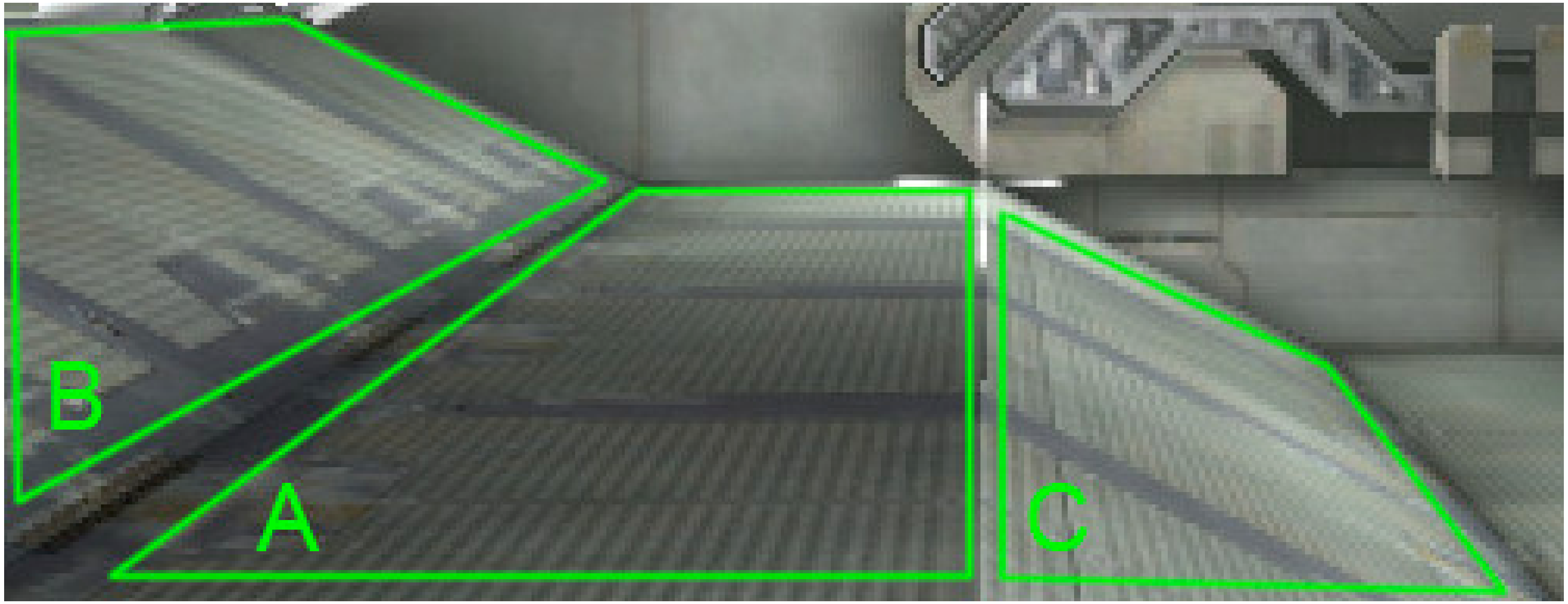
# Real-world texture maps: Bike decals


# Applying texture maps: Quality matters

"Good" algorithm. B and C look blurry.




"Better" algorithm. B and C are detailed.



# Putting it All Together ...

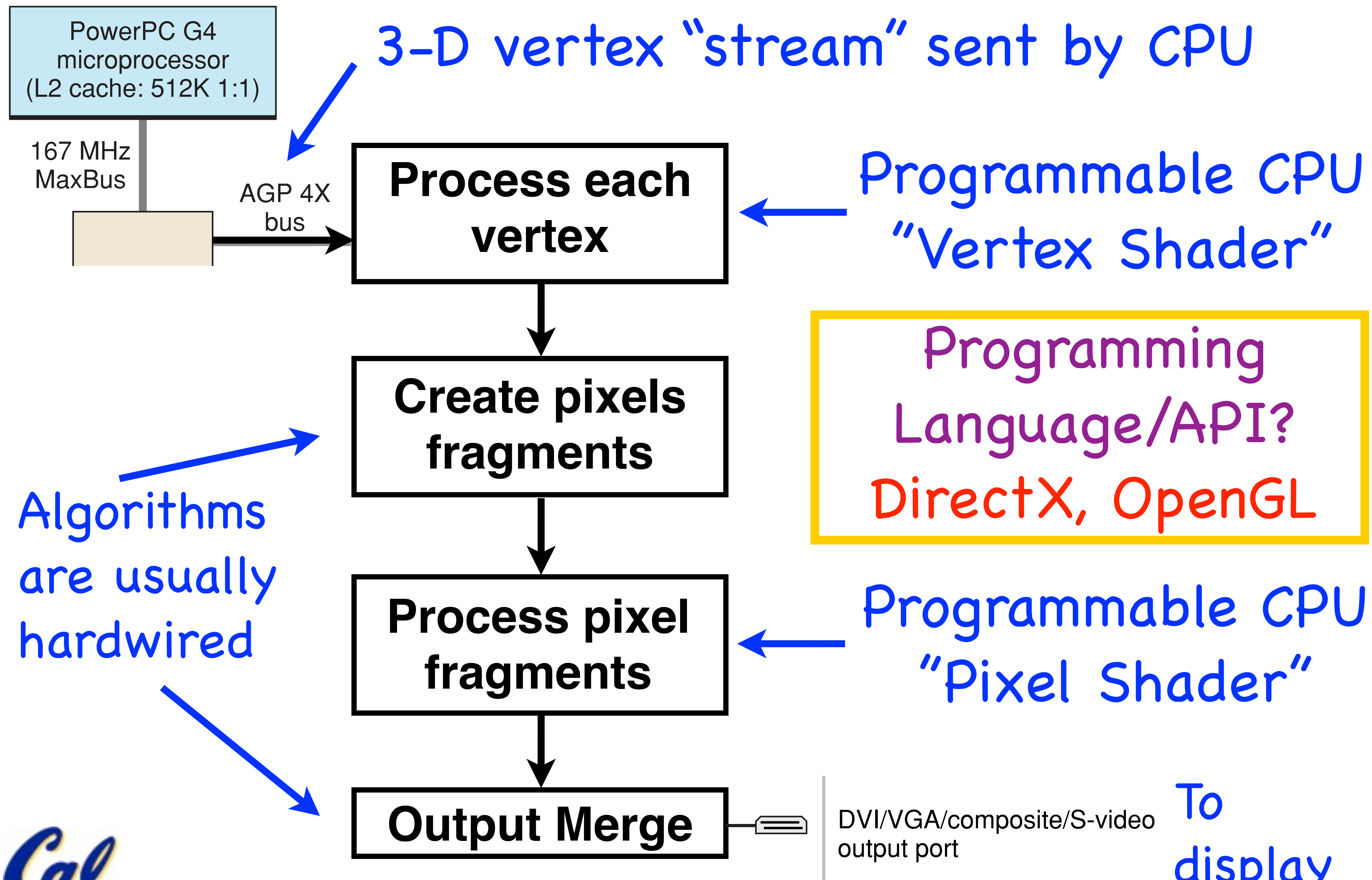
Luxo, Jr: Short movie made by Pixar, shown at SIGGRAPH in 1986.

First Academy Award given to a computer graphics movie.

