GPU 架构与技术详解

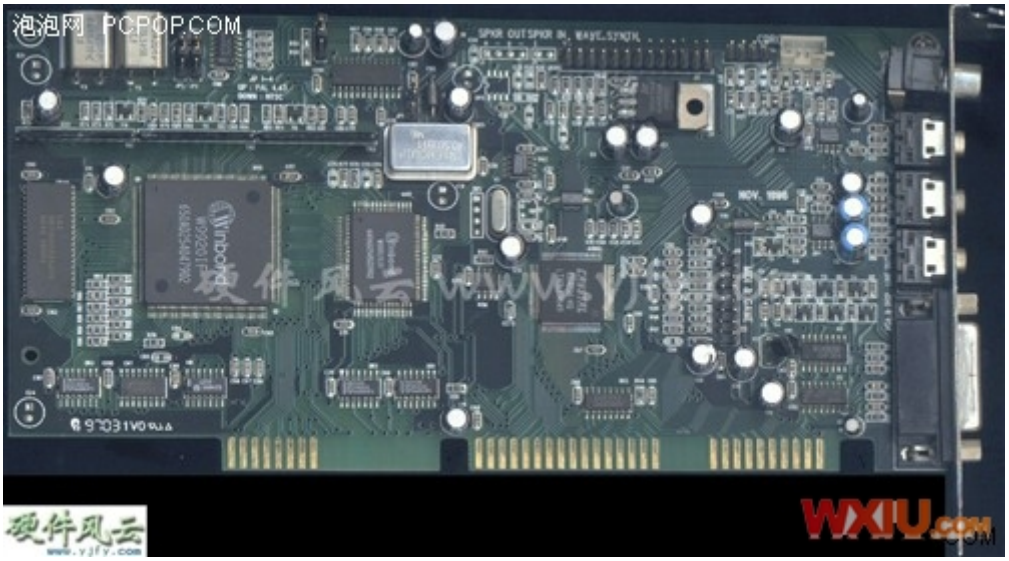
来源: wxiu.com 时间: 2010-06-22 作者: apollo

GPU 英文全称 Graphic Processing Unit，中文翻译为“图形处理器”。GPU 是相对于 CPU 的一个概念，由于在现代的计算机中（特别是家用系统，游戏的发烧友）图形的处理变得越来越重要，需要一个专门的图形的核心处理器。我们从 GPU 的发展历程来看看显卡 GPU 的架构和技术的发展。

整合 VCD/DVD/HD/BD 解压卡

在了解了CPU的发展历程之后，我们再来看看GPU的发展过程，其实GPU很多重大改进都与CPU的技术架构相类似。比如最开始我们介绍了古老的CPU协处理器，下面再介绍一个被遗忘的产品——解压卡，资历较老的玩家应该记得。

十多年前，电脑的 CPU 主频很低，显卡也多为 2D 显示用，当 VCD 兴起的时候，好多电脑（主频为 100MHz 以下）无法以软解压的方式看 VCD 影片，根本运行不起来!



ISA 接口的 VCD 解压卡

这时，VCD 解压卡就出现了，此卡板载专用的解码处理器和缓存，实现对 VCD 的硬解码，不需要 CPU 进行解码运算，所以，即使在 386 的电脑上也可以看 VCD 了。

PCI 接口的 DVD 解压卡

随后，显卡进入了 3D 时代，并纷纷加入支持 VCD 的 MPEG 解码，而且 CPU 的主频也上来了，无论 CPU 软解还是显卡辅助解码都可以流畅播放视频，所以 VCD 解压卡就退出了市场！

但 DVD 时代来临后，分辨率提高很多，而且编码升级至 MPEG2，对于 CPU 和显卡的解码能力提出了新的要求，此时出现了一些 DVD 解压卡，供老机器升级之用，但由于 CPU 更新换代更加频繁，性能提升很大，DVD 解压卡也是昙花一现，就消失无踪了。

