CS 150
Digital Design

Lecture 26 – Graphics Processors

2012-4-19

Professor John Wawrzynek

today's lecture by John Lazzaro

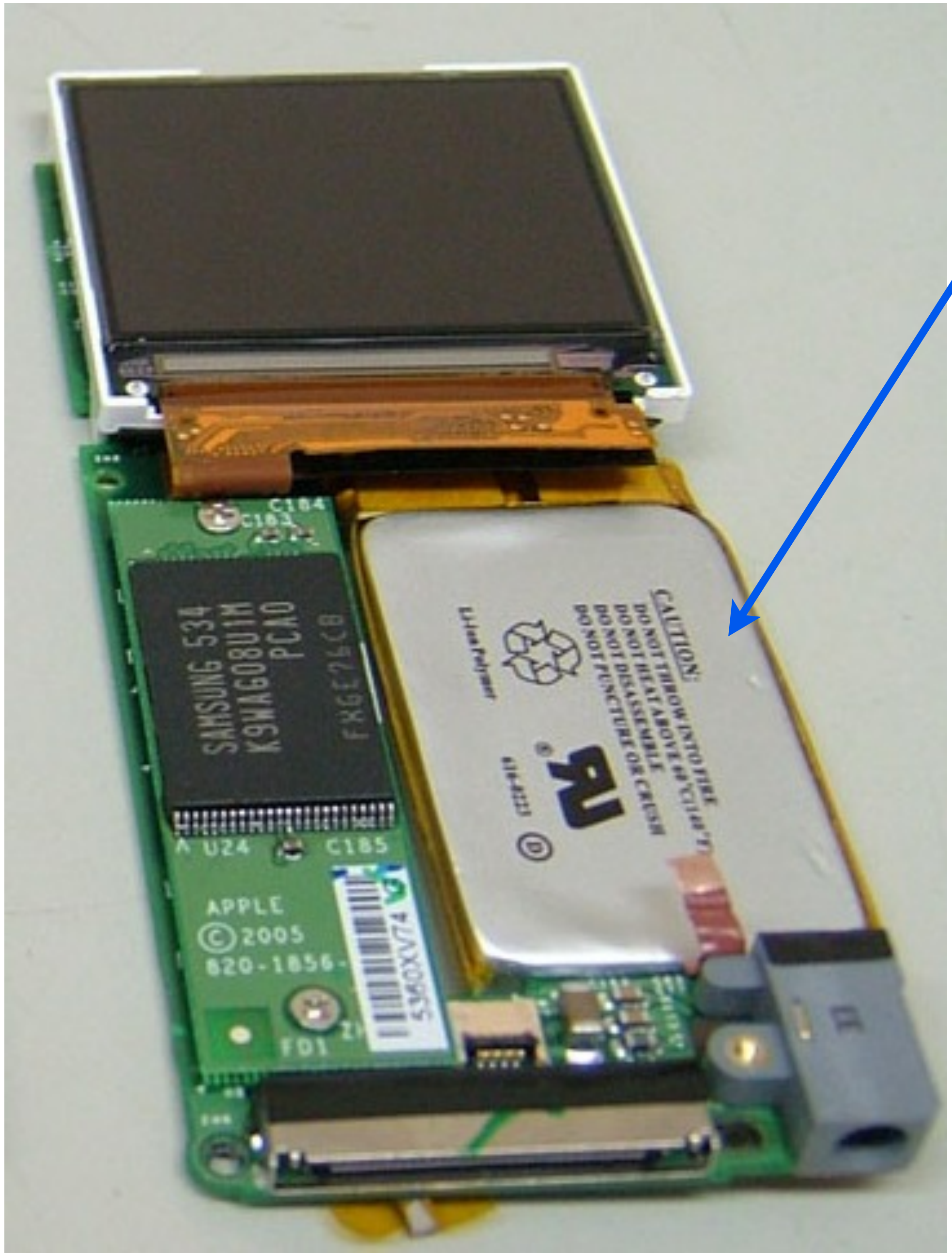
TAs: Shaoyi Cheng, Daiwei Li, James Parker

Play

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

Power lecture errata

Powering an iPod nano (2005 edition)



1.2 W-hour battery:
Can supply 1.2 watts
of power for 1 hour.

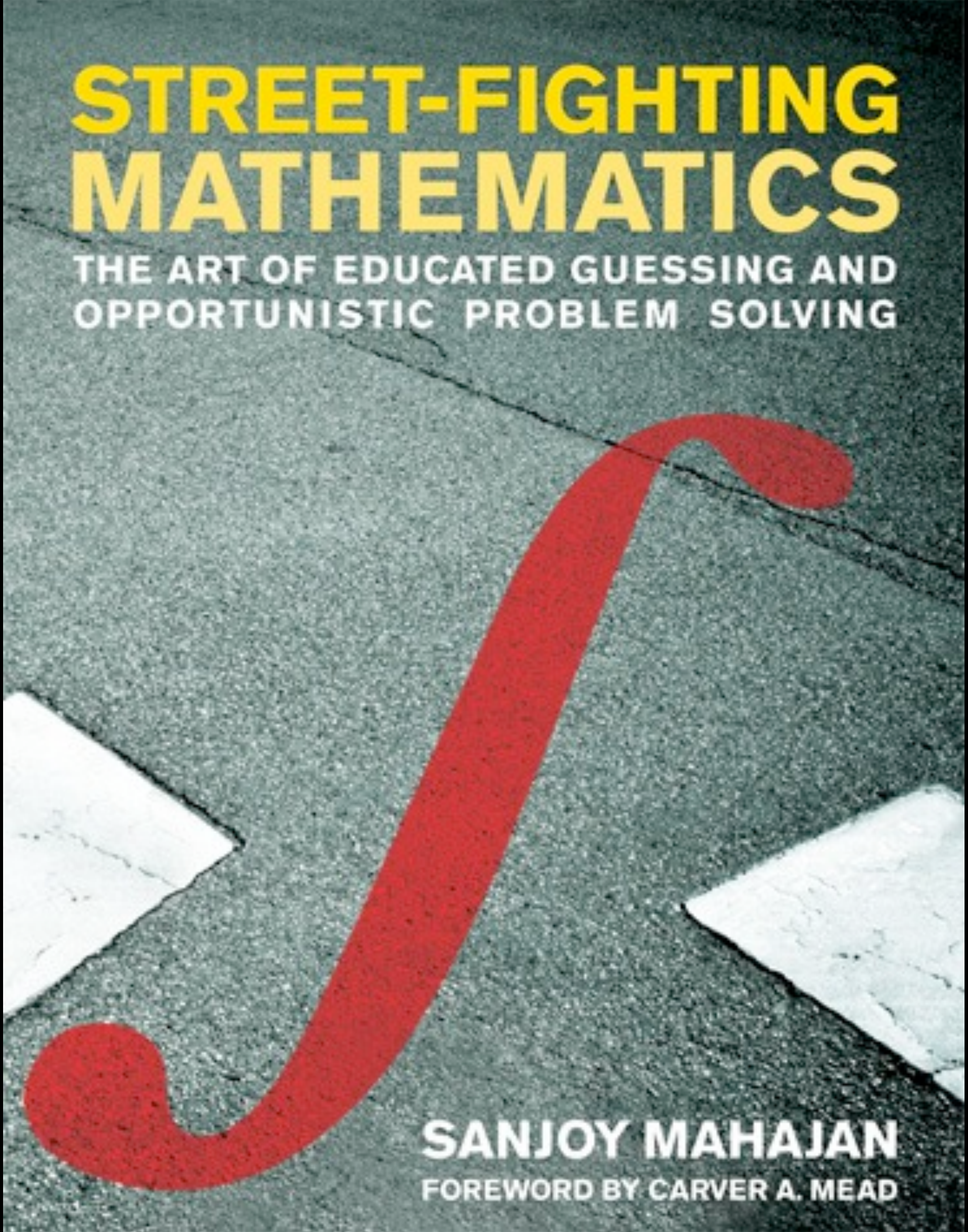
1.2 W-hr / 5 W ≈ 15 minutes.

More W-hours require bigger battery and thus bigger "form factor" -- it wouldn't be "nano" anymore :-).

Real specs for iPod nano :

14 hours for music,
4 hours for slide shows.

85 mW for music.
300 mW for slides.

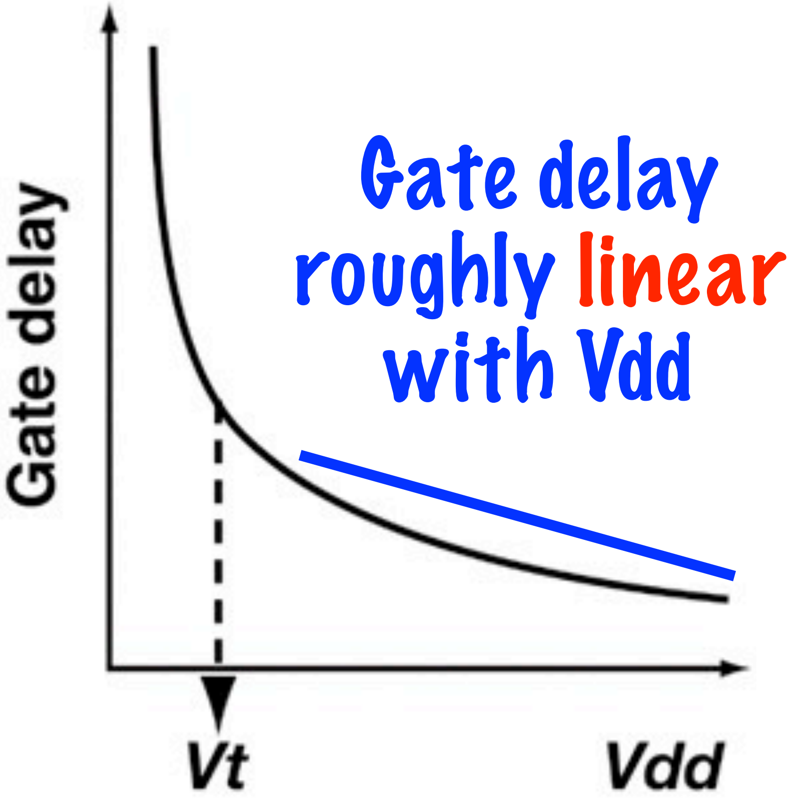


STREET-FIGHTING MATHEMATICS

THE ART OF EDUCATED GUESSING AND OPPORTUNISTIC PROBLEM SOLVING

SANJOY MAHAJAN

FOREWORD BY CARVER A. MEAD



Gate delay roughly linear with Vdd

And so, we can transform this:

Vdd
Logic Block

$P \sim F \times V_{dd}^2$

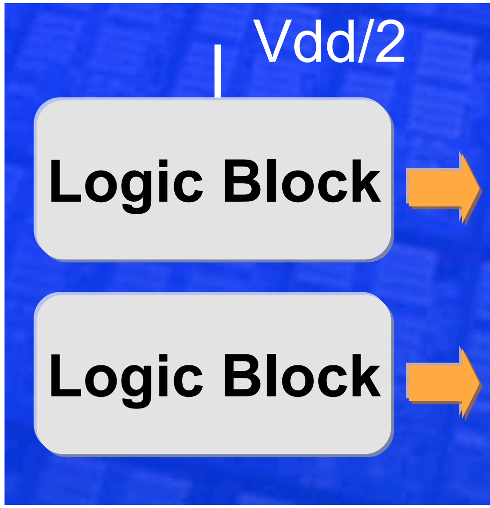
$P \sim 1 \times 1^2$

Freq = 1
Vdd = 1
Throughput = 1
Power = 1
Area = 1
Pwr Den = 1

Block processes stereo audio. 1/2 of clocks for "left", 1/2 for "right".

Into this:

Top block processes "left", bottom "right".



Freq = 0.5
Vdd = 0.5
Throughput = 1
Power = 0.25
Area = 2
Pwr Den = 0.125

$P \sim \#blks \times F \times V_{dd}^2$

$P \sim 2 \times 1/2 \times 1/4 = 1/4$

CV^2 power only

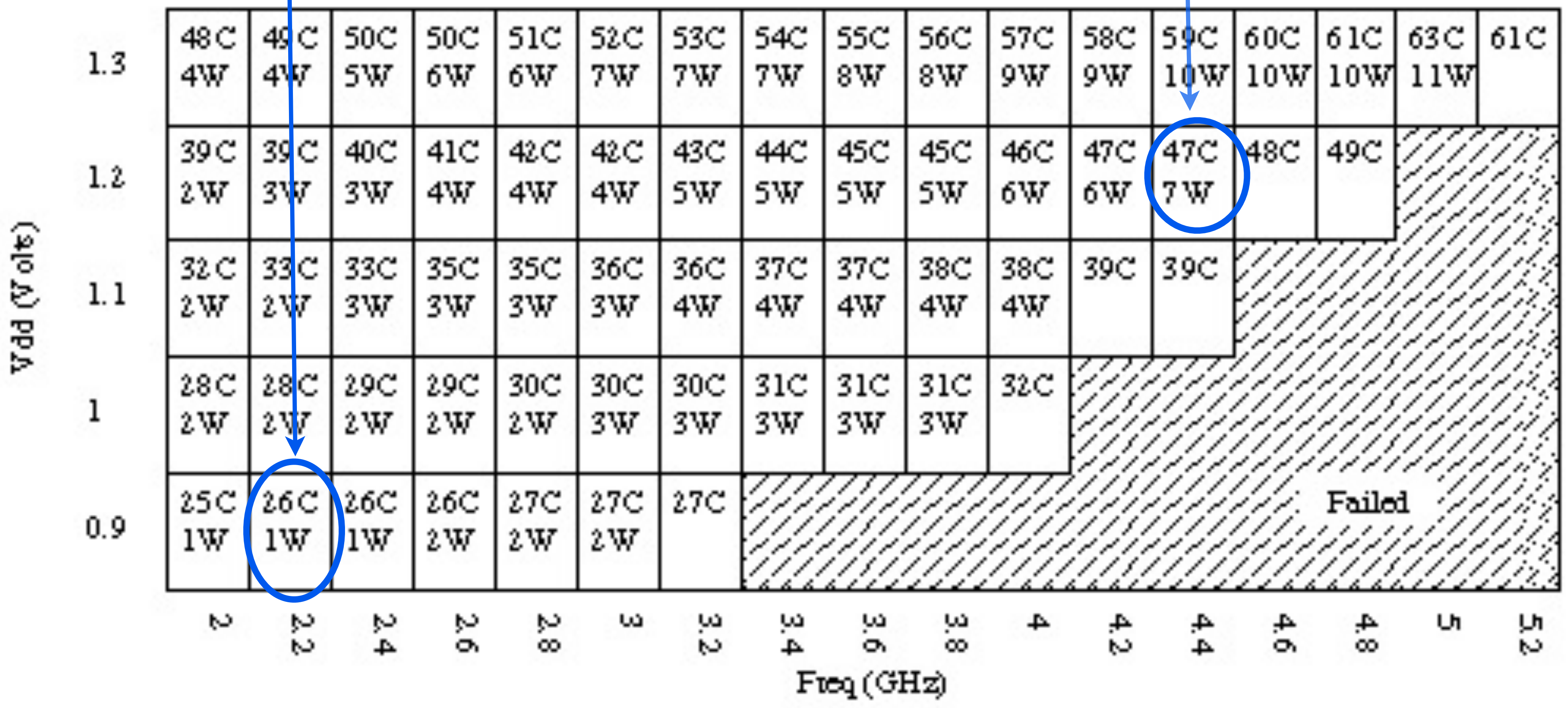
THIS MAGIC TRICK BROUGHT TO YOU BY CORY HALL ...

Scaling V and f *does* lower energy/op

1 W to get 2.2 GHz performance. 26 C die temp.

7W to reliably get 4.4 GHz performance. 47C die temp.

If a program that needs a 4.4 Ghz CPU can be recoded to use two 2.2 Ghz CPUs ... big win.



Vdd (Volts)	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8	4	4.2	4.4	4.6	4.8	5	5.2
1.3	48C 4W	49C 4W	50C 5W	50C 6W	51C 6W	52C 7W	53C 7W	54C 7W	55C 8W	56C 8W	57C 9W	58C 9W	59C 10W	60C 10W	61C 10W	63C 11W	61C
1.2	39C 2W	39C 3W	40C 3W	41C 4W	42C 4W	42C 4W	43C 5W	44C 5W	45C 5W	45C 5W	46C 6W	47C 6W	47C 7W	48C	49C		
1.1	32C 2W	33C 2W	33C 3W	35C 3W	35C 3W	36C 3W	36C 4W	37C 4W	37C 4W	38C 4W	38C 4W	39C	39C				
1	28C 2W	28C 2W	29C 2W	29C 2W	30C 2W	30C 3W	30C 3W	31C 3W	31C 3W	31C 3W	32C						
0.9	25C 1W	26C 1W	26C 1W	26C 2W	27C 2W	27C 2W	27C								Failed		

Freq (GHz)

