# 　谈GPU的作用、原理及与CPU、DSP的区别

GPU是显示卡的“心脏”，也就相当于CPU在电脑中的作用，它决定了该显卡的档次和大部分性能，同时也是2D显示卡和3D显示卡的区别依据。2D显示芯片在处理3D图像和特效时主要依赖CPU的处理能力，称为“软加速”。3D显示芯片是将三维图像和特效处理功能集中在显示芯片内，也即所谓的“硬件加速”功能。显示芯片通常是显示卡上最大的芯片（也是引脚最多的）。GPU使显卡减少了对CPU的依赖，并进行部分原本CPU的工作，尤其是在3D图形处理时。GPU所采用的核心技术有硬体T&L、立方环境材质贴图和顶点混合、纹理压缩和凹凸映射贴图、双重纹理四像素256位渲染引擎等，而硬体T&L技术可以说是GPU的标志。

　　GPU 能够从硬件上支持T&L（TransformandLighting，多边形转换与光源处理）的显示芯片，因为T&L是3D渲染中的一个重要部分，其作用是计算多边形的3D位置和处理动态光线效果，也可以称为“几何处理”。一个好的T&L单元，可以提供细致的3D物体和高级的光线特效；只不过大多数PC中，T&L的大部分运算是交由CPU处理的（这就也就是所谓的软件T&L），由于CPU的任务繁多，除了 T&L之外，还要做内存管理、输入响应等非3D图形处理工作，因此在实际运算的时候性能会大打折扣，常常出现显卡等待CPU数据的情况，其运算速度远跟不上今天复杂三维游戏的要求。即使CPU的工作频率超过1GHz或更高，对它的帮助也不大，由于这是PC本身设计造成的问题，与CPU的速度无太大关系。

　　**主要作用**

　　今天，GPU已经不再局限于3D图形处理了，GPU通用计算技术发展已经引起业界不少的关注，事实也证明在浮点运算、并行计算等部分计算方面，GPU可以提供数十倍乃至于上百倍于CPU的性能，如此强悍的“新星”难免会让CPU厂商老大英特尔为未来而紧张，NVIDIA和英特尔也经常为CPU和GPU谁更重要而展开口水战。GPU通用计算方面的标准目前有 OPEN CL、CUDA、ATI STREAM。其中，OpenCL（全称Open Computing Language，开放运算语言）是第一个面向异构系统通用目的并行编程的开放式、免费标准，也是一个统一的编程环境，便于软件开发人员为高性能计算服务器、桌面计算系统、手持设备编写高效轻便的代码，而且广泛适用于多核心处理器（CPU）、图形处理器（GPU）、Cell类型架构以及数字信号处理器（DSP） 等其他并行处理器，在游戏、娱乐、科研、医疗等各种领域都有广阔的发展前景，AMD-ATI、NVIDIA现在的产品都支持OPEN CL。 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NV显卡的芯就用这个新名字GPU来称呼。GPU使显卡减少了对CPU的依赖，并进行部分原本CPU的工作，尤其是在3D图形处理时。GPU所采用的核心技术有硬体T&L、立方环境材质贴图和顶点混合、纹理压缩和凹凸映射贴图、双重纹理四像素256 位渲染引擎等，而硬体T&L技术可以说是GPU的标志。

　　**工作原理**

　　简单说GPU就是能够从硬件上支持T&L（Transform and Lighting，多边形转换与光源处理）的显示芯片，因为T&L是3D渲染中的一个重要部分，其作用是计算多边形的3D位置和处理动态光线效果，也可以称为“几何处理”。一个好的T&L单元，可以提供细致的3D物体和高级的光线特效；只不过大多数PC中，T&L的大部分运算是交由CPU处理的（这就也就是所谓的软件T&L），由于CPU的任务繁多，除了T&L之外，还要做内存管理、输入响应等非3D图形处理工作，因此在实际运算的时候性能会大打折扣，常常出现显卡等待CPU数据的情况，其运算速度远跟不上今天复杂三维游戏的要求。即使CPU的工作频率超过 1GHz或更高，对它的帮助也不大，由于这是PC本身设计造成的问题，与CPU的速度无太大关系。

　　**GPU与DSP区别**

　　GPU在几个主要方面有别于DSP（Digital Signal Processing，简称DSP（数字信号处理）架构。其所有计算均使用浮点算法，而且目前还没有位或整数运算指令。此外，由于GPU专为图像处理设计，因此存储系统实际上是一个二维的分段存储空间，包括一个区段号（从中读取图像）和二维地址（图像中的X、Y坐标）。此外，没有任何间接写指令。输出写地址由光栅处理器确定，而且不能由程序改变。这对于自然分布在存储器之中的算法而言是极大的挑战。最后一点，不同碎片的处理过程间不允许通信。实际上，碎片处理器是一个SIMD数据并行执行单元，在所有碎片中独立执行代码。