现在已经是 1080p 全高清时代了，高清视频解码依然是非常消耗 CPU 资源的应用之一，于是几年前 NVIDIA 和 ATI 就在 GPU 当中整合了专用的视频解码模块，NVIDIA 将其称为 VP（Video Processor，视频处理器），ATI 将其称为 UVD（Unified Video Decoder，通用视频解码器），相应的技术被叫做 PureVideo 和 AVIVO。

硬解码几乎不消耗 CPU 和 GPU 的资源，看高清视频时接近于待机状态

虽然 VP 和 UVD 都被整合在了 GPU 内部，实际上它们的原理和作用与当年的协处理器/解压卡芯片没有实质性区别，都是为了减轻/分担处理器的某项特定任务。如今 NVIDIA 和 ATI 的 GPU 硬解码技术都能够支持高分辨率、高码率、多部影片同时播放，性能和兼容性都很出色。

如今多核 CPU 的性能已经相当强大了，软解高清视频简直轻松加愉快，但要论效率的话，依然是 GPU 硬件解码更胜一筹，专用模块解码消耗资源更少，整机功耗发热更小，因此手持设备和移动设备都使用硬件解码，而桌面电脑 CPU 软解和 GPU 硬解就无所谓了。

ShaderModel 指令集的扩充与发展

掐指一算，从 GPU 诞生至今双方都已推出了十代产品，每一代产品之间的对决都令无数玩家心动不已，而其中最精彩的战役往往在微软 DirectX API 版本更新时出现，几乎可以说是微软 DirectX 左右着 GPU 的发展，而历代 DirectX 版本更新时的核心内容，恰恰包含在了 ShaderModel 当中：

ShaderModel 1.0 → DirectX 8.0
ShaderModel 2.0 → DirectX 9.0b
ShaderModel 3.0 → DirectX 9.0c
ShaderModel 4.0 → DirectX 10
ShaderModel 5.0 → DirectX 11

Shader（译为渲染或着色）是一段能够针对 3D 对象进行操作、并被 GPU 所执行的程序，ShaderModel 的含义就是“优化渲染引擎模式”，我们可以把它理解成是 GPU 的渲染指令集。

高版本的 ShaderModel 是一个包括了所有低版本特性的超集，对一些指令集加以扩充改进的同时，还加入了一些新的技术。可以说，GPU 的 ShaderModel 指令集与 CPU 的 MMX、SSE 等扩展指令集十分相似。

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 regis-	≥96	≥256	≥256	16x4096
ters	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.
†specification released in 2002, hardware in 2004; ‡texture load + arithmetic instructions; §texture + color registers; dashed line separates vertex shader (above) from pixel shader (below)

随着 ShaderModel 指令集的扩充与改进，GPU 的处理资源和计算精度与日俱增，于是就有能力渲染出更加精美的图像，并且不至于造成性能的大幅下降。就拿最近几个版本来讲，新指令集并没有带来太多新的特效，但却凭借优秀的算法提升了性能，是否支持 DX10.1（ShaderModel 4.1）可能游戏画面上没有差别，但速度就很明显了。

DirectX 11 Shader Model 5.0 新指令		
特性	**功能**	**效果**
覆盖采样	直接为像素着色器输出覆盖采样信息	边缘侦测更加精确 抗锯齿运算效率和效果更佳
Gather 函数加速纹理拾取	在一个纹理指令里读取 4 点采样值 可针对特定颜色分别采样 自动识别能做阴影映射的值	更快/更好的阴影过滤 实现环境光遮蔽（SSAO）
粗糙偏导数	用数学函数定义简单的二维纹理图案，如方格地毯 或用数学函数定义随机高度场，生成表面粗糙纹理即几何纹理	高性能、高画质纹理过滤
类型转化类指令	将数据值在 32bit 浮点和 16bit 浮点之间相互转换	简单高效译码 双精度支持
位操作类指令	转位、排序、封包、压缩、解压	数据压缩解压缩速度提升 传输率提高

