



nVIDIA®

GeForce 8 系列GPU架构解析

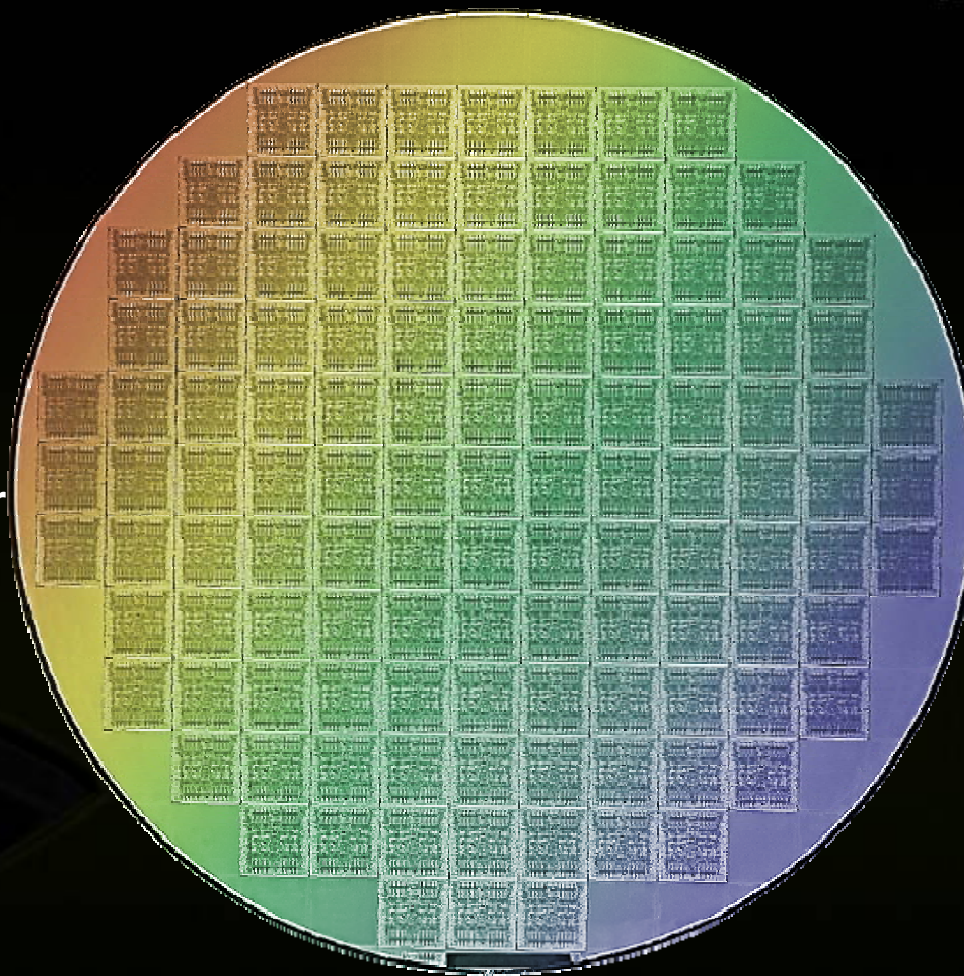
Ashu Rege

Director of Developer Technology

GeForce 8800



- **DirectX 10 / Shader Model 4**
- **统一架构(Unified Architecture)**
- **标量算术单元架构(Scalar ALU Architecture)**
- 非凡的性能
- 世界顶级图像质量
- 为有效利用能源而设计



惊人的真实感图像



华丽、丰富的虚拟世界



出众的物理效果



顶级游戏平台的核心



GeForce 8800

统一架构



图形管线的20年

功能运算



统一管线



Geometry
(新)

物理

顶点

贴图 +
浮点数
运算器

像素

未来

ROP

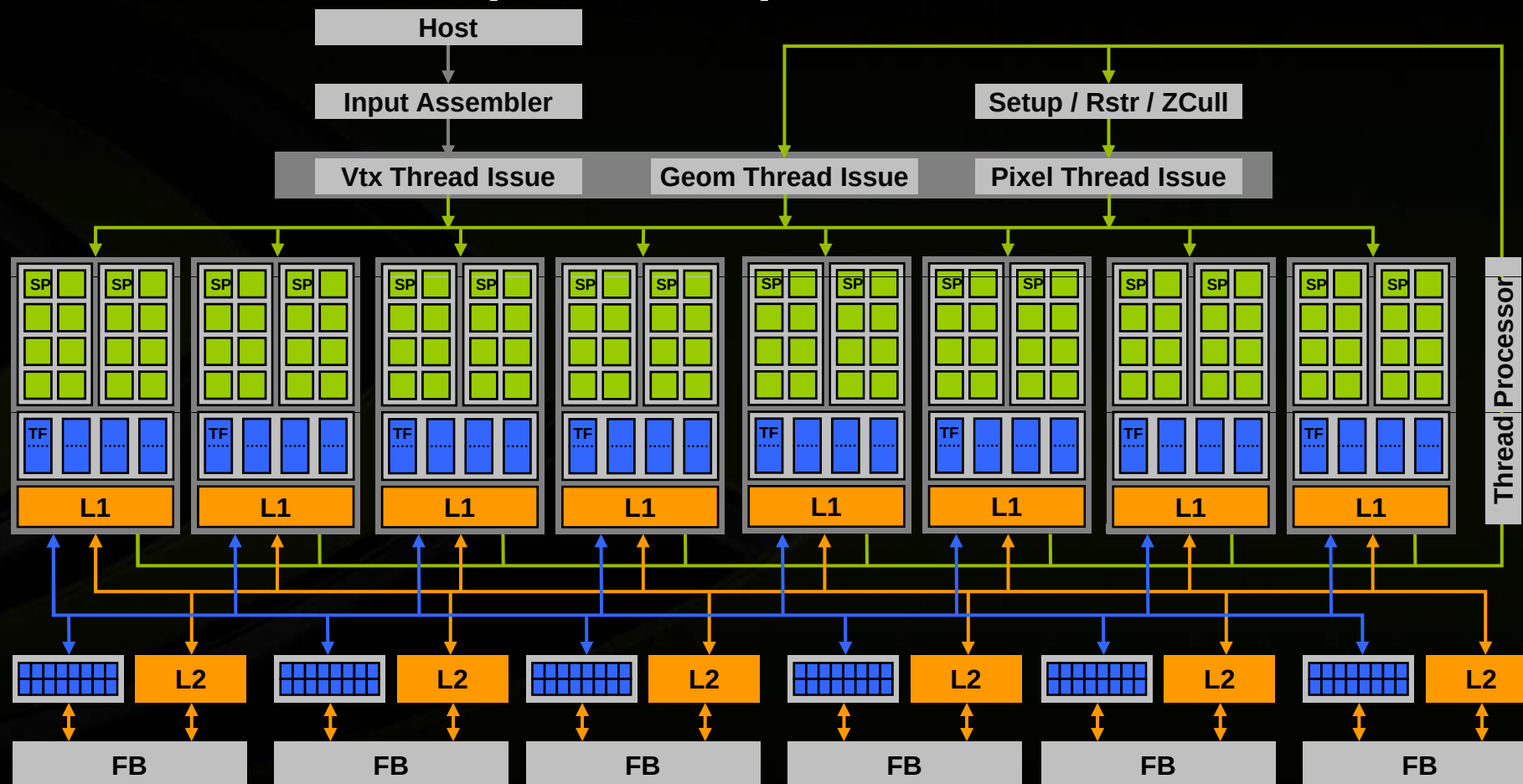
内存



GeForce 8800更新了原有管线的模型



- GPU的未来在于可编程处理
- 所以 – 围绕着处理单元(**Processor**)进行架构

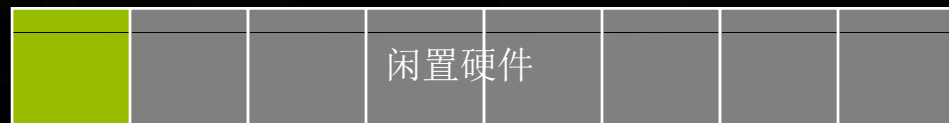


为何要统一？

Vertex Shader



Pixel Shader



Vertex Shader



Pixel Shader



大量几何数据
负载量 = 4



大量像素
负载量 = 8

为何要统一?

统一Shader

		顶点负载			
					像素

统一Shader

		像素负载			
					顶点



大量几何数据
负载量 = 11



大量像素
负载量 = 11

统一



动态负载平衡 - 《英雄连》



较少几何数据



较多几何数据

高pixel shader使用量
低vertex shader使用量



平衡pixel shader
和vertex shader

统一Shader的分配



GeForce 8800

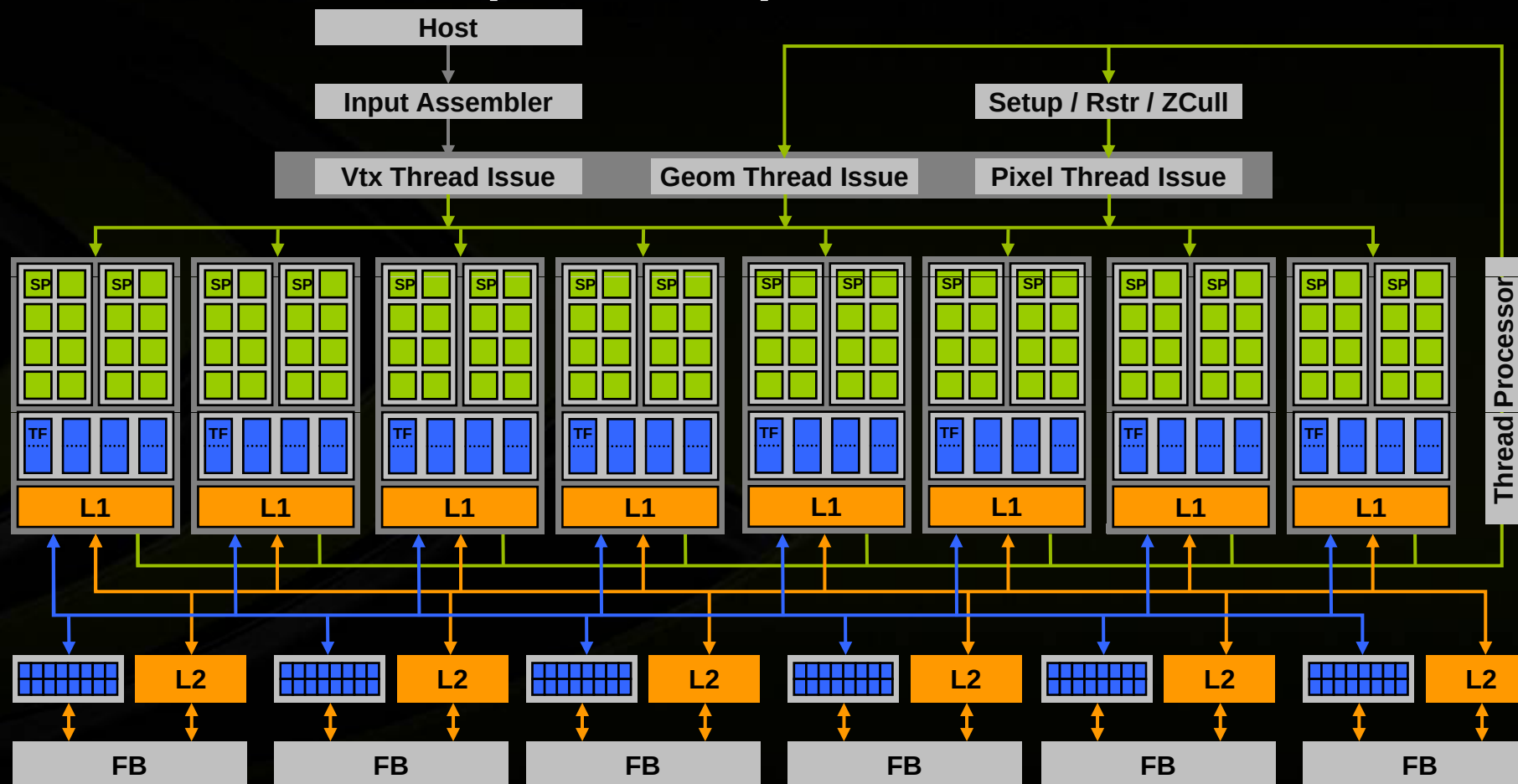
架构细节



GeForce 8800更新了原有管线的模型

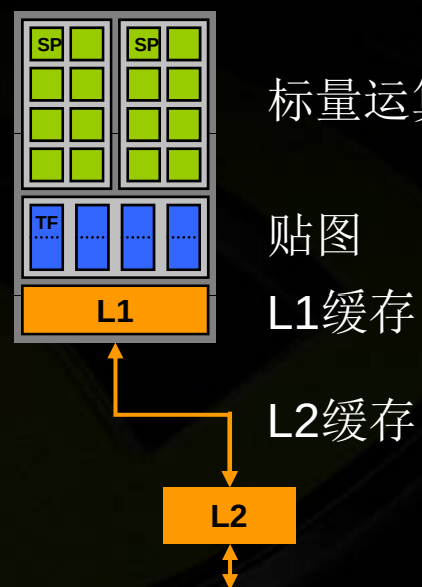


- GPU的未来在于可编程处理
- 所以 – 围绕着处理单元(**Processor**)进行架构





基本运算单元: Texture-Shader Unit



- GeForce 8系列GPU的基本运算单元
 - GeForce 8800 GTX含有8个此种单元
- 每单元含有16个标量流处理运算器(Scalar Streaming Processors, 简称SP)
 - GeForce 8800 GTX总共128个SP
- 每单元含有4个双线性贴图采样器
 - GeForce 8800 GTX总共有32个
- 在GeForce 8800 GTX中, 运行于1.35 GHz
- 完全统一和通用
 - Vertex, Geometry和Pixel Shaders
- 完全分离的贴图和浮点操作
- 出色的分支指令性能
- 完全符合IEEE 754浮点数标准