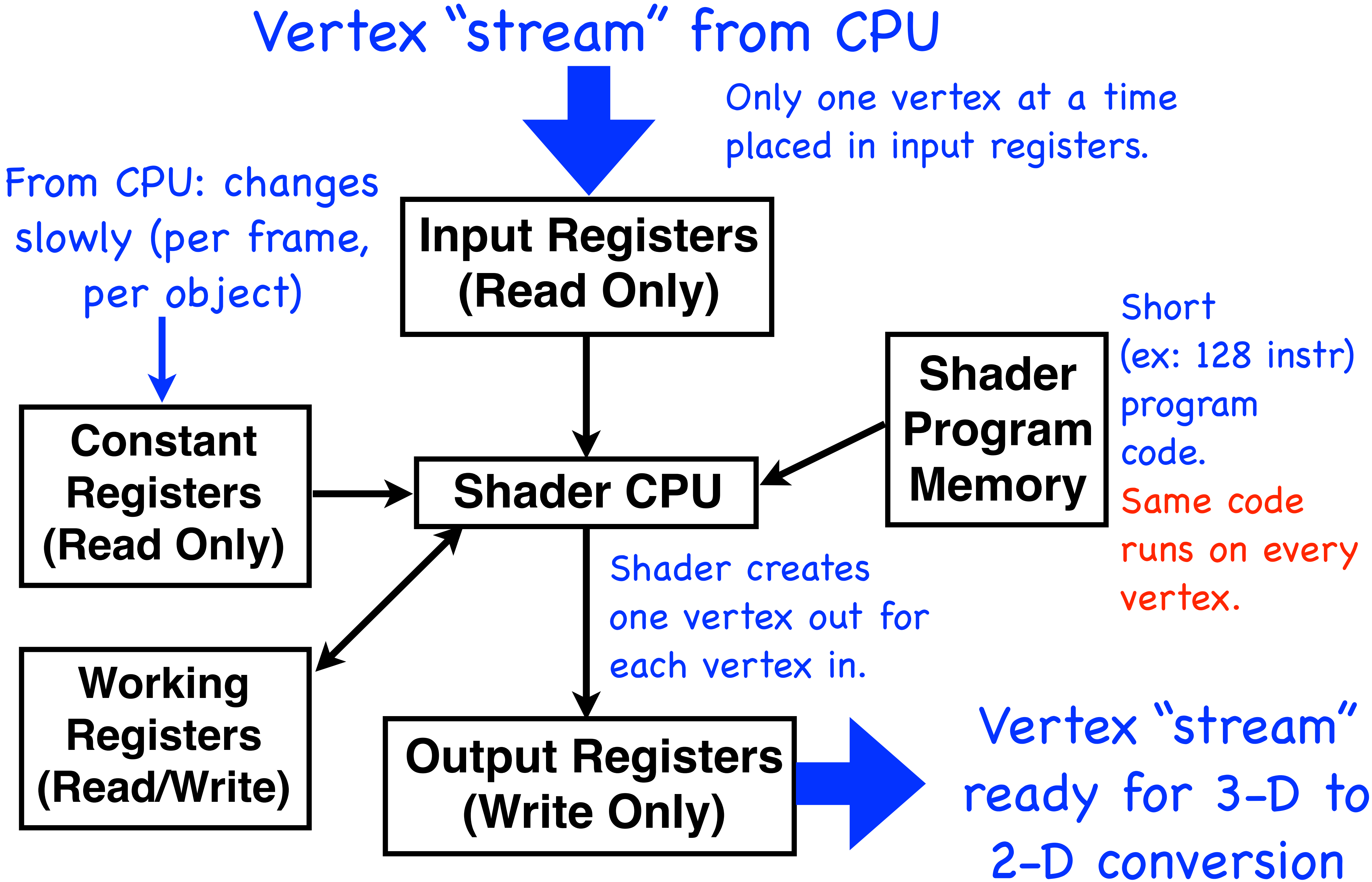
# Graphics Acceleration

**Next:** **Back to architecture ...**

# The graphics pipeline in hardware (2004)



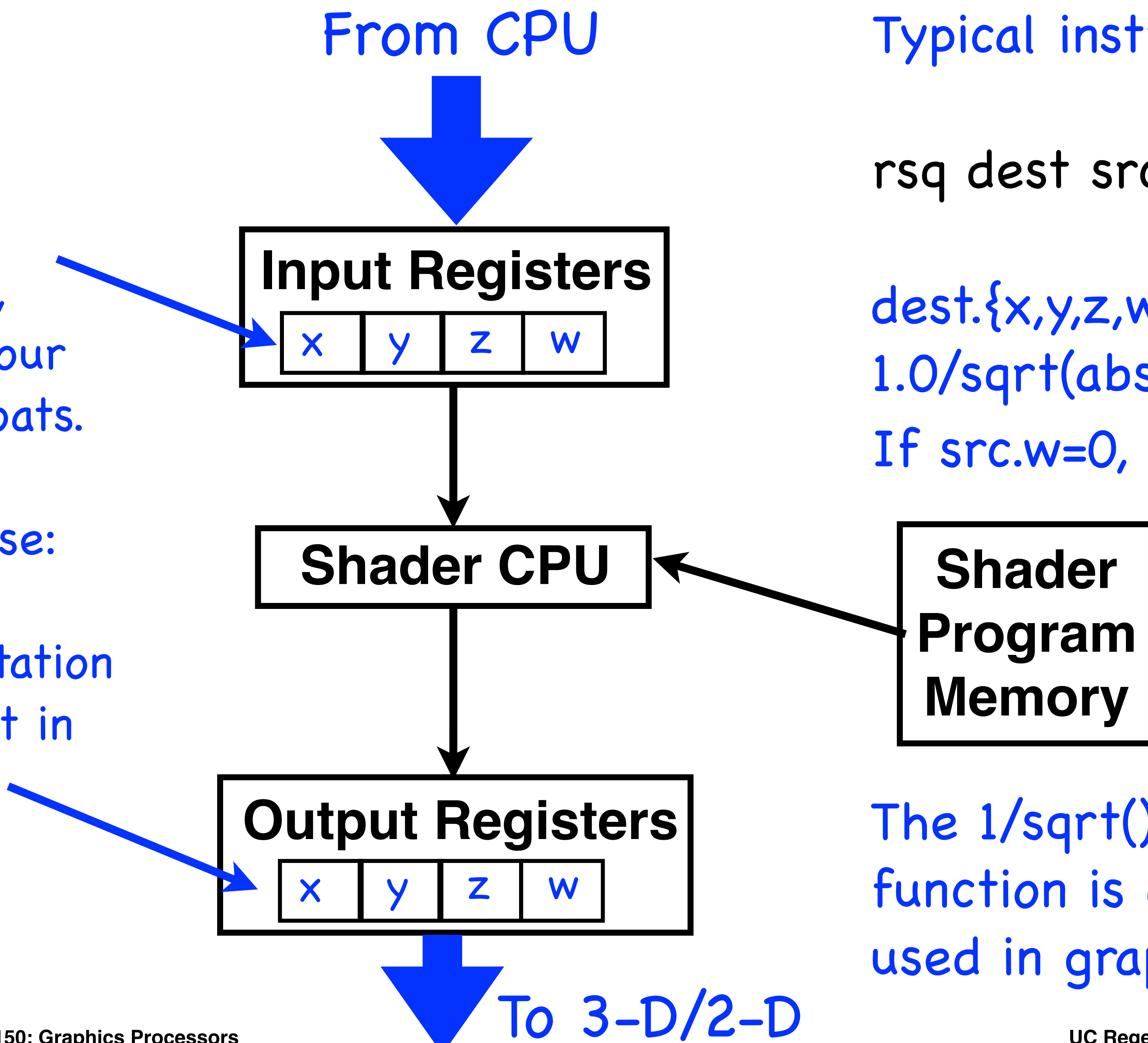
# Vertex Shader: A "stream processor"



# Optimized instructions and data formats

128-bit registers, holding four 32-bit floats.

Typical use: (x,y,z,w) representation of a point in 3-D space.




Typical instruction:

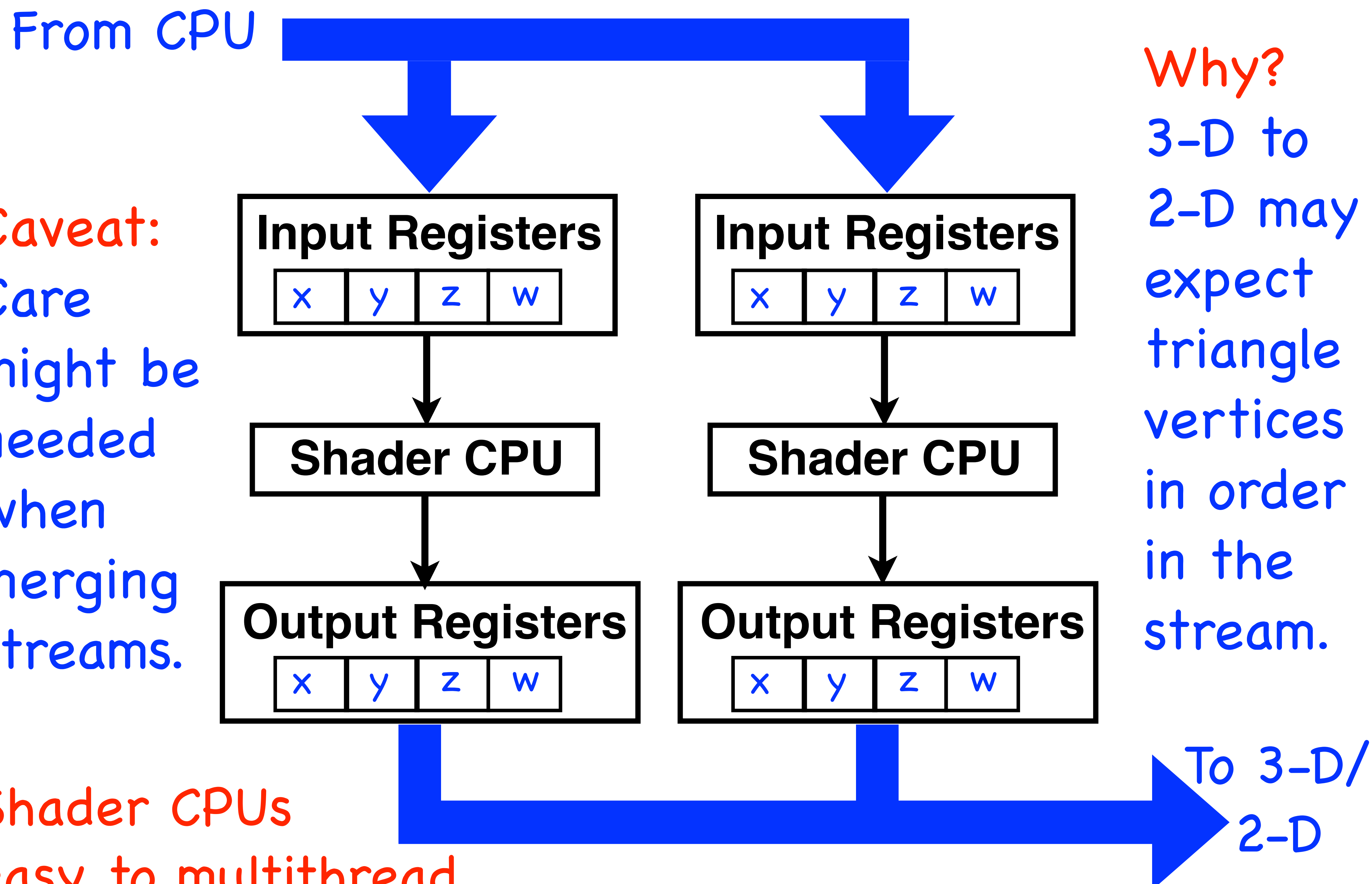
rsq dest src

dest.{x,y,z,w} = 1.0/sqrt(abs(src.w)).

If src.w=0, dest ∞.

The 1/sqrt() function is often used in graphics.

# Easy to parallelize: Vertices independent




Caveat: Care might be needed when merging streams.

Why? 3-D to 2-D may expect triangle vertices in order in the stream.

Shader CPUs easy to multithread.

# Pixel shader specializations ...

PowerPC G4 microprocessor (L2 cache: 512K 1:1)

167 MHz MaxBus

AGP 4X bus

Process each vertex

Create pixels fragments

"Pixel Shader" CPU →

Process pixel fragments

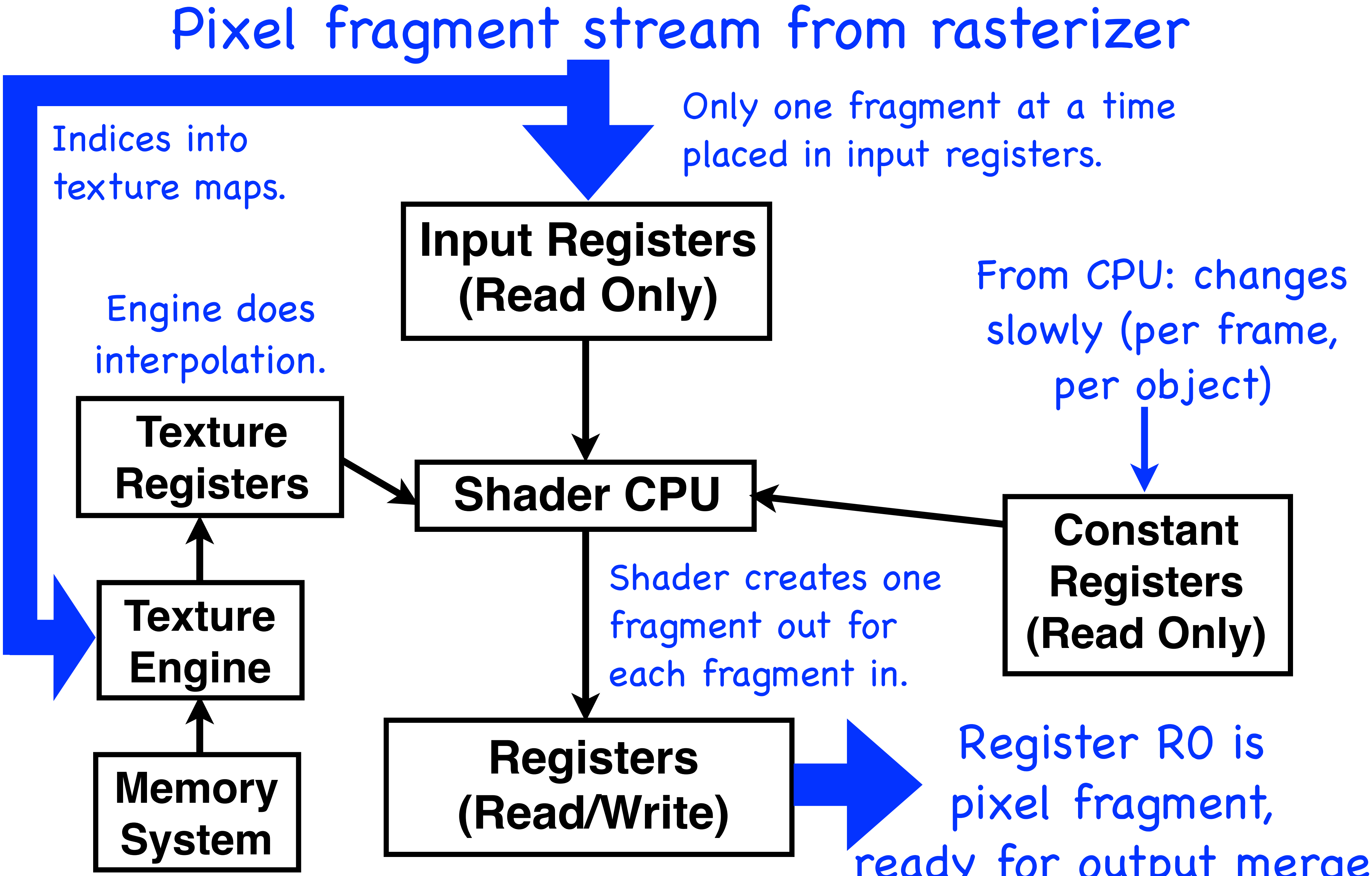
Output Merge

DVI/VGA/composite/S-video output port

Texture maps (look-up tables) play a key role.