DirectCompute 10 和 11 的区别			
特性	**10.0**	**11.0**	**优势**
线程派遣	2D	3D	用单一的 3D 阵列取代 2D 线性阵列组 快速高效节约资源
线程限制	768	1024	同时可执行的线程数提升 1/3 并行计算效能提升
线程组共享缓存	16KB	32KB	线程间的共享资源翻倍
共享缓存存取	写入限制 256bit	完整 32KB 可用 读写不限	减轻 I/O 压力，缓存利用率更高
原子操作	不支持	支持	允许每个线程在受保护的内存区域内操作，避免被其他线程打断，程序员现在可以像 CPU 那样对 GPU 进行编程
双精度	不支持	支持	支持 IEEE754 标准的 64bit 浮点运算
附加缓冲	不支持	支持	一种新的数据缓存格式，位于传统缓存数据列表末尾，可提高缓存空间利用率，降低显存压力
计算着色器顺序无关存取	1	8	允许计算着色器同时访问 8 个位置的缓存
像素着色器顺序无关存取	不支持	8	允许像素着色器直接访问计算着色器的数据，提高图形相关通用计算的运行效能
Gather4	不支持	支持	纹理数据提拾取速度提高 4 倍

此外，DX11 中的关键技术 DirectCompute 通用计算技术就是通过调用 ShaderModel 5.0 中的新指令集来提高 GPU 的运算效率，很多基于 DirectCompute 技术的图形后处理渲染特效也都要用到 SM5.0 指令集来提高性能。

真正的双核/四核 GPU

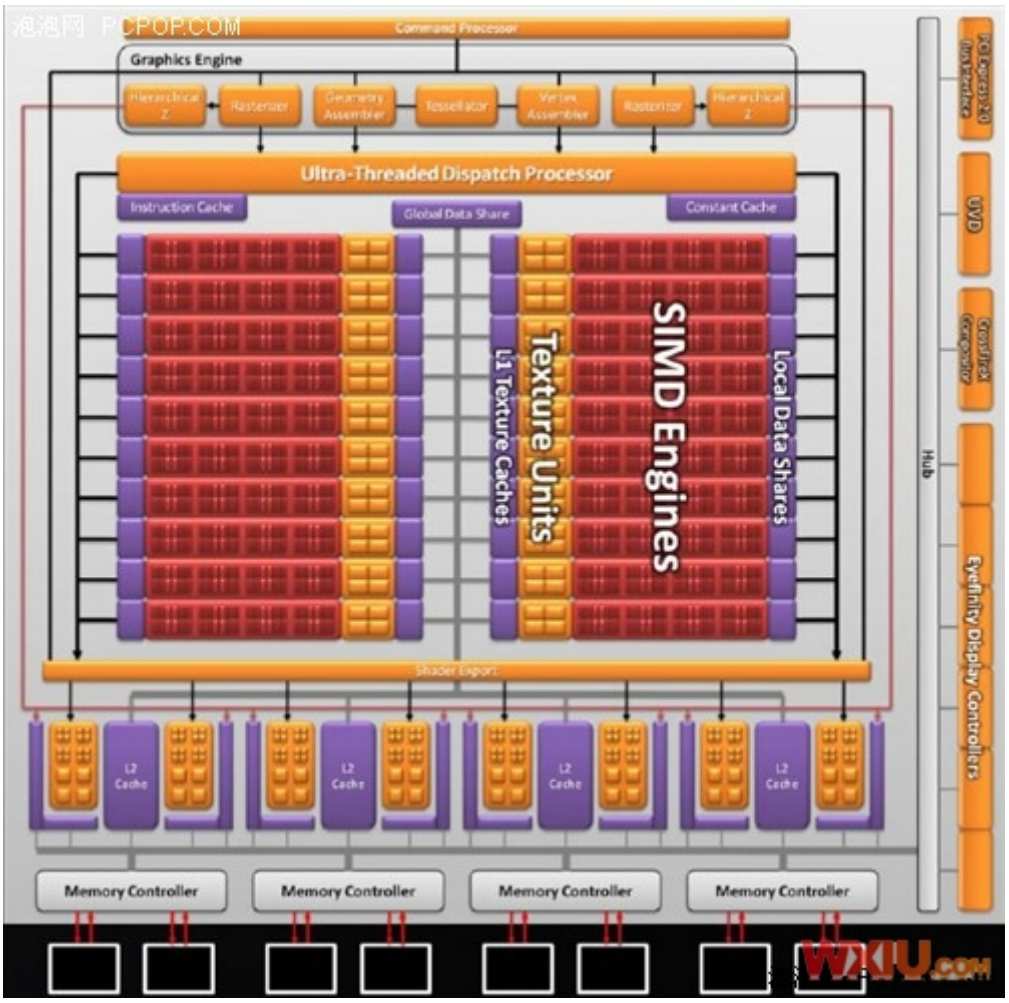
从以往的多处理器系统到现在的双核、四核、六核，CPU 只能依靠增加核心数量来提升性能。而 GPU 从一开始就是作为并行渲染的管线式架构，GPU 性能的强弱主要就是看谁的管线、流处理器数量更多。

