湖南大學

HUNAN UNIVERSITY

计算机设计

学 生 姓 名 李博文

学 生 学 号 201708010602

专 业 班 级 智 能 1702

指 导 老 师 吴 强

论 文 题 目 TeslaGPU架构分析

一 、 GPU架 构

1、GPU的起源

GPU缩写为Graphics Processing Unit的，一般称为视觉处理单元。GPU被广泛用于嵌入式系统、移动电话、个人电脑、工作站和电子游戏解决方案当中。现代的GPU对图像和图形处理是十分高效率的，这是因为GPU被设计为很高的并行架构这样使得比通用处理器CPU在大的数据块并行处理算法上更具有优势。

1985年 8月20日 ATi公司成立，同年10月ATi使用ASIC技术开发出了第一款图形芯片和图形卡，1992年 4月 ATi发布了 Mach32 图形卡集成了图形加速功能，1998年 4月 ATi被IDC评选为图形芯片工业的市场领导者，但那时候这种芯片还没有GPU的称号，很长的一段时间ATI都是把图形处理器称为VPU，直到AMD收购ATI之后其图形芯片才正式采用GPU的名字。

NVIDIA公司在1999年发布GeForce 256图形处理芯片时首先提出GPU的概念。从此NVIDIA显卡的芯片就用这个新名字GPU来称呼。GPU使显卡削减了对CPU的依赖，并执行部分原本CPU的工作，尤其是在3D图形处理时。GPU所采用的核心技术有钢体T&L、立方环境材质贴图与顶点混合、纹理压缩及凹凸映射贴图、双重纹理四像素256位渲染引擎等，而硬体T&L技术能够说是GPU的标志。

2、工作原理

GPU工作流程简介

GPU的图形（处理）流水线完成如下的工作:

① 顶点处理：这阶段GPU读取描述3D图形外观的顶点数据并根据顶点数据确定3D图形的形状及位置关系，建立起3D图形的骨架。在支持DX8和DX9规格的GPU中，这些工作由硬件实现的VertexShader（定点着色器）完成。

② 光栅化计算：显示器实际显示的图像是由像素组成的，我们需要将上面生成的图形上的点和线通过一定的算法转换到相应的像素点。把一个矢量图形转换为一系列像素点的过程就称为光栅化。例如，一条数学表示的斜线段，最终被转化成阶梯状的连续像素点。

③ 纹理帖图：顶点单元生成的多边形只构成了3D物体的轮廓，而纹理映射（texturemapping）工作完成对多变形表面的帖图，通俗的说，就是将多边形的表面贴上相应的图片，从而生成“真实”的图形。TMU（Texturemapping unit）即是用来完成此项工作。

④ 像素处理: 这阶段 (在对每个像素进行光栅化处理期间) GPU完成对像素的计算和处理, 从而确定每个像素的最终属性。在支持DX8和DX9规格的GPU中，这些工作由硬件实现的Pixel Shader（像素着色器）完成最终输出，由ROP（光栅化引擎）最终完成像素的输出，1帧渲染完毕后，被送到显存帧缓冲区。

⑤ 在GPU出现之前，CPU一直负责着计算机中主要的运算工作，包括多媒体的处理工作。CPU的架构是有利于X86指令集的串行架构，

CPU从设计思路上适合尽可能快的完成一个任务。

但是如此设计的CPU在多媒体处理中的缺陷也显而易见：多媒体计算通常要求较高的运算密度、多并发线程和频繁地存储器访问，而由于X86平台中CISC（Complex Instruction Set Computer）架构中暂存器数量有限，CPU并不适合处理这种类型的工作。

以Intel为代表的厂商曾经做过许多改进的尝试，从1999年开始为X86平台连续推出了多媒体扩展指令集SSE（Streaming SIMD Extensions）的一代到四代版本，但由于多媒体计算对于浮点运算和并行计算效率的高要求，CPU从硬件本身上就难以满足其巨大的处理需求，仅仅在软件层面的改并不能起到根本效果。