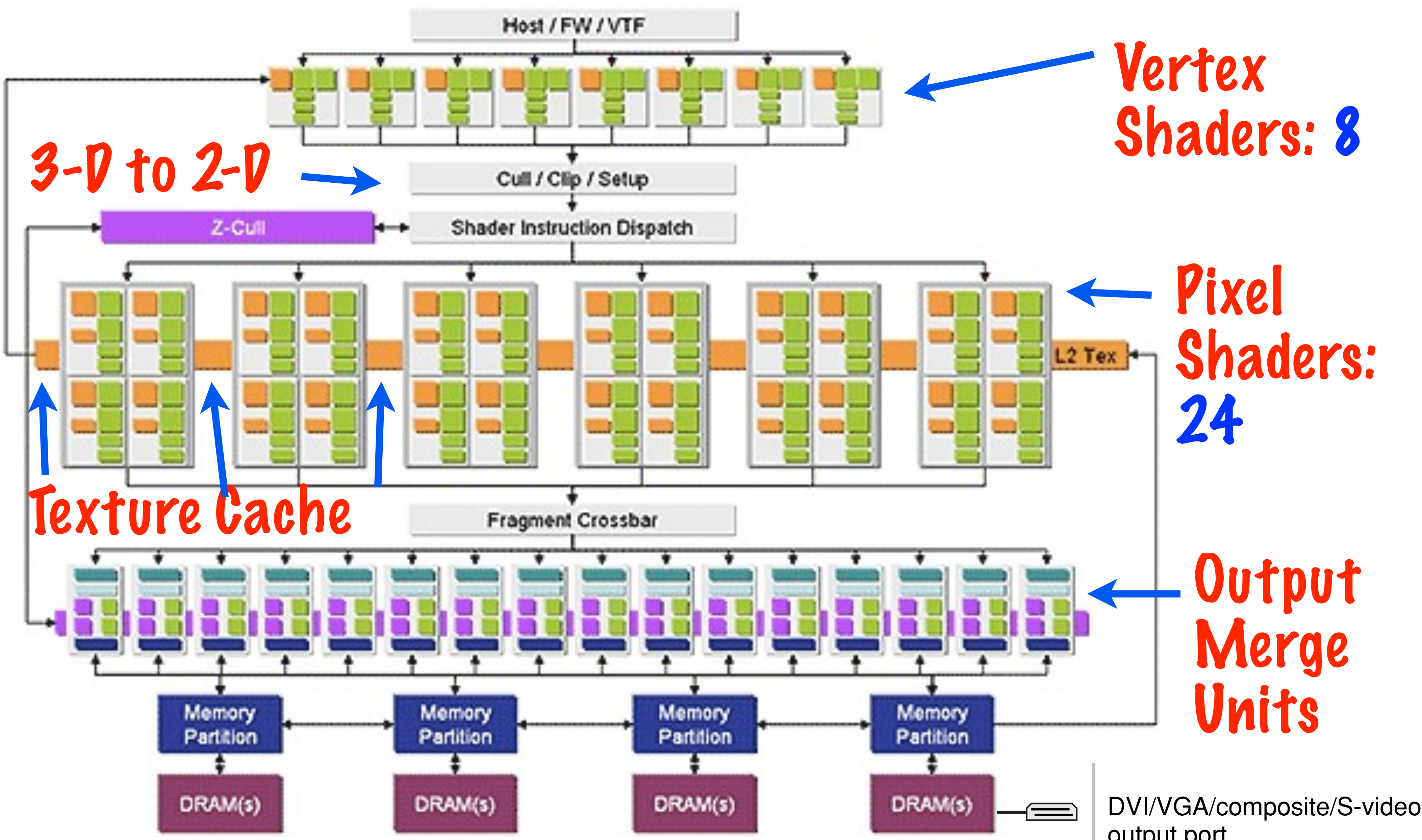
Pixel shader needs fast access to the map of Europe on teapot (via graphics card RAM).

# Pixel Shader: Stream processor + Memory



# Example (2006): Nvidia GeForce 7900

278 Million Transistors, 650 MHz clock, 90 nm process



# Break Time ...

Play

**Next:** **Unified architectures**

Basic idea: Replace specialized logic (vertex shader, pixel shader, hardwired algorithms) with many copies of one unified CPU design.

# Unified Architectures

Consequence: You no longer "see" the graphics pipeline when you look at the architecture block diagram.

Designed for: DirectX 10 (Microsoft Vista), and new non-graphics markets for GPUs.

# DirectX 10 (Vista): Towards Shader Unity

**Earlier APIs:** Pixel and Vertex CPUs very different ...

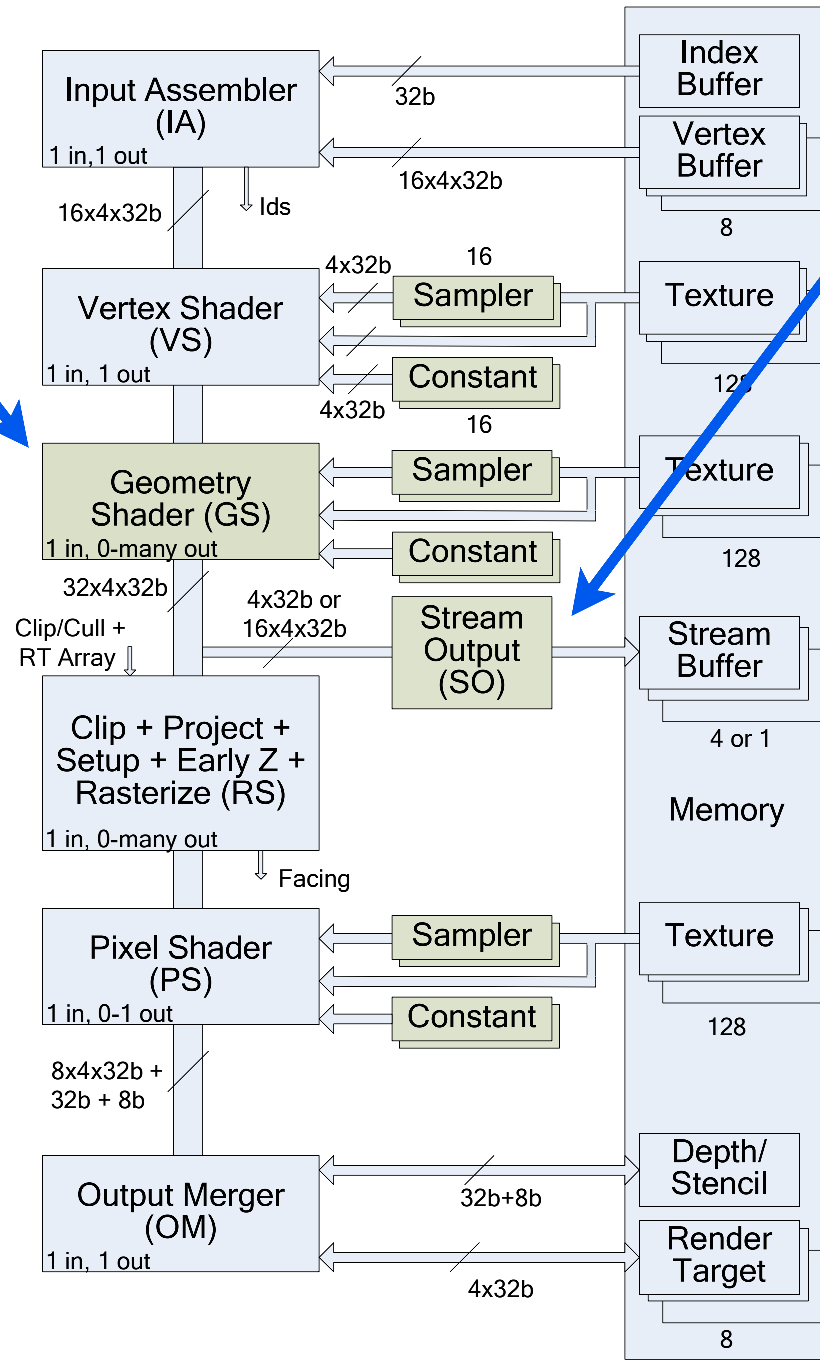
| Feature | 1.1 2001 | 2.0 2002 | 3.0 2004† | 4.0 2006 |
|---|---|---|---|---|
| instruction slots | 128 | 256 | ≥512 | ≥64K |
| | 4+8‡ | 32+64‡ | ≥512 | |
| constant registers | ≥96 | ≥256 | ≥256 | 16x4096 |
| | 8 | 32 | 224 | |
| tmp registers | 12 | 12 | 32 | 4096 |
| | 2 | 12 | 32 | |
| input registers | 16 | 16 | 16 | 16 |
| | 4+2§ | 8+2§ | 10 | 32 |
| render targets | 1 | 4 | 4 | 8 |
| samplers | 8 | 16 | 16 | 16 |
| textures | | | 4 | 128 |
| | 8 | 16 | 16 | |
| 2D tex size | | | 2Kx2K | 8Kx8K |
| integer ops | | | | ✓ |
| load op | | | | ✓ |
| sample offsets | | | | ✓ |
| transcendental ops | ✓ | ✓ | ✓ | ✓ |
| | | ✓ | ✓ | |
| derivative op | | | ✓ | ✓ |
| flow control | | static | stat/dyn | dynamic |
| | | | stat/dyn | |