不过双显卡甚至多显卡也成为提升电脑游戏性能的一种途径，通过 SLI 和 CrossFire 技术能够轻松让 3D 性能倍增，于是双核心的显卡成为 NVIDIA 和 AMD 双方角逐 3D 性能王者宝座的杀手锏，近年来的旗舰级显卡几乎都是双核心设计的。

但与 CPU 单芯片整合多核心的设计不同，显卡一般是单卡多 GPU 设计，很少有单一 GPU 多核心设计，因为 GPU 性能提升的瓶颈主要在于制造工艺，只要工艺跟得上，那么他们就有能力在 GPU 内部植入尽可能多的流处理器。

★ 双核心设计的 Cypress 核心：

不管 GPU 架构改不改，流处理器数量总是要扩充的，准确的说是以级数规模增长，这样才能大幅提升理论性能。在流处理器数量急剧膨胀之后，如何管理好如此庞大的规模、并与其它模块协调工作成为新的难题。



RV870 的双核心模块设计

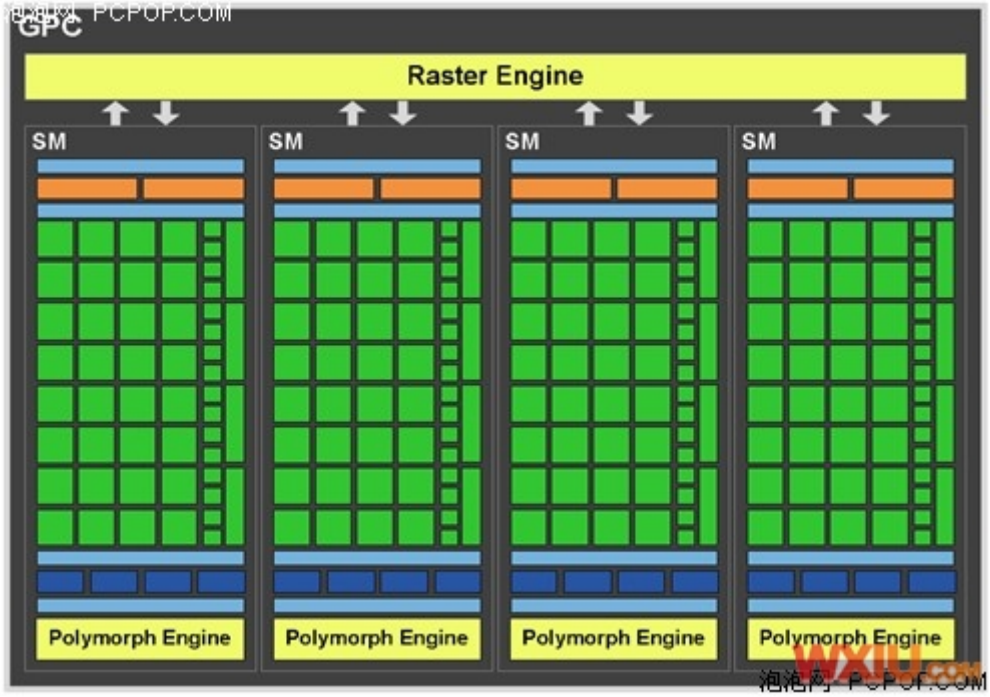
ATI RV870 包括流处理器在内的所有核心规格都比 RV770 翻了一倍，ATI 选择了“双核心”设计，几乎是并排放置两颗 RV770 核心，另外在装配引擎内部设计有两个 Rasterizer（光栅器）和 Hierarchial-Z（多级 Z 缓冲模块），以满足双倍核心规格的胃口。