对于GPU来说，它的任务是在屏幕上合成显示数百万个像素的图像，也就是同时拥有几百万个任务需要并行处理，因此GPU被设计成可并行处理很多任务，而不是像CPU那样完成单任务。
因此CPU和GPU架构差异很大，CPU功能模块很多，能适应复杂运算环境；GPU构成则相对简单，目前流处理器和显存控制器占据了绝大部分晶体管。

CPU中大部分晶体管主要用于构建控制电路（比如分支预测等）和Cache，只有少部分的晶体管来完成实际的运算工作。而GPU的控制相对简单，且对Cache的需求小，所以大部分晶体管可以组成各类专用电路、多条流水线，使得GPU的计算速度有了突破性的飞跃，拥有了更强大的处理浮点运算的能力。

图2-1 CPU和GPU架构

从硬件设计上来讲，CPU 由专为顺序串行处理而优化的几个核心组成。另一方面，GPU则由数以千计的更小、更高效的核心组成，这些核心专为同时处理多任务而设计。

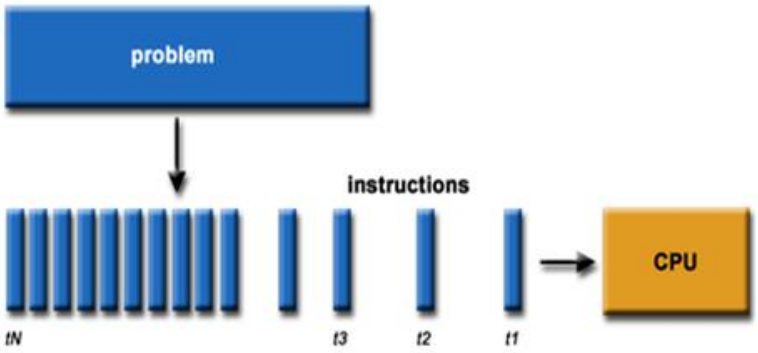


图2-2 串行运算示意图

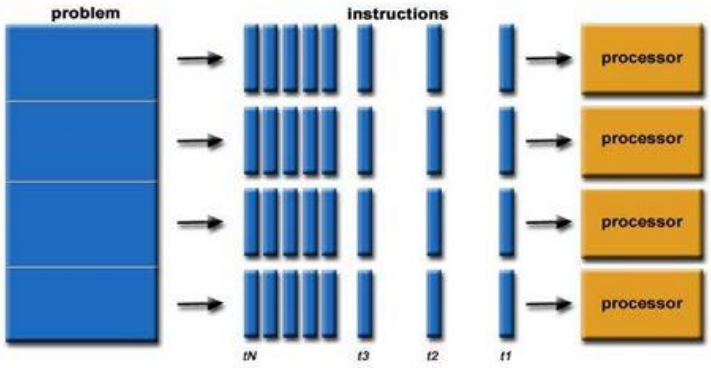


图2-3 并行运算示意图

通过上图我们可以较为容易地理解串行运算和并行运算之间的区别。传统的串行编写软件具备以下几个特点：要运行在一个单一的具有单一中央处理器（CPU）的计算机上；一个问题分解成一系列离散的指令；指令必须一个接着一个执行；只有一条指令可以在任何时刻执行。