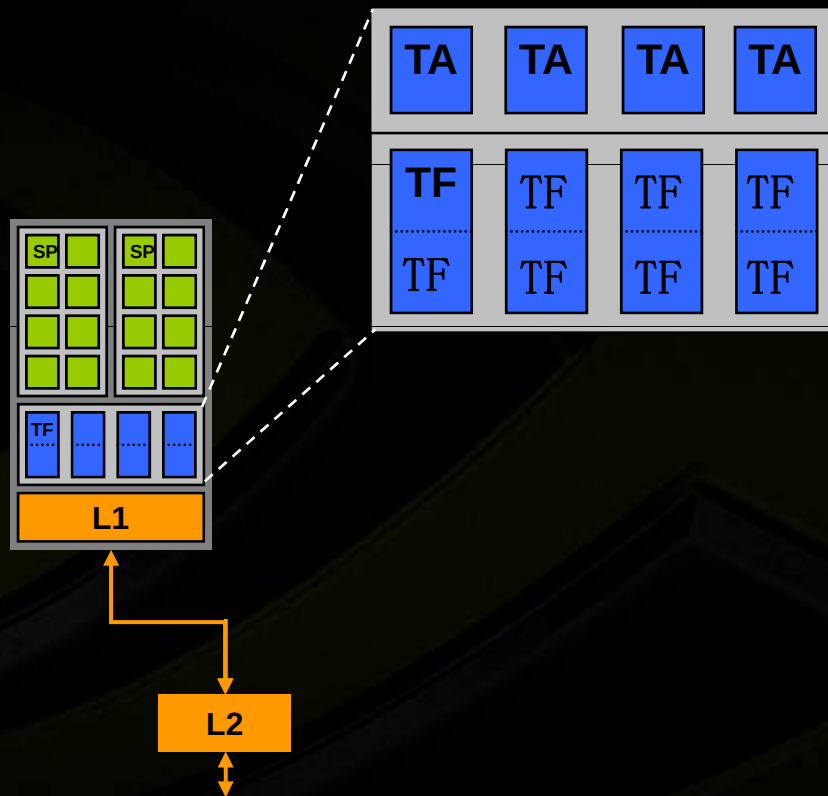


GeForce 8800

贴图处理



贴图



- 贴图地址: **32**采样/周期
- 贴图过滤: **64**采样/周期
 - **36.8 GBilerps/秒**
- 为**HDR**优化
- 支持**fp16**和**fp32**格式
- 全速各向异性过滤 (**2:1**)
- 全速**fp16 HDR**
- 最佳图像质量

贴图读取速率(Texture Fill Rates)



32b Dual Tex	7900gtx	8800gtx	8800GTX / 7900GTX
Bilinear	7571	9063	1.2x
2x Aniso	3834	9062	2.4x
16x Aniso	553	1147	2.1x
2x Aniso Perf	51%	100%	
64b Dual Tex			
Bilinear	3838	9058	2.4x
2x Aniso	1925	4568	2.4x
16x Aniso	283	575	2x
64b Bilinear Perf	51%	100%	



Crysis. Images courtesy of Crytek

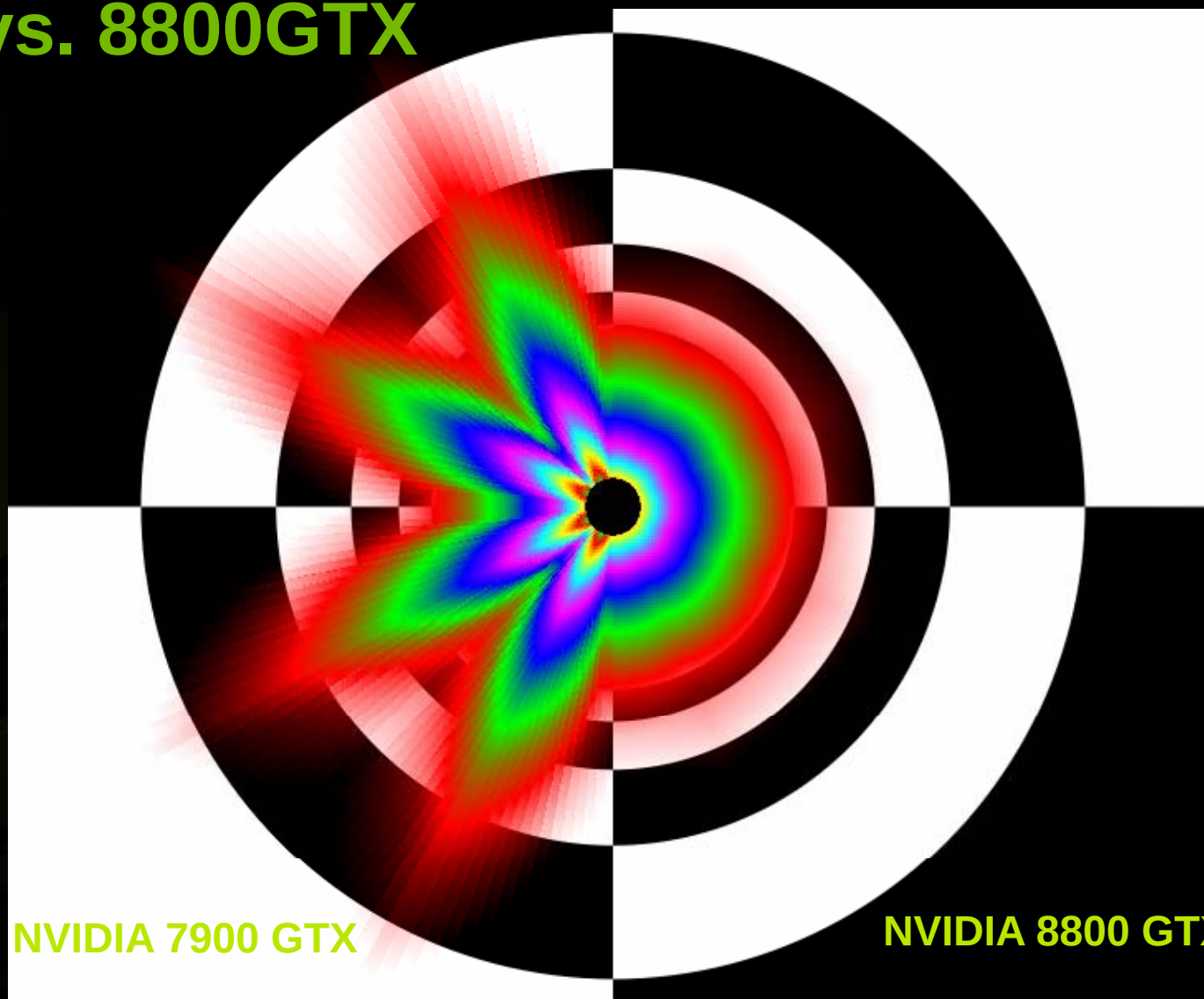
贴图过滤质量

7900GTX vs. 8800GTX



NVIDIA 7900 GTX

NVIDIA 8800 GTX



最高图像质量对比



x1950xtx LOD质量

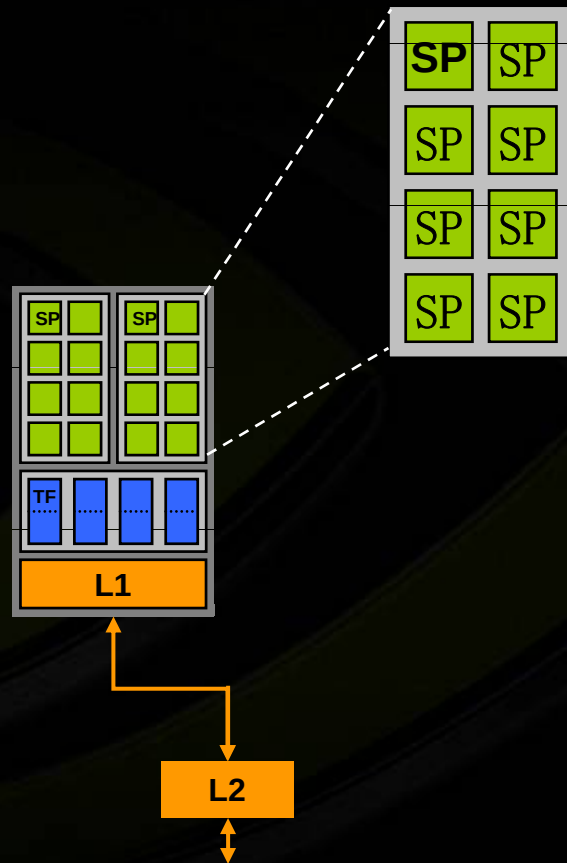
8800 GTX LOD质量



GeForce 8800 Shader



Shader



- 128标量流处理运算器(Scalar Streaming Processors)
- 在GeForce 8800中运行于1.35 GHz
- 完全统一和通用
- 完全分离的贴图和浮点操作
- 出色的分支指令性能
- 完全符合IEEE 754浮点数标准

一些原始数据

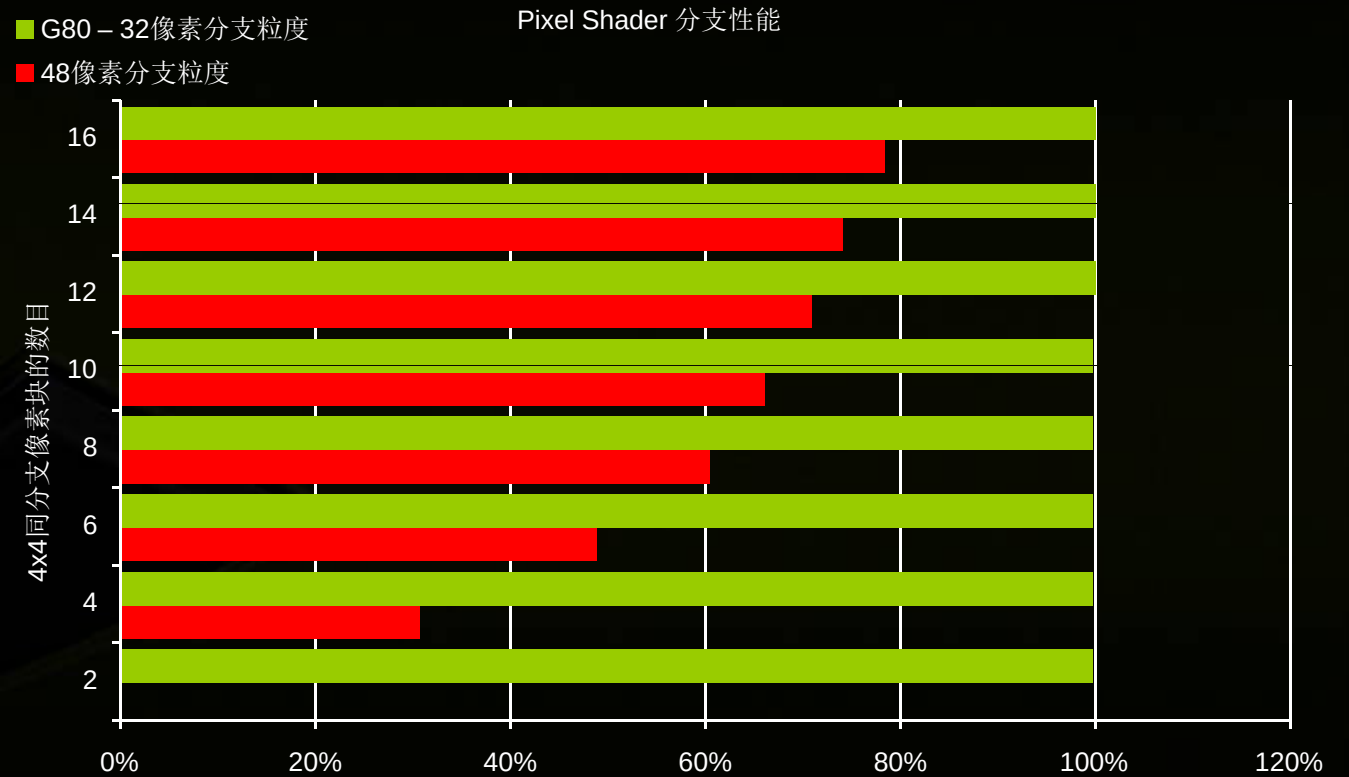


	Geforce 7900	GeForce 8800
结构	Vec3+scalar MAD	Scalar MAD + Scalar MUL
运算单元数目	48	128
主频(GHz)	.65	1.35
GMuls	125	345

分支性能



- GeForce 8800的分支粒度为32 像素
- 与分支粒度为48像素的架构作性能比较



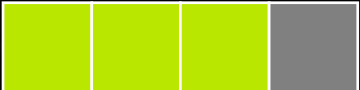
为何用标量Shader?

● 矢量算术单元(Vector ALU) – 效率时高时低

●

● 4  MAD r2.xyzw, r0.xyzw, r1.xyzw – 100%

●

● 3  DP3 r2.w, r0.xyz, r1.xyz – 75%

●

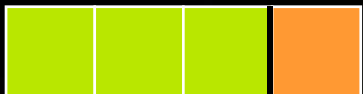
● 2  MUL r2.xy, r0.xy, r1.xy – 50%

●

● 1  ADD r2.w, r0.x, r1.x – 25%

为何用标量Shader?

- 矢量算术单元使用并行指令(co-issue) – 效率有所改善但不完美

4  DP3 r2.x, r0.xyz, r1.xyz } 100%
ADD r2.w, r0.w, r1.w

3  DP3 r2.w, r0.xyz, r1.xyz

1  ADD r2.w, r0.w, r2.w

不能使用并行指令

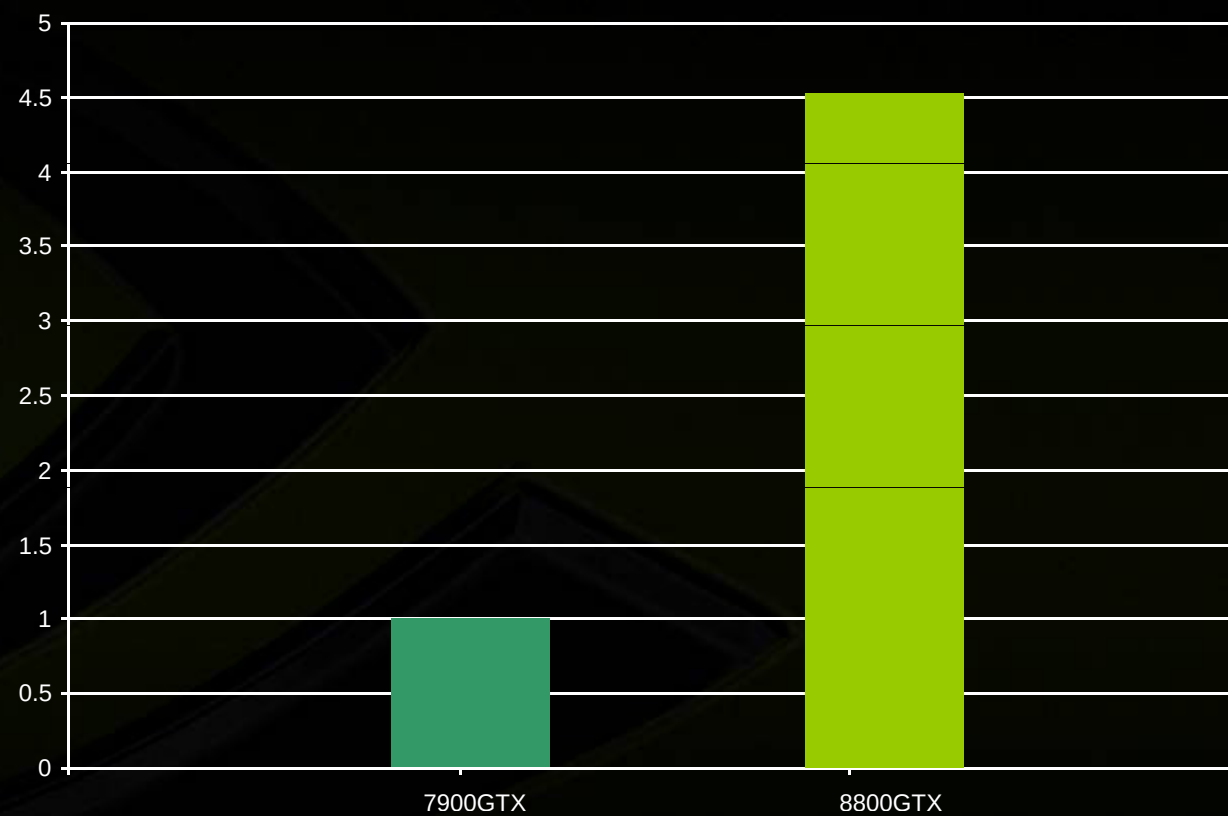
- 矢量/超长指令字(Vector/VLIW)架构 – 需要复杂的编译器支持
- G8x : 标量 – 总是100%高效, 更易于编译
- 最高可达到矢量构架的吞吐率的两倍