★ 四核心设计的 GF100 核心：

GF100 可以看作是四核心设计

如果说 Cypress 是双核心设计的话，那么 GF100 的流处理器部分就是“四核心”设计，因为 GF100 拥有四个 GPC（图形处理器集群）模块，每个 GPC 内部包含一个独立的 Raster Engine（光栅化引擎），而在以往都是整颗 GPU 共享一个 Raster Engine。

我们知道 RV870 的 Rasterizer 和 Hierarchial-Z 双份的，而 GF100 则是四份的，虽然命名有所不同但功能是相同的。



GF100 的每个 GPC 都可以看作是一个自给自足的 GPU

GF100 的四个 GPC 是完全相同的，每个 GPC 内部囊括了所有主要的图形处理单元。它代表了顶点、几何、光栅、纹理以及像素处理资源的均衡集合。除了 ROP 功能以外，GPC 可以被看作是一个自给自足的 GPU，所以说 GF100 就是一颗四核心的 GPU。

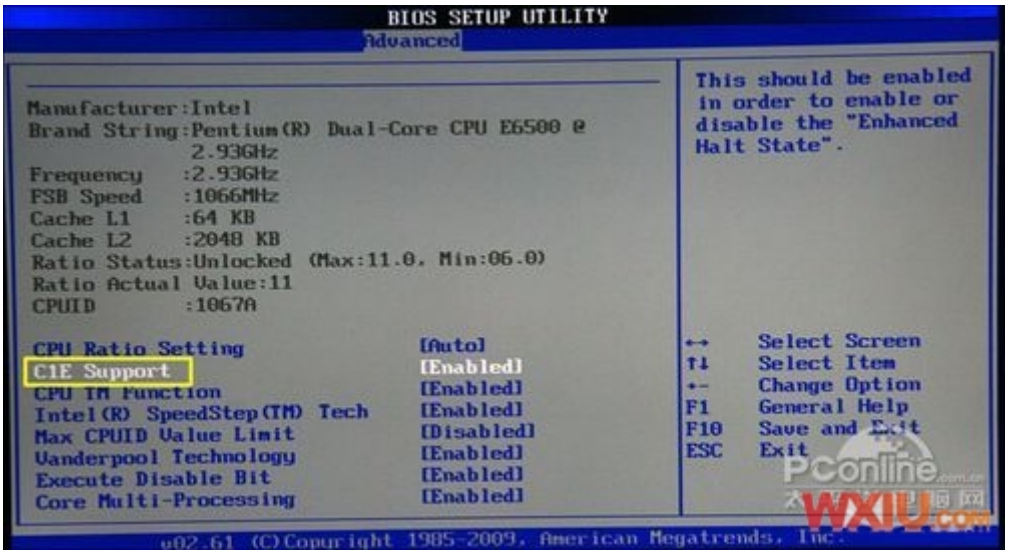
CPU 三大节能技术简单介绍

来源: http://www.wxiu.com/ 时间: 2010-09-03 作者: apollo

随着低碳概念的推广，节能技术在生活中开始变成无处不在，特别是在电子设备领域，各种节能技术运用地非常多。现在很多朋友在选购电脑的时候都会将节能省电放在考虑因素里面，那么电脑中有那些节能技术呢？下面我们先看看 CPU 的三大节能技术。