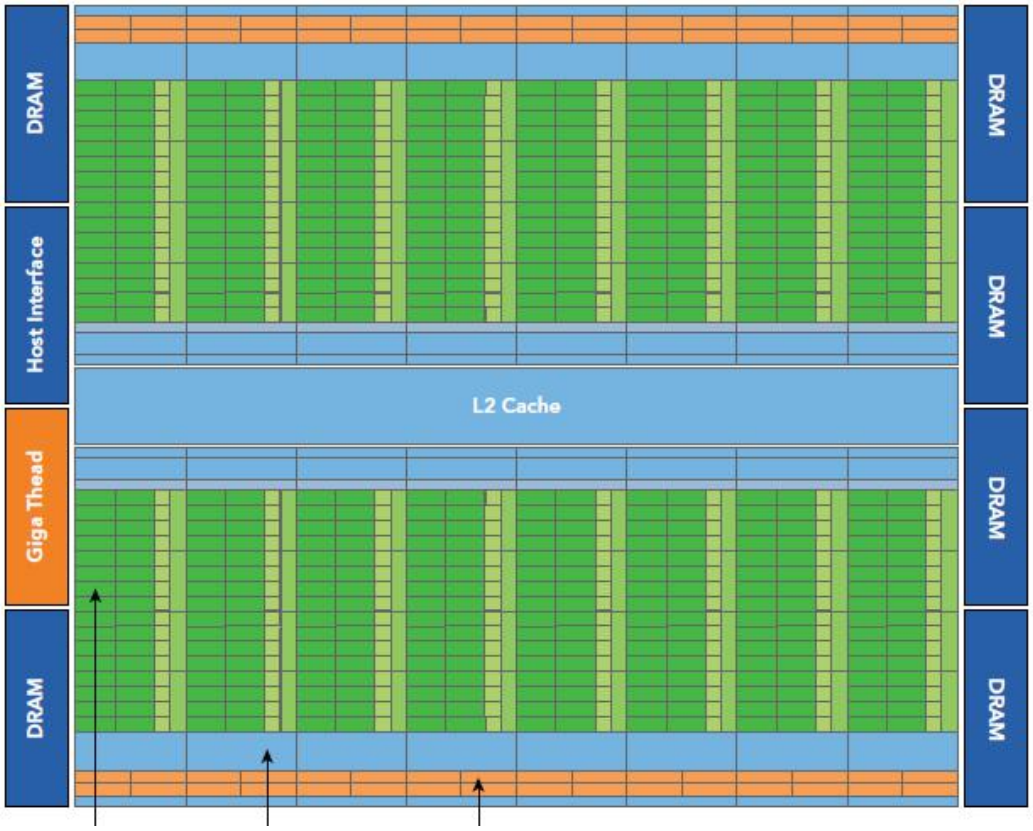
而并行计算则改进了很多重要细节：要使用多个处理器运行；一个问题可以分解成可同时解决的离散指令；每个部分进一步细分为一系列指示；每个部分的问题可以同时在不同处理器上执行。提高了算法的处理速度。

二、Fermi架构

Fermi是第一个完整的GPU计算架构。

- 512个accelerator cores即所谓CUDA cores（包含ALU和FPU）
- 16个SM，每个SM包含32个CUDA core
- 六个384位 GDDR5 DRAM，支持6GB global on-board memory
- GigaThread engine（图左侧）将thread blocks分配给SM调度
- 768KB L2 cache
- 每个SM有16个load/store单元，允许每个clock cycle为16个thread（即所谓half-warp，不过现在不提这个东西了）计算源地址和目的地址
- Special function units（SFU）用来执行sin cosine 等

- 每个SM两个warp scheduler两个instruction dispatch unit，当一个block被分配到一个SM中后，所有该block中的thread会被分到不同的warp中。
- Fermi（compute capability 2.x）每个SM同时可处理48个warp共计1536个thread。