复杂Shader的性能

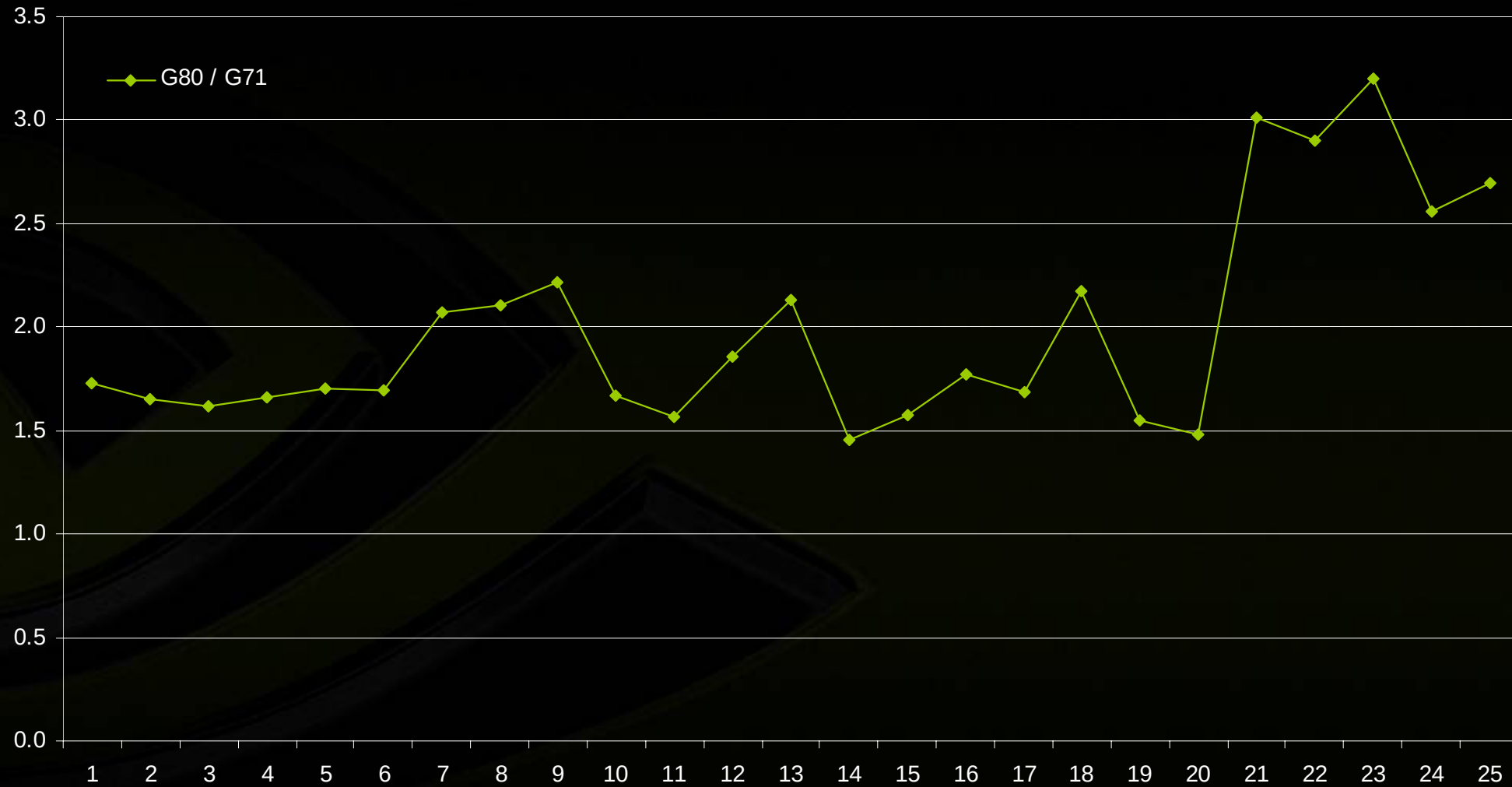
过程Perlin Noise火焰



过程火焰



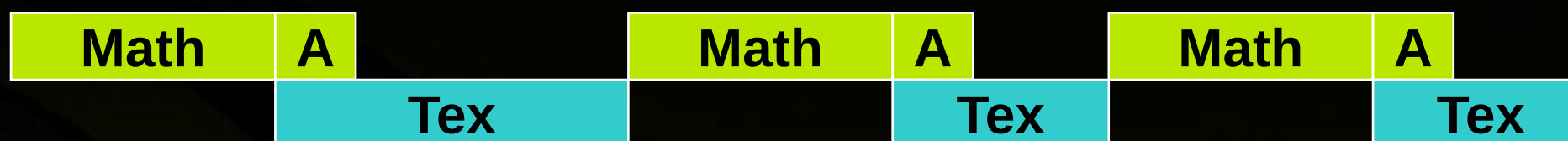
Shadermark性能测试 8800GTX / 7900GTX





Shader: 分离化 与 多线程化

- GeForce 7 系列架构



- GeForce 8 系列架构



- GeForce 8 架构: 上千条线程并行执行

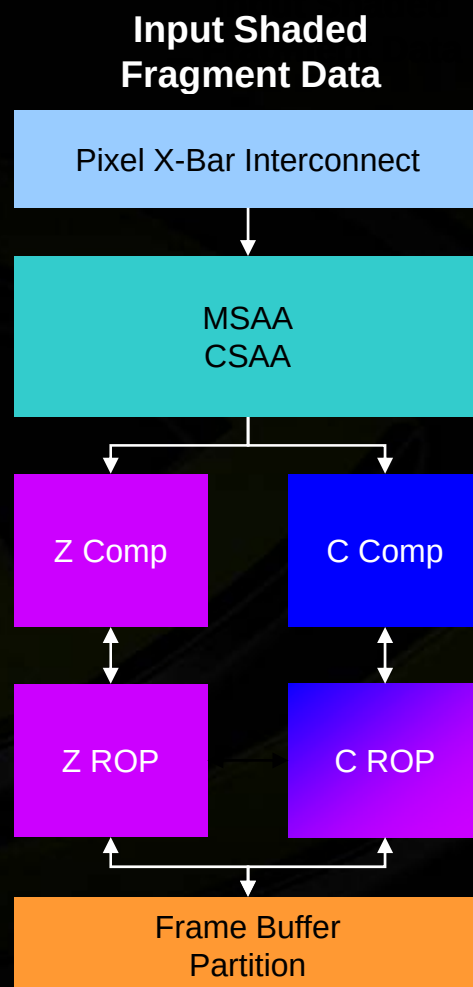


GeForce 8800 ROP





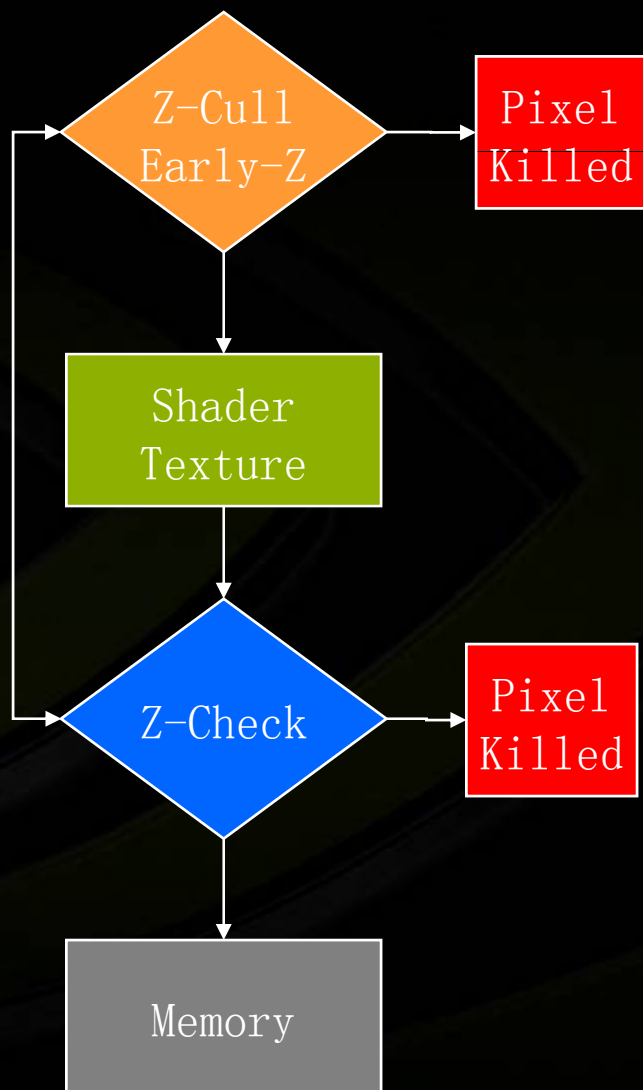
一个ROP单元的像素处理



- 支持**FP16**和**FP32**格式
- 最高支持**16**倍多重采样(抗锯齿)
- 可同时使用**AA**(抗锯齿)和**HDR**(高动态光照)
- 每显存分区**16**色彩采样 + **16**深度采样
- 单独深度采样: 最高达**32**采样/周期(每显存分区)
- 提供新型色彩和深度的带宽压缩(**2x**)
- **MRT**最多支持更多**Render Targets (8)**
- 每时钟周期**96**色彩采样 + **96**深度
- 单独深度采样: **192**采样/周期



深度预处理(Early Z)



- 深度缓存(Z-Buffer)可以帮助提前移除不可见的像素
- 深度剔除(Z-Culling)可以以极高的速度移除不可见区域
- 深度预处理(Early-Z)可以在单个像素被渲染之前进行深度比较，移除不可见像素



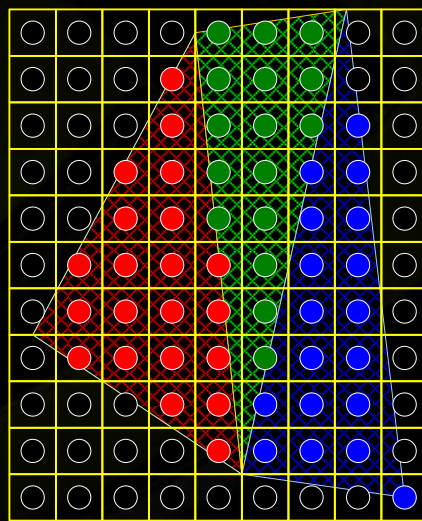
新一代抗锯齿

- 高质量、高性能抗锯齿
- 当前最高质量的抗锯齿效果
- 可同时使用抗锯齿和**HDR**
- 覆盖采样抗锯齿(**CSAA**)

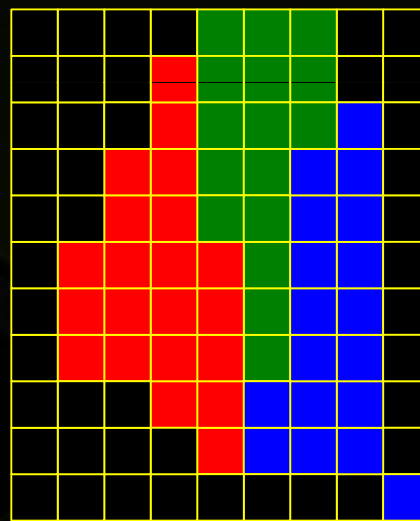


回顾：没有抗锯齿的渲染

- 一个像素只有一个采样
 - 覆盖率，色彩，深度和模板
- 精确的计算无法弥补采样不足



三角形



没有抗锯齿的结果

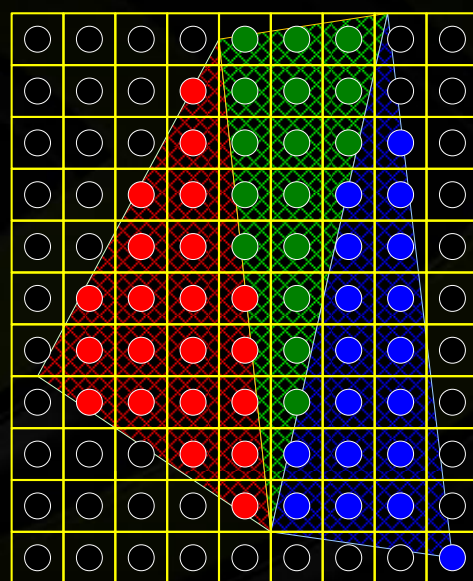


抗锯齿的要点

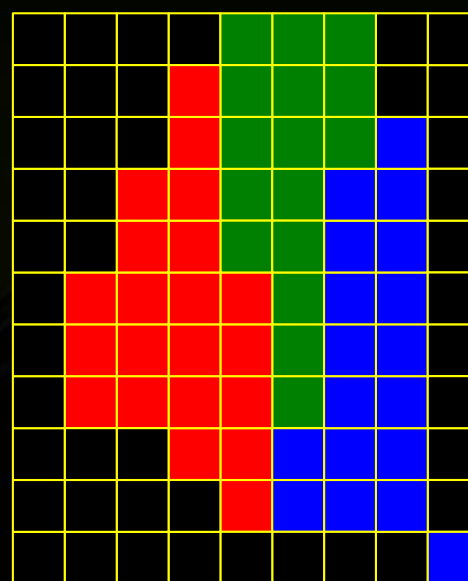
- 图像中的每个像素都代表了一块有限大小的区域，而不是一个无限小的采样点
- 理想的抗锯齿应当是一个作用于像素区域的平滑卷积
 - 卷积对于一般的场景来说太昂贵
- 所以我们用大量采样来代替
- 采样越多，效果越好

抗锯齿

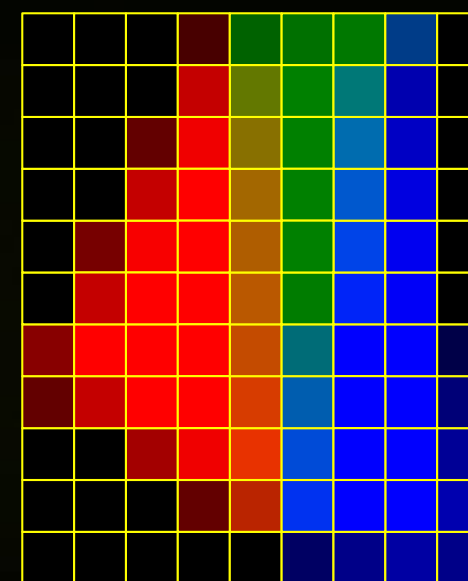
- 抗锯齿应处理所有与某个像素相交的几何形体，并正确地计算各形体对这个像素产生的影响