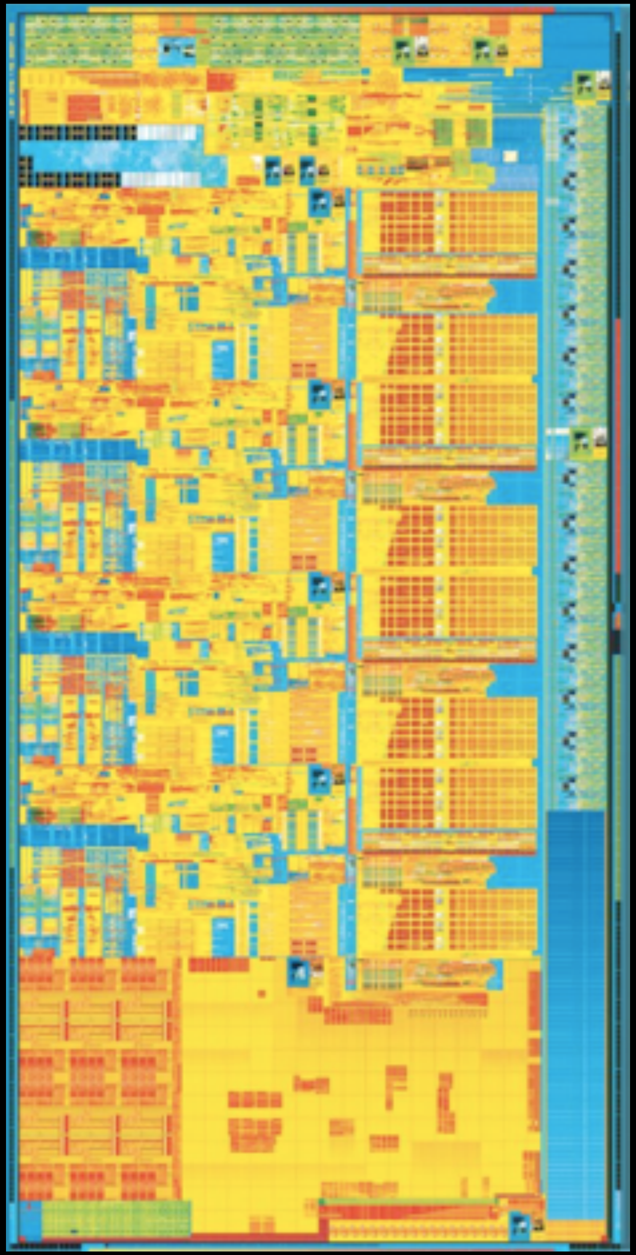
Today: Graphics Processors

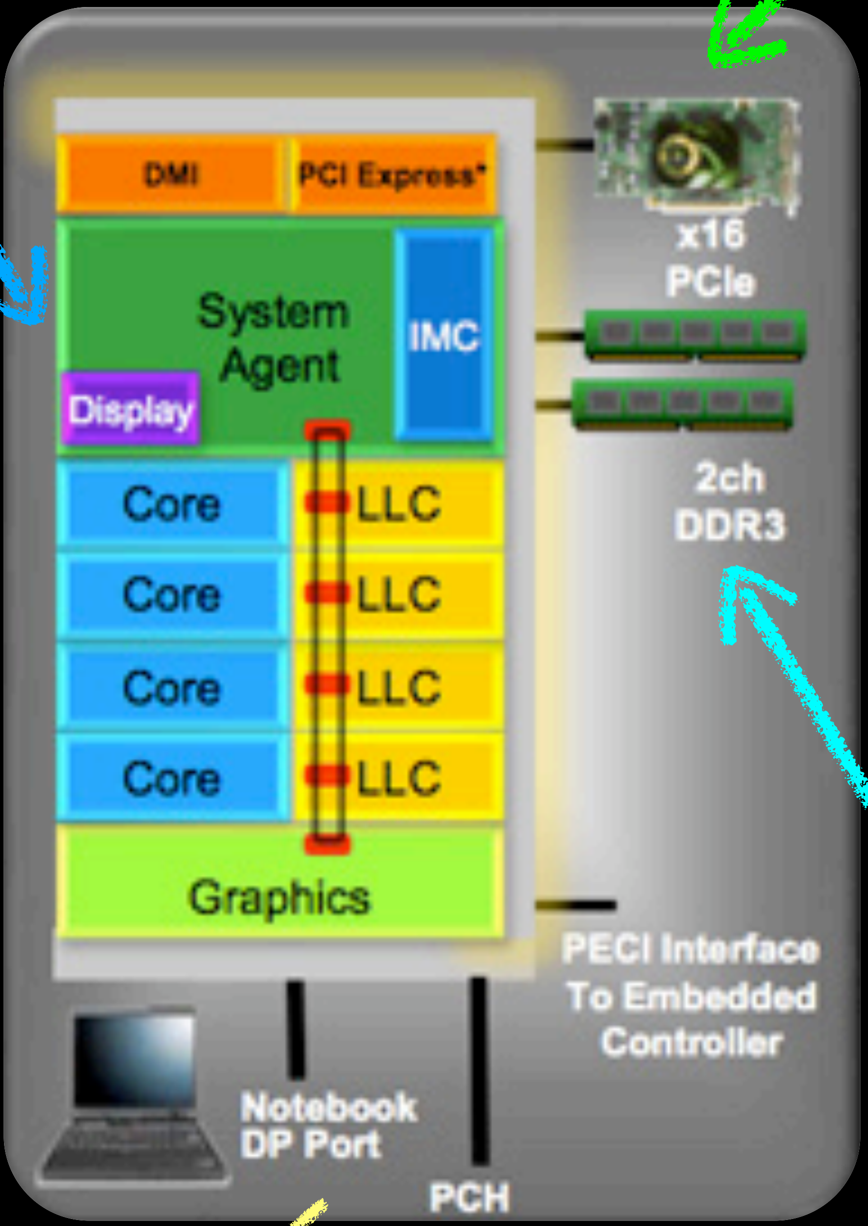
- **Computer Graphics.** A brief introduction to "the pipeline".
- **Stream Processing.** Casting the graphics pipeline into hardware.
- **Unified Pipelines.** GeForce 8800, from Nvidia, introduced in 2006.
- **Kepler.** The latest generation from Nvidia, released last month.

PC Graphics, 2012 Edition

PC Graphics Architecture

Core i5
CPU/IGP

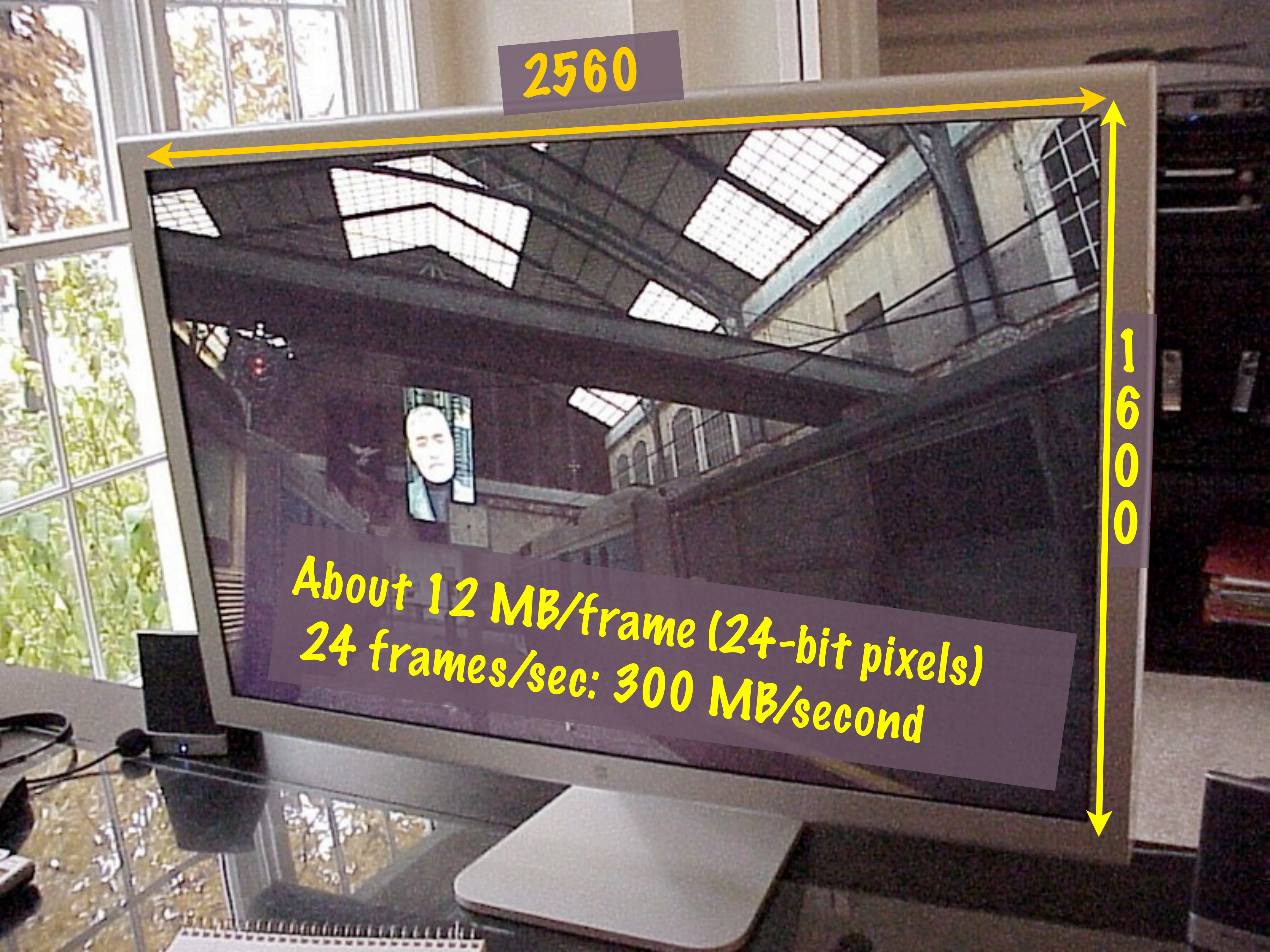




PCIe bus supports discrete GPU, with dedicated RAM and monitor outputs.

IGP uses system DRAM as graphics memory.

Display Out



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

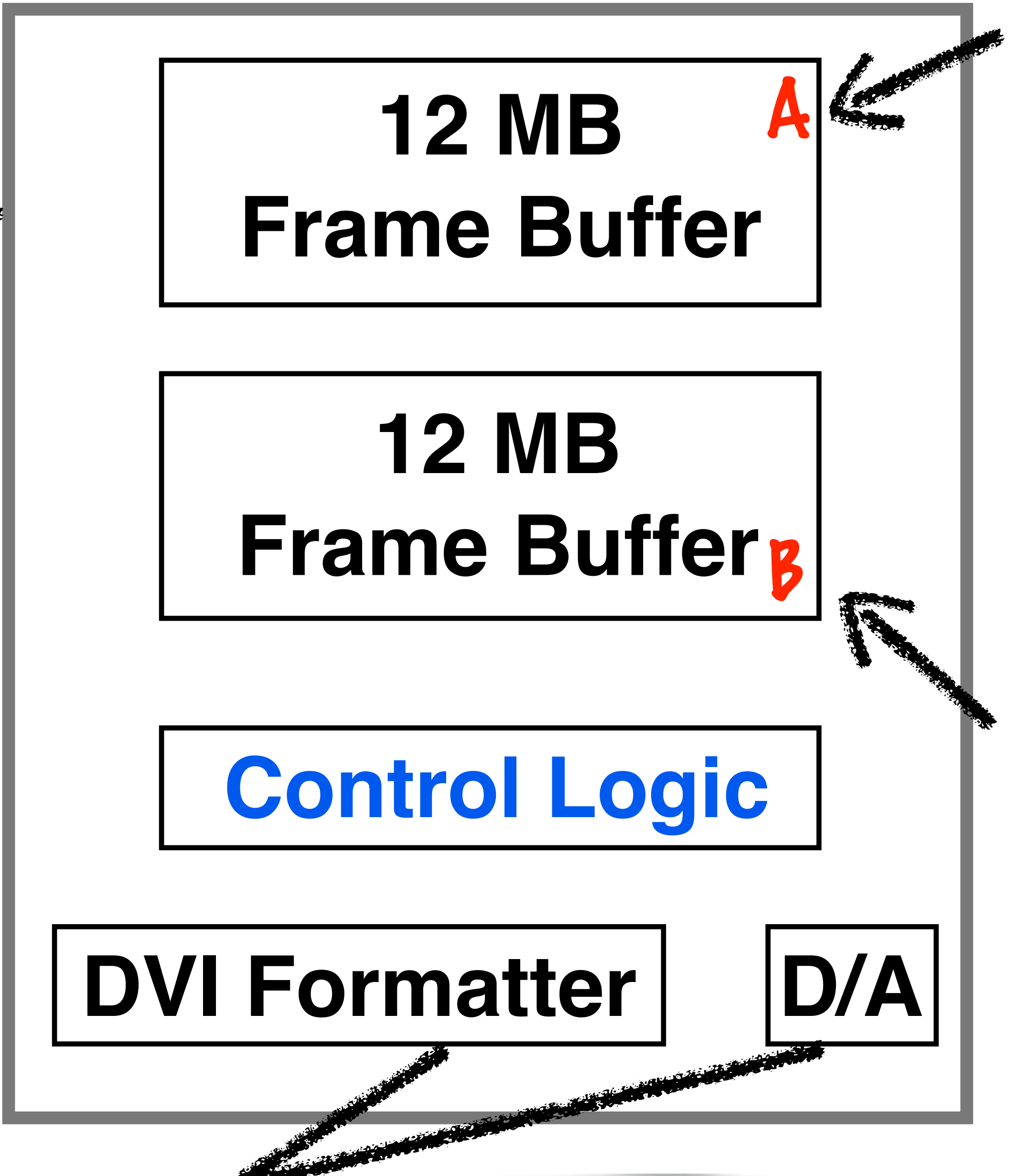
The “unaccelerated” graphics board ...

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.



Double Buffering:

CPU writes A frame in one buffer.

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

Display Out

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).

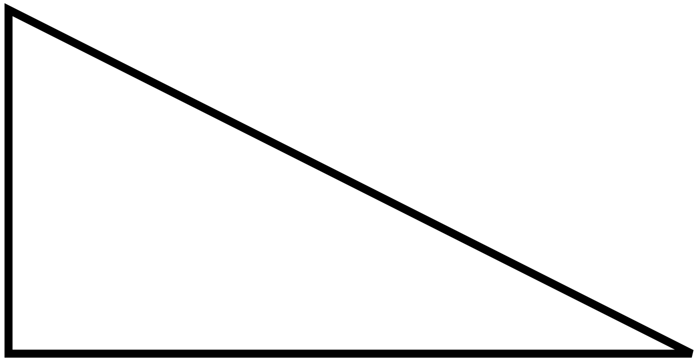
Graphics Acceleration

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.

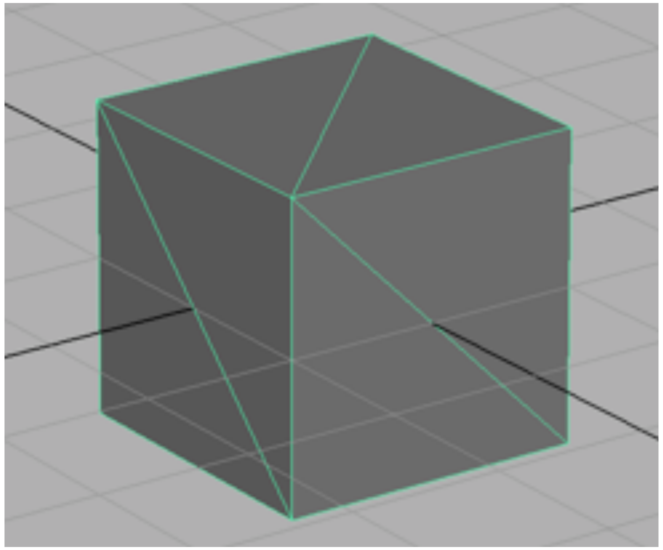
Next: An intro to 3-D graphics.

The Triangle ...



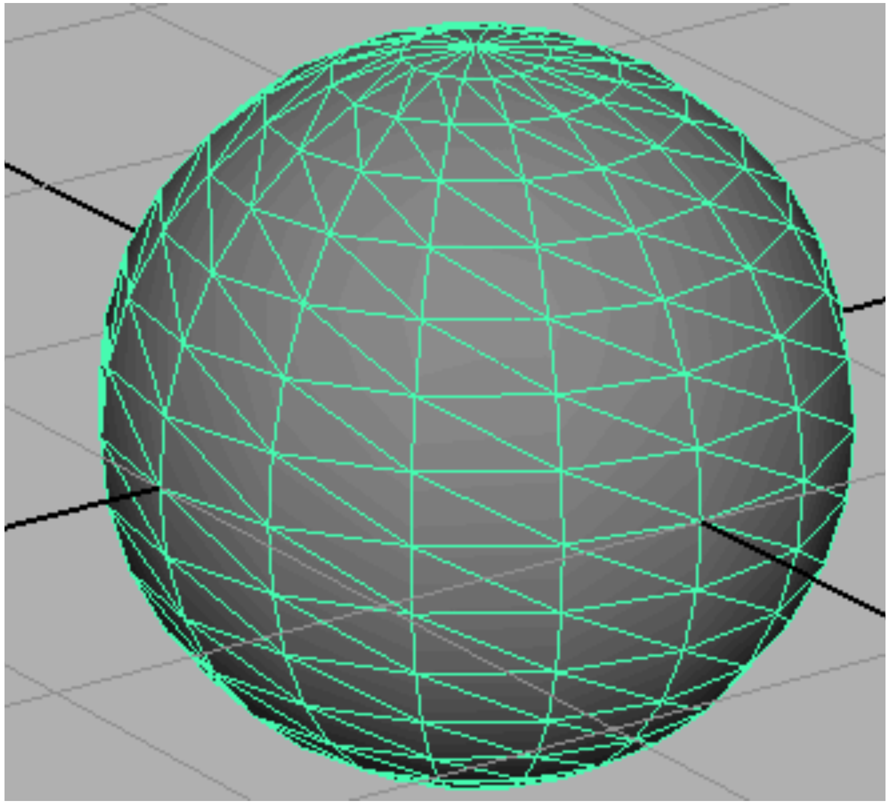
Simplest closed shape that may be defined by straight edges.

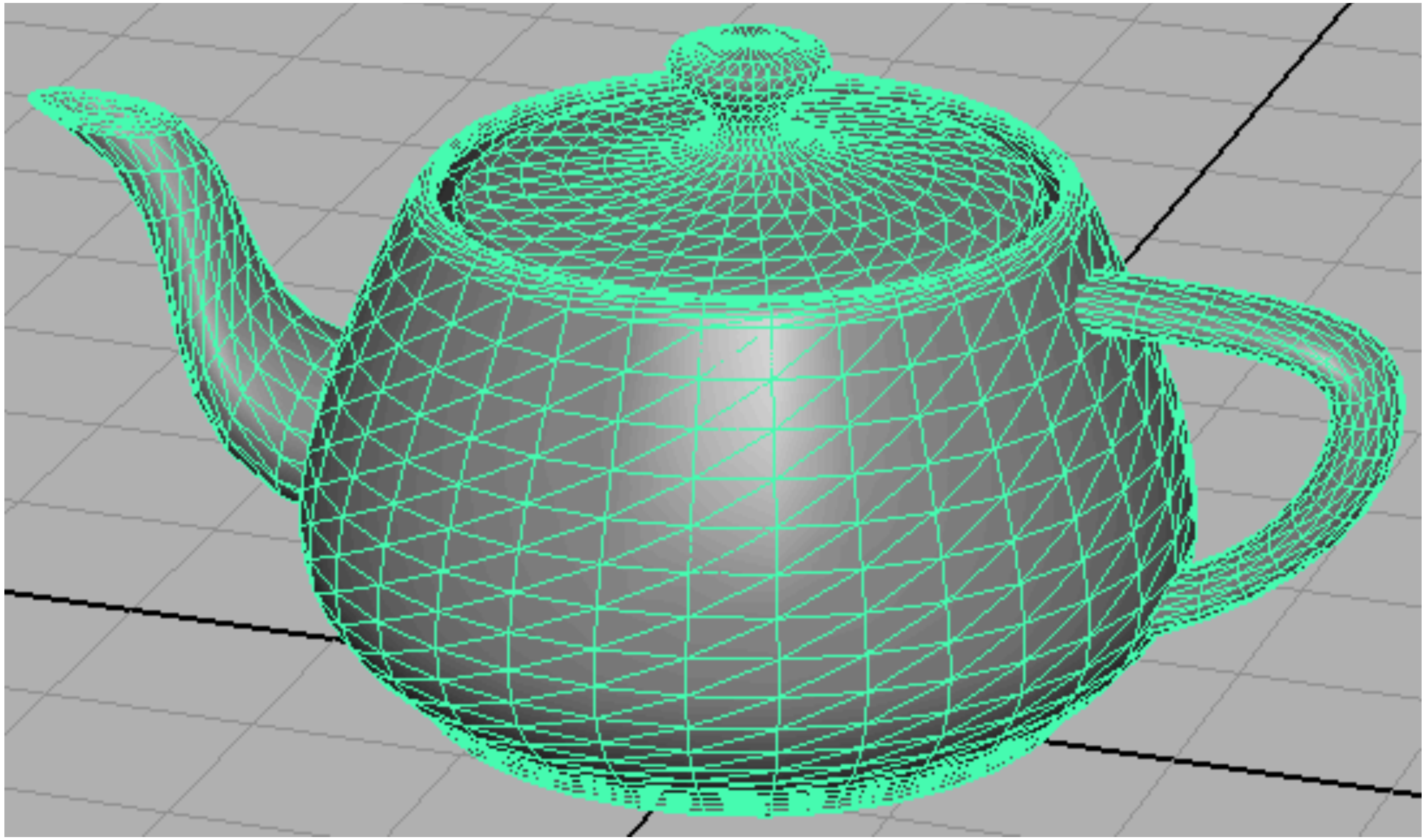
With enough triangles, you can make anything.



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.