每个SM由一下几部分组成：

- 执行单元（CUDA cores）
- 调度分配warp的单元
- shared memory，register file，L1 cache

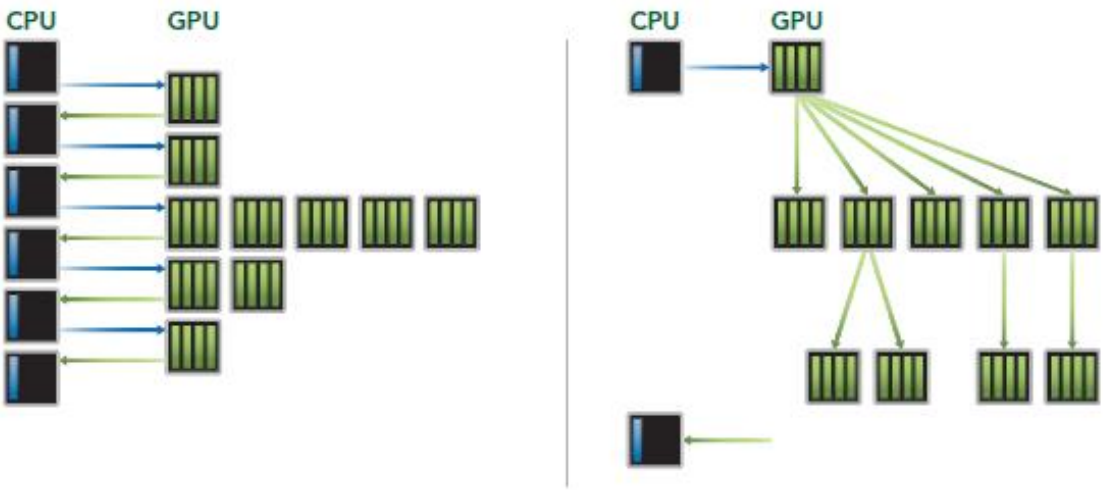
三、Kepler 架构

1、Kepler相较于Fermi更快，效率更高，性能更好。

- 15个SM
- 6个64位memory controller
- 192个单精度CUDA cores，64个双精度单元，32个SFU，32个load/store单元（LD/ST）
- 增加register file到64K
- 每个Kepler的SM包含四个warp scheduler、八个instruction dispatchers，使得每个SM可以同时issue和执行四个warp。
- Kepler K20X（compute capability 3.5）每个SM可以同时调度64个warp共计2048个thread。

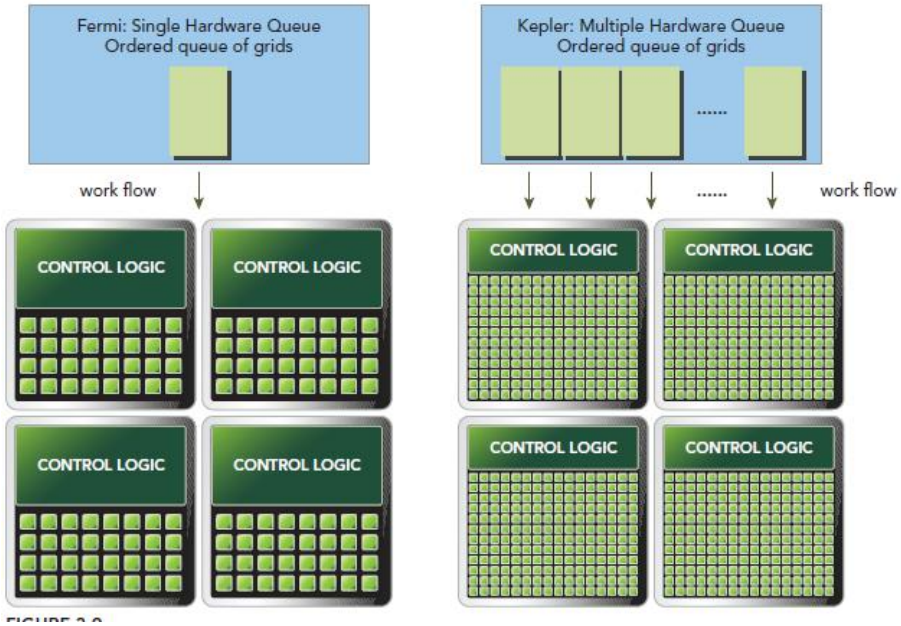
2、Dynamic Parallelism

Dynamic Parallelism是Kepler的新特性，允许GPU动态的启动新的Grid。有了这个特性，任何kernel内都可以启动其它的kernel了。这样直接实现了kernel的递归以及解决了kernel之间数据的依赖问题。也许D3D中光的散射可以用这个实现。



3、Hyper-Q

Hyper-Q是Kepler的另一个新特性，增加了CPU和GPU之间硬件上的联系，使CPU可以在GPU上同时运行更多的任务。这样就可以增加GPU的利用率减少CPU的闲置时间。Fermi依赖一个单独的硬件上的工作队列来从CPU传递任务给GPU，这样在某个任务阻塞时，会导致之后的任务无法得到处理，Hyper-Q解决了这个问题。相应的，Kepler为GPU和CPU提供了32个工作队列。