三角形



没有抗锯齿的结果



有抗锯齿的结果



过采样(Supersampling)

- 对每一个像素中的每一个子像素(采样)存储一个色彩值和深度值。每画一个三角形，为每一个像素中的每一个子像素单独计算一个色彩值和深度值
- 优点
 - 通过蛮力达到稳固的图像质量
- 缺点
 - 低效：对每个子像素都要进行**pixel shader**的处理以及贴图操作。多数情况下，临近子像素的色彩接近，产生极大的浪费

多重采样 (SGI, 1993)

- 对每一个像素中的每一个子像素(采样)存储一个色彩值和深度值。但是，每个像素中的所有子像素只采用同一色彩值。
- 优点
 - 每画一个三角形，对每个像素只计算一个色彩值
 - 深度和模板值精确到每个子像素；可正确处理相交和悬面
- 弱点
 - 占用N倍于1x的内存
 - 超过8x以上的多重采样过于昂贵



CSAA的目标

- 多重采样的质量按1 → 2 → 4倍的顺序提升
- 高于每像素4个子像素后，内存占用量的增长远大于图像质量的增长
- 打开HDR后，情况更为严重
 - 每个子像素要占用64bit和128bit！
- 对于绝大多数物体边缘的像素来说，用两个色彩值来表示已经足够了
 - 更重要的是提供更多的覆盖信息



基于覆盖率采样的抗锯齿

- 每个像素取**16**个子像素，分别计算是否被覆盖，将布尔值存储下来
- 压缩冗余的色彩、深度和模板信息，使得内存使用量和带宽降低到**4倍或8倍**多重采样的水平
- 用**4倍**多重抗锯齿的开销实现**16倍**抗锯齿的质量

CSAA质量级别

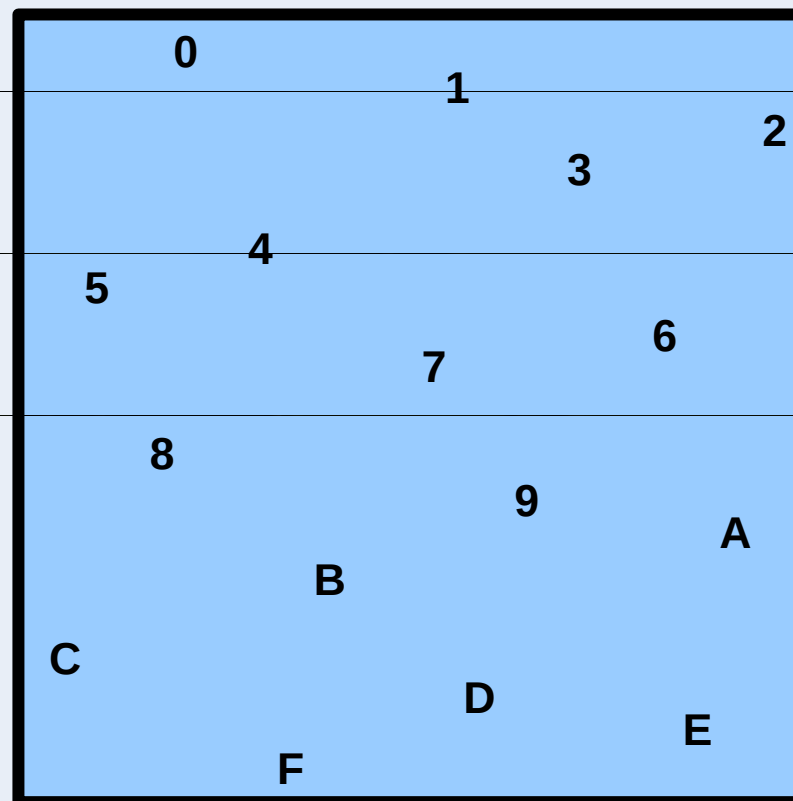


质量级别	4x	8x	8xQ	16x	16xQ
贴图/Shader使用倍数	1	1	1	1	1
存储色彩/深度采样数量	4	4	8	4	8
覆盖率采样数	4	8	8	16	16

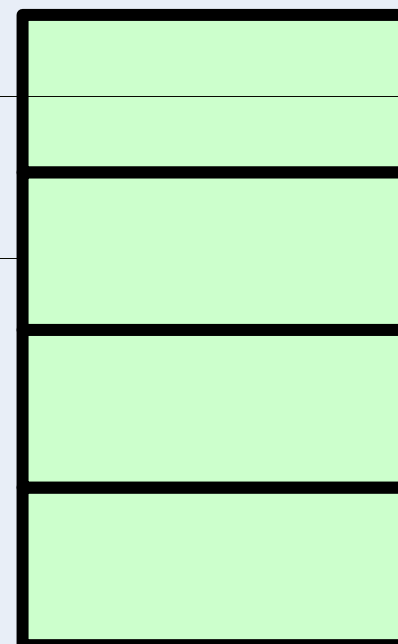
CSAA 16x 内存存储方式



采样分布



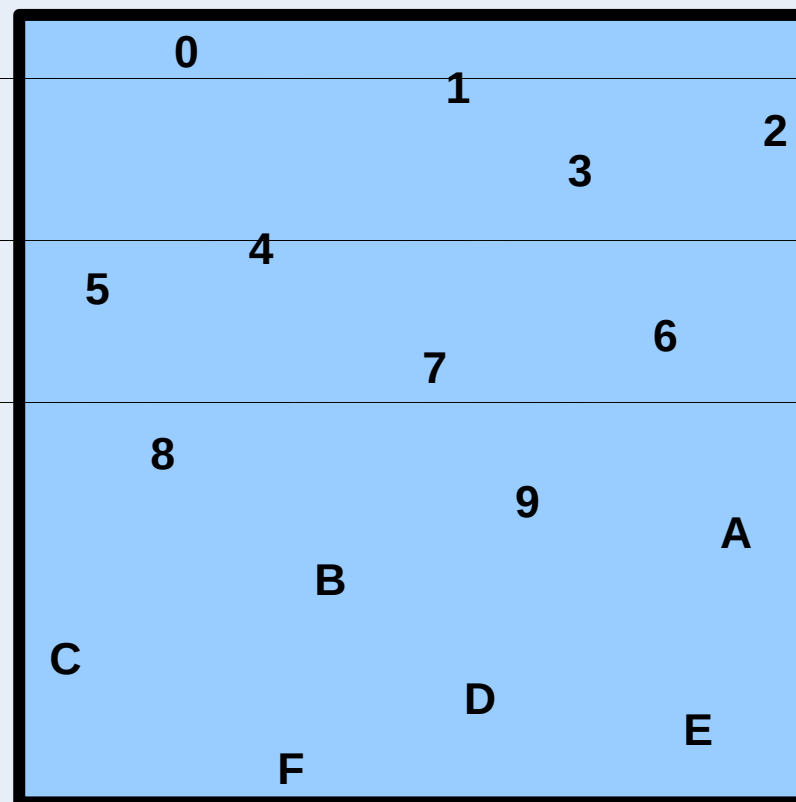
色彩存储



CSAA 16xQ 内存存储方式



采样分布



色彩存储





CSAA优点

- 低开销的覆盖率采样
- 可以兼容现有的各种渲染技术
 - **HDR**, 模板缓存算法
- 有效利用**shader**和贴图硬件
- 覆盖率数据使用布尔值表示, 而不是标量
 - 不会有渗透的副作用
- 在深度/模板高对比的情况下, 返回相应的多重采样(**4x或8x**)的质量
 - 三角形相交(**Inter-penetrating triangles**)
 - 模板阴影(**Stencil shadow volumes**)

抗锯齿模式比较



抗锯齿模式:	过采样			多重采样			覆盖采样		
质量等级:	1x	4x	16x	1x	4x	16x	1x	4x	16x
贴图/Shader使用 倍数	1	4	16	1	1	1	1	1	1
存储色彩/深度采样 数量	1	4	16	1	4	16	1	4	4
覆盖率采样数量	1	4	16	1	4	16	1	4	16

多重采样减少了shader和贴图
开销

覆盖采样减少了内存使用量和带宽需求量

《半条命2》



NVIDIA 4x



NVIDIA 16xQ

FarCry



NVIDIA 16xQ

Far Cry - Crytek



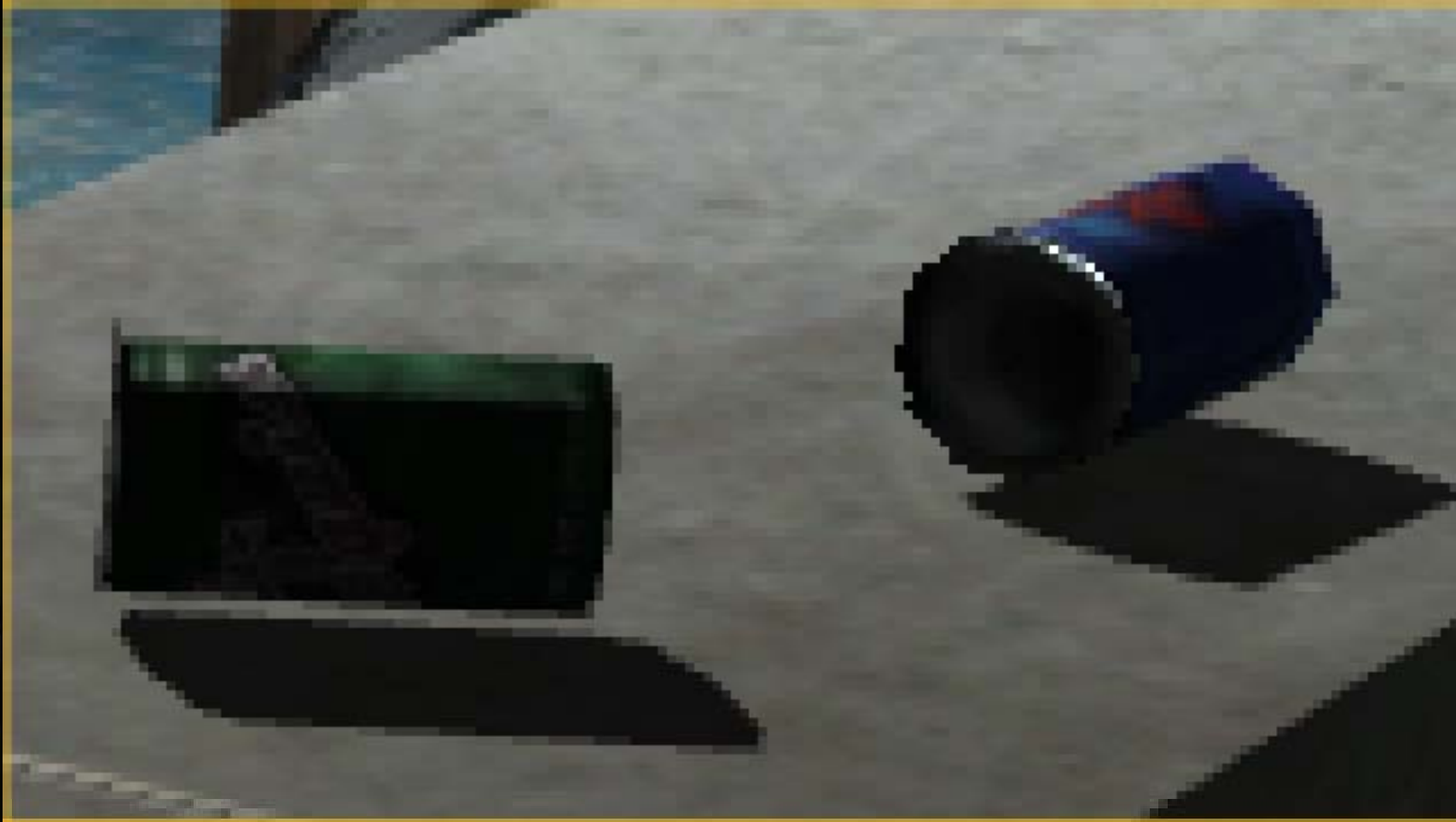
阴影边缘抗锯齿

- 模板阴影(**Stencil shadow volume**)算法假定每个子像素都有一个独立的模板值
 - **16x**: 每像素4个模板值
 - **16xQ**: 每像素8个模板值
- 阴影边缘的质量取决于子像素的数量