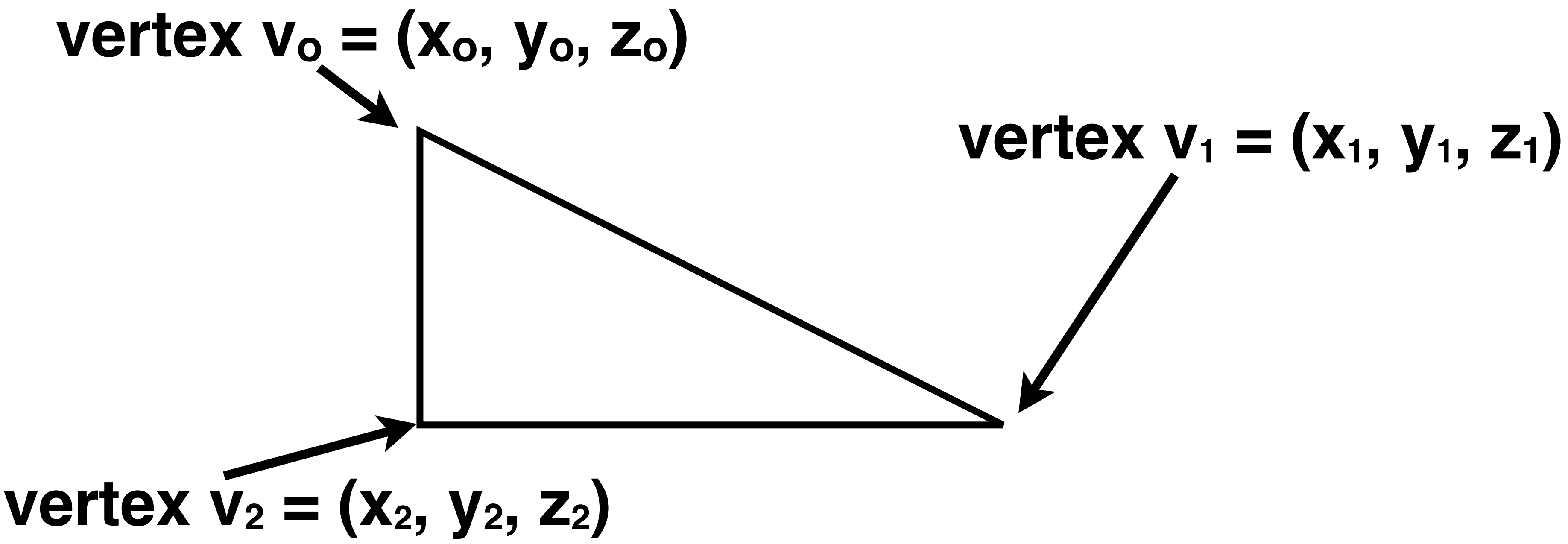




A teapot (famous object in computer graphics history). A “wire-frame” of triangles can capture the 3-D shape of complex, man-made objects.

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.



vertex $v_0 = (x_0, y_0, z_0)$

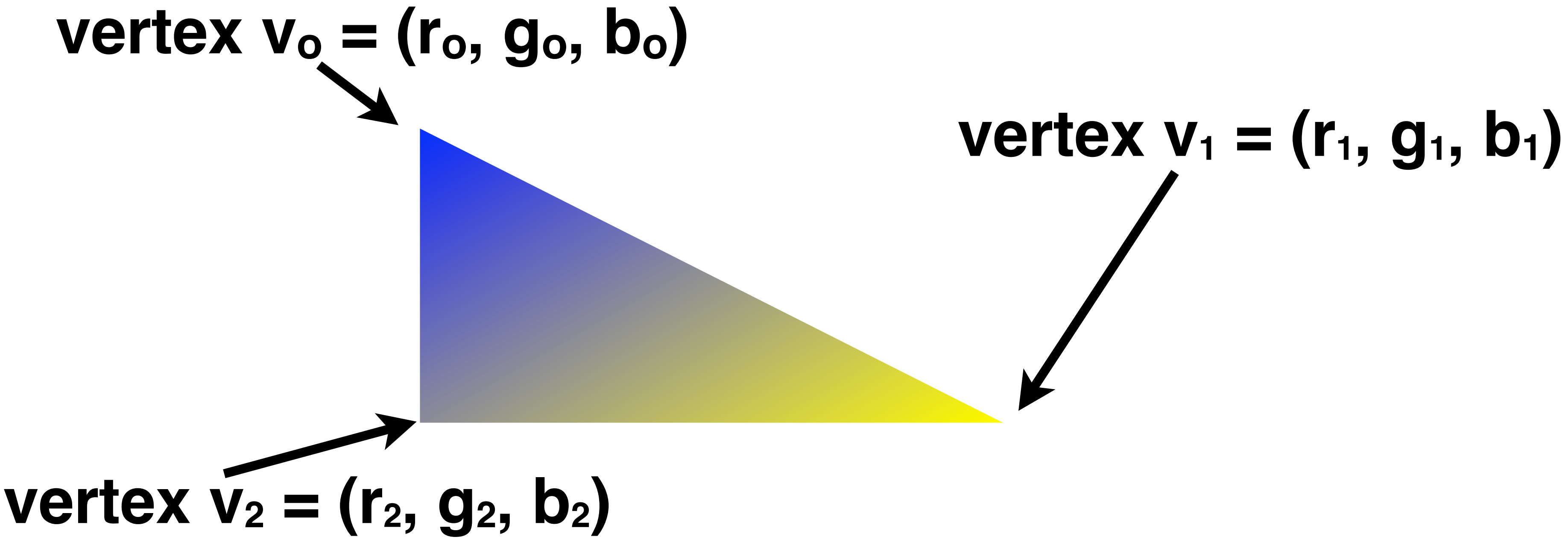
vertex $v_1 = (x_1, y_1, z_1)$

vertex $v_2 = (x_2, y_2, z_2)$

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.



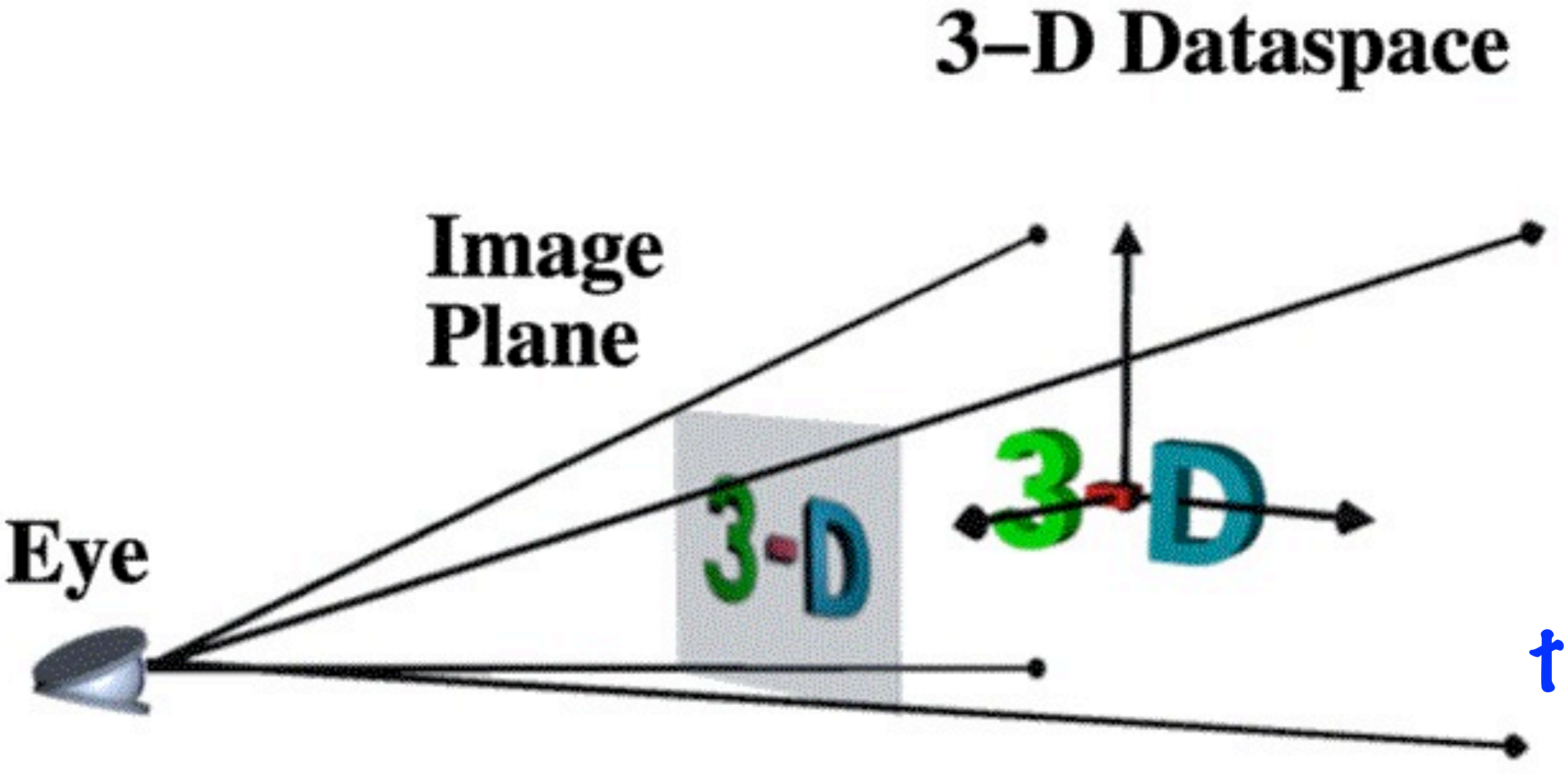
vertex $v_0 = (r_0, g_0, b_0)$

vertex $v_1 = (r_1, g_1, b_1)$

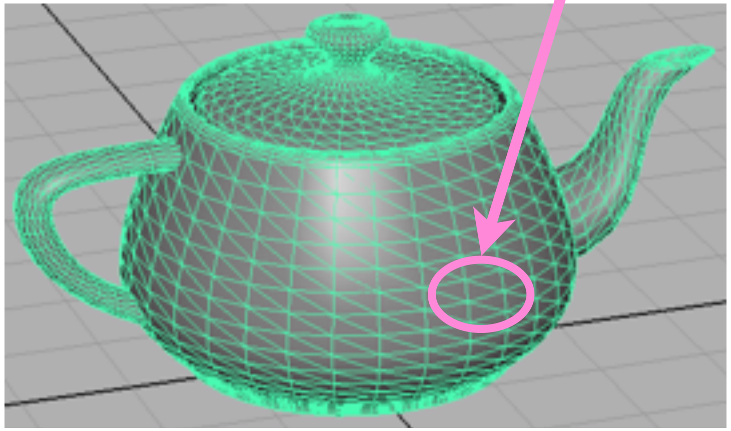
vertex $v_2 = (r_2, g_2, b_2)$

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



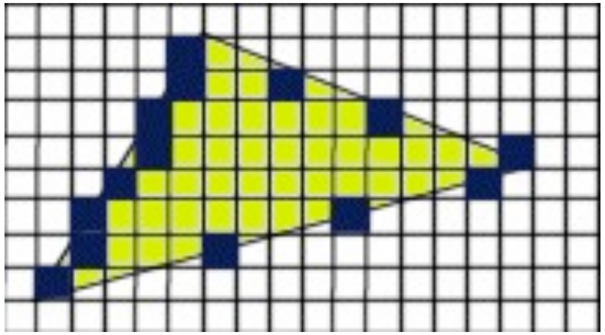
Let's follow one 3-D triangle.



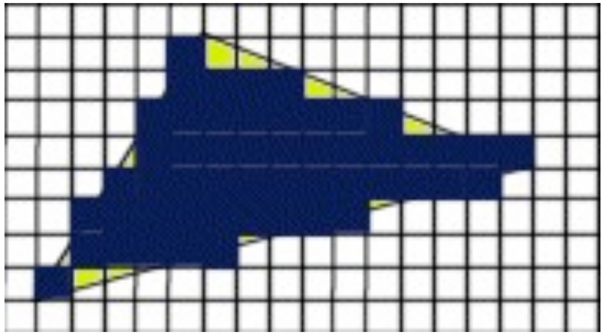
From 3-d triangles to screen pixels

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



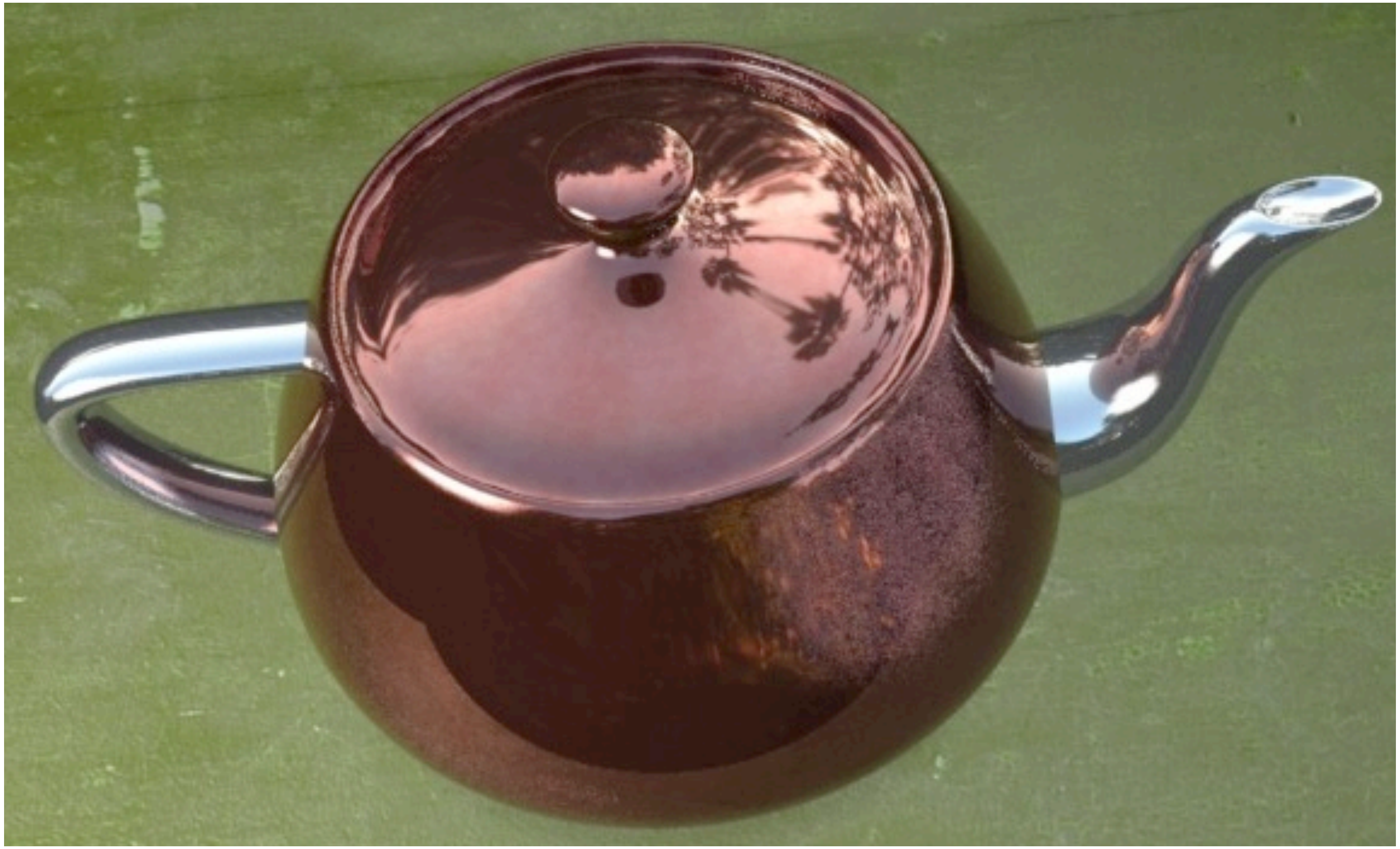
Then, create "pixel fragments" to fill in the triangle (rasterization).



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.

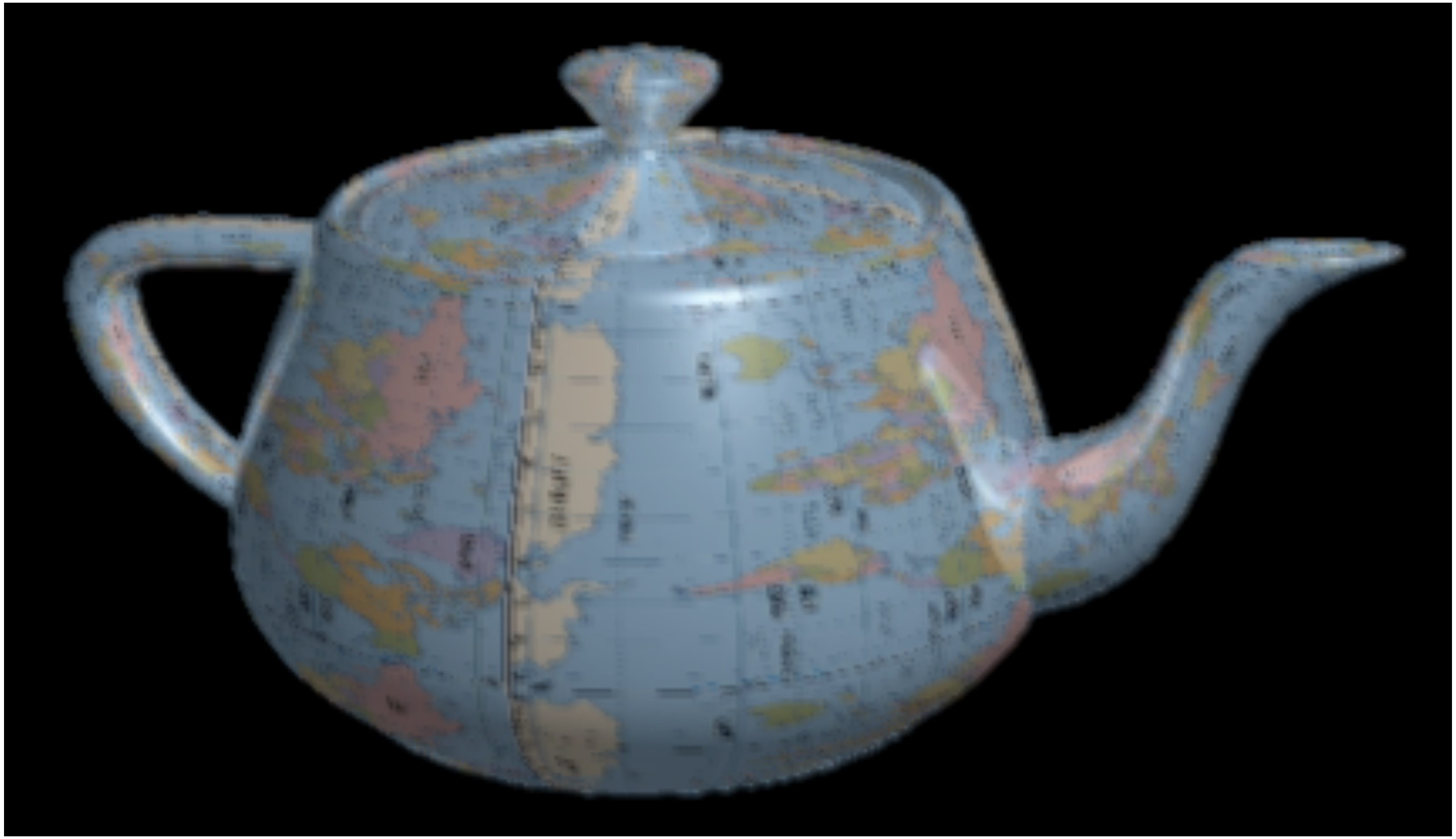


Process each fragment to "shade" it.