不同arch的主要参数对比

ARCHITECTURE SPECIFICATIONS	COMPUTE CAPABILITY			
	2.0	2.1	3.0	3.5
Number of cores for integer and floating-point arithmetic function operations per multiprocessor	32	48	192	
Number of special function units for single-precision floating-point transcendental functions per multiprocessor	4	8	32	
Number of warp schedulers per multiprocessor	2		4	
Number of instructions issued at once by scheduler	1	2	2	
Number of load/store units per multiprocessor	16		32	

Load/Store address width	64 B	64 B
L2 cache	768 K	1,536 K
On-chip memory per multiprocessor	64 K	64 K
Shared memory per multiprocessor (configurable)	48 K or 16 K	48 K/32 K/16 K
L1 cache per multiprocessor (configurable)	16 K or 48 K	48 K/32 K/16 K
Read-only data cache	N/A	48 K
Global memory	Up to 6 GB	Up to 12 GB

四、Volta架构

- 6个GPCs
- 84个Volta SM
- 42个TPC（每个包括2个SM）
- 8个512位内存控制器（总共4096位）
- 每个SM 有64个 FP32核、64个INT32核、32个FP64核和8个新张量核。
- 每个SM也包括四个纹理单元。
- 一个完整的GV100 GPU共有5376个FP32核、5376个INT32核，2688个FP64核、672个张量核和336个纹理单元。
- 每个内存控制器连接到768 KB的L2高速缓存，每个HBM2DRAM堆栈由一对内存控制器控制。
- 完整的GV100 GPU共6144KB L2高速缓存。

Volta SM的架构是设计来提供更高的性能的，它的设计比过去的SM设计降低了指令和高速缓存的延迟，并且包括了新的功能来加速沈度学习的应用。 主要特征包括：

- 专为深度学习矩阵算法建造的新混合精度FP16 / FP32张量核
- 增强的L1数据缓存，达到更高的性能和更低的延迟
- 简单的解码和减少指令延迟的精简指令集
- 更高的频率和更高的功率效率。

类似于Pascal GP100，GV100 每个SM包含64个FP32核和32个FP64核。然而，GV100 SM采用一种新的划分方法，提高SM的利用率和整体性能。GP100 SM被划分成两个处理模块，每个有32个 FP32核，16个FP64核，一个指令缓冲器，一个warp调度，两个派发单元，和一个128 kb的登记文件。GV100 SM被划分成四个处理块，每组16个FP32核、8个FP6416核，16个Int32核，2个为深度学习矩阵运算设计的新的混合精度张量核，新的10指令缓存，一个warp调度，一个派发单元，以及一个64 kb的登记文件。请注意，新的L0指令缓存，现在使用在每个分区内，来提供比以前的NVIDIA GPU的指令缓冲器更高的效率。