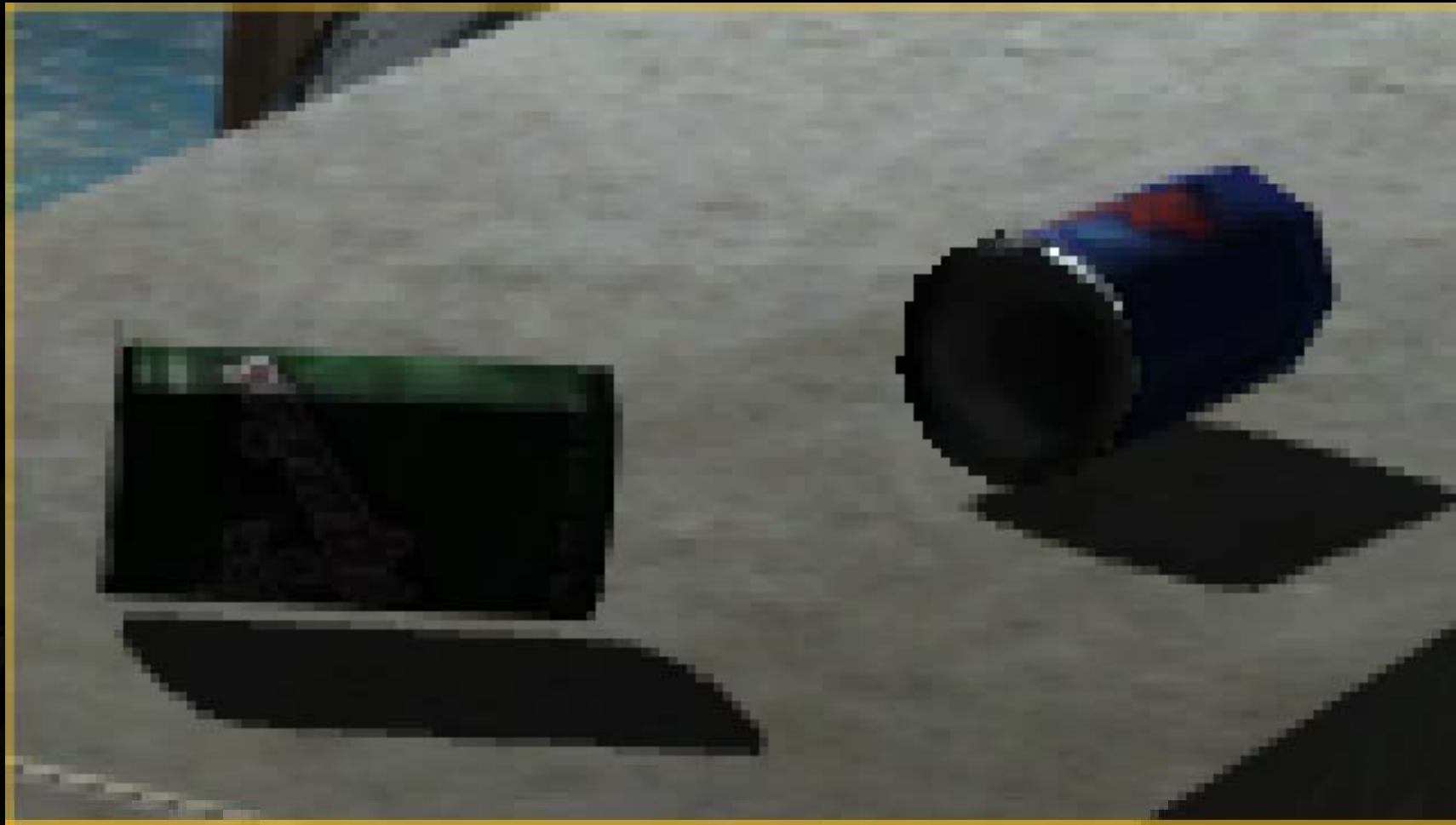
FEAR中的阴影边缘



FEAR 4x



FEAR 16x



更高质量的物体边缘

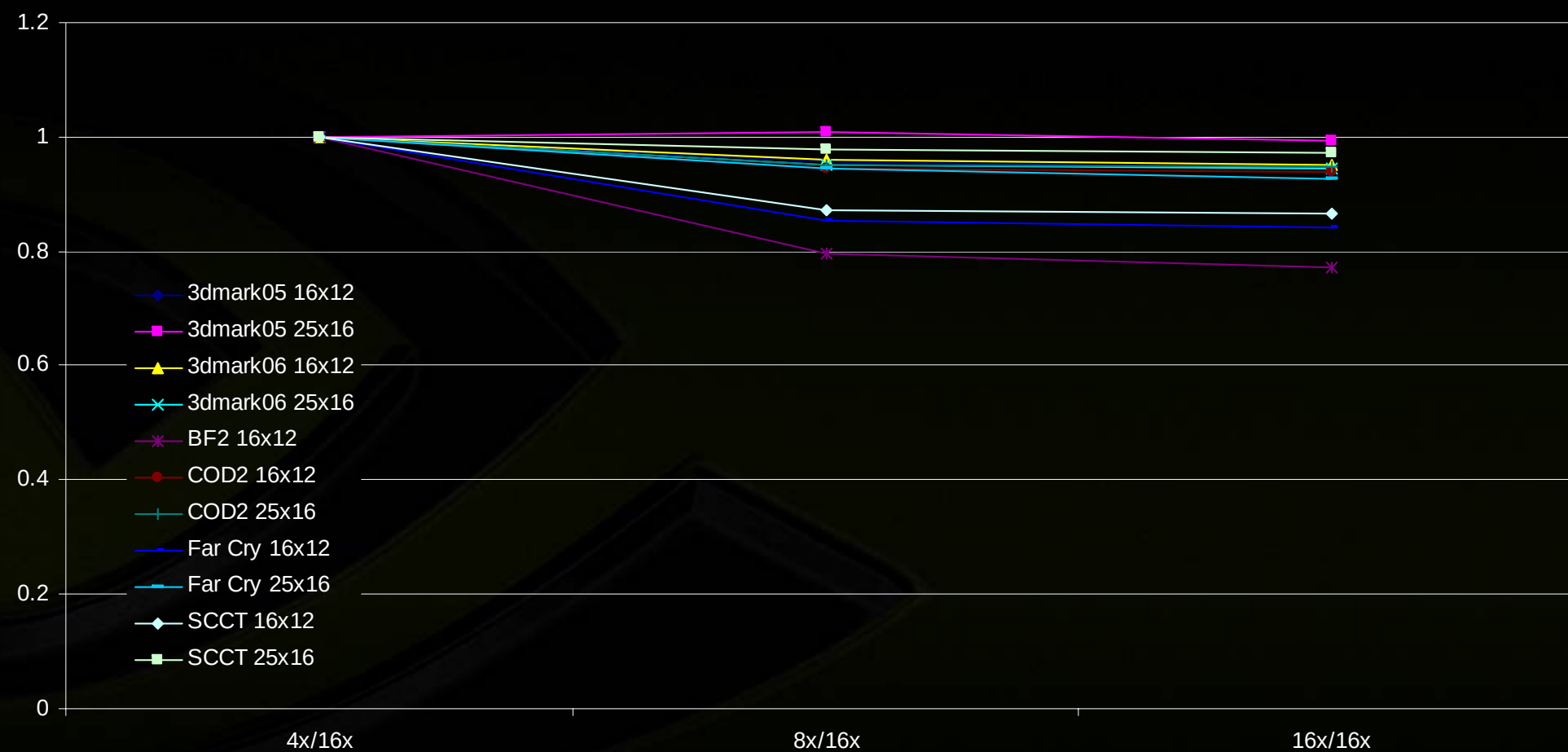
FEAR - Monolith



CSAA优点总结

- 只需对场景进行一次遍历
- 紧凑、固定的显存使用量
- 可处理物体相交的情况
- **Pixel shaders**只在需要色彩细节的地方才会被调用
- **16倍不同位置的覆盖率数据**
- **16倍的质量 ~ 4倍的开销**

抗锯齿性能





NVIDIA CUDA



应用程序的复杂度不断增长



指令中心型

- 不同的任务
- 非相干数据
- 顺序处理



CPU的更新



指令中心型
*CPU*的更新

多核
多线程
多任务

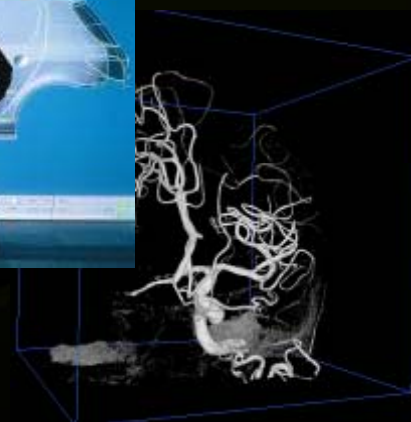


数据的复杂度不断增长

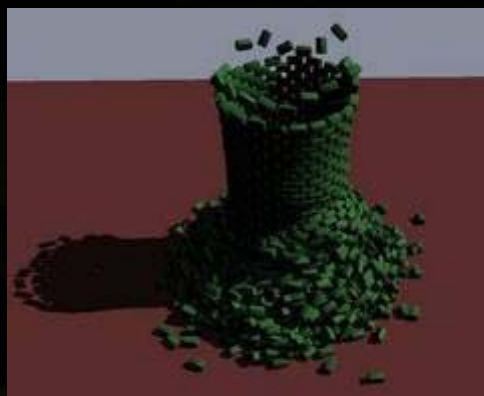


数据中心型

- 高重复性
- 高度规则数据
- 固有的并行性



GPU的更新



数据中心型
*GPU*的更新

通用GPU计算

浮点运算

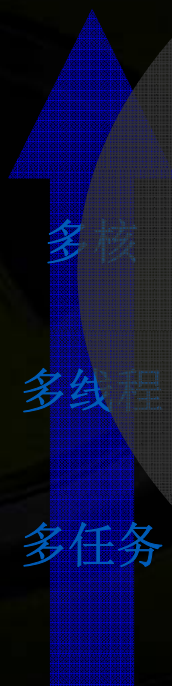
可编程管线

并行管线

NVIDIA CUDA

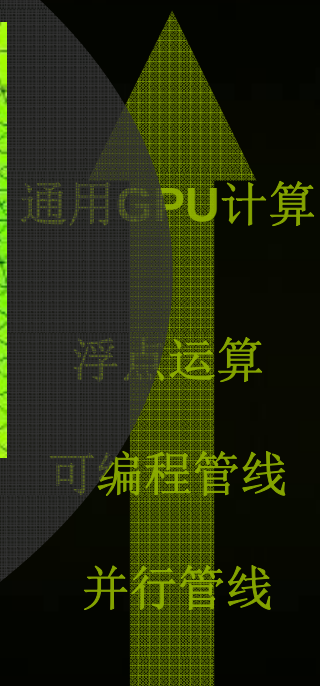
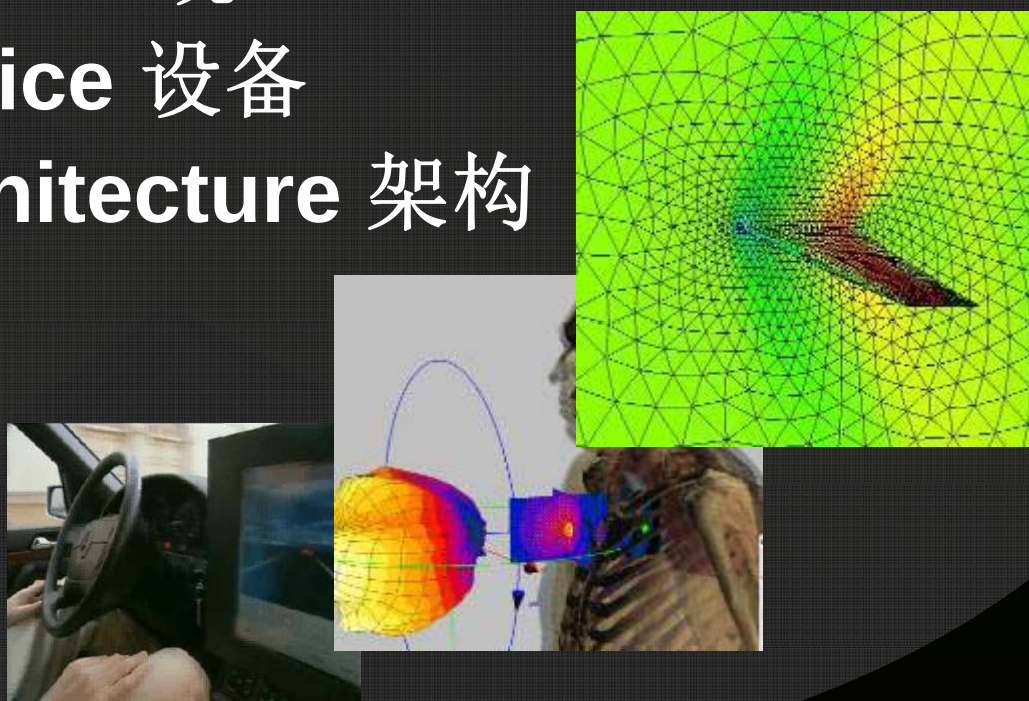


指令中心型
CPU 的更新



Compute 计算
Unified 统一
Device 设备
Architecture 架构

数据中心型
GPU 的更新





CUDA线程计算

- 革命性的硬件与软件架构
 - 专门为大规模计算而进行的设计
 - 全套开发方案
- 针对**GPU**的**C**语言编译器
 - 内置多线程编程模型，为并行处理而设计
 - 为提高**GPU**性能而优化
- 带来了大量新型应用
 - 独特的用于解决大规模运算问题的架构
 - 应用广泛，架构稳固

应用举例 – 物理计算

目标：计算每个运动盒子的位置

新位置 = 旧位置 + 速度 * 时间

$$\mathbf{P}_n' = \mathbf{P} + \mathbf{V} * t$$

对每个盒子...

$$\mathbf{P}_1 = \mathbf{P}_{1old} + \mathbf{V}_1 * t$$

$$\mathbf{P}_2 = \mathbf{P}_{2old} + \mathbf{V}_2 * t$$

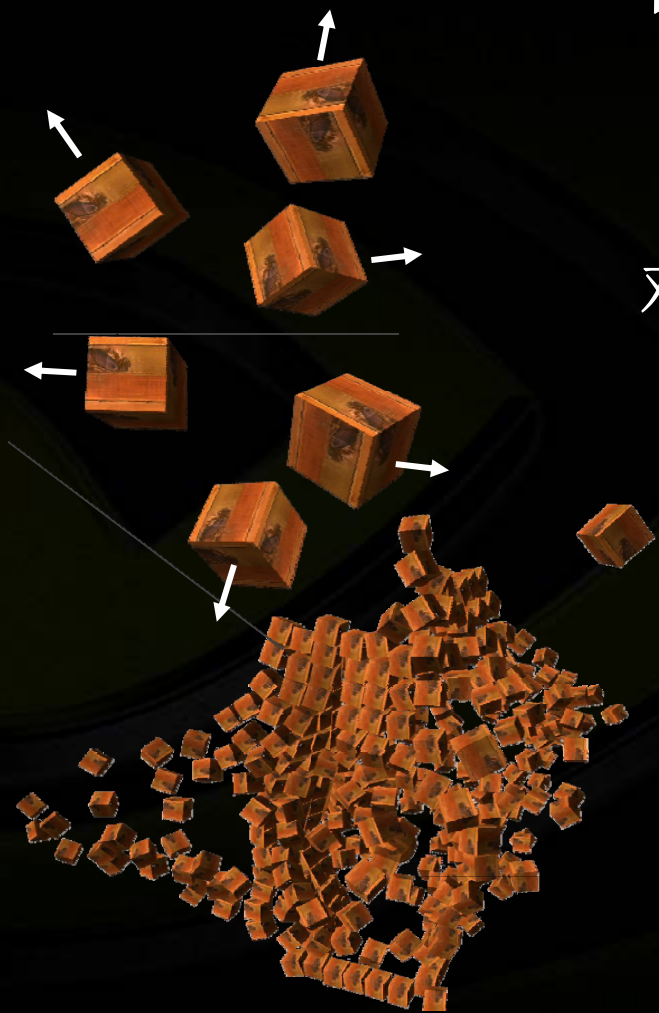
$$\mathbf{P}_3 = \mathbf{P}_{3old} + \mathbf{V}_3 * t$$