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

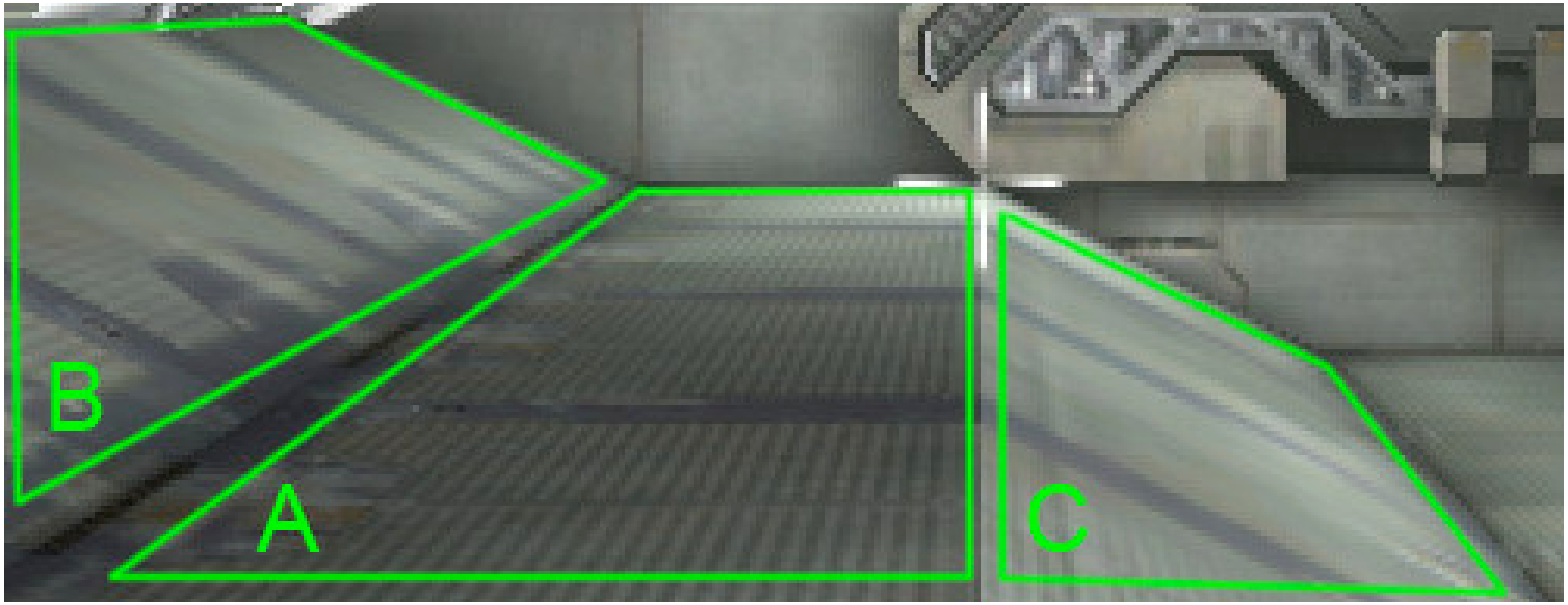
Final step: Output Merge. Assemble pixel fragments to make final 2-d image pixels.

Real-world texture maps: Bike decals

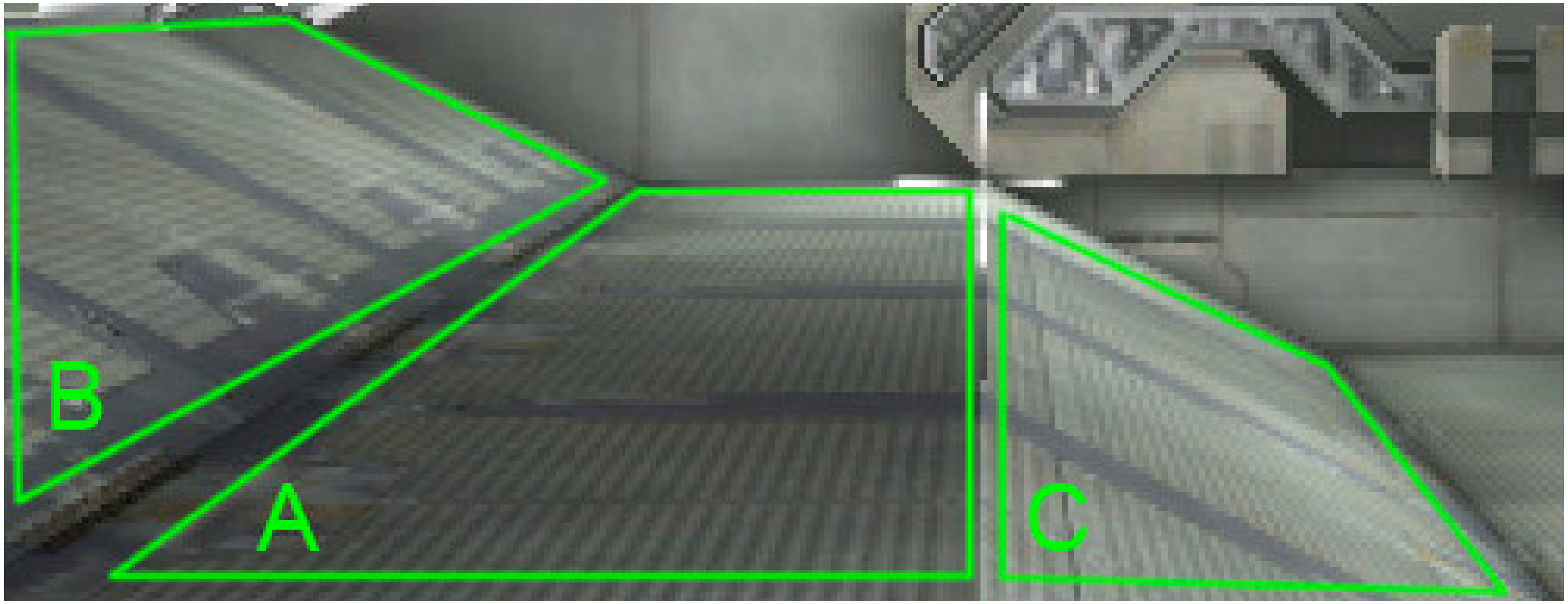


Applying texture maps: Quality matters

“Good” algorithm. B and C look blurry.



“Better” algorithm. B and C are detailed.



Putting it All Together ...

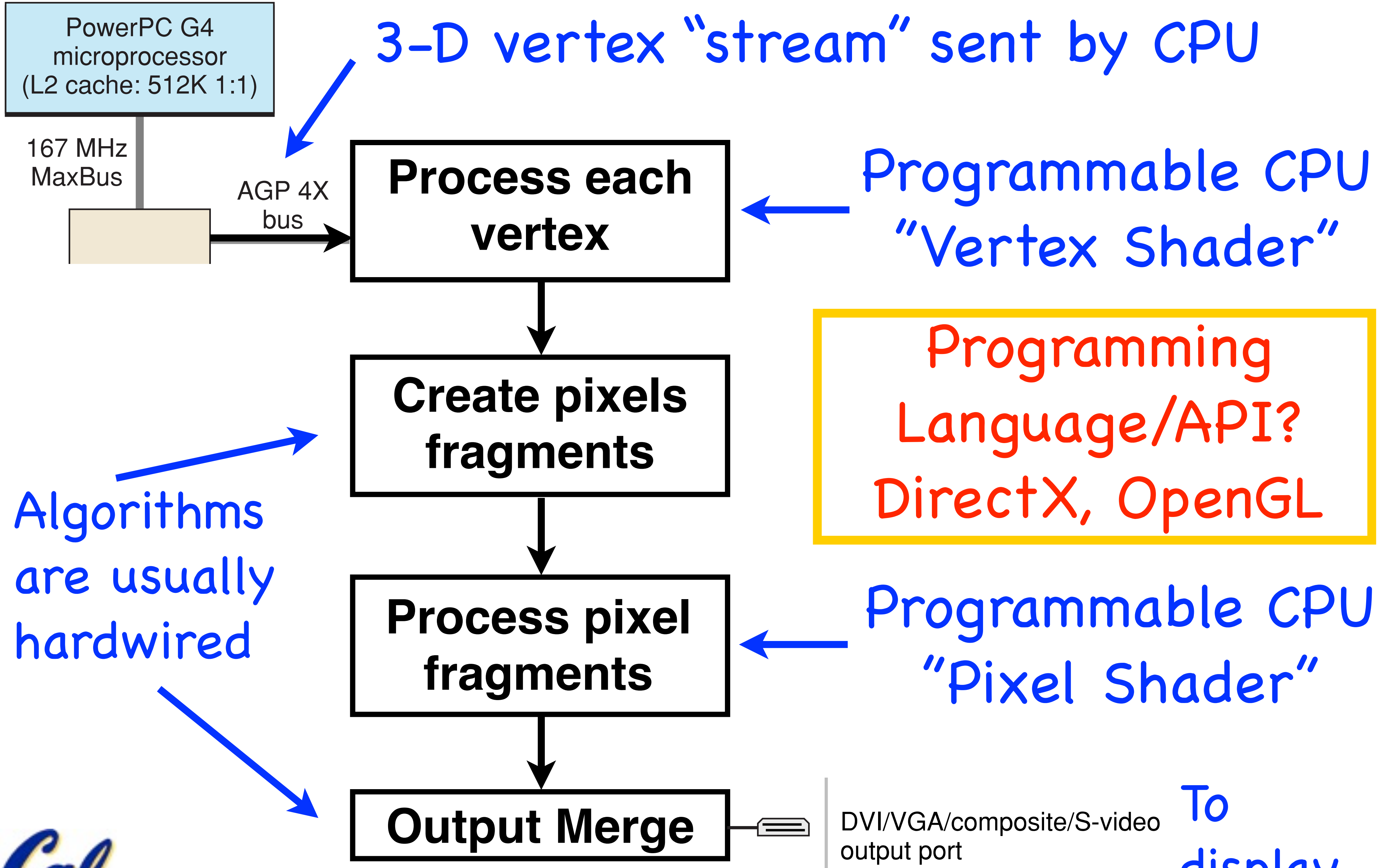
Luxo, Jr: Short movie made by Pixar, shown at SIGGRAPH in 1986.

First Academy Award given to a computer graphics movie.

Graphics Acceleration

Next: Back to architecture ...

The graphics pipeline in hardware (2004)



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

167 MHz MaxBus

AGP 4X bus

3-D vertex "stream" sent by CPU

Process each vertex

Programmable CPU "Vertex Shader"

Create pixels fragments

Programming Language/API? DirectX, OpenGL

Algorithms are usually hardwired

Process pixel fragments

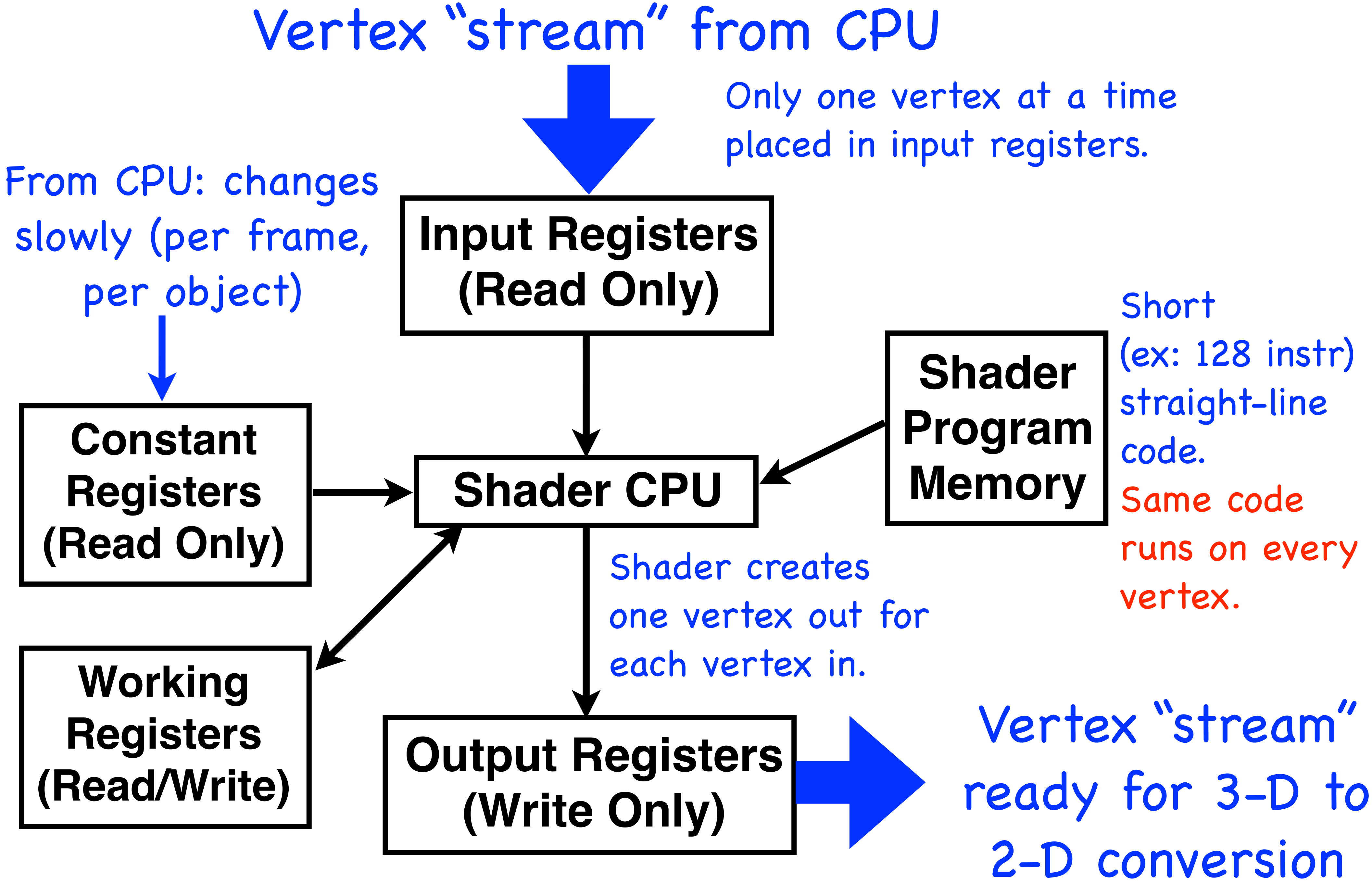
Programmable CPU "Pixel Shader"

Output Merge

DVI/VGA/composite/S-video output port

To display

Vertex Shader: A “stream processor”



Vertex “stream” from CPU

Only one vertex at a time placed in input registers.

From CPU: changes slowly (per frame, per object)

Input Registers (Read Only)

Constant Registers (Read Only)

Shader CPU

Shader Program Memory

Short (ex: 128 instr) straight-line code.

Same code runs on every vertex.

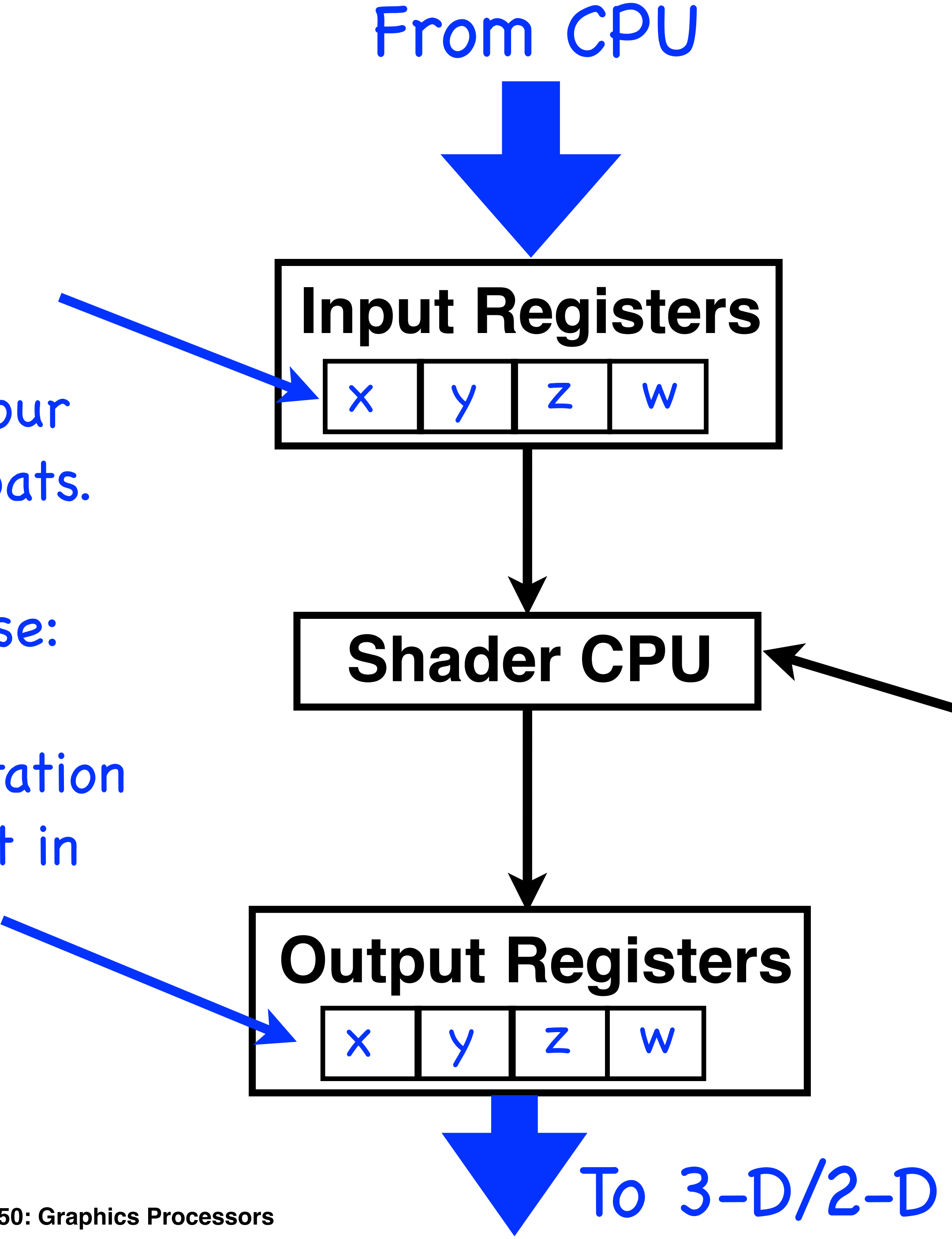
Working Registers (Read/Write)

Shader creates one vertex out for each vertex in.

Output Registers (Write Only)

Vertex “stream” ready for 3-D to 2-D conversion

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.