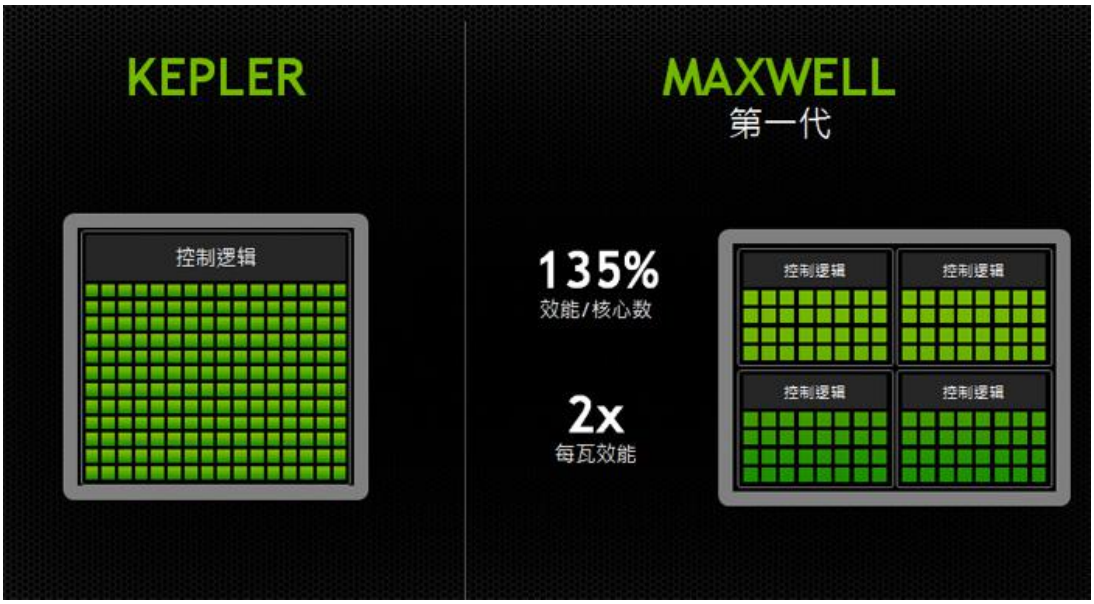
尽管GV100 SM与Pascal GP100 SM具有相同数量的寄存器，整个GV100 GPU拥有更多的SM，从而整体上有更多的寄存器。总的来说，GV100支持多线程，变形，和与之前的GPU相比，具有了线程块。在整个GV100 GPU上，由于SM数增加，以及每个SM的共享内存的潜力增加到96KB，相比GP100的64 KB，全局共享内存也有所增加。Pascal GPU无法同时执行FP32和Int32指令，与它不同的Volta GV100 SM包括单独的FP32和INT32核，允许在全吞吐量上同时执行FP32和INT32的操作，但同时也增加了指令问题的吞吐量。相关的指令问题

延迟也通过核心FMA的数学操作得到减少，Volta只需要四个时钟周期，而Pascal需要六个。

五、Maxwell架构

Maxwell架构简介：SMM重组，更精确的控制

Kepler架构发布到现在接近2年的时间了，期间的GK104和GK110核心虽然差异极大，不过内部组成上只算是小修补，GK110增大了位宽、提升了双精度运算单元及SMX单元数量，而Maxwell架构是全新的，NVIDIA重新设计了Kepler上的SMX单元，现在称为SMM（Maxwell Streaming Multiprocessor）单元，每组SMM单元的CUDA核心数量有所减少，但是每个SMM单元将将拥有更多的逻辑控制电路，便于精确控制，这使得GM107核心的每核心效能提升了35%，每瓦功耗比提升了一倍。