$$\mathbf{P}_4 = \mathbf{P}_{4old} + \mathbf{V}_4 * t$$

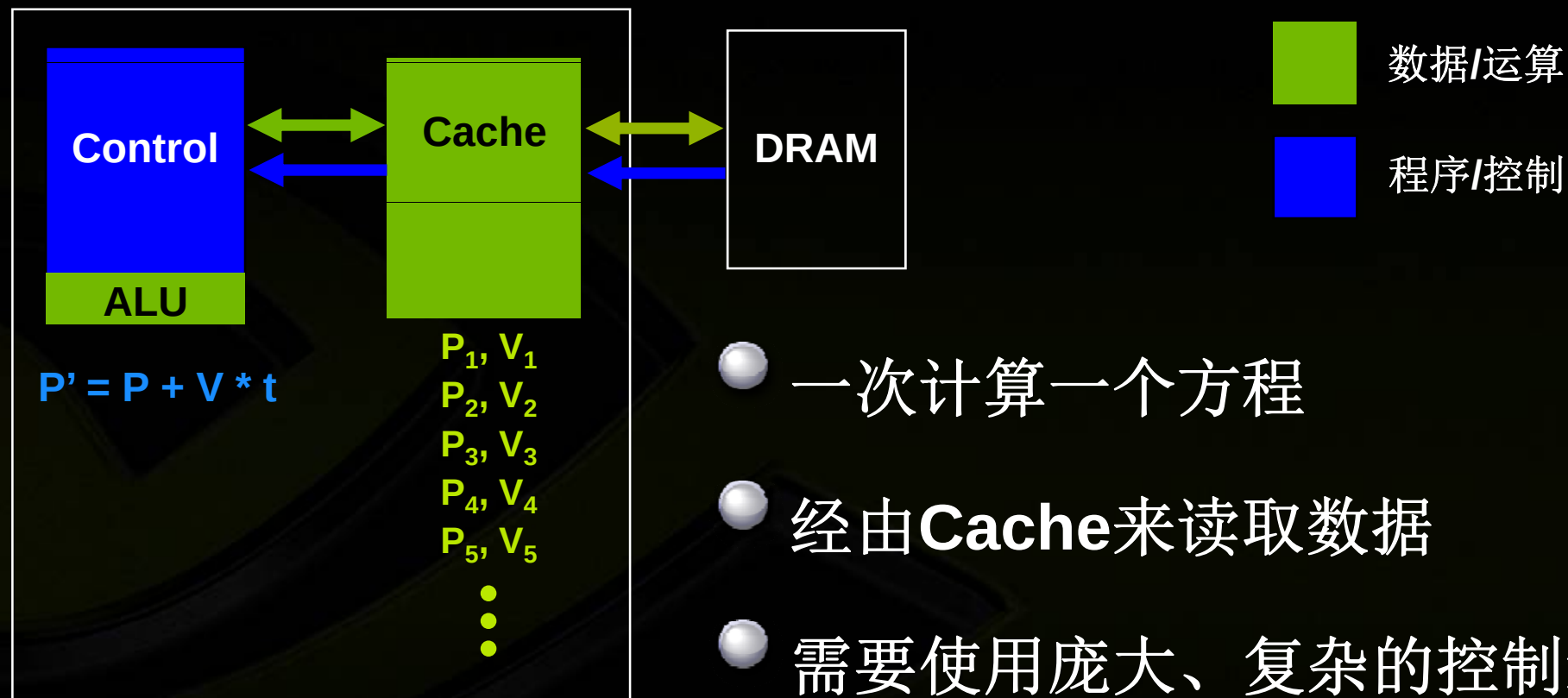
$$\mathbf{P}_5 = \mathbf{P}_{5old} + \mathbf{V}_5 * t$$

$$\mathbf{P}_6 = \mathbf{P}_{6old} + \mathbf{V}_6 * t$$

⋮

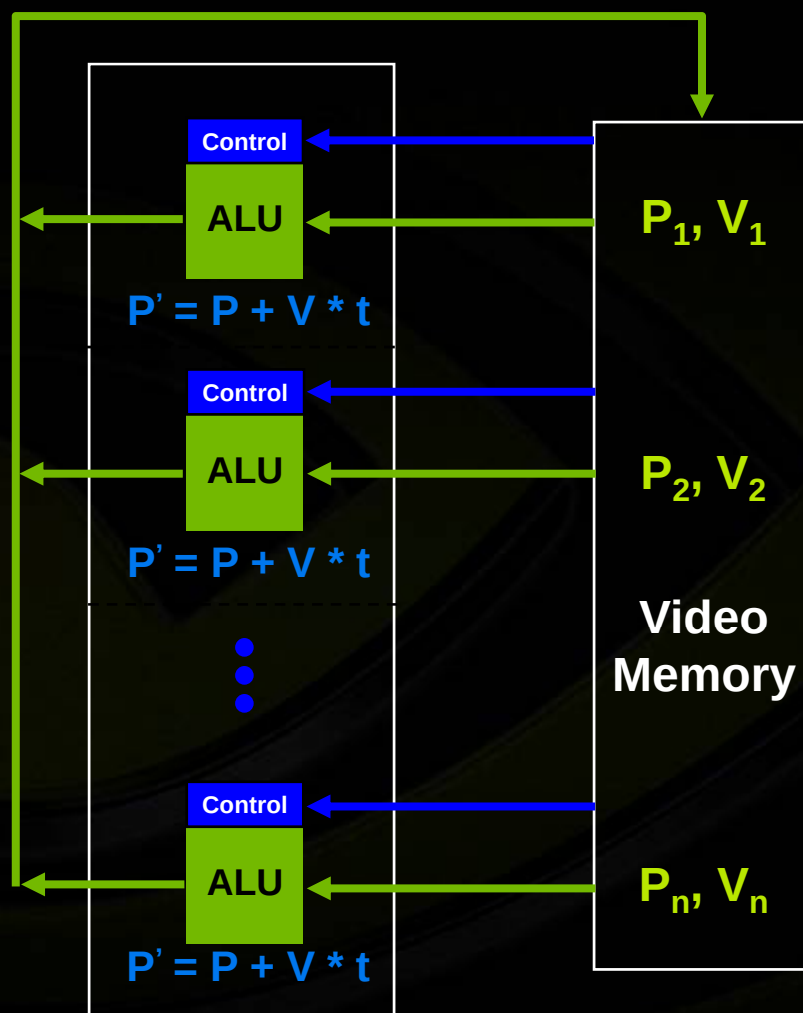


CPU



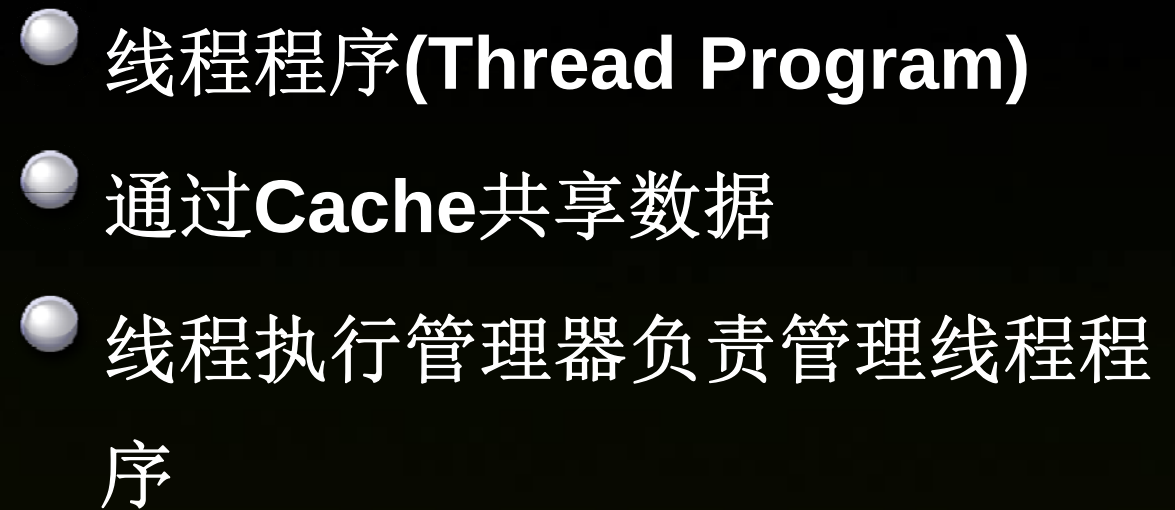
- 一次计算一个方程
- 经由Cache来读取数据
- 需要使用庞大、复杂的控制逻辑来保证CPU的利用率

GPGPU



- 多shader并行求解方程组
- 数据通过显示内存共享
- 流式填充多个shader

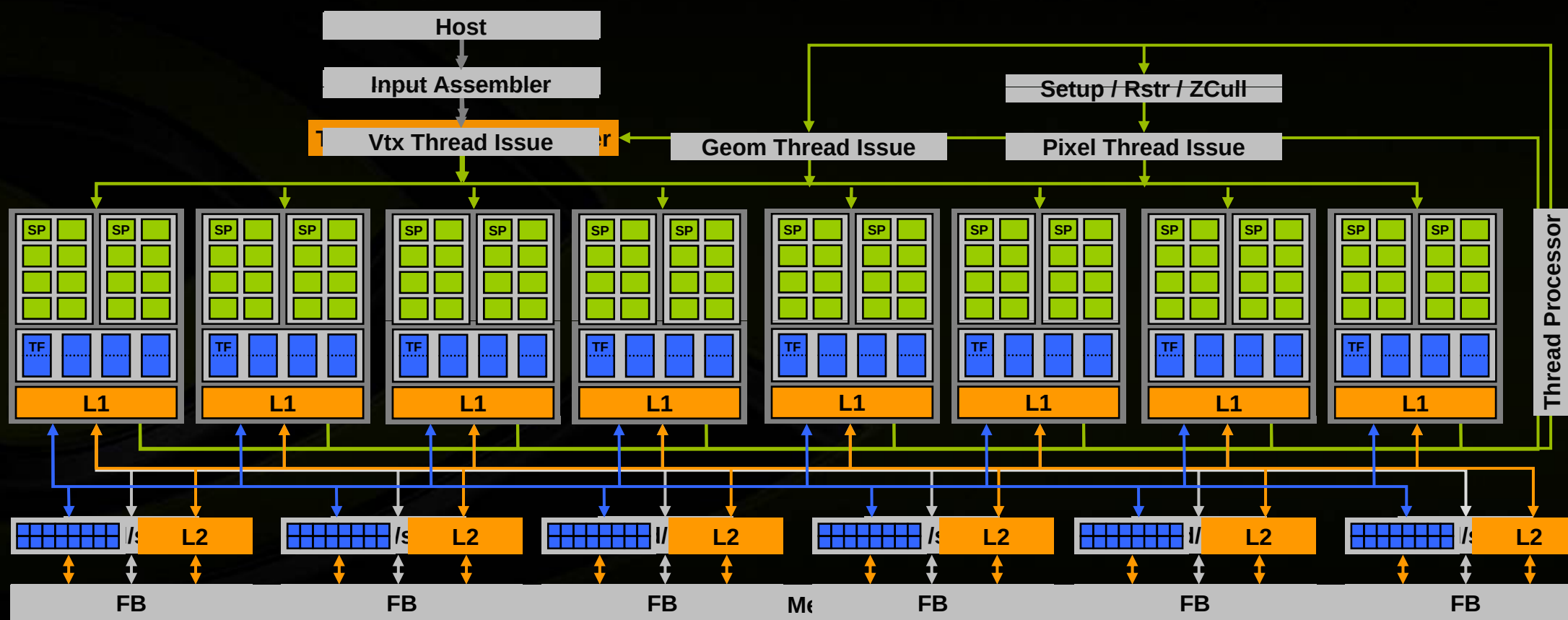




A diagram consisting of two colored squares. The top square is light blue and contains the text "数据/运算" (Data/Operations). The bottom square is light red and contains the text "程序/控制" (Program/Control). The two squares are stacked vertically, illustrating the relationship between data/operations and program/control.

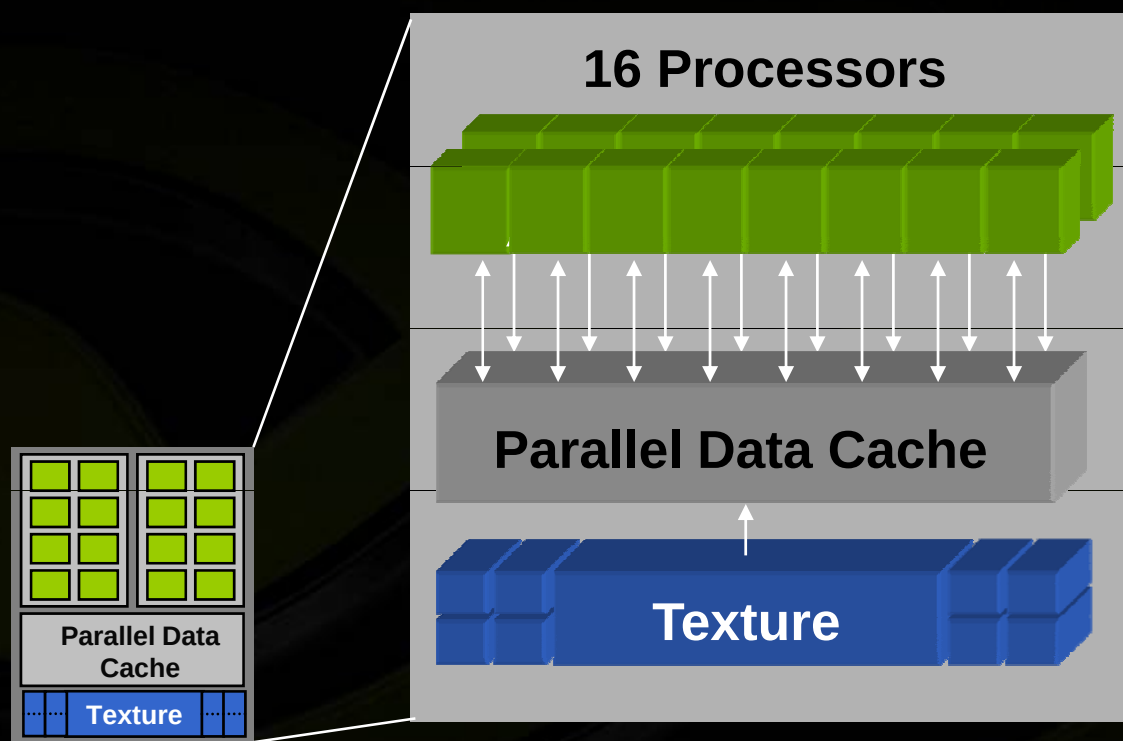
G80线程计算管线

- 处理的未来在计算管线管理
- 成为计算资源设置特殊操作模式(special mode)进行架构





线程计算的处理单元(Processors)



- 128个1.35 GHz处理单元
- 每16个处理单元拥有一套并行数据缓存(Parallel Data Cache)
- 并行处理
- 贴图访问
- IEEE 754浮点数支持
- 使用C语言编程

CUDA SDK



标准库:
FFT, BLAS,...