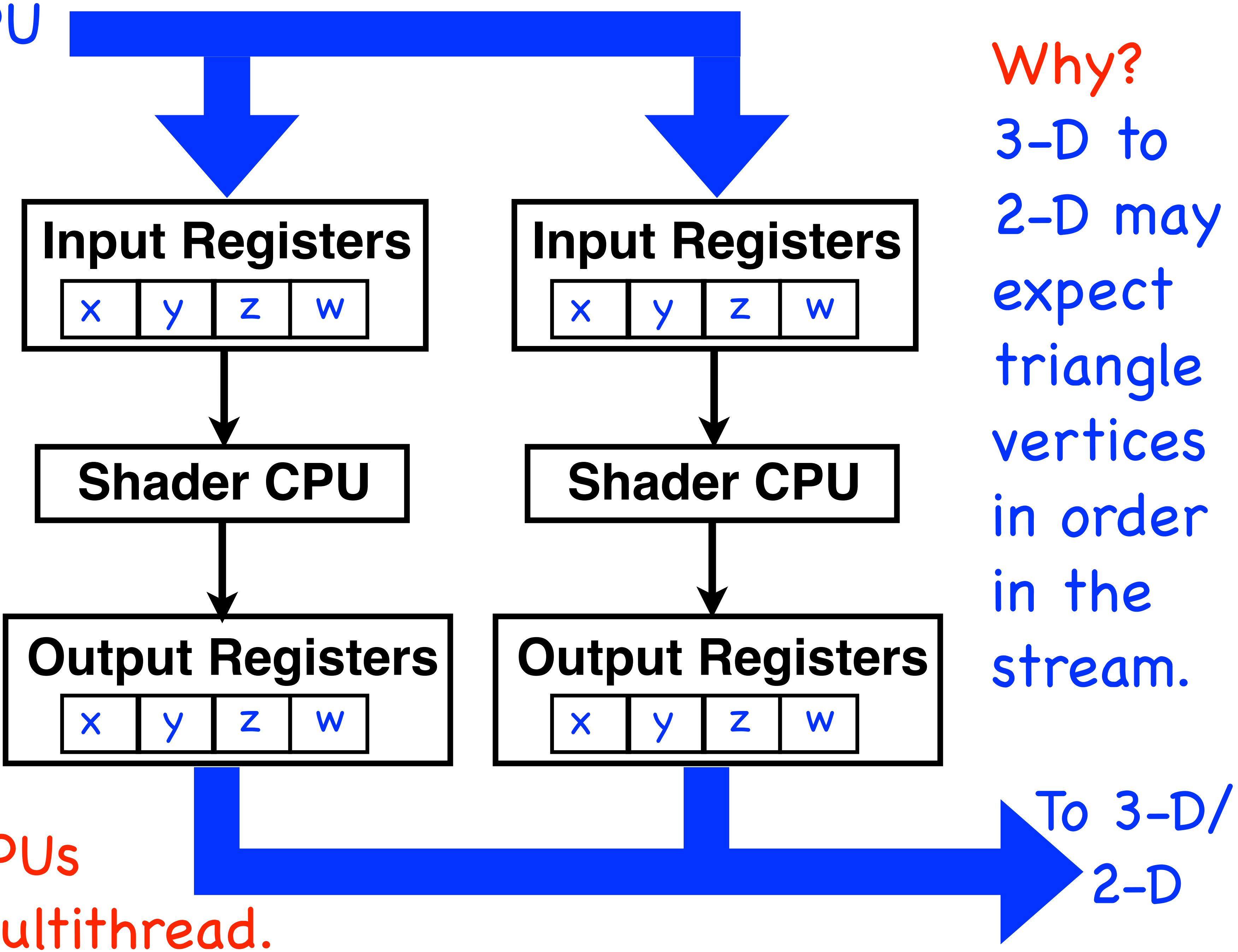
Typical instruction:

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

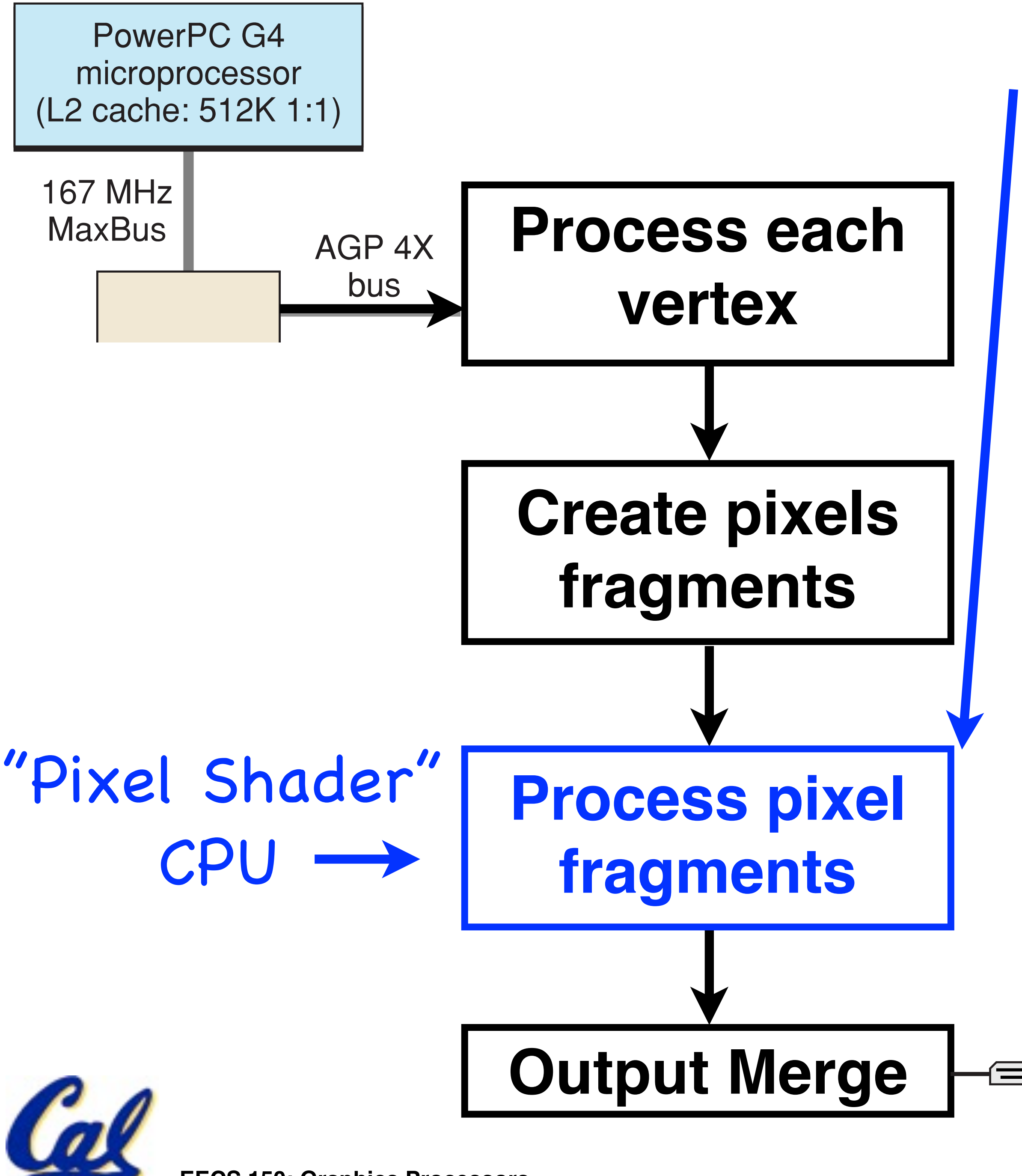


Caveat: Care might be needed when merging streams.

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

Shader CPUs easy to multithread.

Pixel shader specializations ...

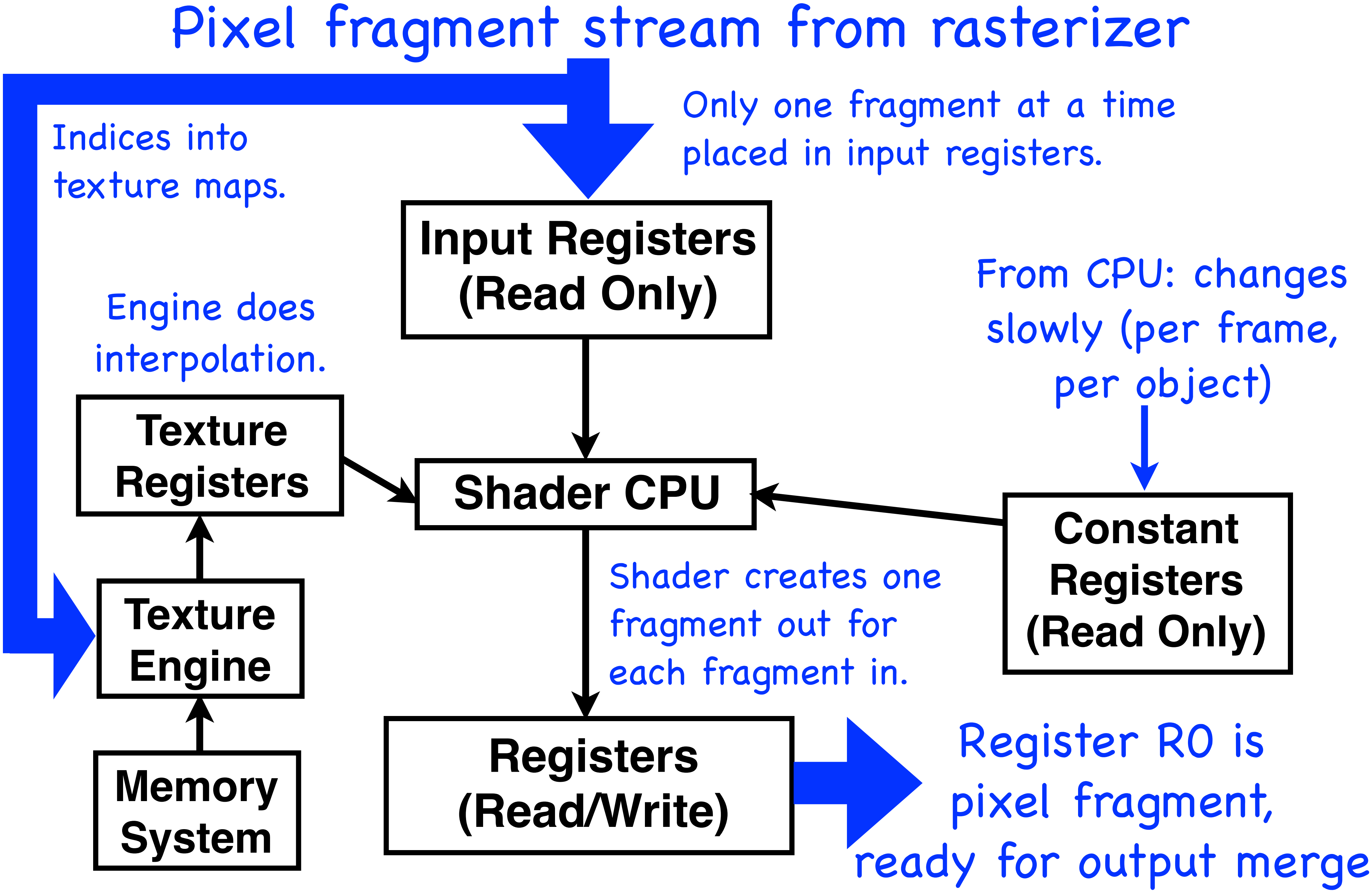


Texture maps (look-up tables) play a key role.

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

DVI/VGA/composite/S-video output port

Pixel Shader: Stream processor + Memory



Pixel fragment stream from rasterizer

Only one fragment at a time placed in input registers.

Indices into texture maps.

Input Registers (Read Only)

From CPU: changes slowly (per frame, per object)

Engine does interpolation.

Texture Registers

Shader CPU

Constant Registers (Read Only)

Texture Engine

Shader creates one fragment out for each fragment in.

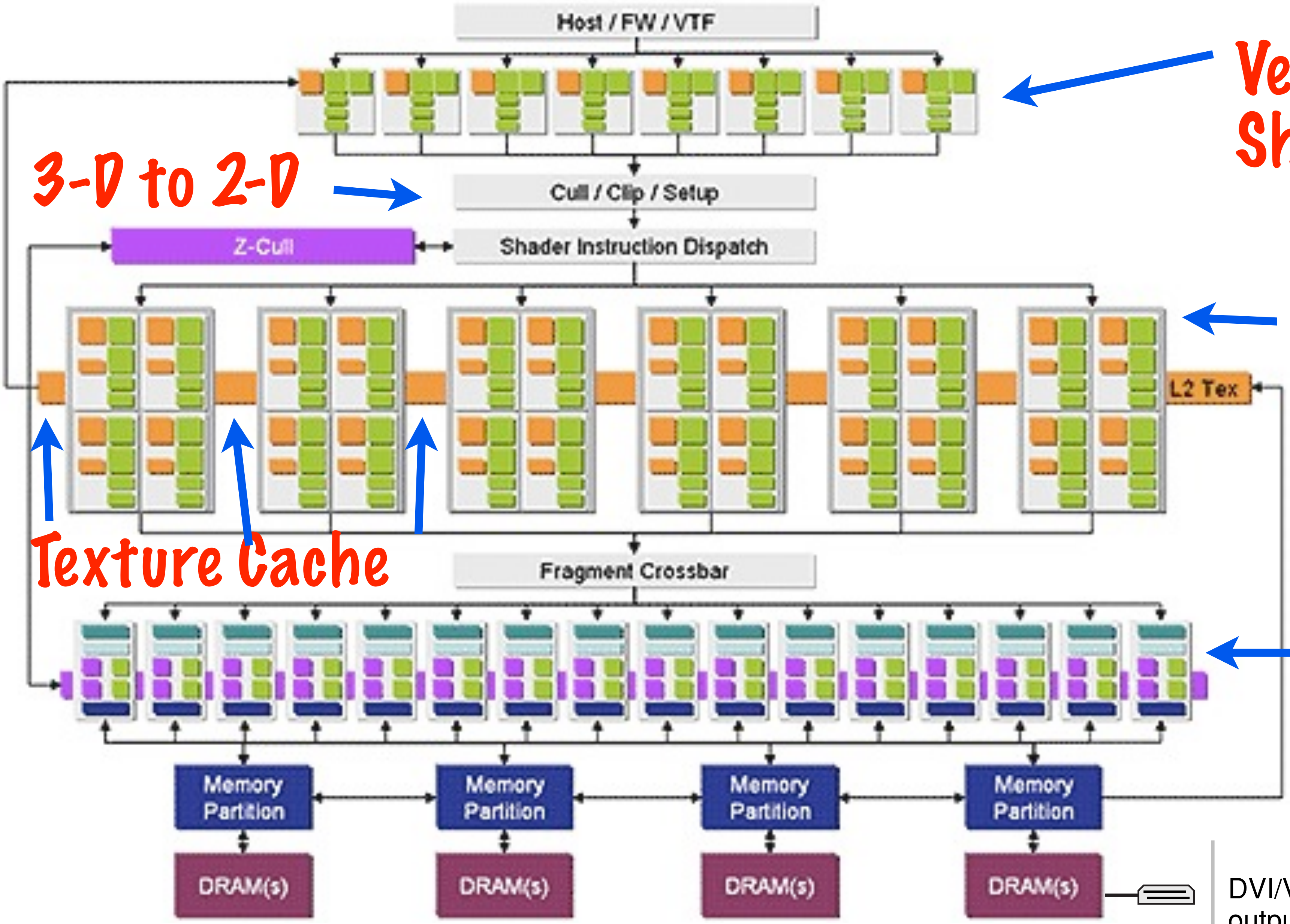
Memory System

Registers (Read/Write)

Register R0 is pixel fragment, ready for output merge

Example Design: Nvidia GeForce 7900

278 Million Transistors, 650 MHz clock, 90 nm process



Vertex Shaders: 8

3-D to 2-D

Pixel Shaders: 24

Texture Cache

Output Merge Units

DVI/VGA/composite/S-video output port

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.

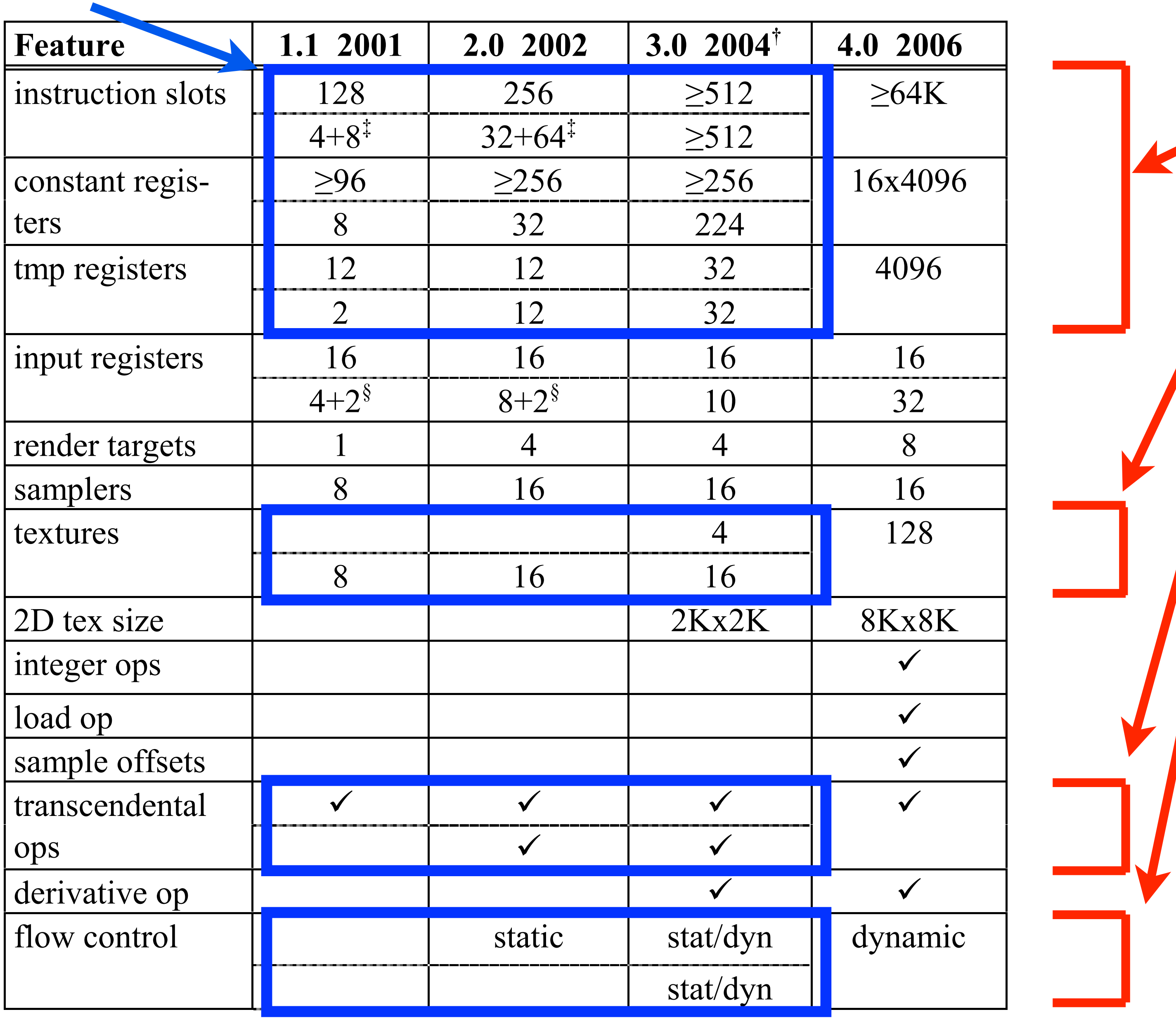
Unified Architectures

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 ...