**Table 1:** Shader model feature comparison summary.

**DirectX 10:** Many specs are identical for Pixel and Vertex CPUs

# DirectX 10 : New Pipeline Features ...

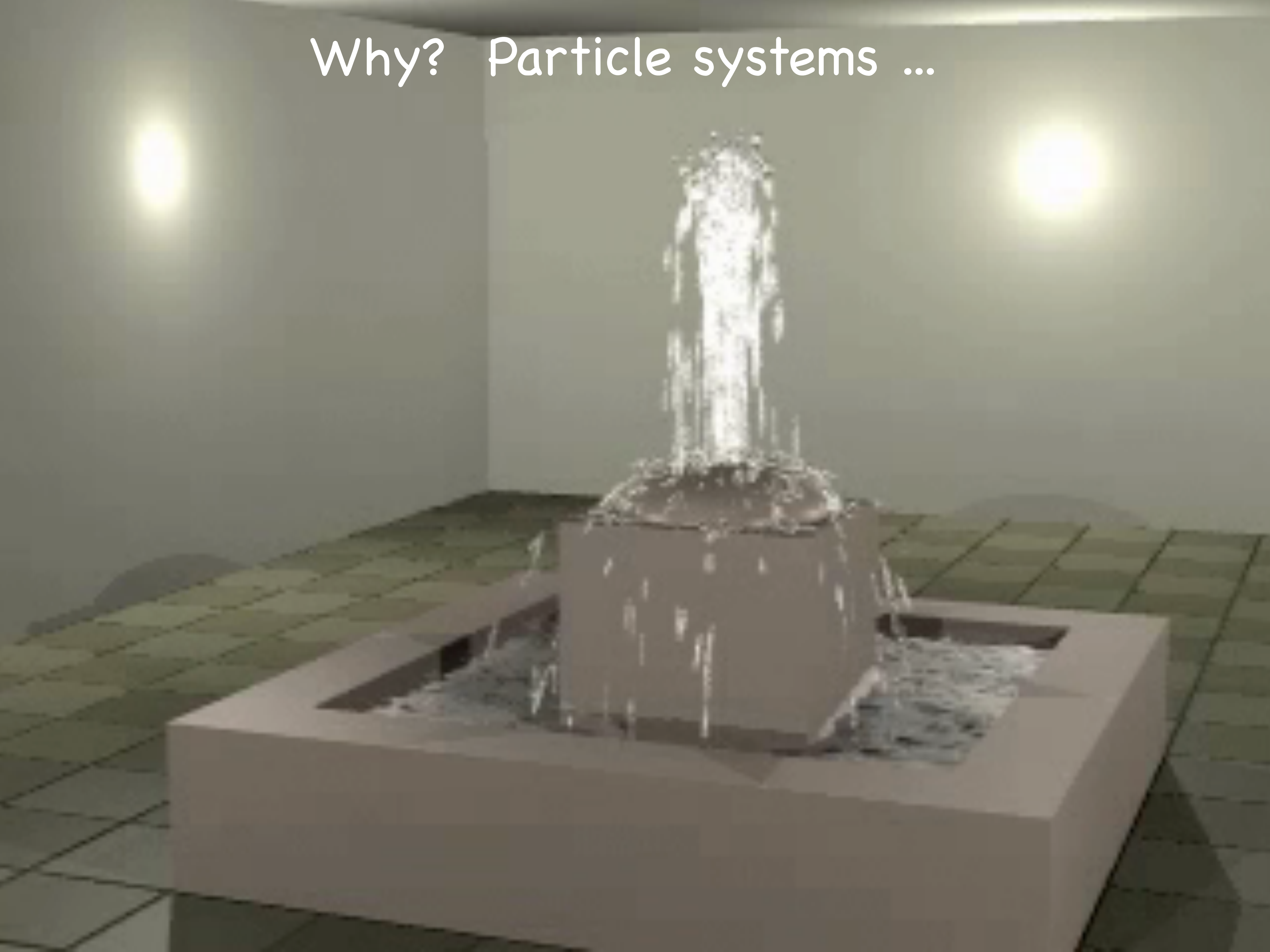
**Geometry Shader:** Lets a shader program create new triangles.



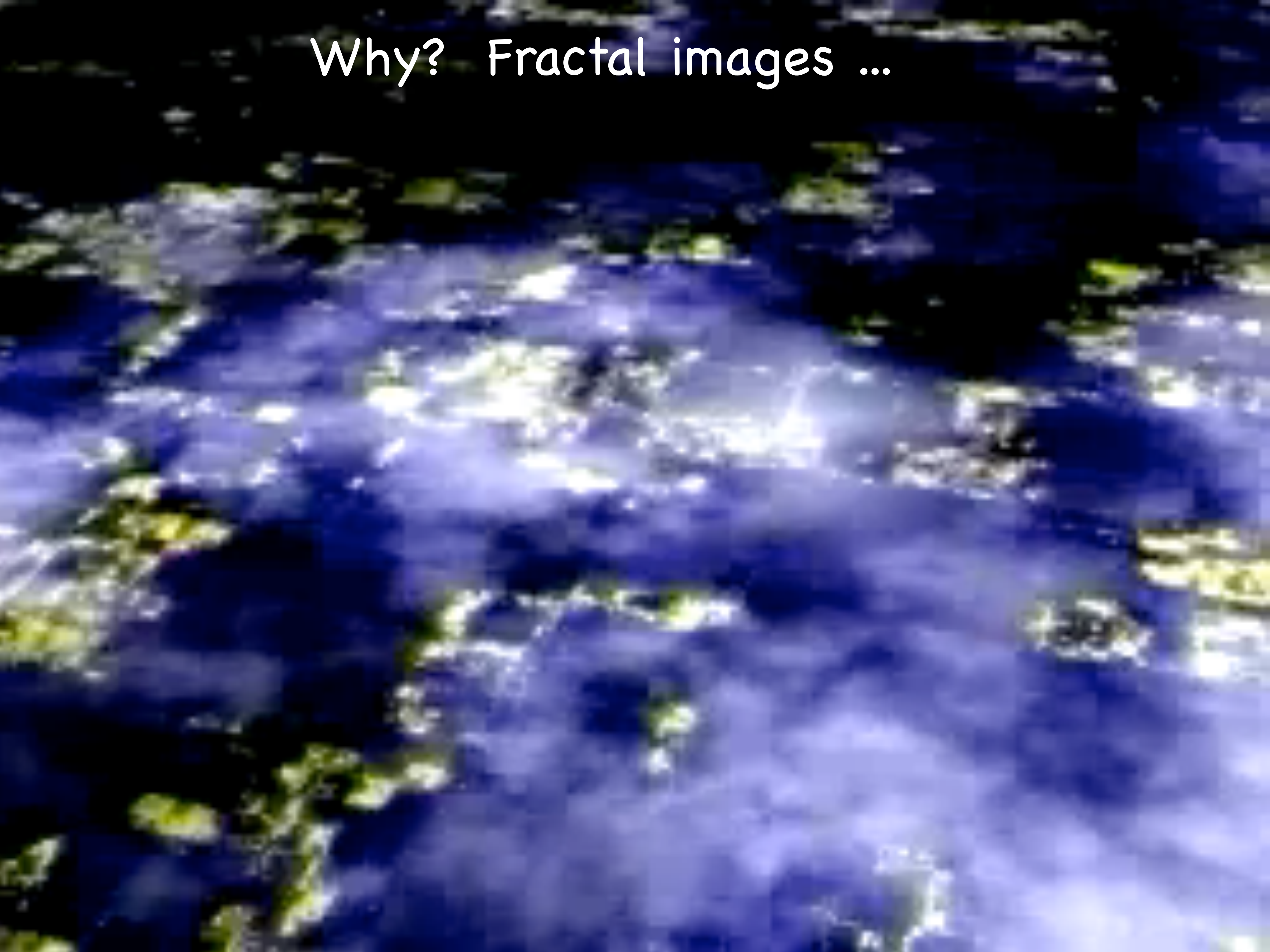

**Stream Output:** Lets vertex stream recirculate through shaders many times ... (and also, back to CPU)

Also: Shader CPUs are more like RISC machines in many ways.