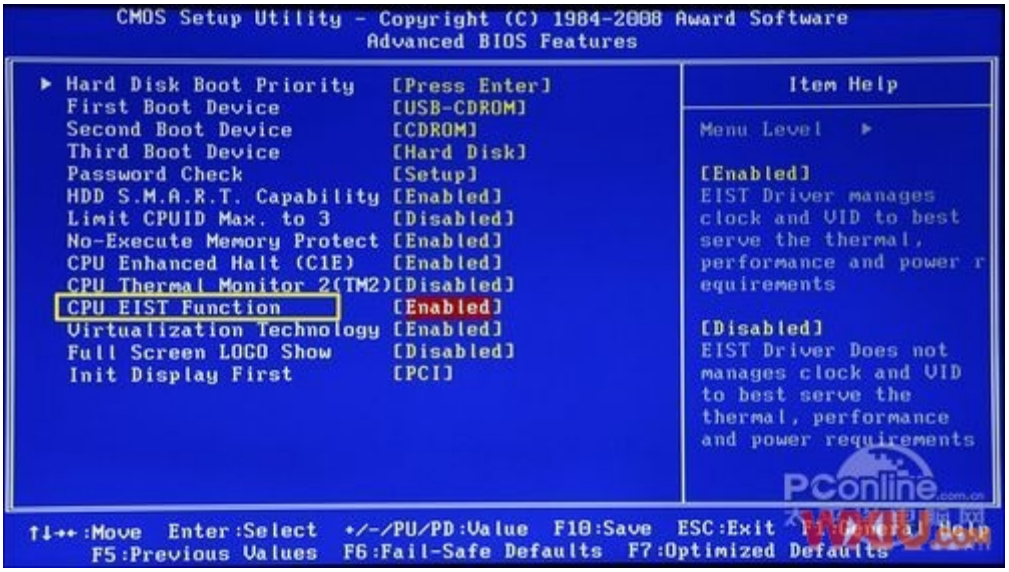
1、C1E 节能 （增强型深度休眠技术）

在当前的主流系统中（包括 Intel 和 AMD 平台），我们都可以看到一个“C1E”的选项。它是一种可以令 CPU 省电的功能，开启后，CPU 在空闲轻负载状态可以降低工作电压与倍频，这样就达到了省电的目的。



2、Intel EIST 技术（增强型电源管理技术）

EIST 全称为“Enhanced Intel SpeedStep Technology”，最早是 Intel 公司专门为移动平台和服务器平台处理器开发的一种节电技术。到后来，新推出的桌面处理器也内置了该项技术，比如 Intel 的 Pentium 4 6xx 系列及 Pentium D 全系列处理器都开始支持 EIST 技术。现在基本上成为了处理器的标配技术，不管是桌面还是移动产品。