集成CPU端和
GPU端的C语言源程序

NVIDIA C语言编译器

NVIDIA计算用汇编语言

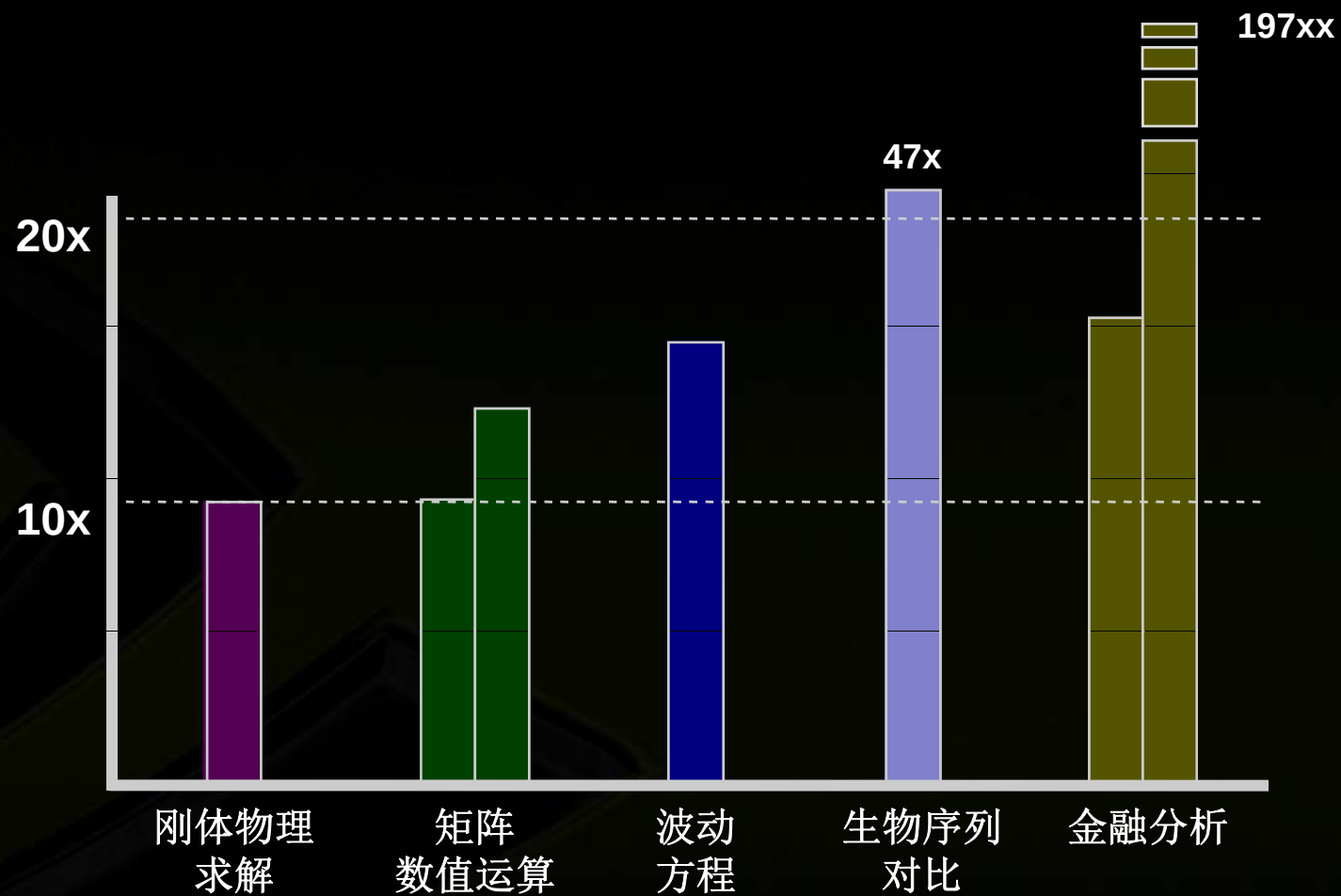
CPU端代码

CUDA运行时库/驱动程序

性能



CUDA
优势



GeForce 8800 vs. 2.66 GHz Dual Core Conroe

CUDA的线程计算

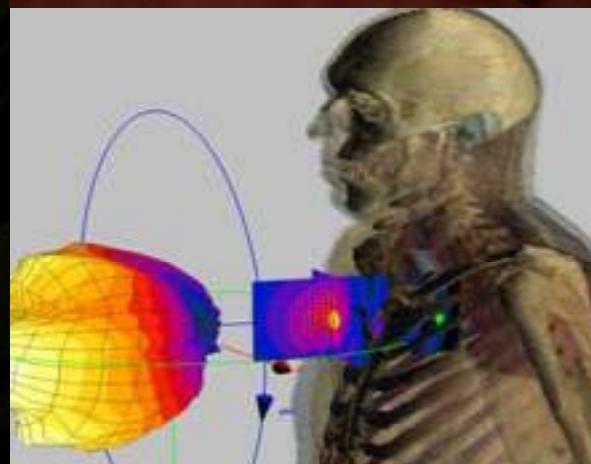
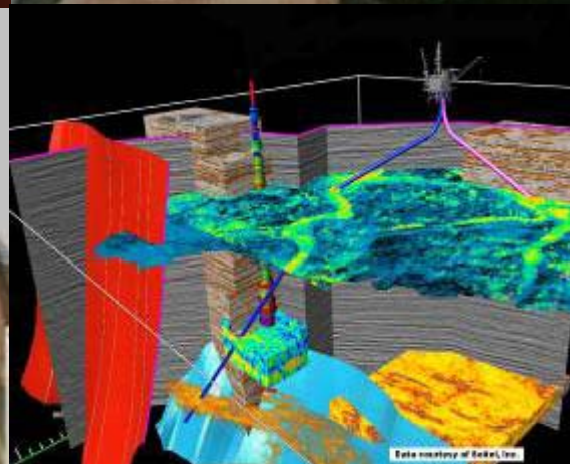


消费领域



嵌入系统

商业领域



高性能计算

线程计算的革命



- GPU上的线程计算
- 使用C言语编程
- 带来了一整套新型的应用

GeForce 8800GTX



	X1950xtx	7950GX2	8800GTX
Shader Model	SM3	SM3	SM4
Vertex Shader单元数量	8	16	128
贴图过滤速度	10.4 GBilerp	24 GBilerp	36.8 GBilerp
Pixel Shader单元数量	48	64	128
Shader数学运算量 (GMUL/s)	124.8	192	345.6
ROP处理速度	Up to 32ppc	Up to 64ppc	Up to 192ppc
显存位宽	256-bit	512-bit	384-bit
显存带宽	64 GBs/ec	76.8 GB/sec	86.4 GB/sec

GeForce 8800 GTX显卡



\$599 e-tail

内核频率	575MHz
流处理单元数目	128
Shader频率	1350MHz
内存频率	900MHz
内存	768MB GDDR3
输出	DL-DVI DL-DVI HDTV-out

GeForce 8800 GTS Graphics Board



\$449 e-tail

内核频率	500MHz
流处理单元数目	96
Shader频率	1200MHz
内存频率	800MHz
内存	640MB GDDR3
输出	DL-DVI DL-DVI HDTV-out

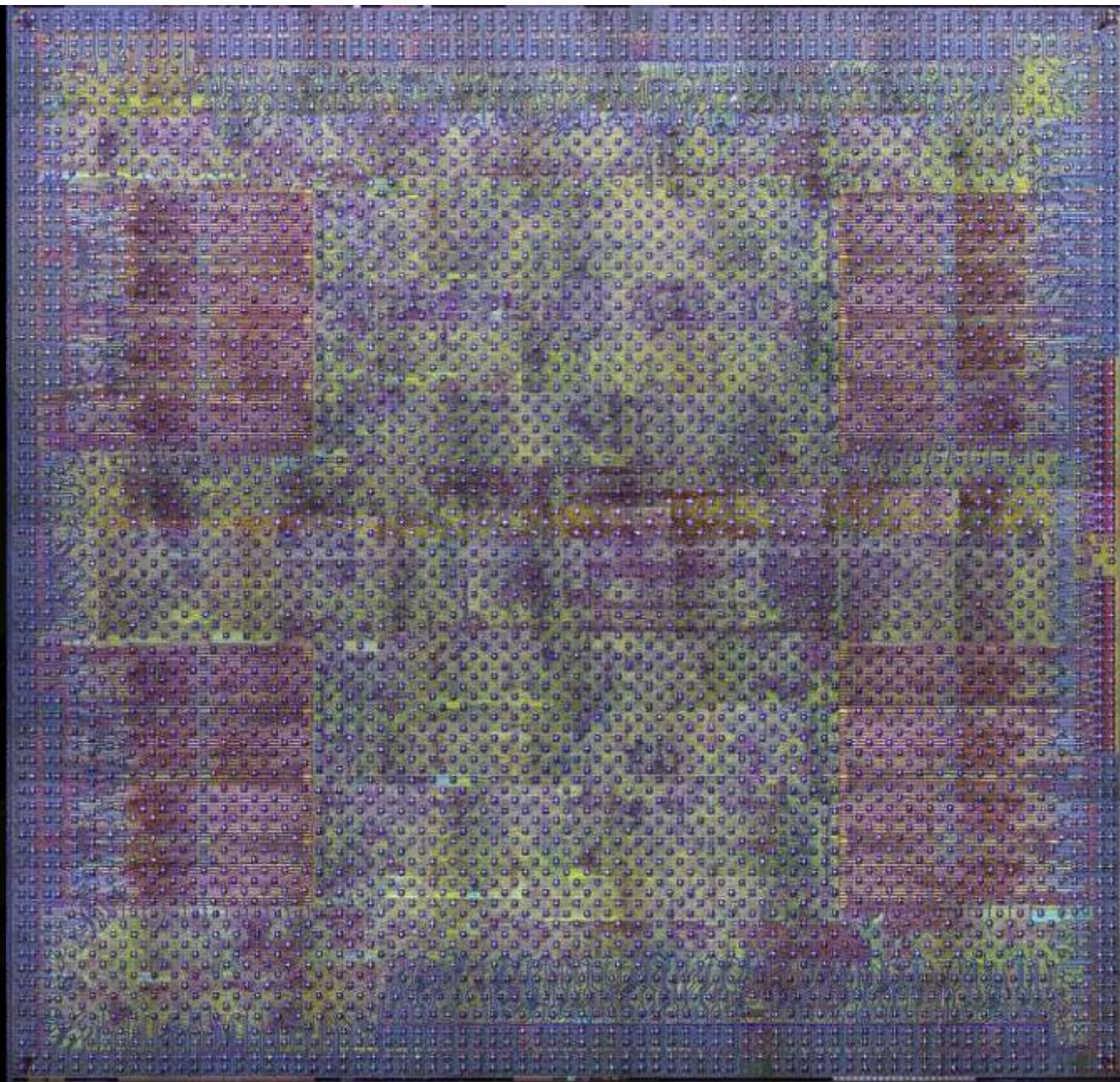


谢谢大家





6.81亿晶体管
看起来是什么样的？





- **Pure Video**
- **Display Processing**
- **Power Efficiency**



GeForce 8800 – Raises the Bar for High-Definition Post-Processing

- The processing power of the GeForce 8800 allows more sophisticated post-processing with HD content:
 - Adds VC-1 & H.264 HD Spatial-Temporal Deinterlacing
 - Adds VC-1 & H.264 HD Inverse Telecine
 - Adds HD Noise reduction
 - Adds HD Edge enhancement

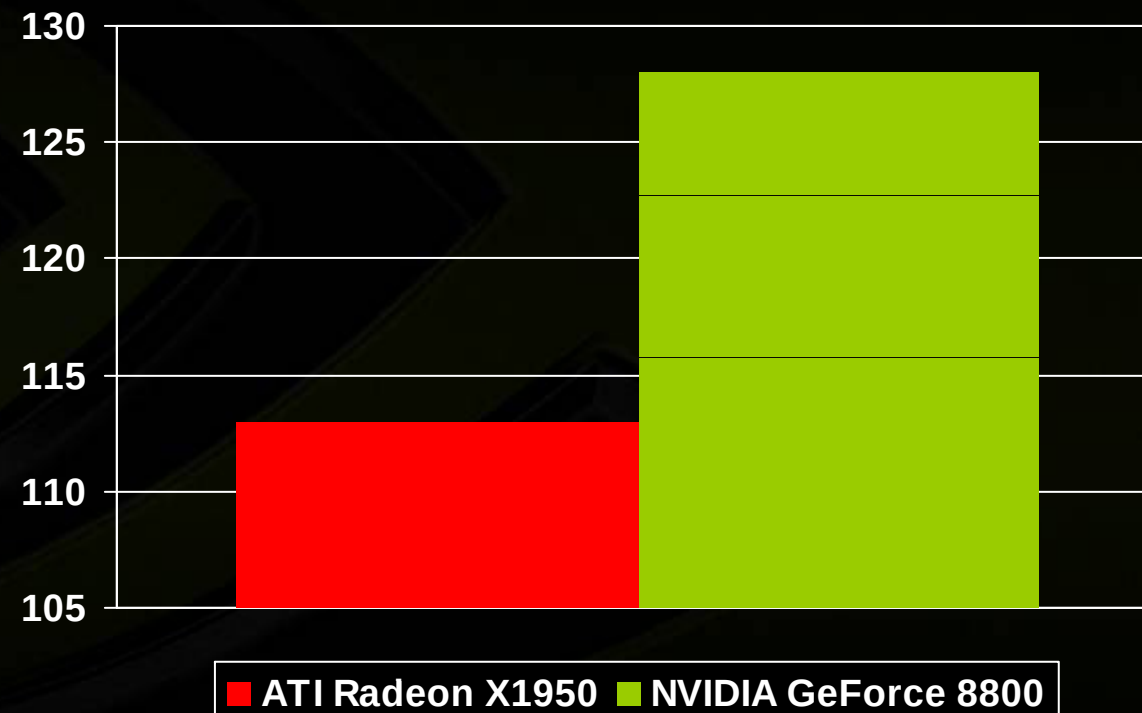


GeForce 8800 – Industry's Best Video Processor

(HQV Score = 128 Points)