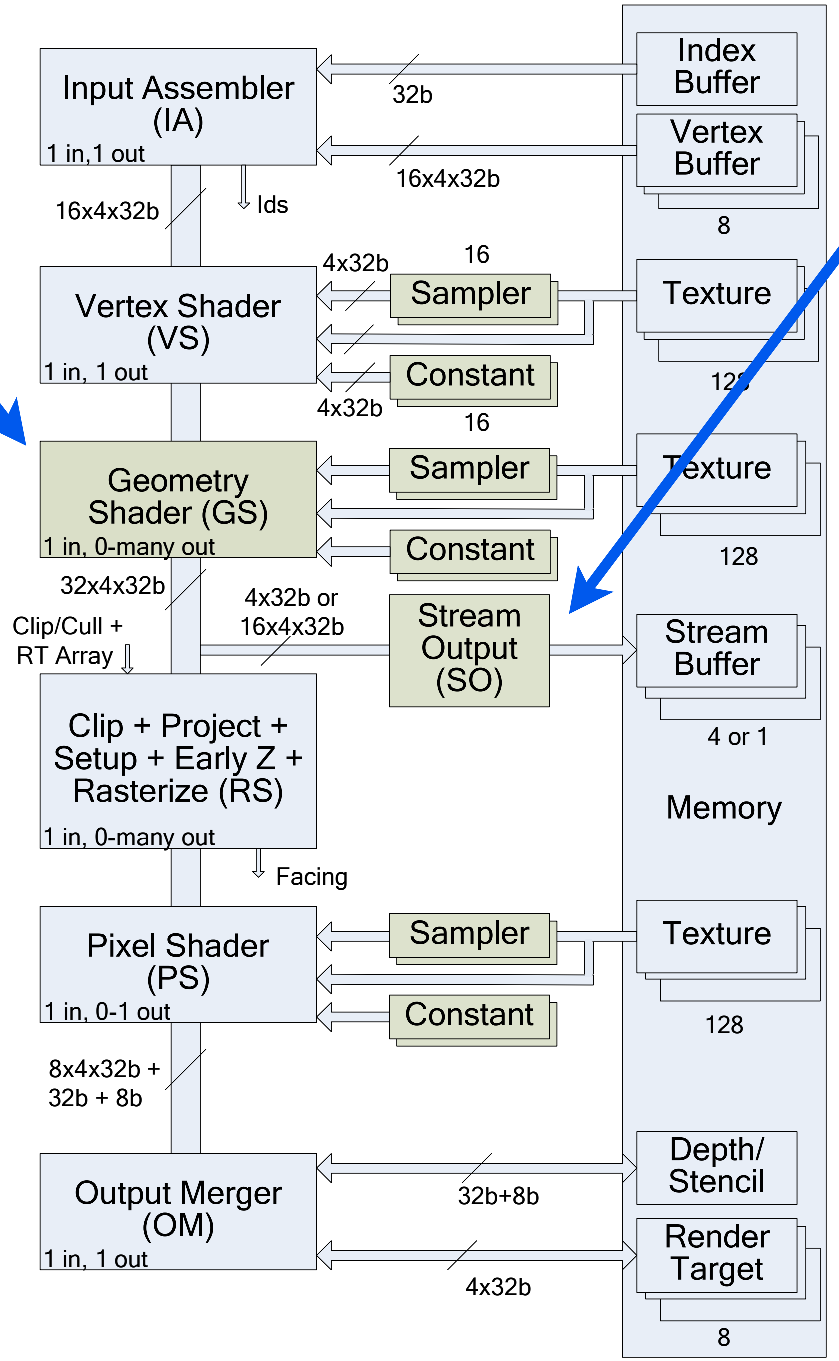
Feature	1.1 2001	2.0 2002	3.0 2004[†]	4.0 2006
instruction slots	128	256	≥512	≥64K
	4+8[‡]	32+64[‡]	≥512	
constant registers	≥96	≥256	≥256	16x4096
	8	32	224	
tmp registers	12	12	32	4096
	2	12	32	
input registers	16	16	16	16
	4+2[§]	8+2[§]	10	32
render targets	1	4	4	8
samplers	8	16	16	16
textures			4	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 ...

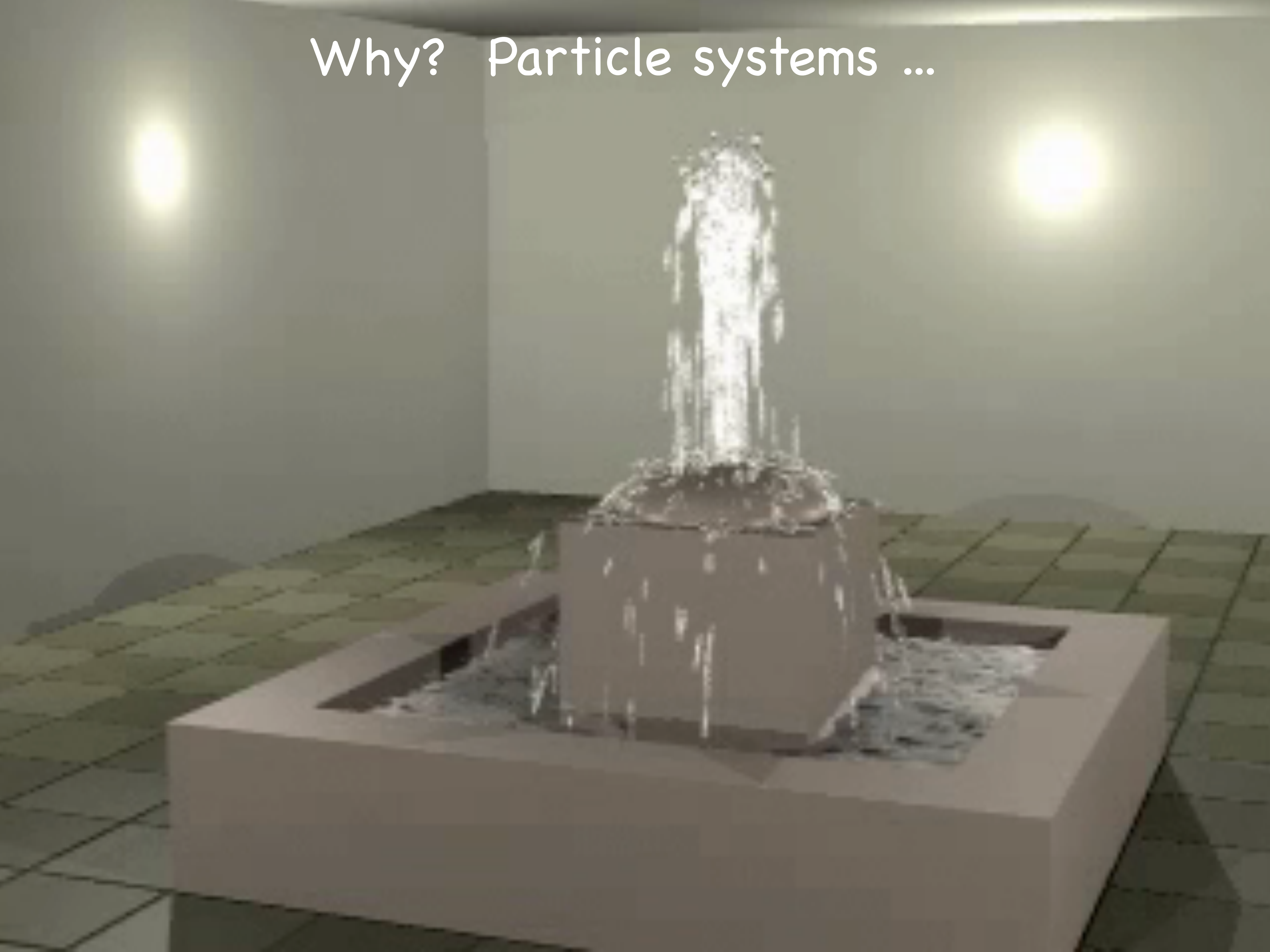
Geometry Shader: Lets a shader program create new triangles.



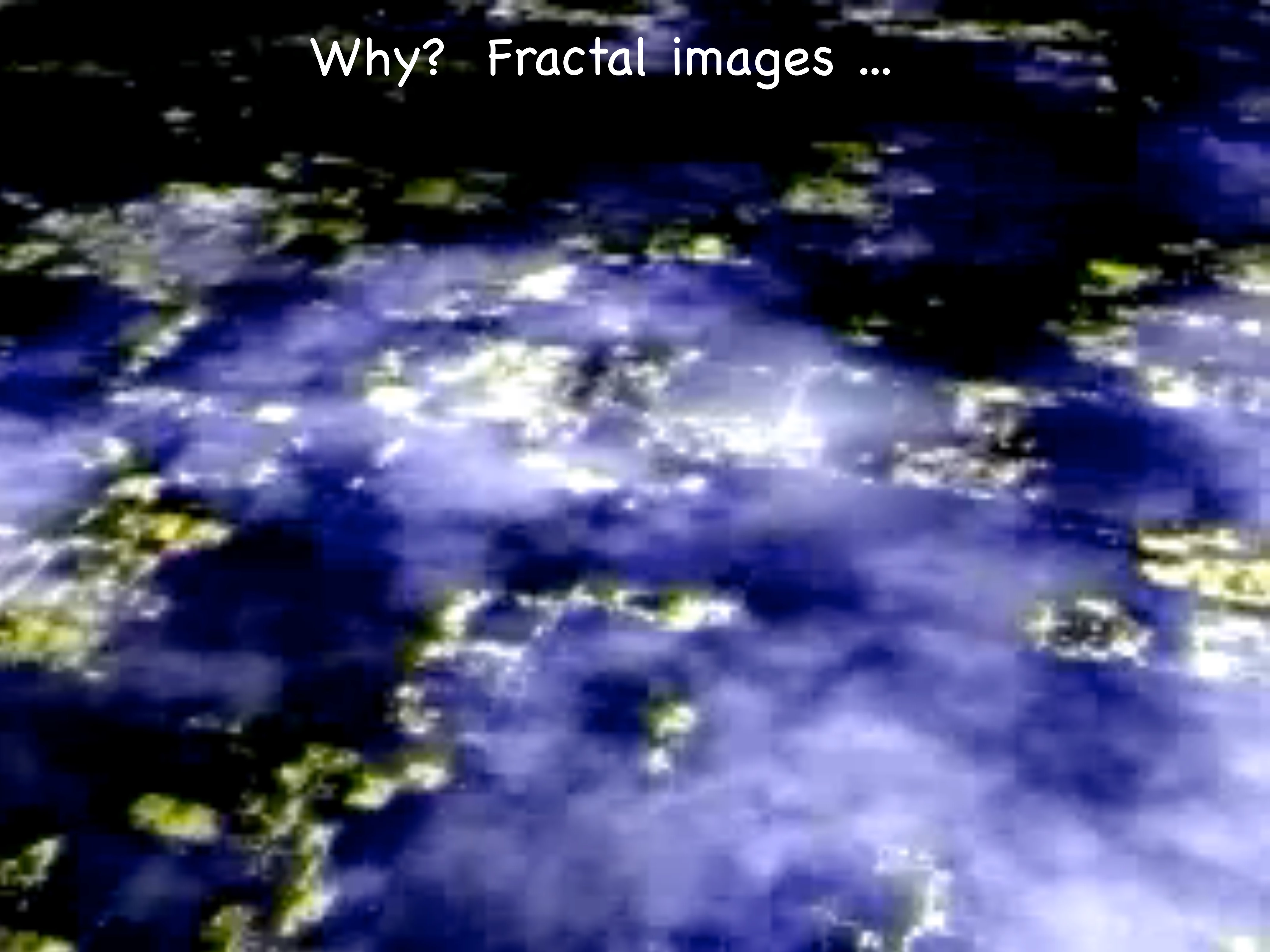
Stream Output: Lets vertex stream recirculate through shaders many times ... (and also, back to CPU)

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

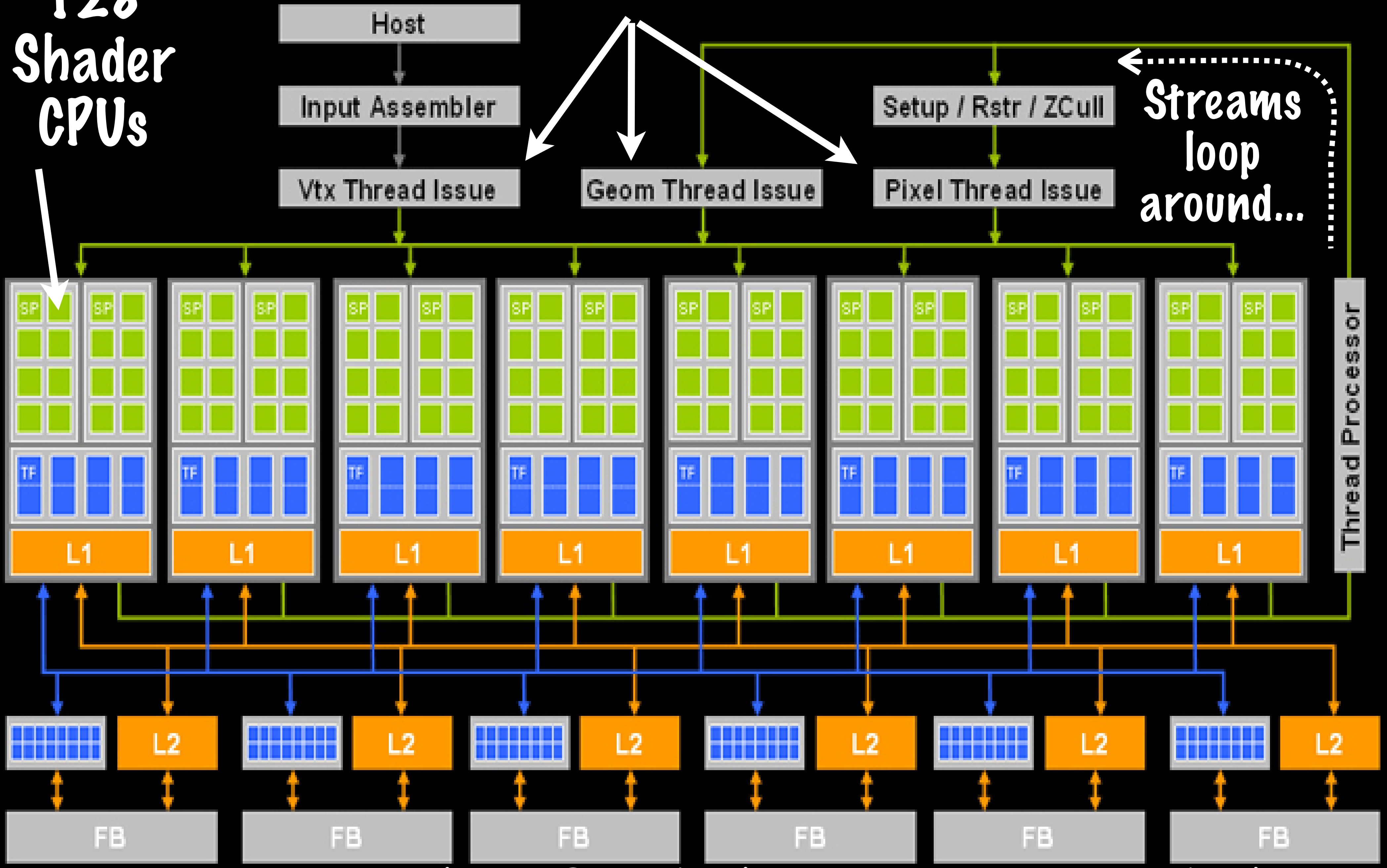
Why? Particle systems ...



Why? Fractal images …



NVidia 8800: Unified GPU, announced Fall 2006



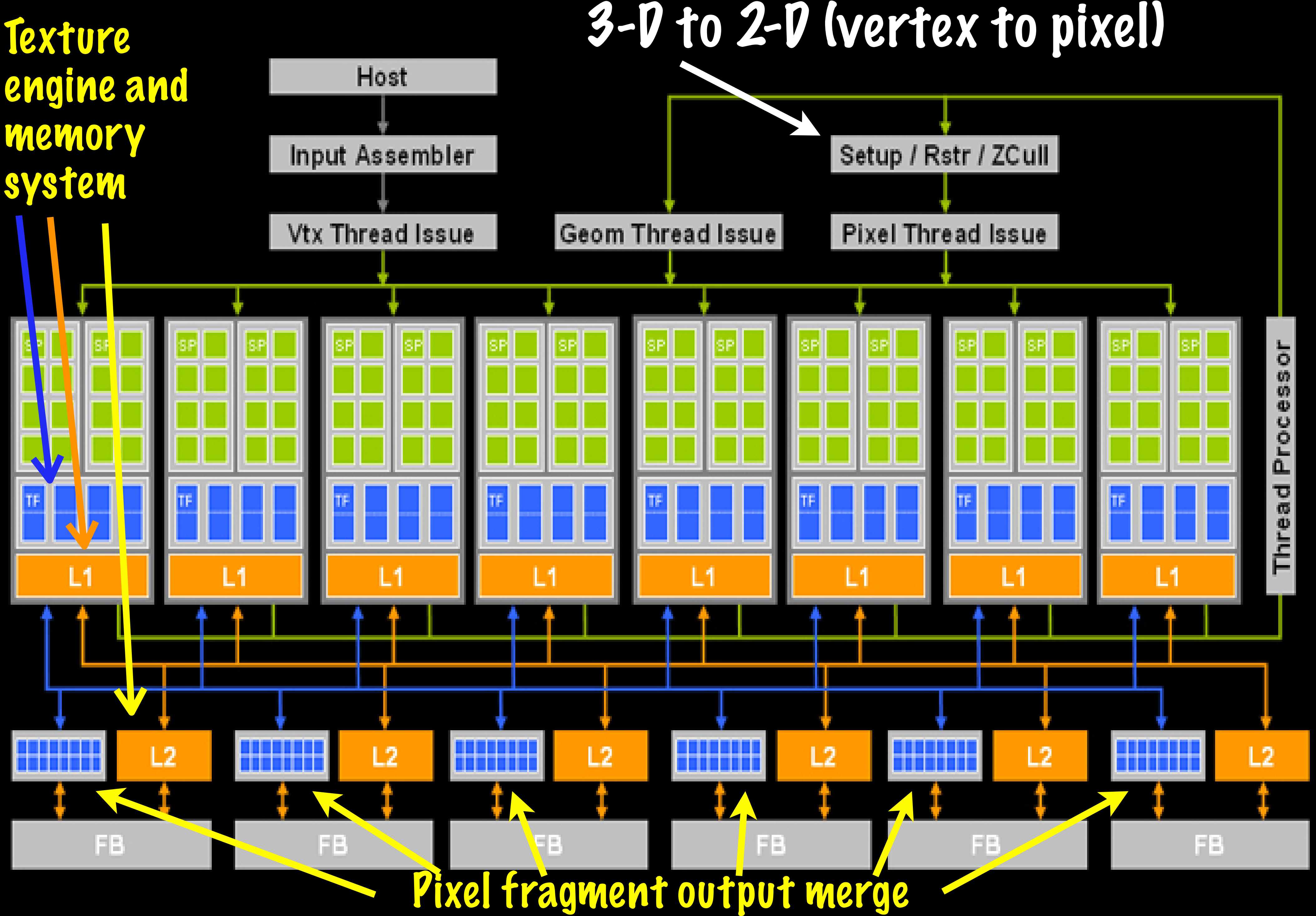
Thread processor sets shader type of each CPU

128 Shader CPUs

Streams loop around...

1.35 GHz Shader CPU Clock, 575 MHz core clock

Graphics-centric functionality ...

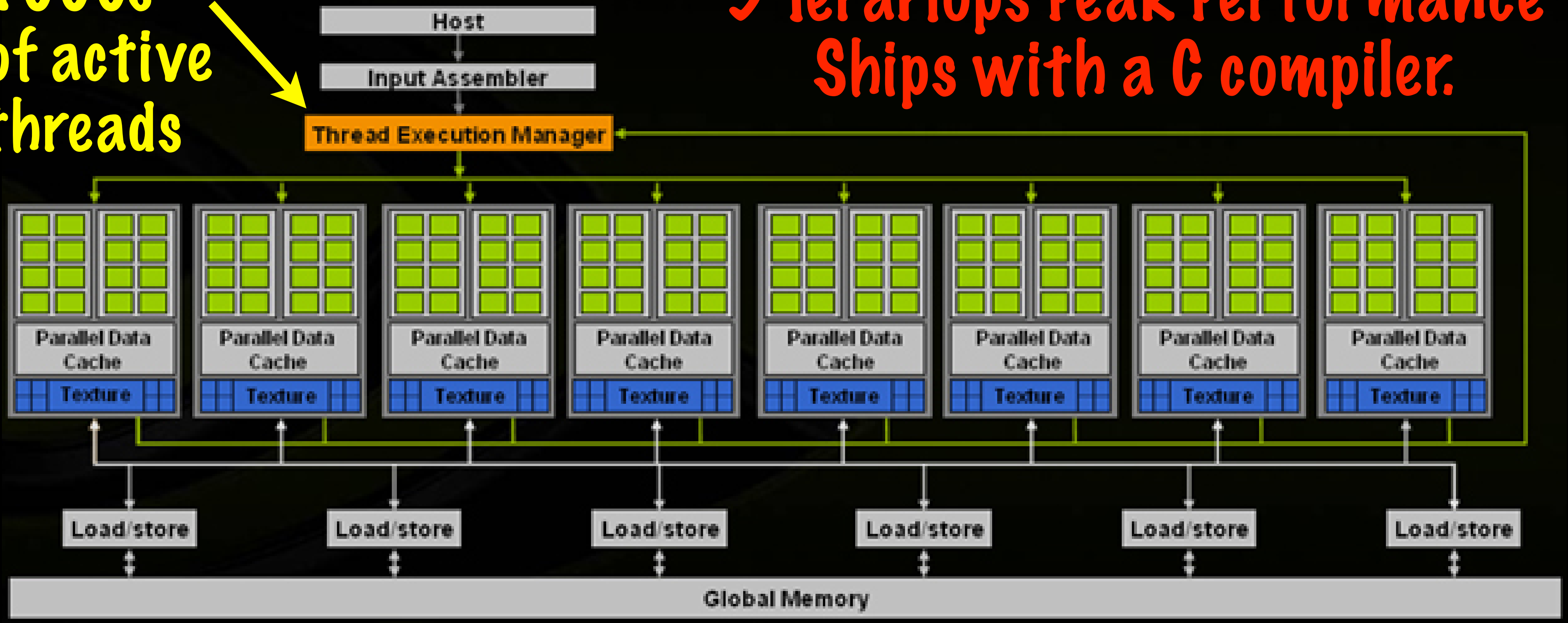


Can be reconfigured with graphics logic hidden ...