　　尽管有上述约束，但是GPU还是可以有效地执行多种运算，从线性代数和信号处理到数值仿真。虽然概念简单，但新用户在使用GPU计算时还是会感到迷惑，因为GPU需要专有的图形知识。这种情况下，一些软件工具可以提供帮助。两种高级描影语言CG和HLSL能够让用户编写类似C的代码，随后编译成碎片程序汇编语言。Brook是专为GPU计算设计，且不需要图形知识的高级语言。因此对第一次使用GPU进行开发的工作人员而言，它可以算是一个很好的起点。Brook是C语言的延伸，整合了可以直接映射到 GPU的简单数据并行编程构造。经 GPU存储和操作的数据被形象地比喻成“流”（stream），类似于标准C中的数组。核心（Kernel）是在流上操作的函数。在一系列输入流上调用一个核心函数意味着在流元素上实施了隐含的循环，即对每一个流元素调用核心体。Brook还提供了约简机制，例如对一个流中所有的元素进行和、最大值或乘积计算。Brook还完全隐藏了图形API的所有细节，并把GPU中类似二维存储器系统这样许多用户不熟悉的部分进行了虚拟化处理。用Brook编写的应用程序包括线性代数子程序、快速傅立叶转换、光线追踪和图像处理。利用ATI的X800XT和Nvidia的GeForce 6800 Ultra型GPU，在相同高速缓存、SSE汇编优化Pentium 4执行条件下，许多此类应用的速度提升高达7倍之多。

　　对GPU计算感兴趣的用户努力将算法映射到图形基本元素。类似Brook这样的高级编程语言的问世使编程新手也能够很容易就掌握GPU的性能优势。访问GPU计算功能的便利性也使得GPU的演变将继续下去，不仅仅作为绘制引擎，而是会成为个人电脑的主要计算引擎。

　　**GPU和CPU的区别是什么？**

　　要解释两者的区别，要先明白两者的相同之处：两者都有总线和外界联系，有自己的缓存体系，以及数字和逻辑运算单元。一句话，两者都为了完成计算任务而设计。

　　两者的区别在于存在于片内的缓存体系和数字逻辑运算单元的结构差异：CPU虽然有多核，但总数没有超过两位数，每个核都有足够大的缓存和足够多的数字和逻辑运算单元，并辅助有很多加速分支判断甚至更复杂的逻辑判断的硬件；GPU的核数远超CPU，被称为众核（NVIDIA Fermi有512个核）。每个核拥有的缓存大小相对小，数字逻辑运算单元也少而简单（GPU初始时在浮点计算上一直弱于CPU）。从结果上导致CPU擅长处理具有复杂计算步骤和复杂数据依赖的计算任务，如分布式计算，数据压缩，人工智能，物理模拟，以及其他很多很多计算任务等。GPU由于历史原因，是为了视频游戏而产生的（至今其主要驱动力还是不断增长的视频游戏市场），在三维游戏中常常出现的一类操作是对海量数据进行相同的操作，如：对每一个顶点进行同样的坐标变换，对每一个顶点按照同样的光照模型计算颜色值。GPU的众核架构非常适合把同样的指令流并行发送到众核上，采用不同的输入数据执行。在 2003-2004年左右，图形学之外的领域专家开始注意到GPU与众不同的计算能力，开始尝试把GPU用于通用计算（即GPGPU）。之后NVIDIA 发布了CUDA，AMD和Apple等公司也发布了OpenCL，GPU开始在通用计算领域得到广泛应用，包括：数值分析，海量数据处理（排序，Map- Reduce等），金融分析等等。

　　简而言之，当程序员为CPU编写程序时，他们倾向于利用复杂的逻辑结构优化算法从而减少计算任务的运行时间，即Latency。当程序员为GPU编写程序时，则利用其处理海量数据的优势，通过提高总的数据吞吐量（Throughput）来掩盖 Lantency。目前，CPU和GPU的区别正在逐渐缩小，因为GPU也在处理不规则任务和线程间通信方面有了长足的进步。另外，功耗问题对于GPU比 CPU更严重。

　　总的来讲，GPU和CPU的区别是个很大的话题，甚至可以花一个学期用32个学时十几次讲座来讲。