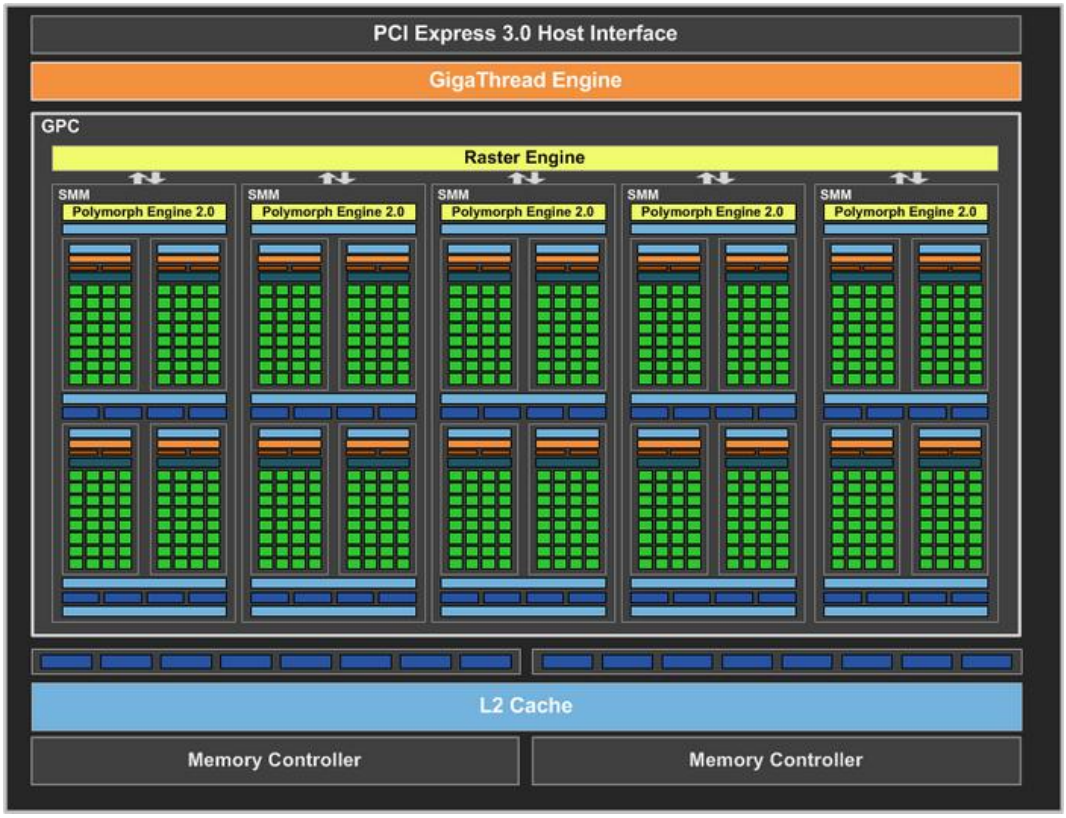


Maxwell架构的SMM单元重新设计，加入了更多的逻辑控制电路回顾Kepler的SMX设计，此前我们在GTX 680显卡的评测中有过详细介绍：Kepler架构解析：3倍核心是怎样练成的，SMX单元的CUDA核心从之前Fermi架构的32（GF110）/48（GF114）跃升到了192个，每组SMX有16个TMU纹理单元，总计8组SMX单元，搭配4条64bit GDDR5内存控制器，32个ROP单元。