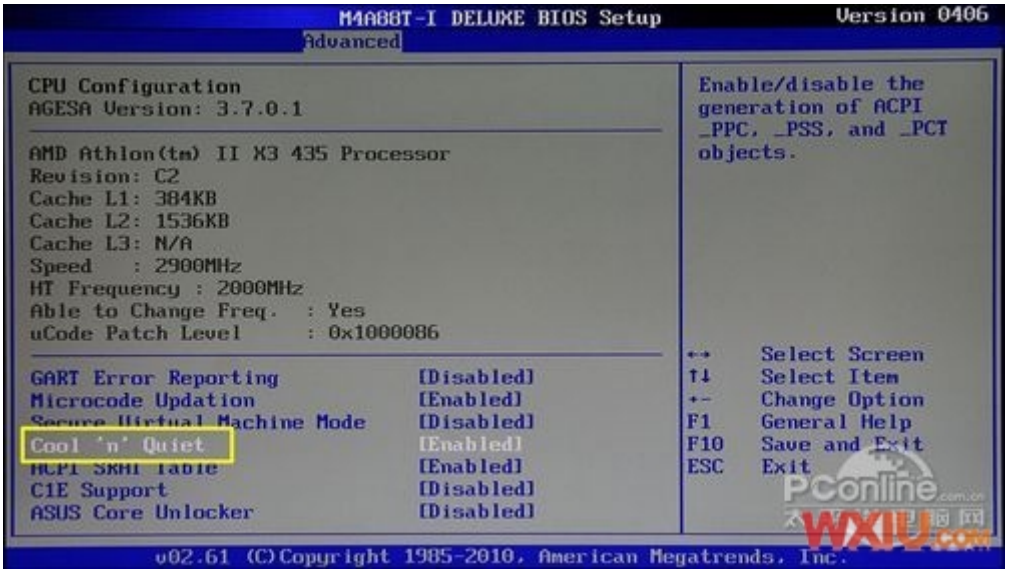


Intel EIST 节能

消费者仅需要在主板 BIOS 中开启“EIST”或“Intel SpeedStep technology”的选项，就能够让 CPU 更具实际使用情况来自己控制频率和电压，进而实现功耗的控制。

3、AMD Cool N' Quiet（“凉又静”）

Cool‘n’Quiet 是 AMD 台式机 CPU 的节能技术，被形象的称为“凉又静”。Cool‘n’Quiet 也是一项能让处理器在闲置状态下自动降低电压与频率的节能技术，与 AMD 移动平台的 PowerNow!非常相似。



AMD Cool‘n’Quiet 节能选项

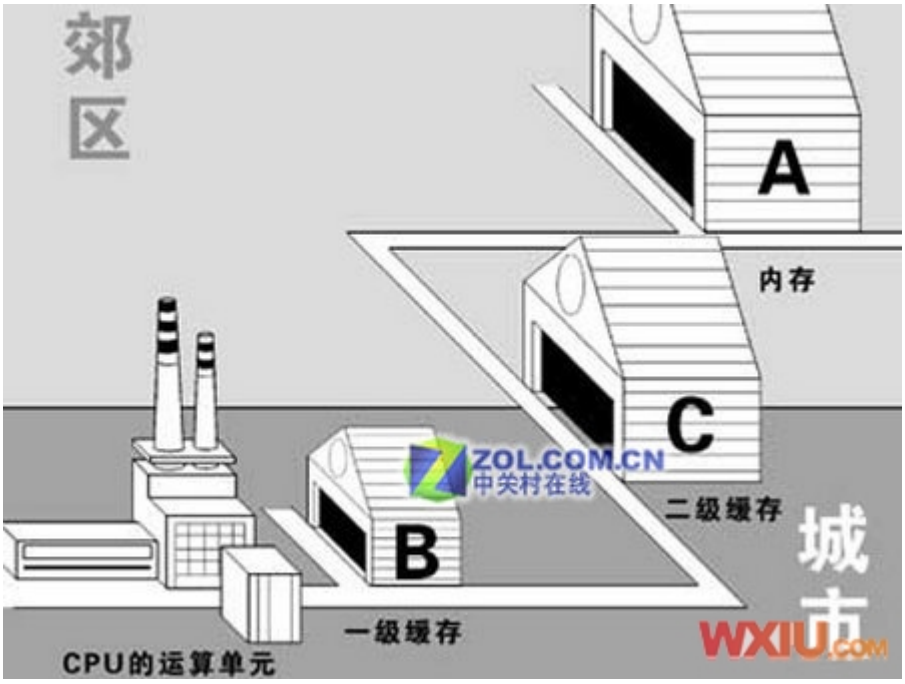
Cool‘n’Quiet 需要处理器硬件、驱动程序和主板 BIOS 三方面的支持，应在主板 BIOS 中将“Power Management”设置页内的“Cool'N'Quiet”选项设成“Auto”，电源使用方案设成“最少电源管理”，并安装 AMD 处理器驱动程序，Cool'N'Quiet 功能才会生效。