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)

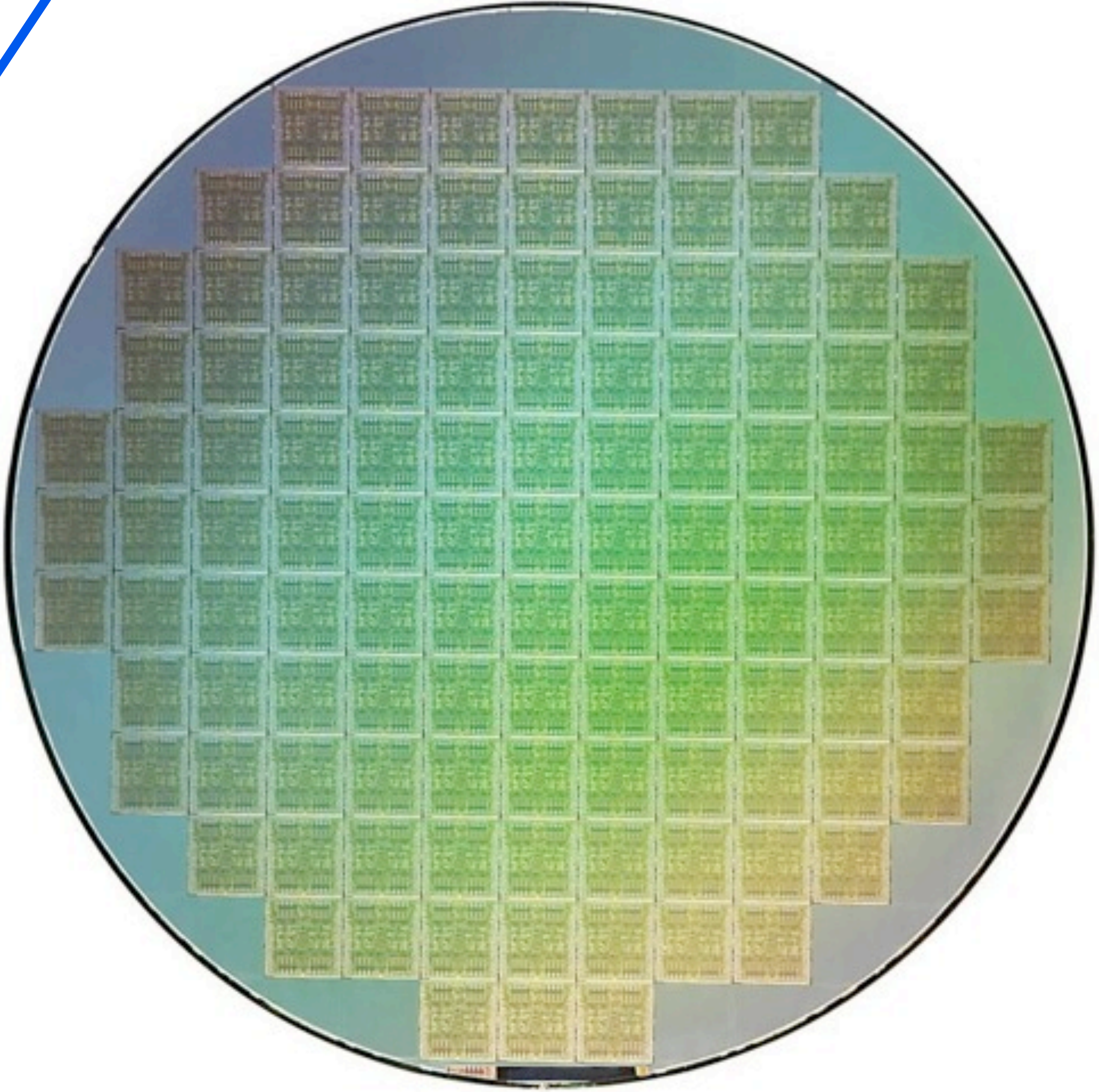
Chip Facts

90nm process

681M Transistors

80 die/wafer (pre-testing)

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



Design Facts

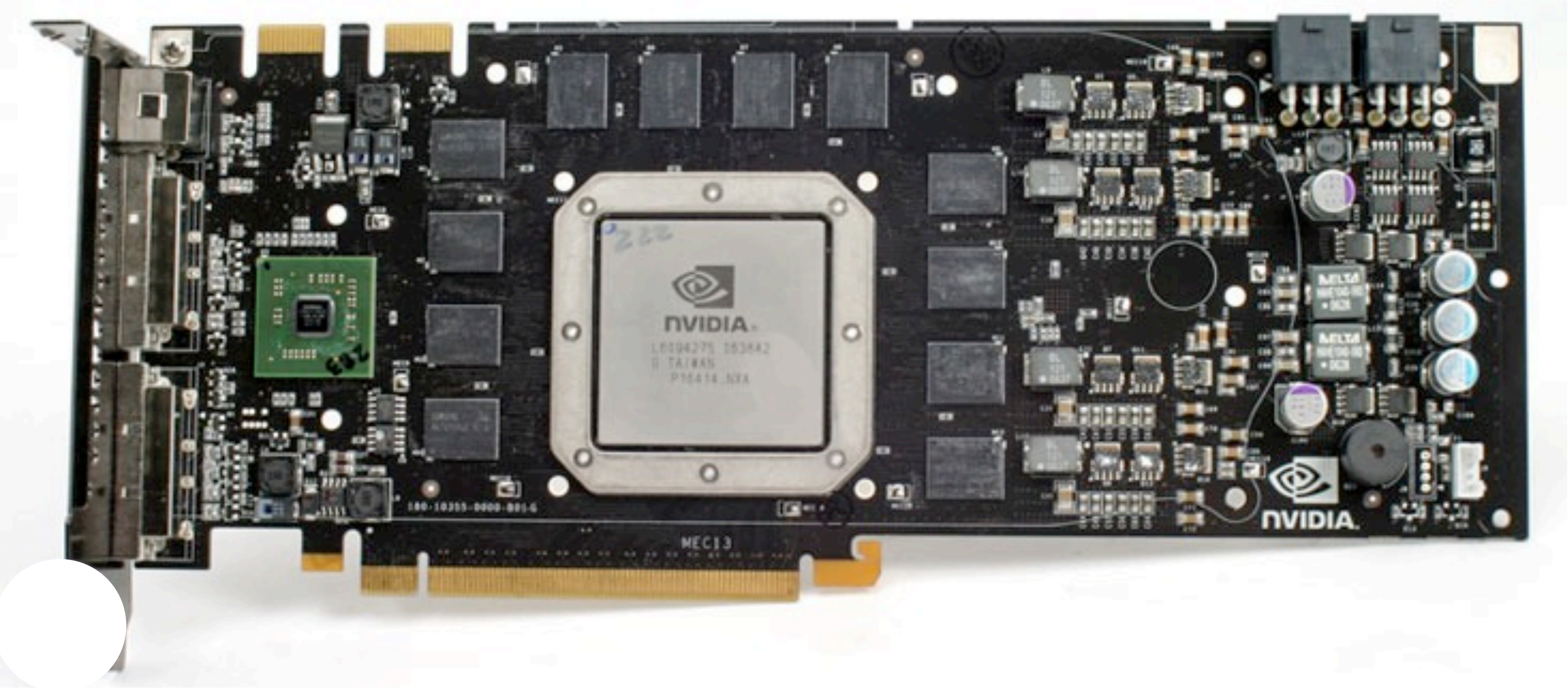
4 year design cycle

$400 Million design budget

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

GeForce 8800 GTX Card: $599 List Price

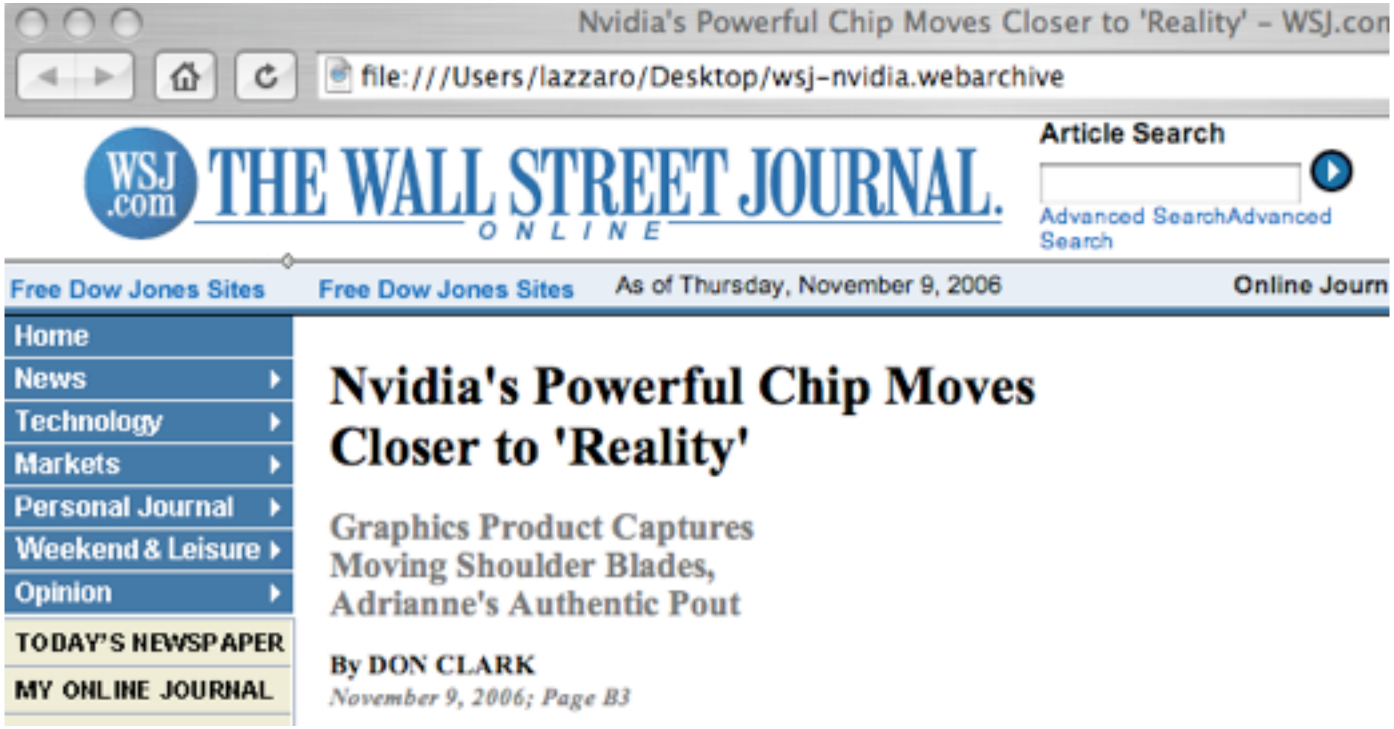
PCI-Express 16X Card - 2 Aux Power Plugs!



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.



Nvidia's Powerful Chip Moves Closer to 'Reality' – WSJ.con

file:///Users/lazzaro/Desktop/wsj-nvidia.webarchive

THE WALL STREET JOURNAL. ONLINE

Article Search

Advanced SearchAdvanced Search

Free Dow Jones Sites Free Dow Jones Sites As of Thursday, November 9, 2006 Online Journ

Nvidia's Powerful Chip Moves Closer to 'Reality'

Graphics Product Captures Moving Shoulder Blades, Adrianne's Authentic Pout

By DON CLARK

November 9, 2006; Page B3

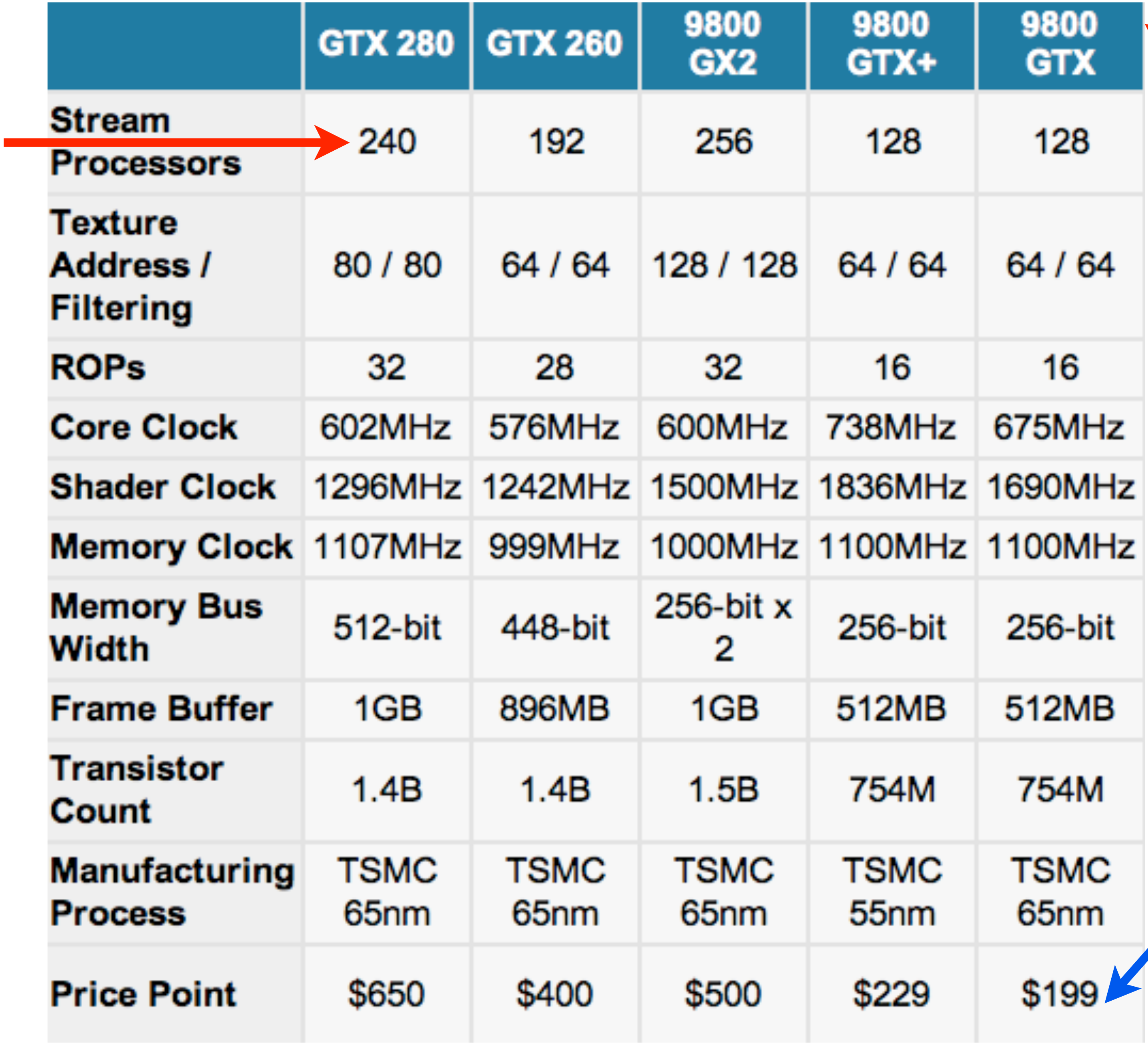
(1) Hope: when chip "shrinks" to 65nm fab process, die will be smaller, yields will improve, profits will rise.

(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



	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

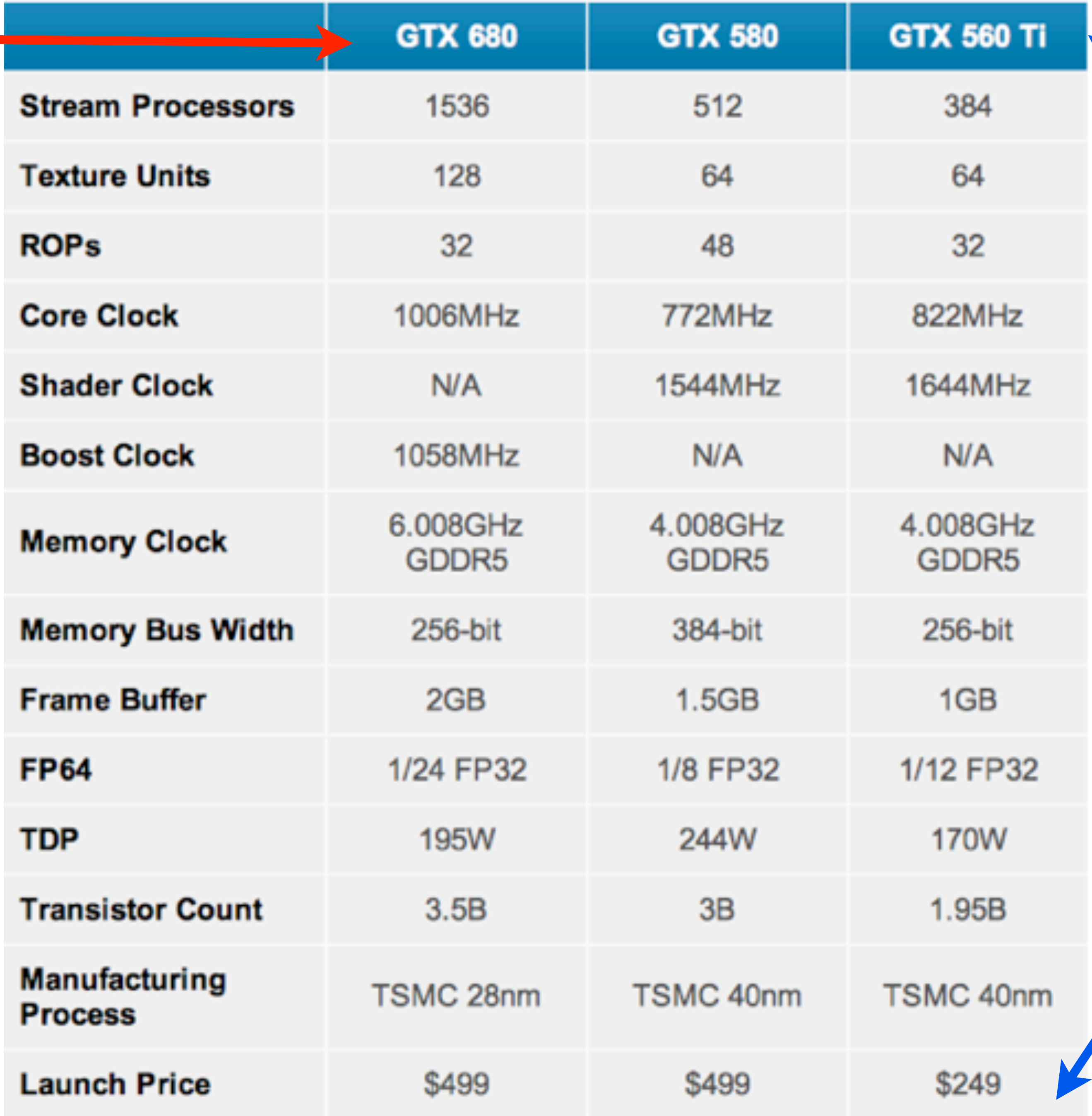
Price similar to 8800, stream CPU count > 2X.