GK110核心中SMX单元数量增加到15组

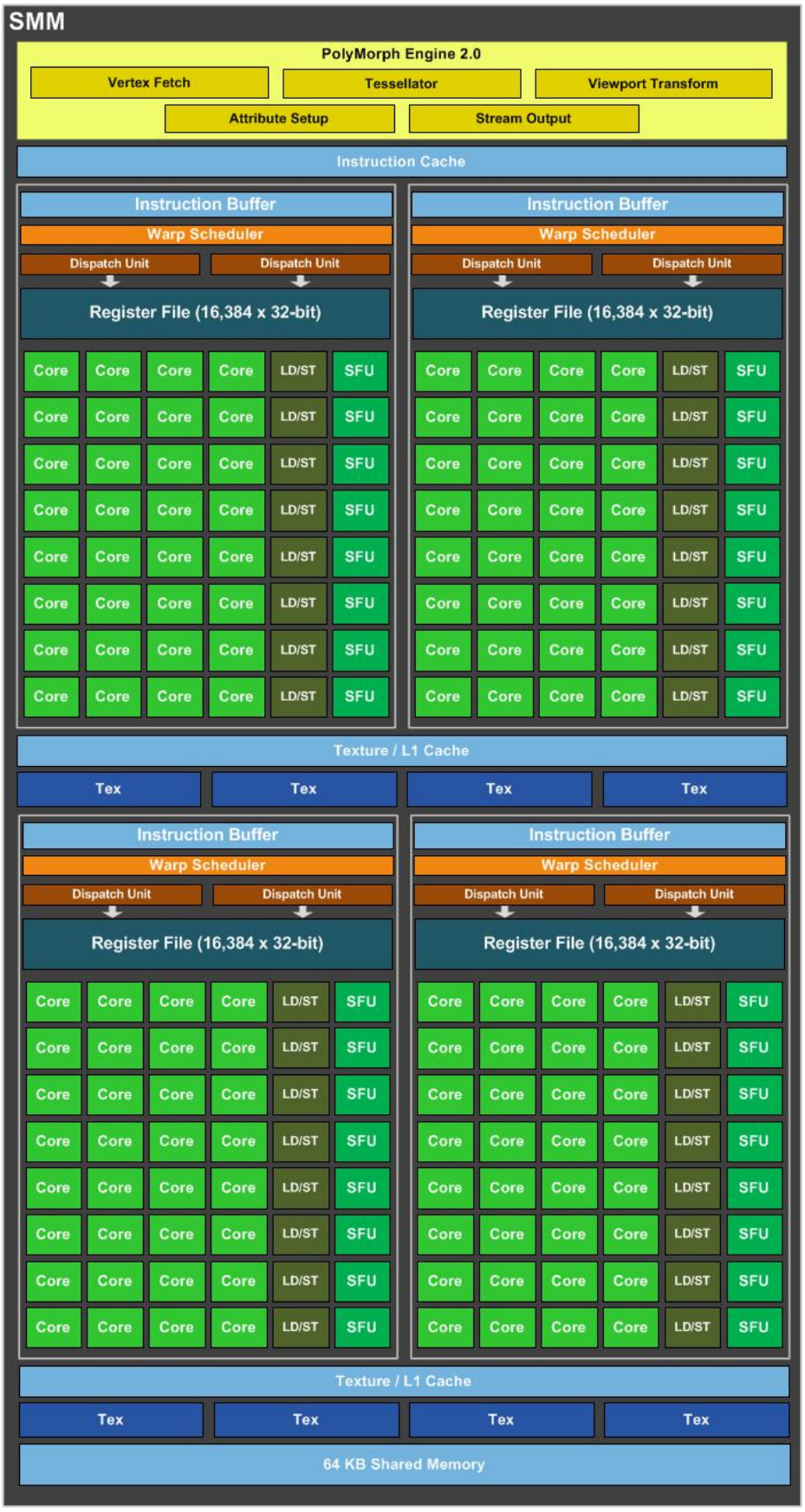
去年的GK110核心与GTX 680的GK104相比，规模大幅提升，15组SMX单元，不过具体的SMX单元变化不大，每组SMX单元还是192个CUDA核心，但是增加了64个DP双精度单元，并引入了

Hyper-Q、动态并行等新功能，因为GK110核心跟GK104核心针对的市场不同，它不仅游戏性能更强，计算性能也是一大重点。



GM107的核心设计

Maxwell这一代首先推出的是GM107核心的，之前的SMX单元变成了SMM，总计5组SMM单元，而每组SMM单元又由4个小单元组成，每组32个CUDA核心，TMU单元又降低到8个，位宽维持128bit，因此CUDA核心总数为640个，纹理单元40个，ROP单元为16个。



每组SMM单元由4组更小的单元组成，每个单元拥有独立的逻辑控制电路

NVIDIA Maxwell架构节能技术

Maxwell效率上的提升主要归功于全新的 Maxwell SM架构，即SMM。这种全新的 SM 架构可大幅提升节能性，而且在着色器有限的工作场合中可令每个CUDA核心的性能提升 35%。实现这些进步需要对架构进行大量重大更改。 NVIDIA重新编写了SM调度器架构和算法，使其更加智能，避免了不必要的停顿，同时进一步降低了调度每条指令所需的能耗。

当然，SMM单元也有很多改进的地方，比如L2缓存容量从之前的256KB大幅增加到2MB, H.264及NVENC编码/解码能力也提升了，指令周期性能也改善了。

这样对比下来，Maxwell架构的SMM单元的CUDA核心实际上是从上代的192个减少到128个，规模还缩减了，而且NVIDIA还没有公布详细的Maxwell架构文档，所以SMM单元的实际运算、调度能力、多边形计算等性能到底提升了多少还不得而知，但是有一点可以肯定的是：在同样的28nm工艺下，Maxwell架构的晶体管密度提升了，单位面积的CUDA核心数也提升了，每瓦性能比也提升了，符合NVIDIA路线图中2倍每瓦性能比的宣传