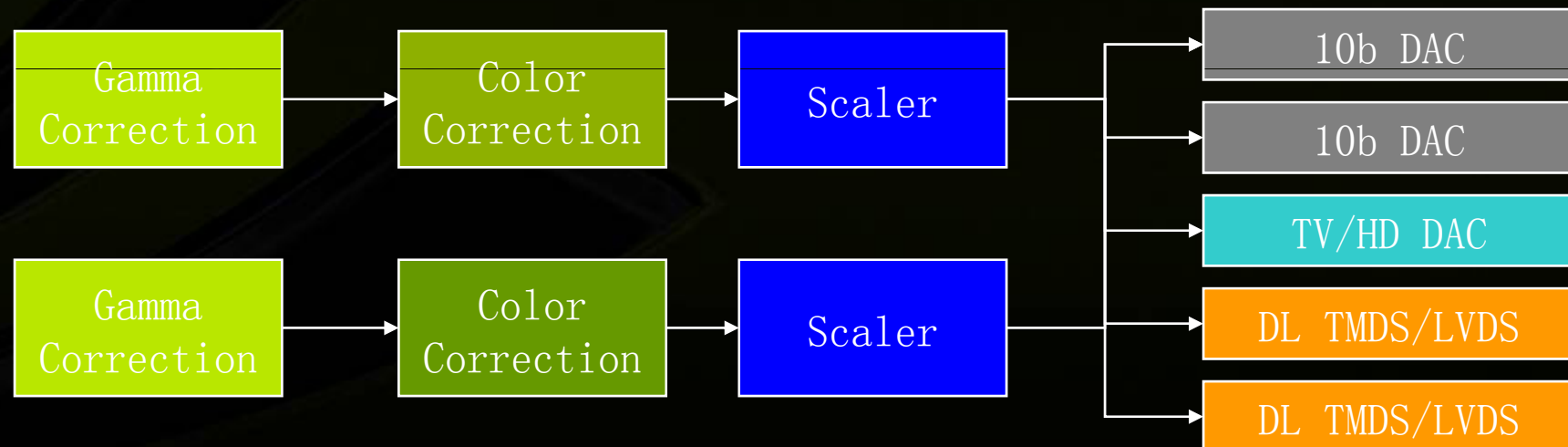
HQV Video Benchmark Results





Next Generation Display

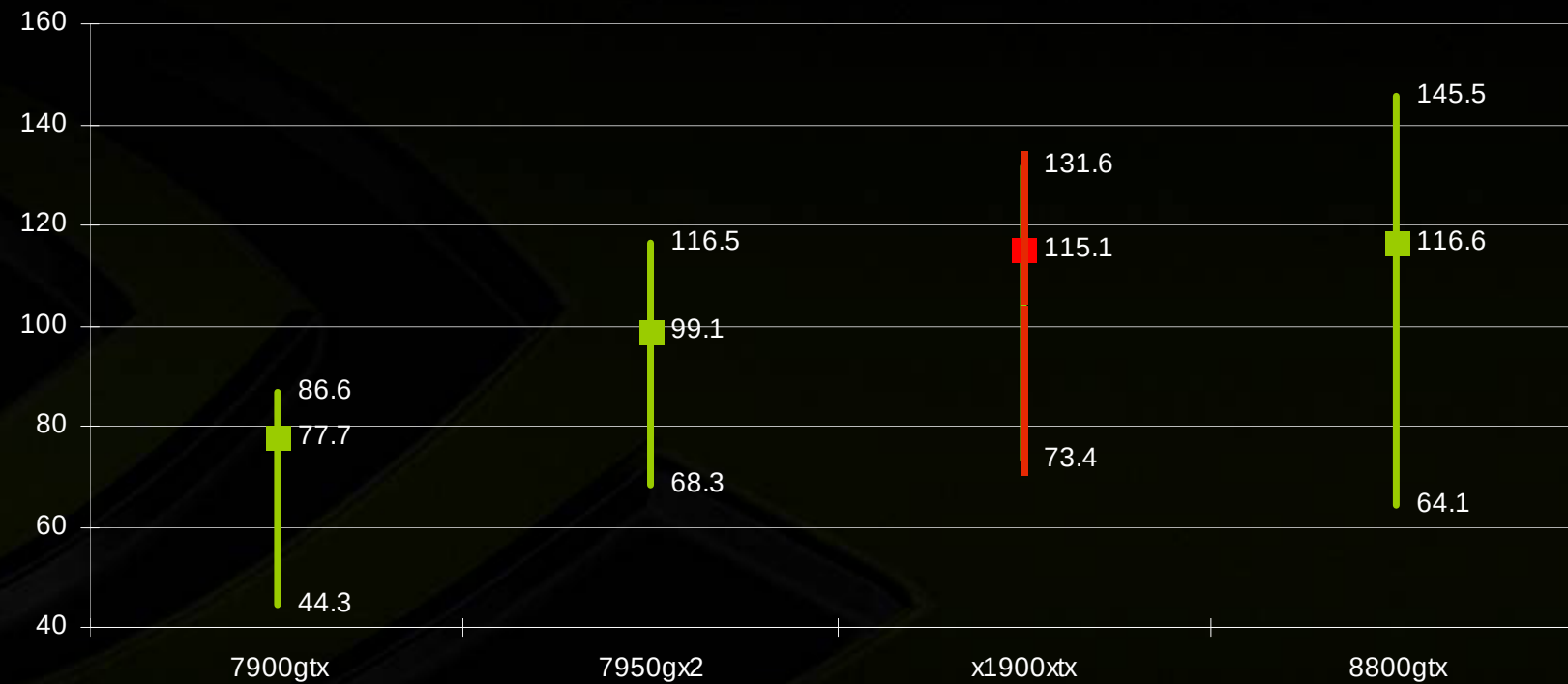
- New from ground up
- Modular architecture
- 10b Display processing pipeline



Architected for Power Efficiency

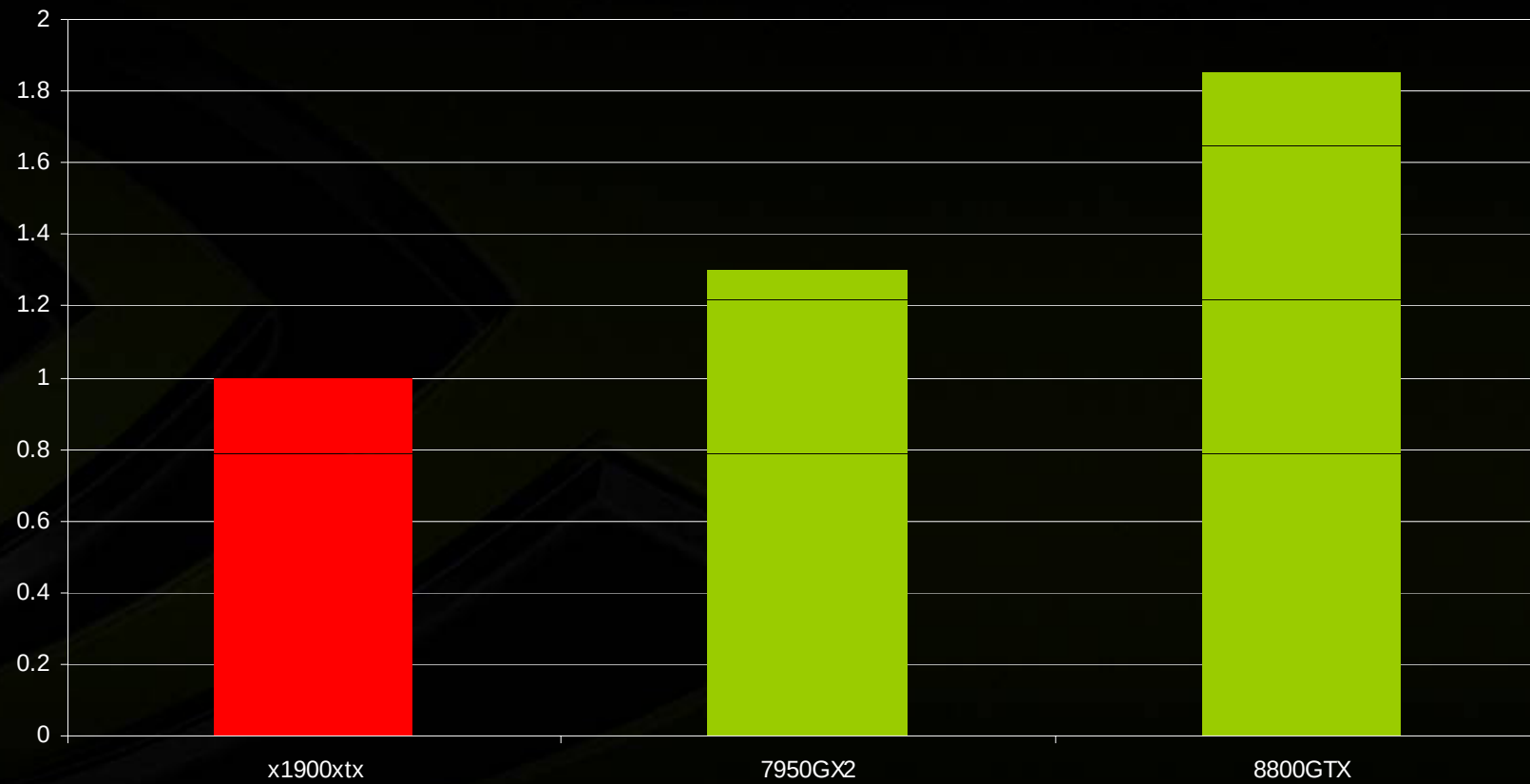


Power in Watts



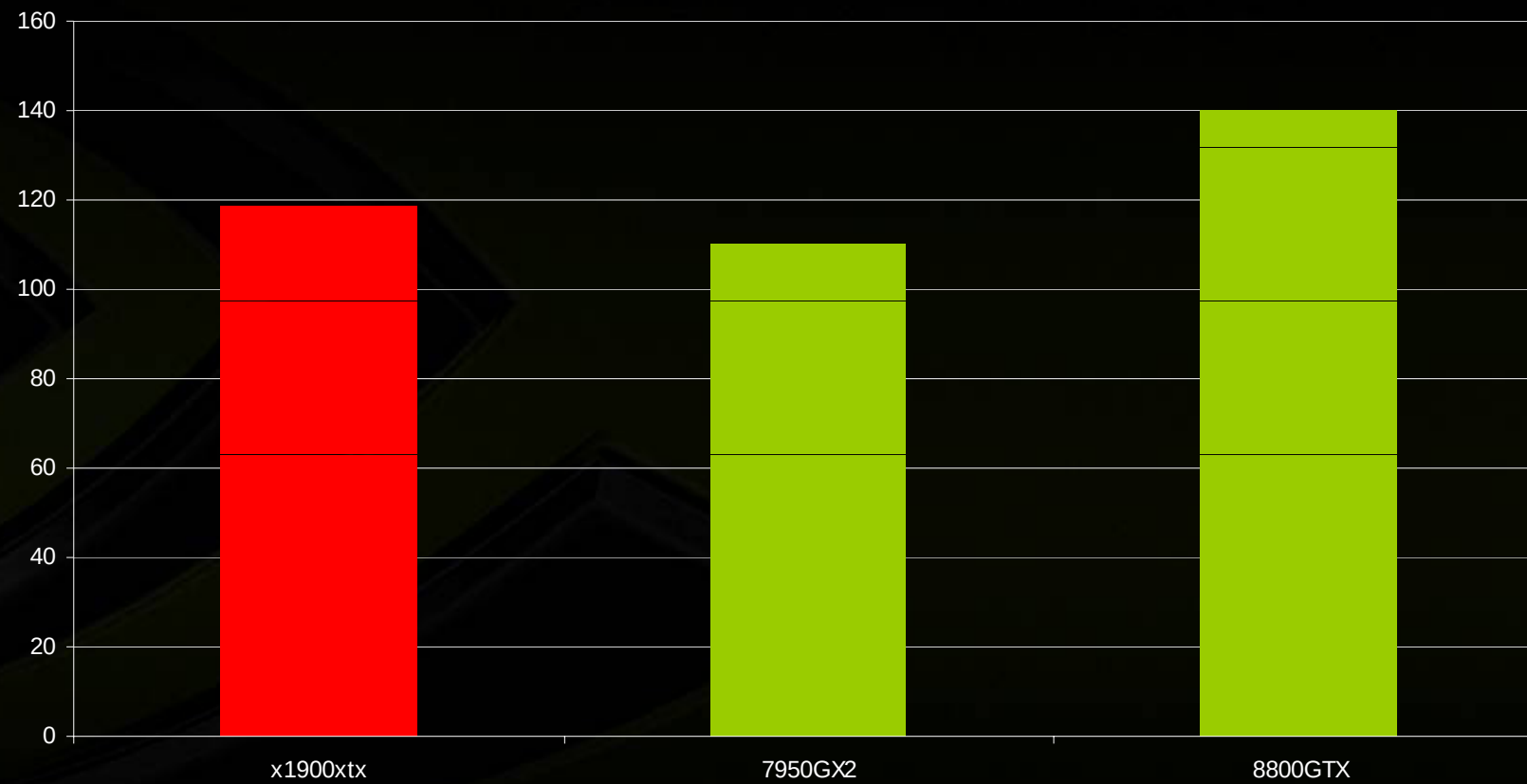
130 App/Mode Combinations
Room Temperature
Core 2 Duo Platform
1600x1200 & 2560x1600

Performance



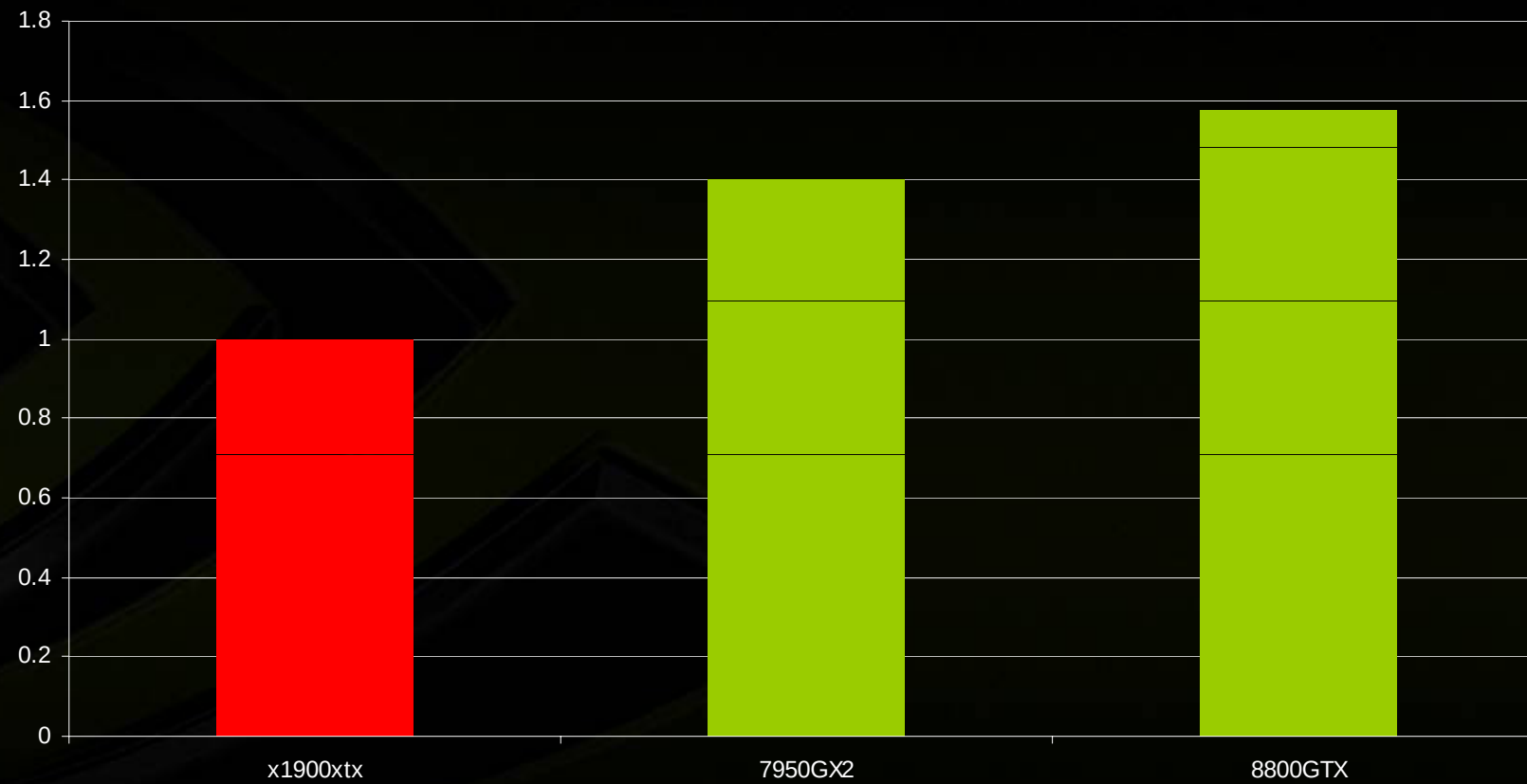
3DMark06
2500x1600 1x/8x
Core2 Duo

Power (Watts)



3DMark06
2500x1600 1x/8x
Core2 Duo

Performance per Watt



3DMark06
2500x1600 1x/8x
Core2 Duo