# Why? Particle systems …



# Why?  Fractal images …

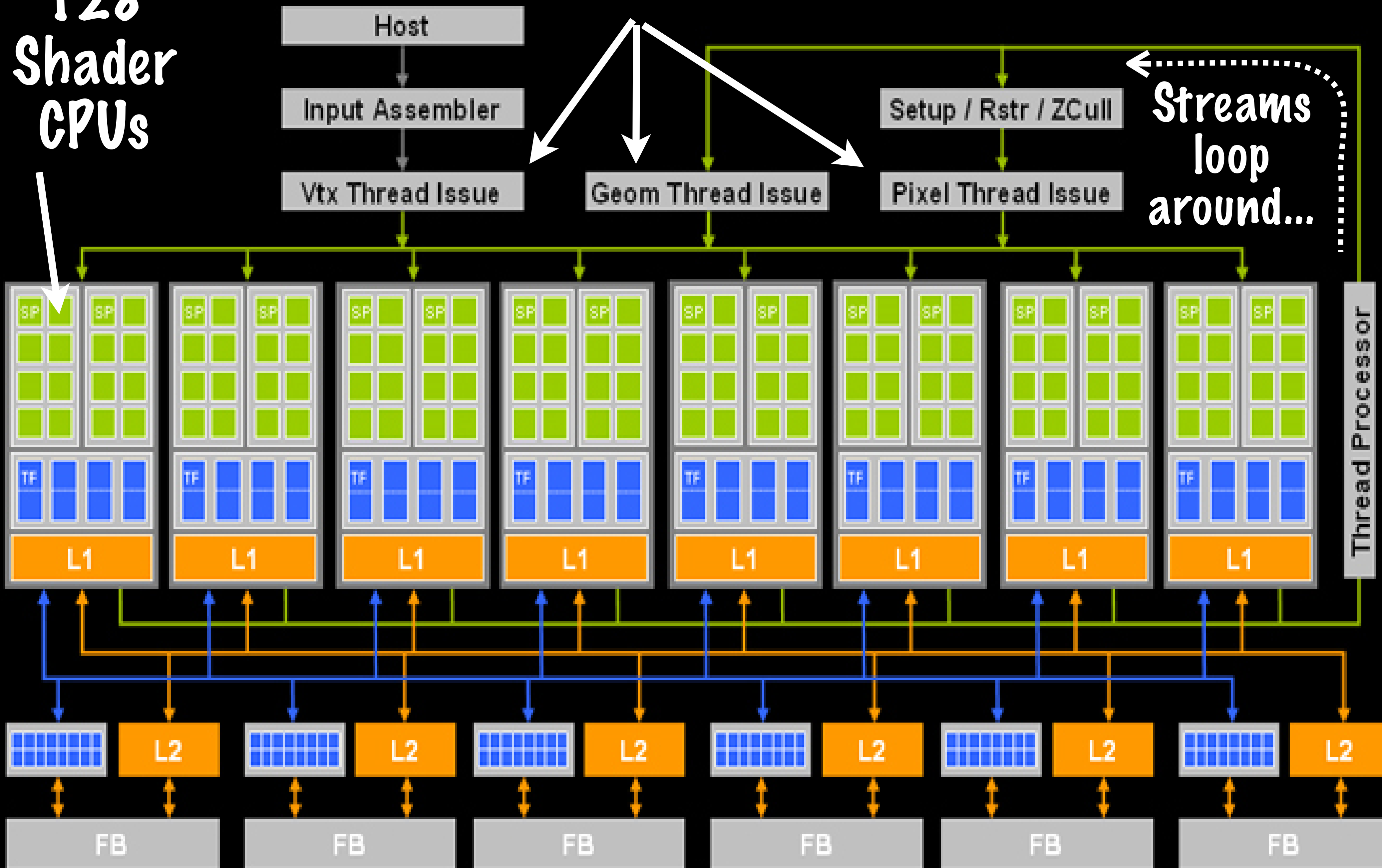


# NVidia 8800: Unified GPU, announced Fall 2006

Thread processor sets shader type of each CPU

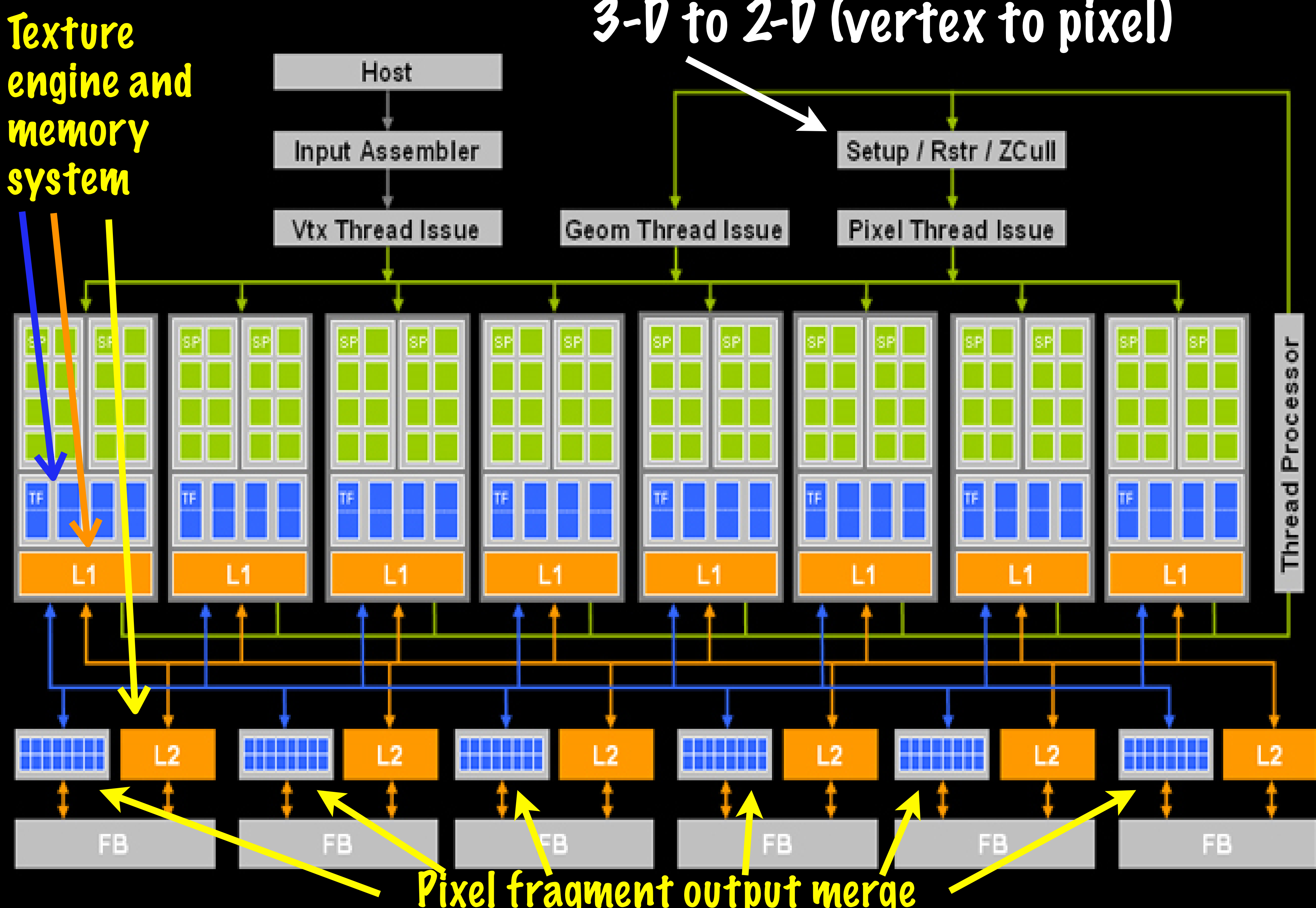
128 Shader CPUs

Streams loop around...




1.35 GHz Shader CPU Clock, 575 MHz core clock

# Graphics-centric functionality ...

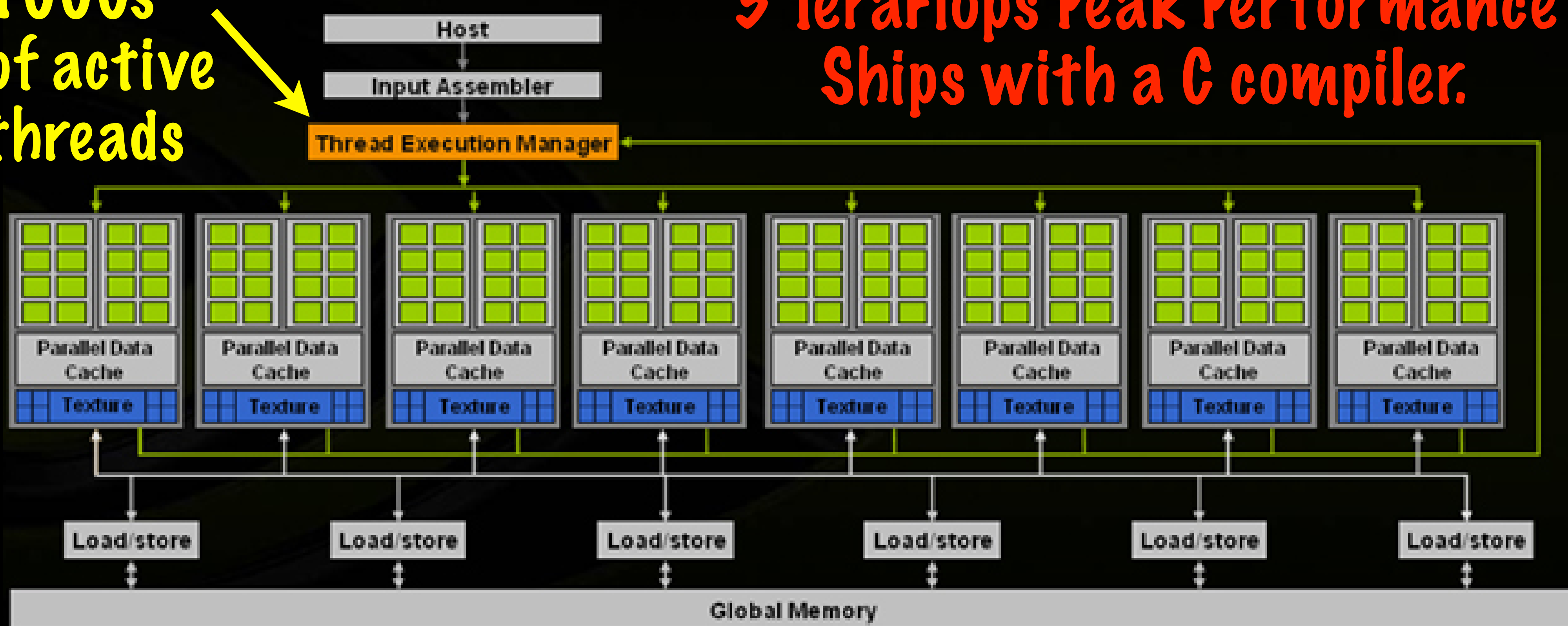


# Can be reconfigured with graphics logic hidden ...

128 scalar 1.35 GHz processors: Integer ALU, dual-issue single-precision IEEE floats.

1000s of active threads

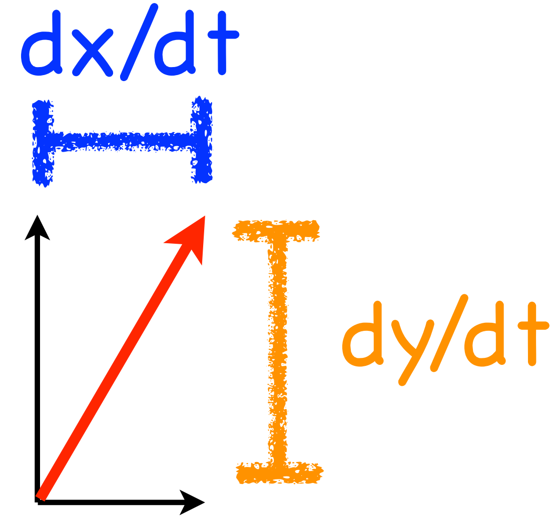
3 TeraFlops Peak Performance
Ships with a C compiler.