9800 GTX

Specs similar to 8800, card sells for $199.

And again in 2012! GTX 680 -- “Kepler”

GTX 680



	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

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

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

GTX 560 Ti

Specs better than GTX 280, sells for $249

GTX 680

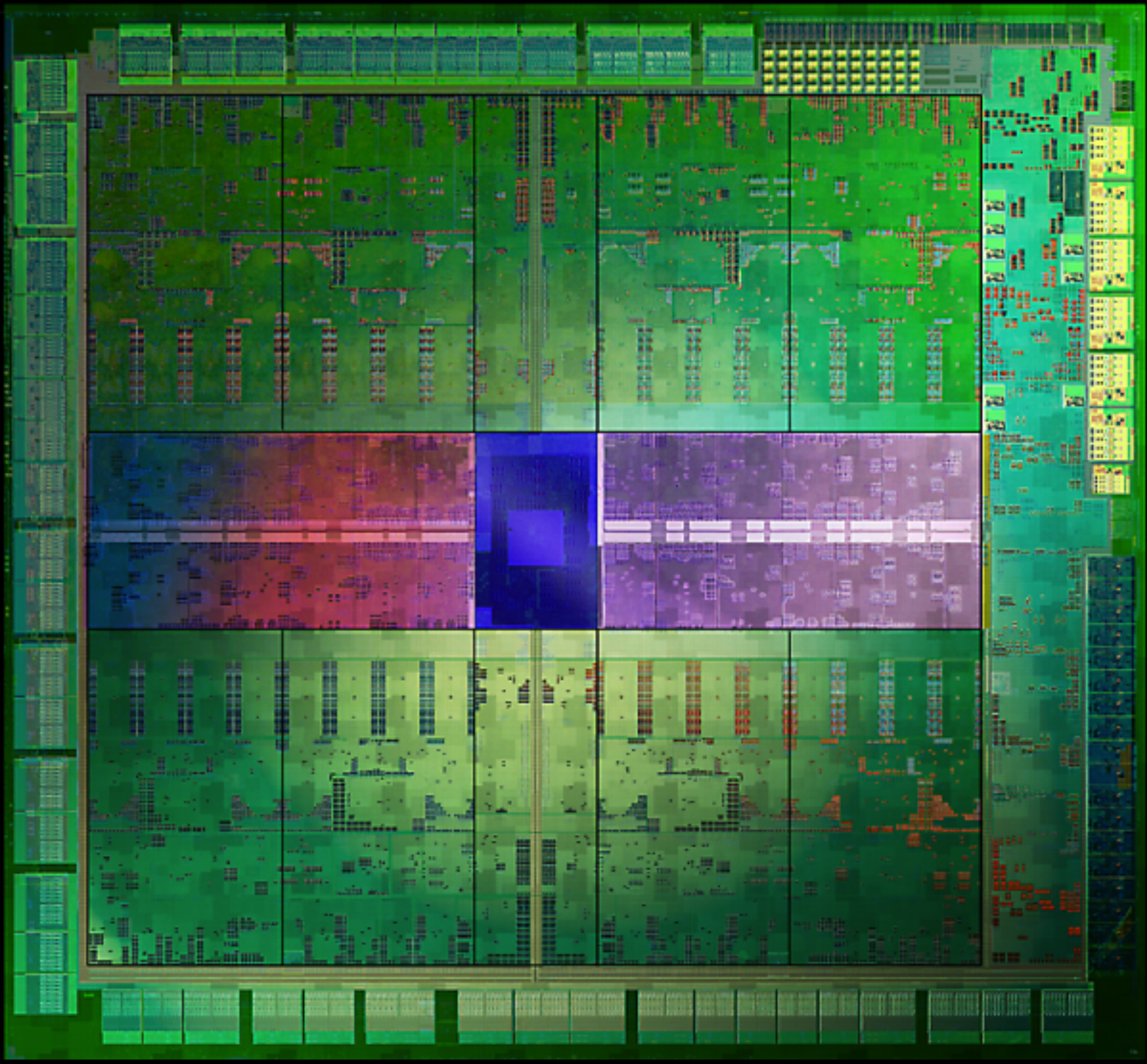
28nm process

3.5 billion transistors

1 GHz core clock

6GHz GDDR5

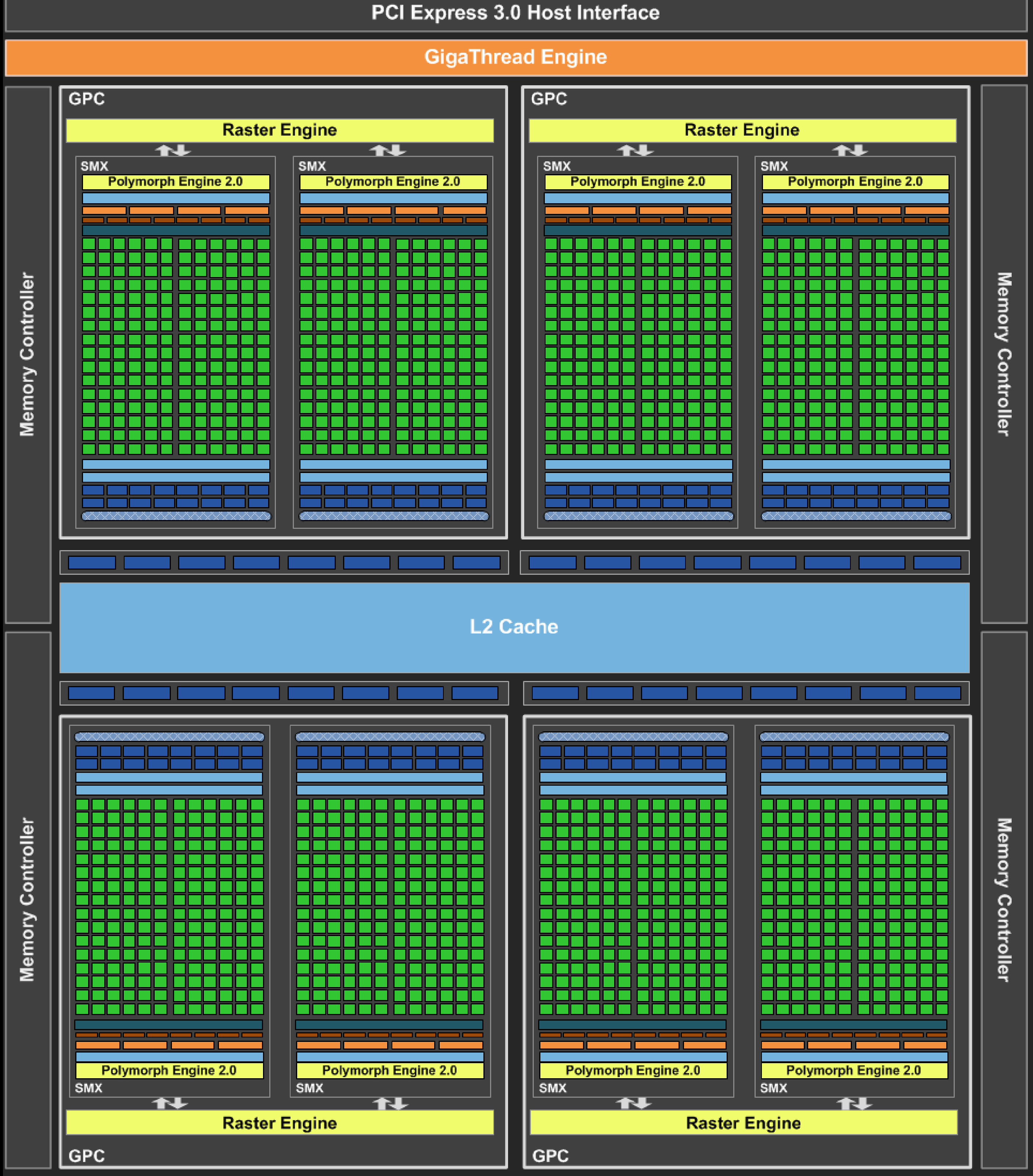
3 years, 1000 engineers



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.

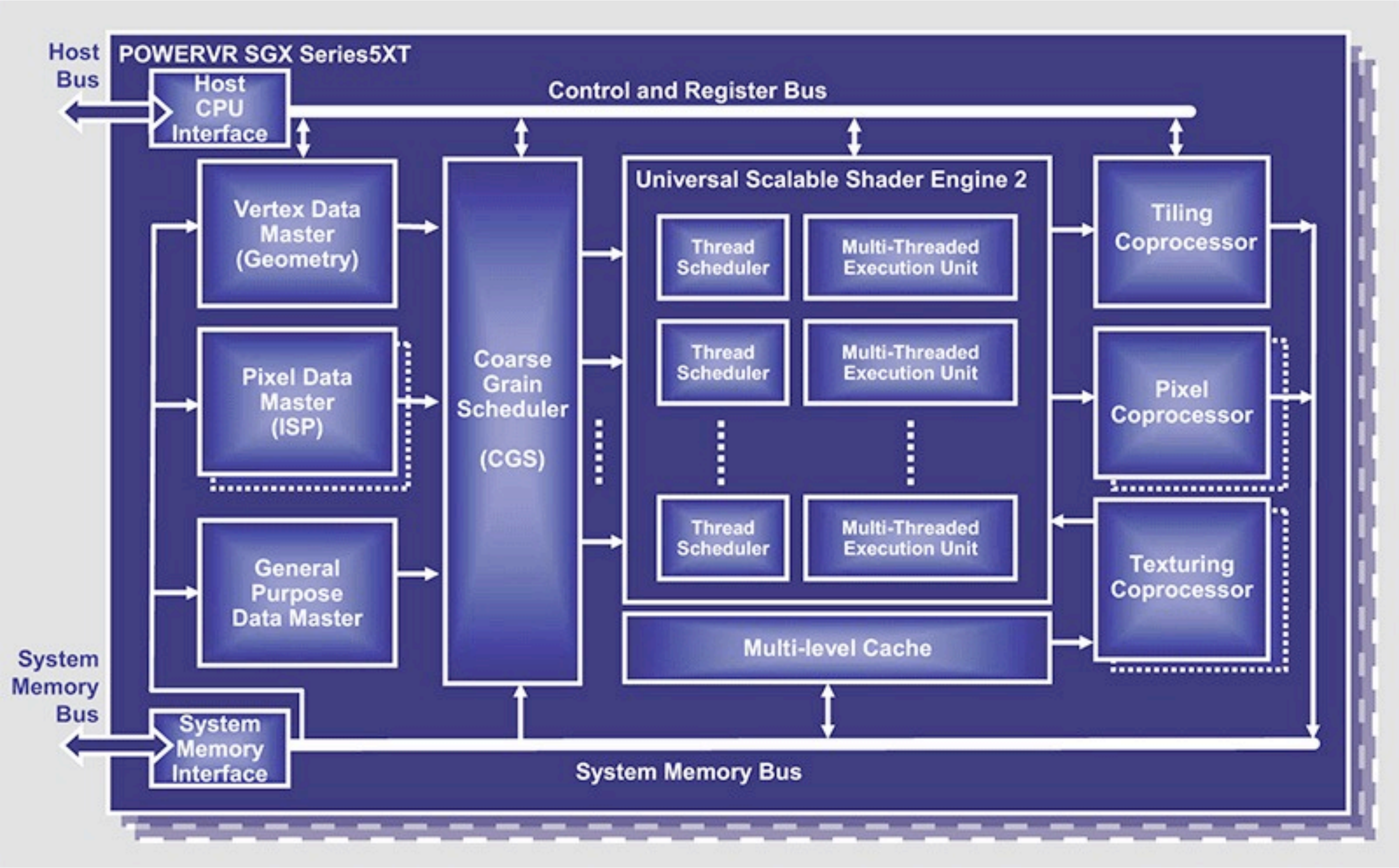
"Wheel of reincarnation" -- Hardwired graphics hardware evolves to look like general-purpose CPU. Ivan Sutherland co-wrote a paper on this topic in 1968!

Samaritan: Direct X-11 demo from Unreal.

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

GPUs on mobile devices

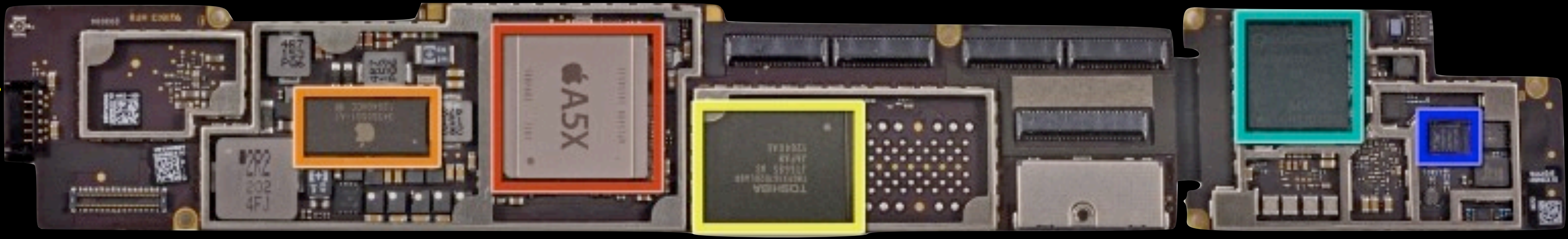
Same ideas, scaled down ...



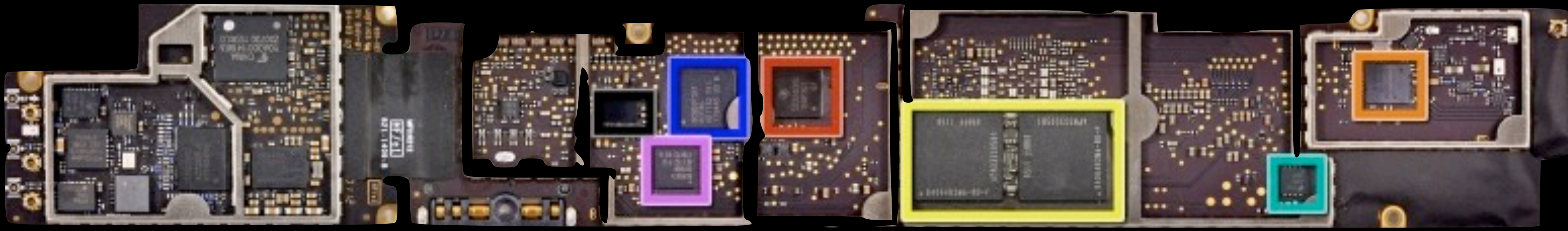
iPad: iPhone++

A5X: 2 ARM Cortex9 Cores, expanded PowerVR GPU

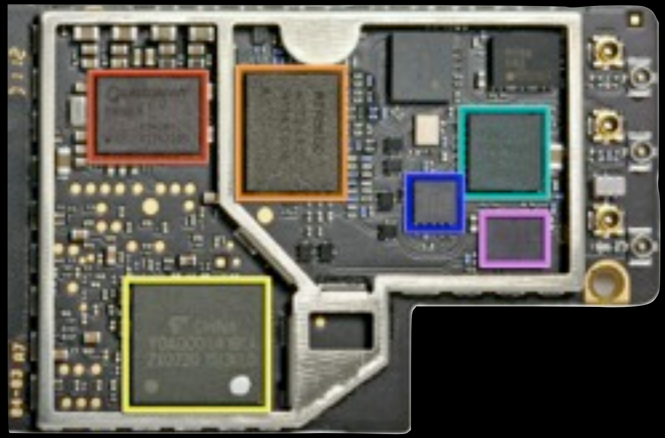
Top



Bottom



Cellular RF



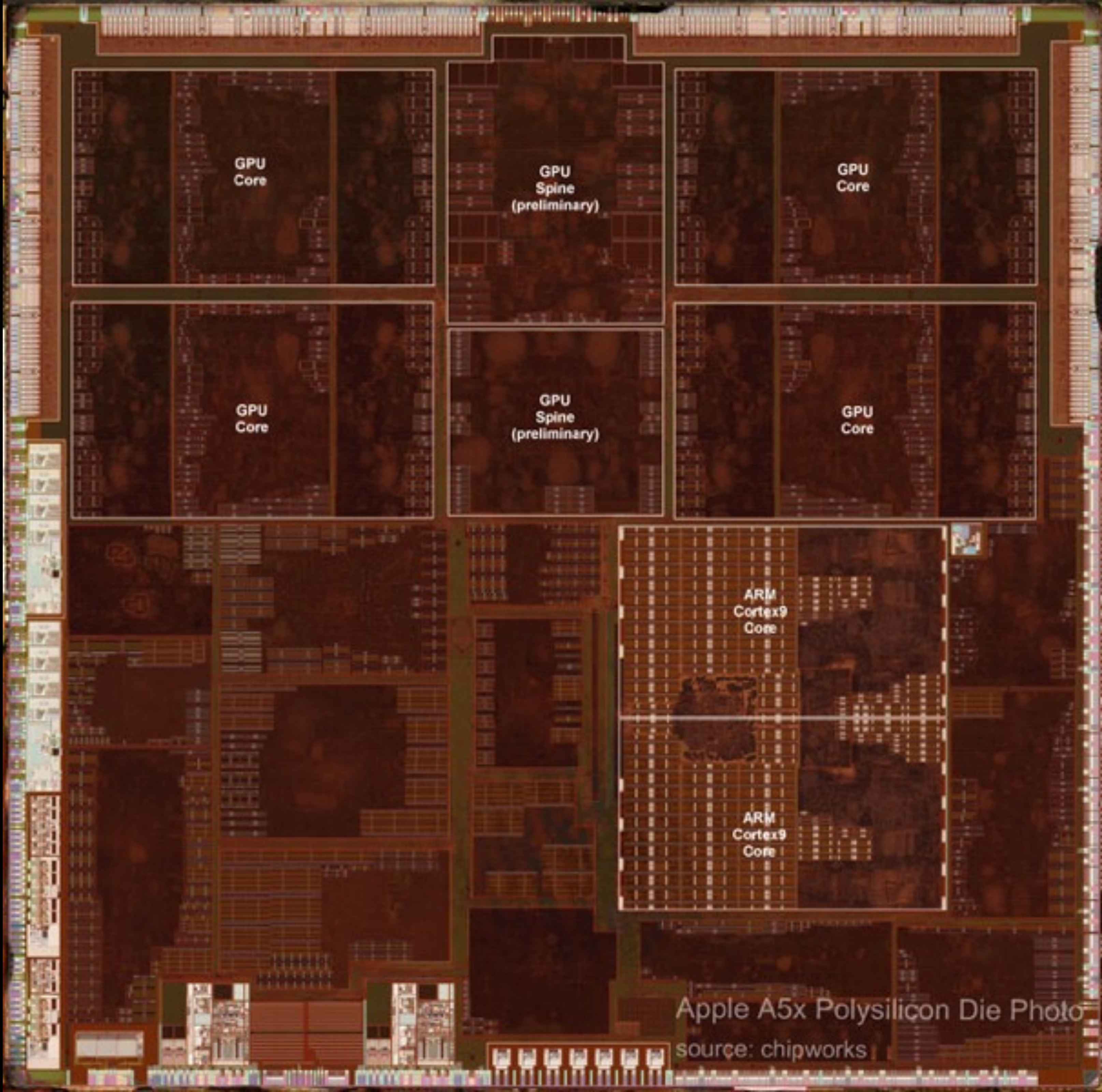
Apple A5X

2012 iPad CPU/IGP.

45 nm,
13 x 13 mm

IGP fills about 40% of die.

IGP: 2% of Kepler (in GFLOPs).



Apple A5x Polysilicon Die Photo
source: chipworks

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.