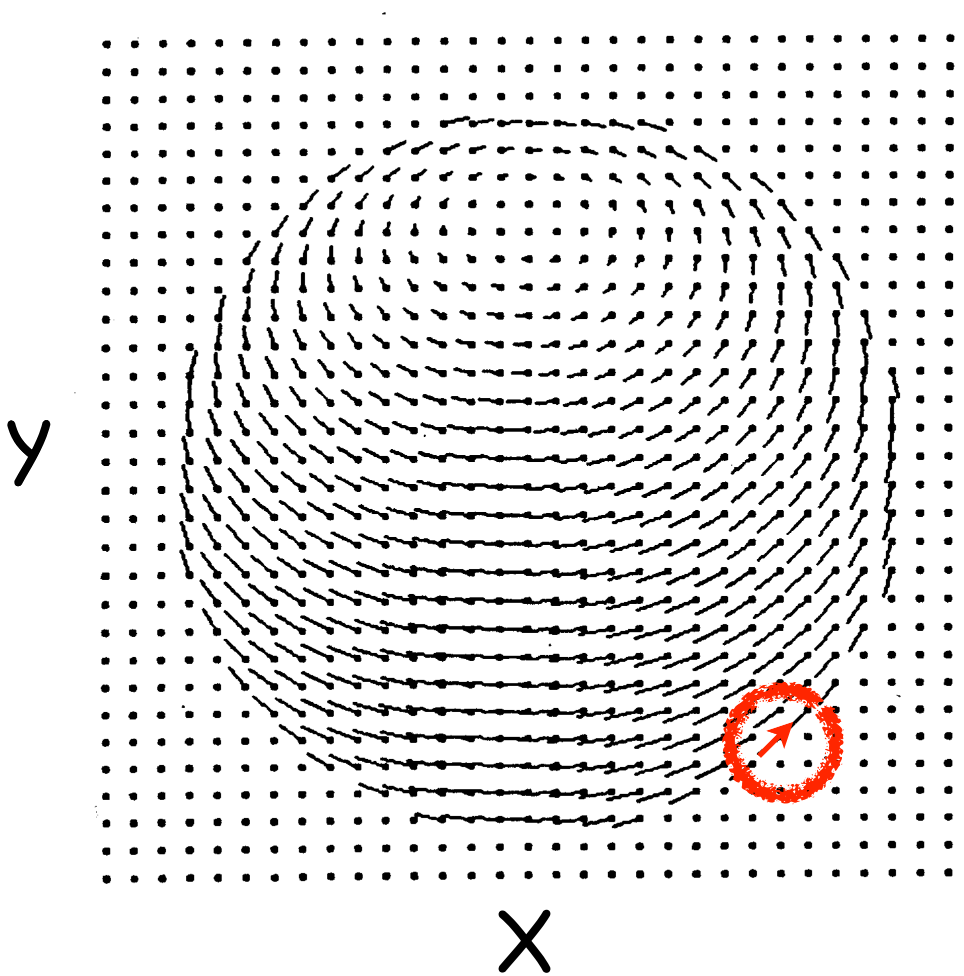


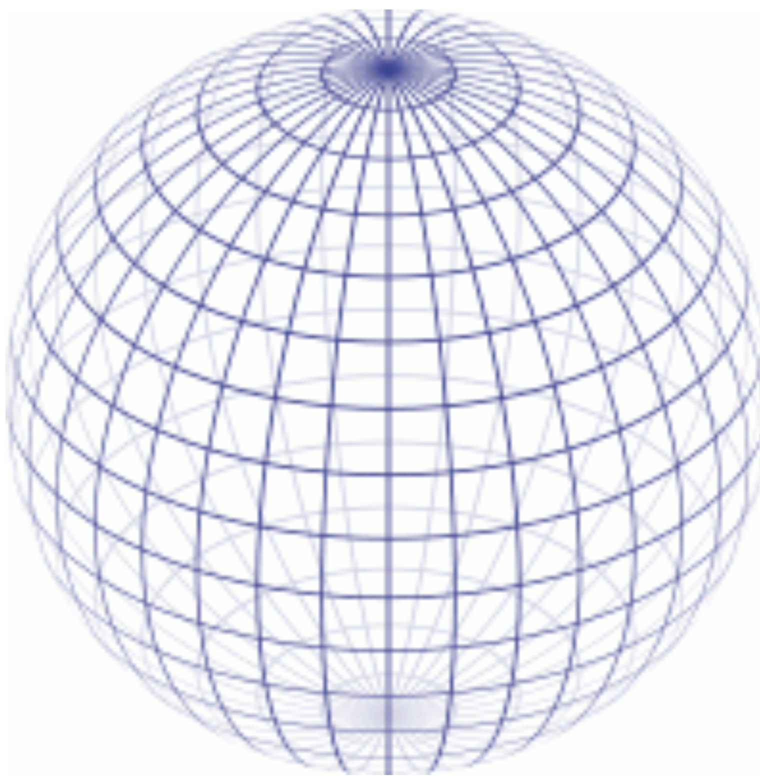

Texture system set up to look like a conventional memory system (768MB GDDR3, 86 GB/s)

# Optic Flow (Computer Vision)

Notate a movie with arrows to show speed and direction.



# Optical Disc

[illegible]

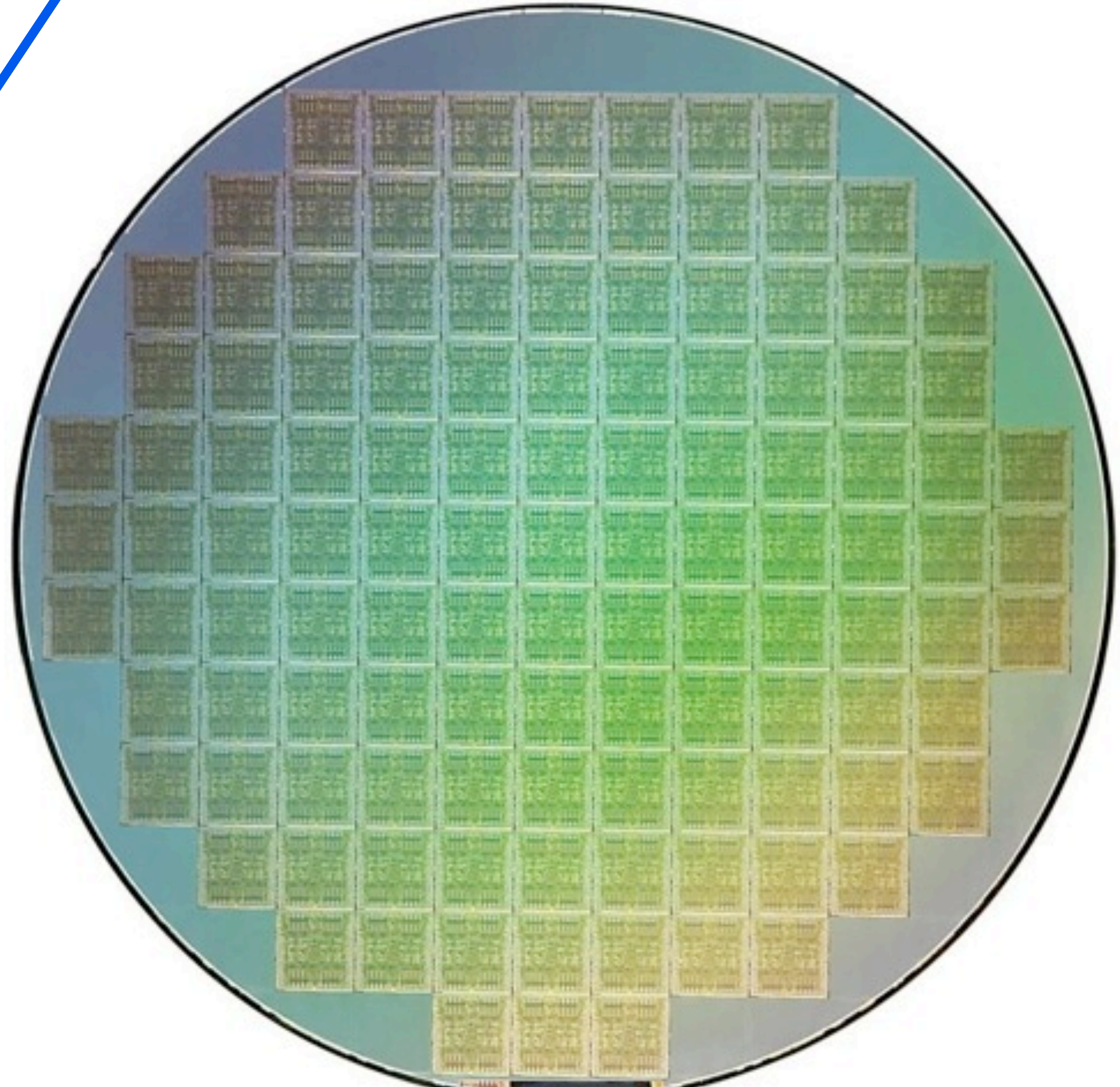
A big die. Many chips will not work (low yield). Low profits.

## Chip Facts

90nm process

681M Transistors

80 die/wafer (pre-testing)



## Design Facts

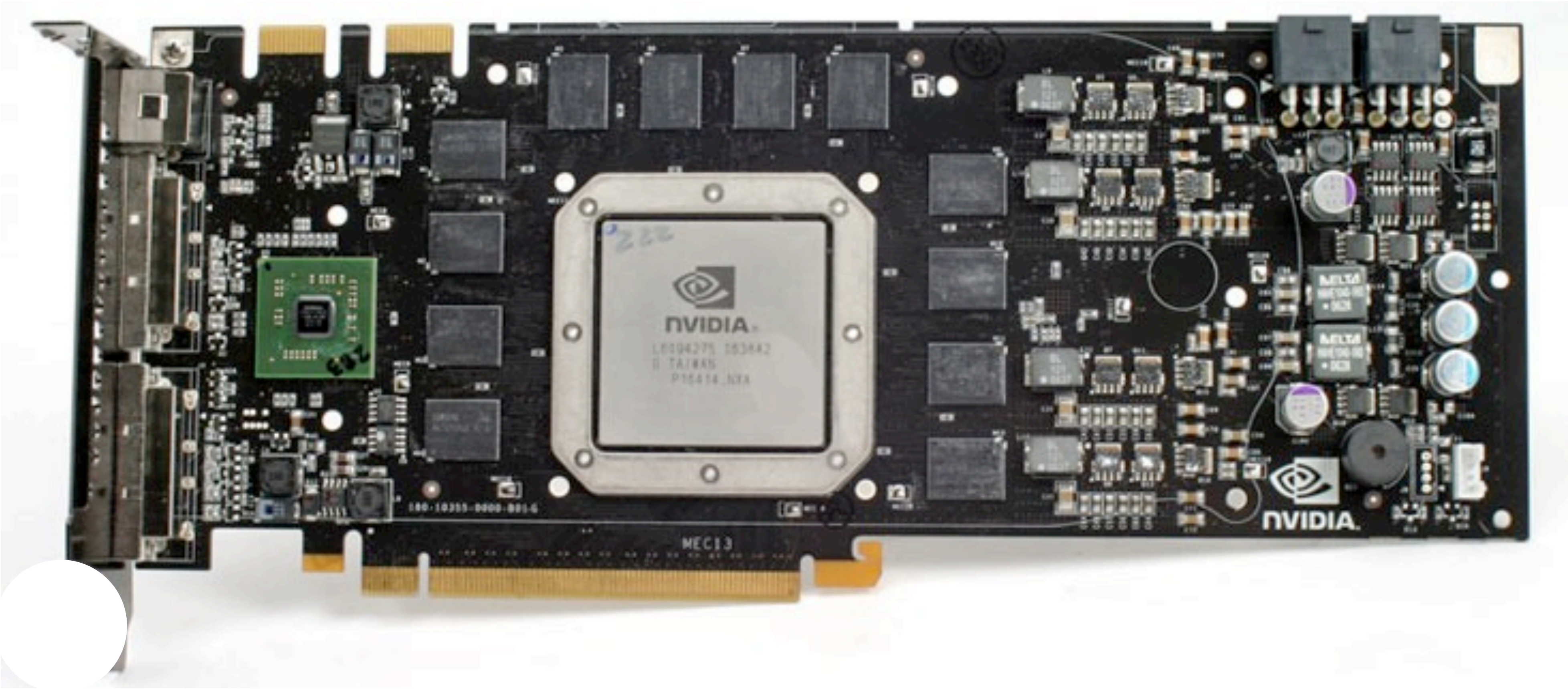
4 year design cycle

$400 Million design budget

600 person-years: 10 people at start, 300 at peak

# GeForce 8800 GTX Card: $599 List Price

PCI-Express 16X Card - 2 Aux Power Plugs!




185 Watts Thermal Design Point (TDP) -- TDP is a “real-world” maximum power spec.

# Some products are “loss-leaders”

Breakthrough product creates “free” publicity you can’t buy.


(1) Hope: when chip “shrinks” to 65nm fab process, die will be smaller, yields will improve, profits will rise.

(2) Simpler versions of the design will be made to create an entire product family, some very profitable.

“We tape out a chip a month”, NVidia CEO quote.

# And it happened! 2008 nVidia products

GTX 280

Price similar to 8800, stream CPU count > 2X.

| | GTX 280 | GTX 260 | 9800 GX2 | 9800 GTX+ | 9800 GTX |
|---|---|---|---|---|---|
| Stream Processors | 240 | 192 | 256 | 128 | 128 |
| Texture Address / Filtering | 80 / 80 | 64 / 64 | 128 / 128 | 64 / 64 | 64 / 64 |
| ROPs | 32 | 28 | 32 | 16 | 16 |
| Core Clock | 602MHz | 576MHz | 600MHz | 738MHz | 675MHz |
| Shader Clock | 1296MHz | 1242MHz | 1500MHz | 1836MHz | 1690MHz |
| Memory Clock | 1107MHz | 999MHz | 1000MHz | 1100MHz | 1100MHz |
| Memory Bus Width | 512-bit | 448-bit | 256-bit x 2 | 256-bit | 256-bit |
| Frame Buffer | 1GB | 896MB | 1GB | 512MB | 512MB |
| Transistor Count | 1.4B | 1.4B | 1.5B | 754M | 754M |
| Manufacturing Process | TSMC 65nm | TSMC 65nm | TSMC 65nm | TSMC 55nm | TSMC 65nm |
| Price Point | $650 | $400 | $500 | $229 | $199 |

9800 GTX

Specs similar to 8800, card sells for $199.

# And again in 2012! GTX 680 -- "Kepler"

GTX 680

3X more effective CPUs as GTX 280, lower price point.

6X more CPUs as 8800, (from 2006).

| | GTX 680 | GTX 580 | GTX 560 Ti |
|---|---|---|---|
| Stream Processors | 1536 | 512 | 384 |
| Texture Units | 128 | 64 | 64 |
| ROPs | 32 | 48 | 32 |
| Core Clock | 1006MHz | 772MHz | 822MHz |
| Shader Clock | N/A | 1544MHz | 1644MHz |
| Boost Clock | 1058MHz | N/A | N/A |
| Memory Clock | 6.008GHz GDDR5 | 4.008GHz GDDR5 | 4.008GHz GDDR5 |
| Memory Bus Width | 256-bit | 384-bit | 256-bit |
| Frame Buffer | 2GB | 1.5GB | 1GB |
| FP64 | 1/24 FP32 | 1/8 FP32 | 1/12 FP32 |
| TDP | 195W | 244W | 170W |
| Transistor Count | 3.5B | 3B | 1.95B |
| Manufacturing Process | TSMC 28nm | TSMC 40nm | TSMC 40nm |
| Launch Price | $499 | $499 | $249 |

GTX 560 Ti

Specs better than GTX 280, sells for $249

# GTX 680

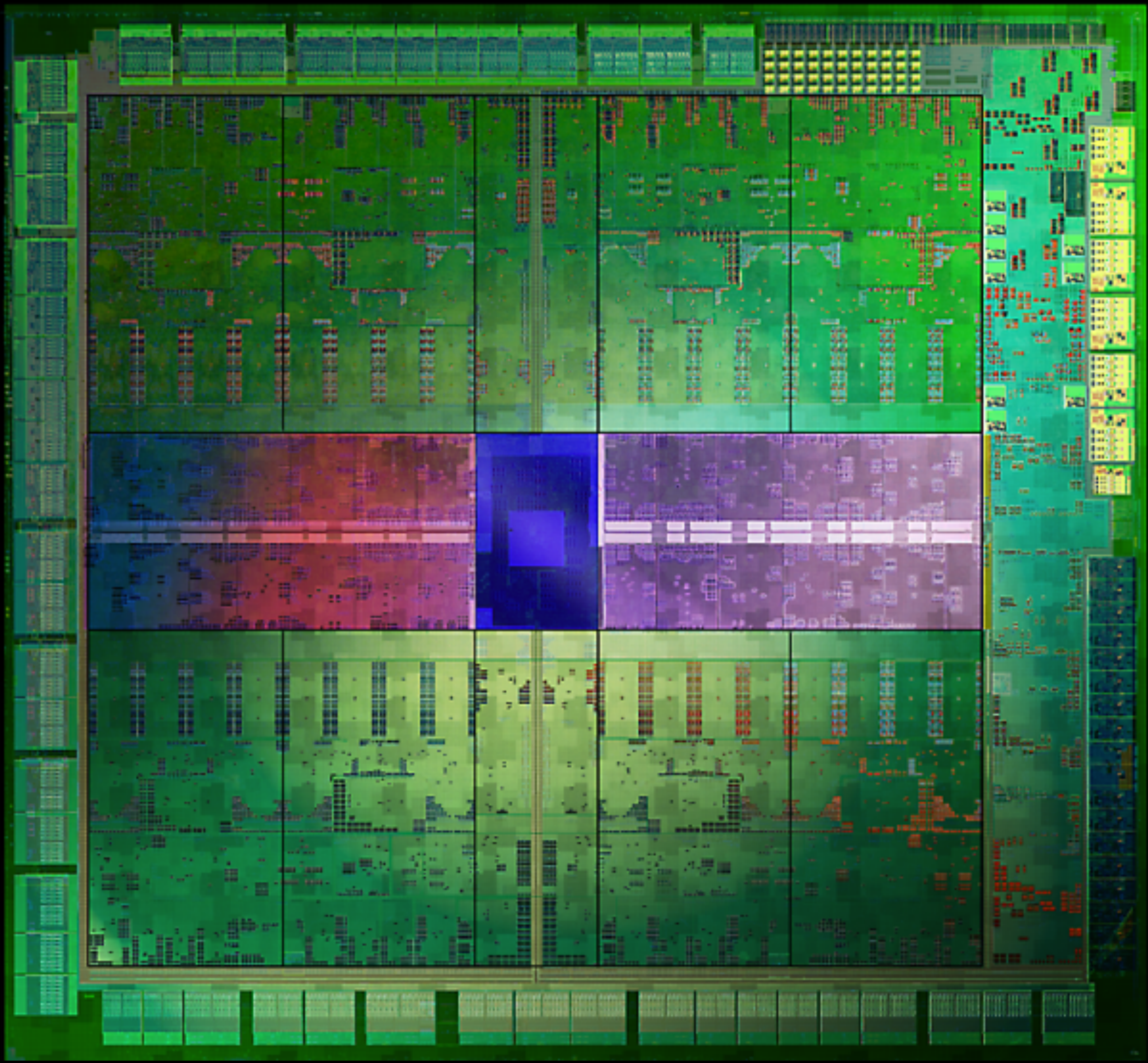
28nm process

3.5 billion transistors

1 GHz core clock

6GHz GDDR5

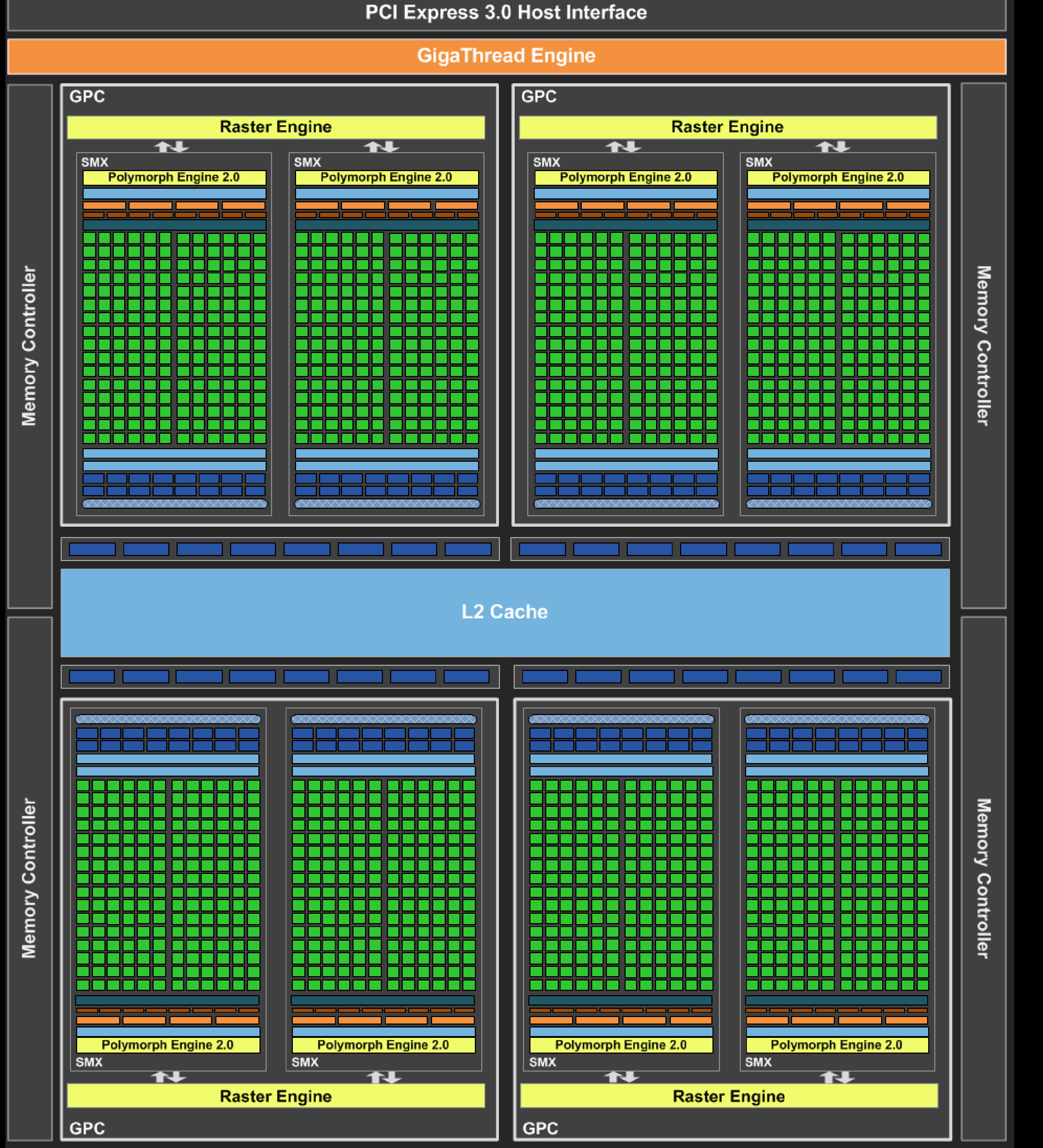
3 years, 1000 engineers



# GTX 680

4X as many shader CPUs, running at 2/3 the clock (vs GTX 560).

Polymorph engine does polygon tessellation. PCIe bus no longer limits triangle count.



# History and Graphics Processors

- **Create standard model from common practice:** Wire-frame geometry, triangle rasterization, pixel shading.

- **Put model in hardware:** Block diagram of chip matches computer graphics math.

- **Evolve to be programmable:** At some point, it becomes hard to see the math in the block diagram.

"Wheel of reincarnation" -- Hardwired graphics hardware evolves to look like general-purpose CPU. Ivan Sutherland co-wrote a paper on this topic in 1968!

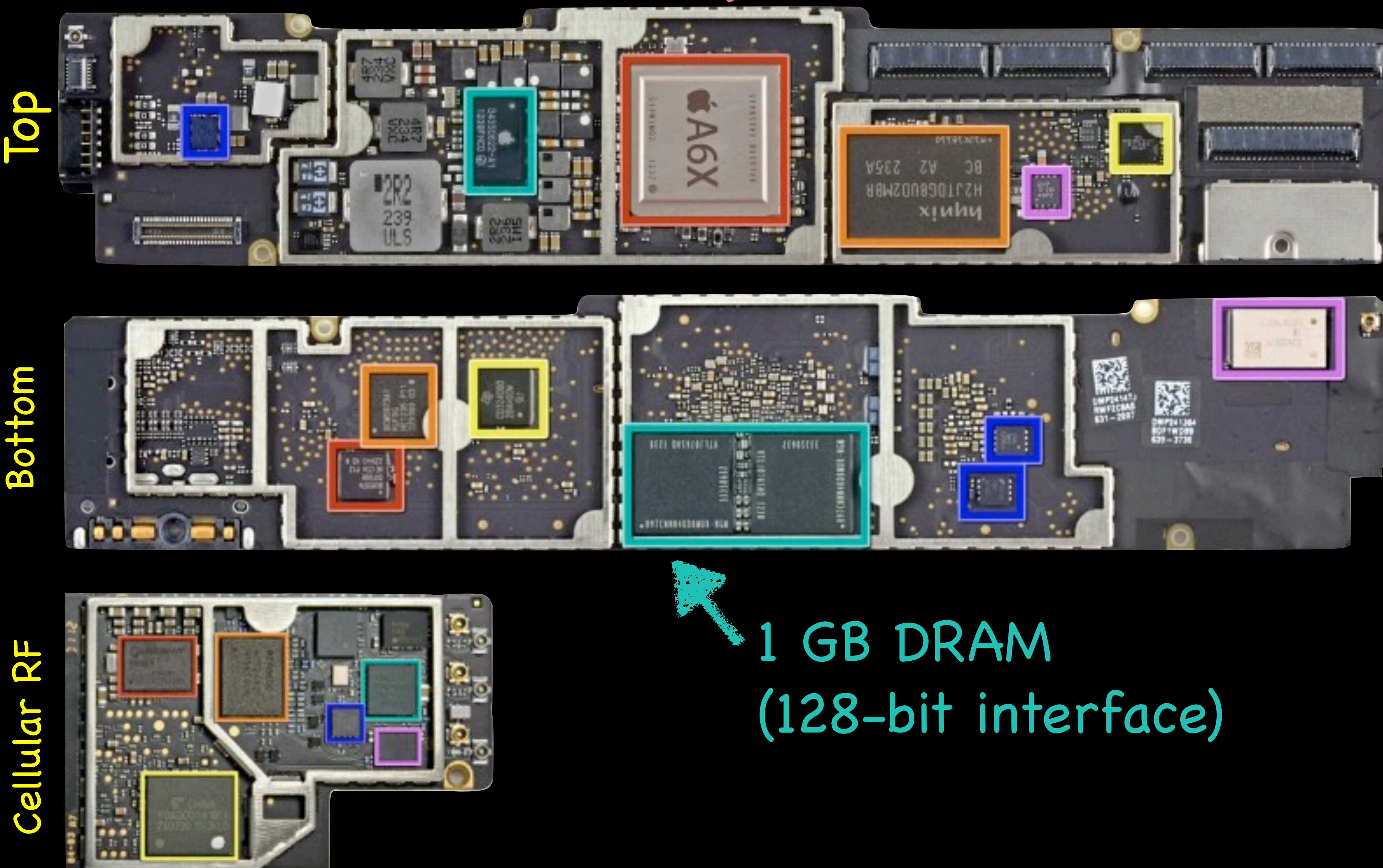
Samaritan: Direct X-11 demo from Unreal.

Runs in real-time on one GTX 680 (barely).

# GPUs on mobile devices

# iPad: iPhone++

A6X: ARMv7 cores,
PowerVR GPUs




1 GB DRAM
(128-bit interface)

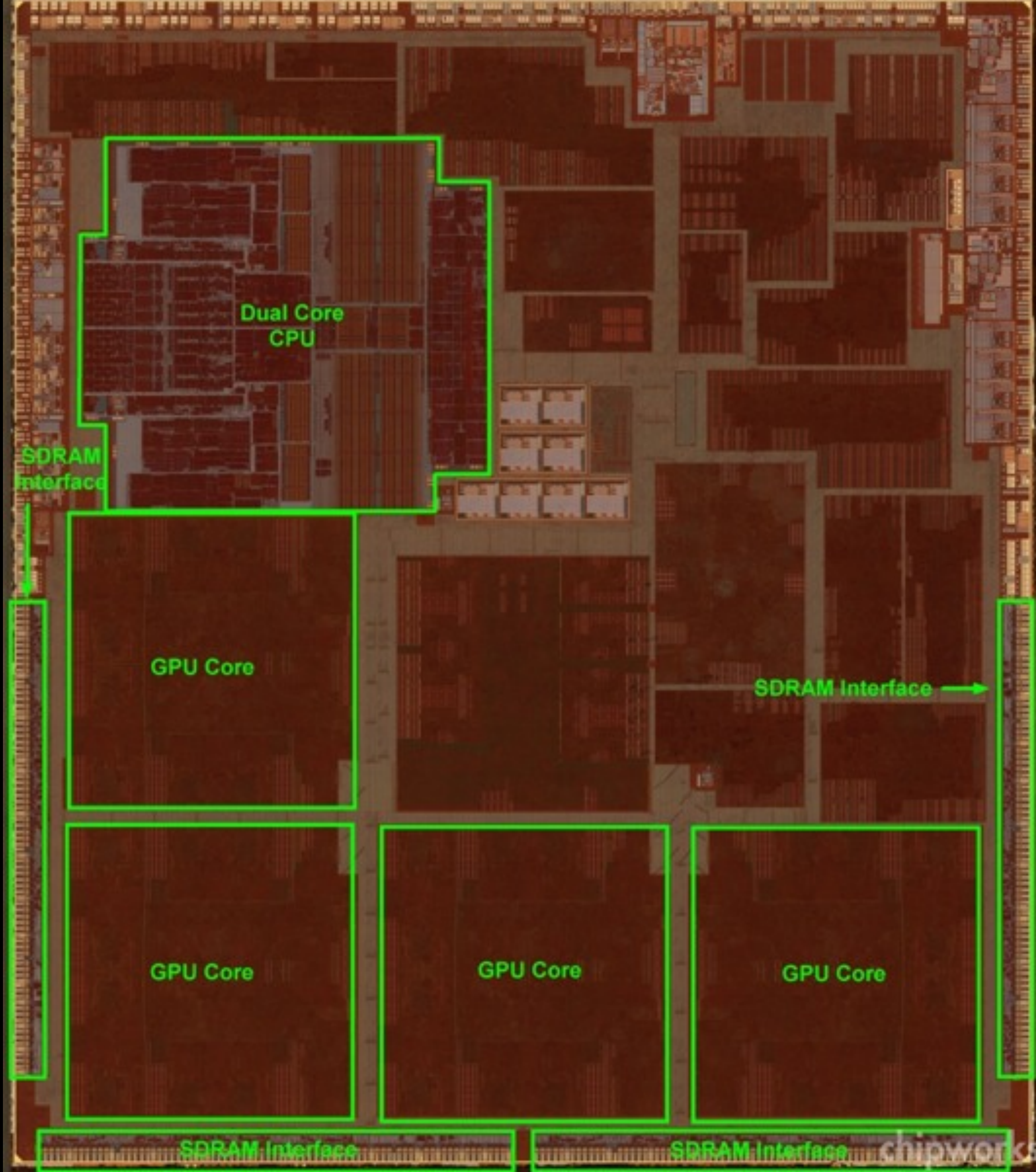
# Apple A6X

2012 iPad CPU/IGP.

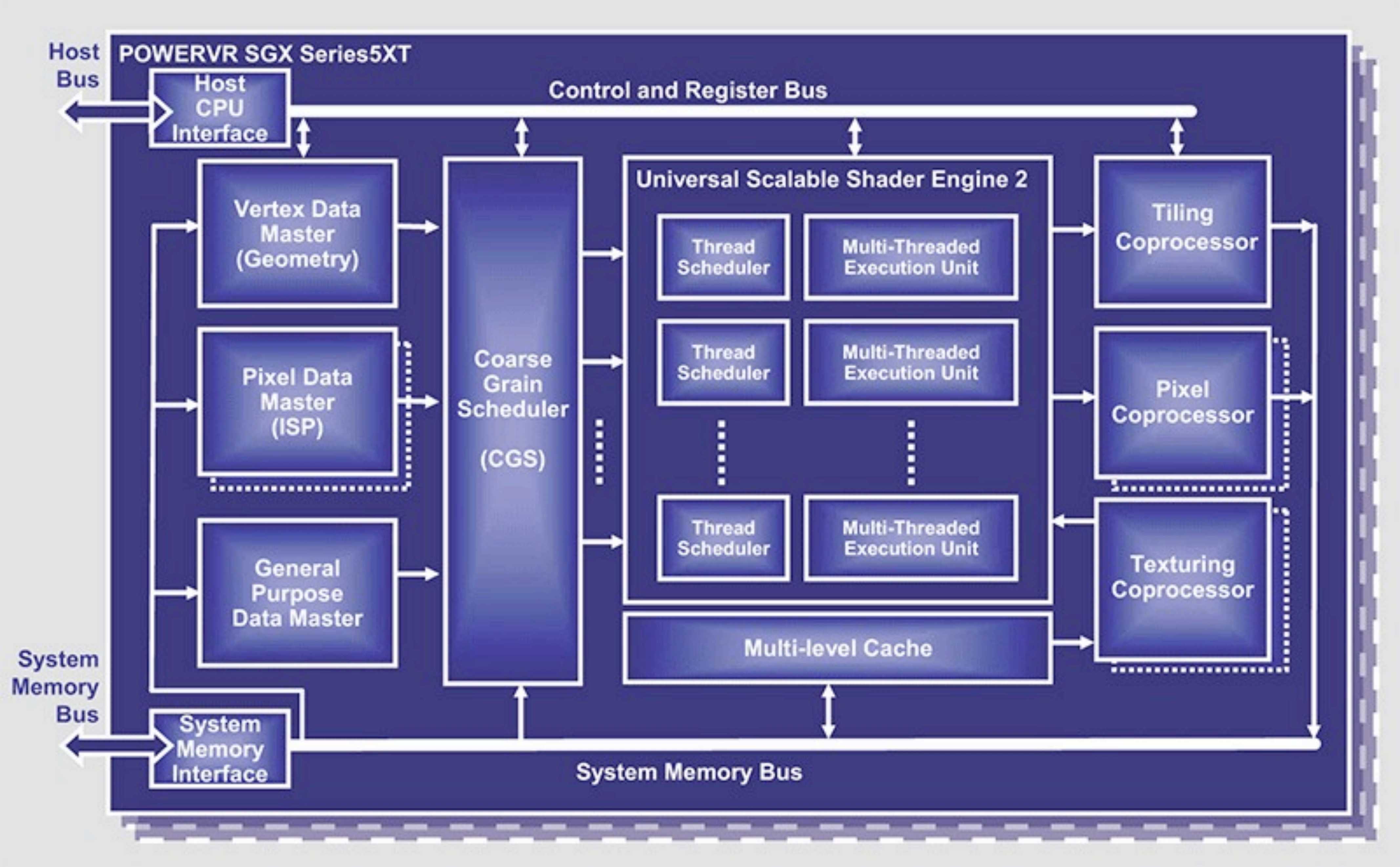
32 nm process, 10.4 x 11.9 mm

IGP fills about 40% of die.

IGP: 2.5% of Kepler (in GFLOPs).



# Mobile GPUs: Same ideas, scaled down.



# Today: Graphics Processors

- **Computer Graphics.** A brief introduction to “the pipeline”.
- **Stream Processing.** Casting the graphics pipeline into hardware.
- **Unified Pipelines.** GeForce 8800, from Nvidia, introduced in 2006.
- **Kepler.** The latest generation from Nvidia, released last month.