CPU 缓存对 CPU 性能的影响

来源: wxiu.com 时间: 2010-07-01 作者: 匿名

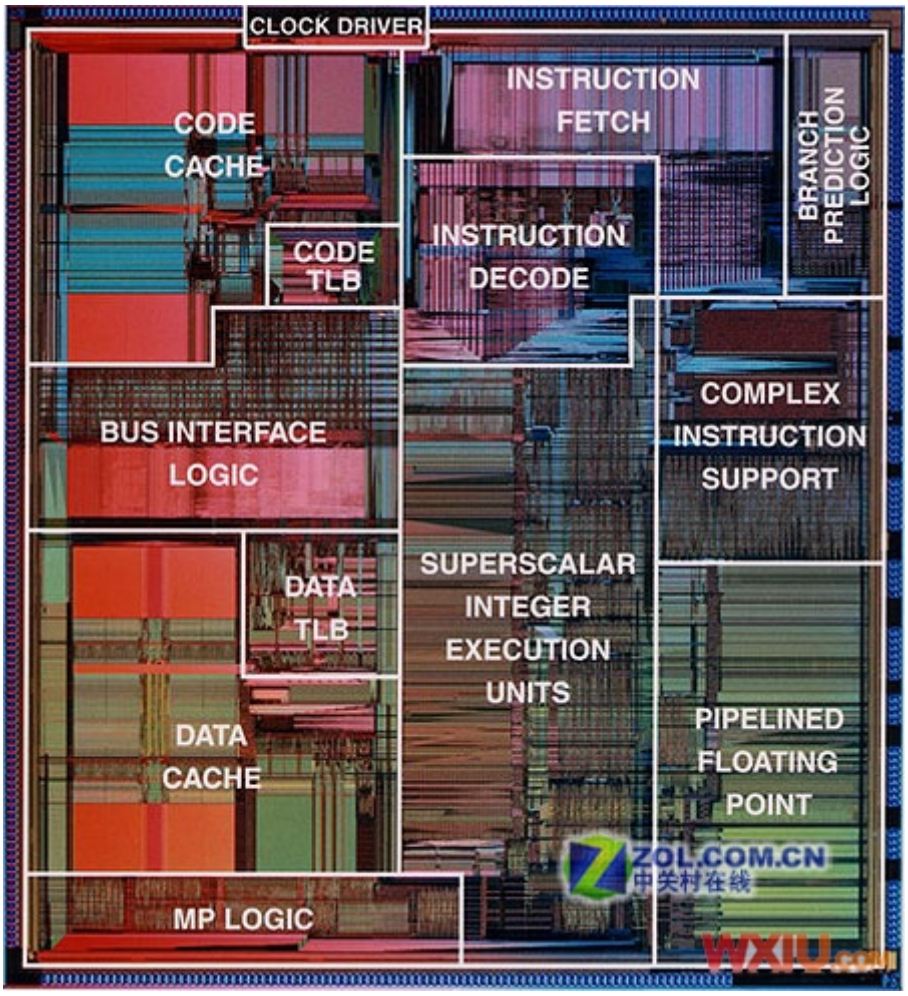
CPU 缓存是什么？CPU 缓存有什么用？CPU 缓存多大才好？这是很多朋友在选购 CPU 时会考虑到的问题。CPU 缓存（Cache Memory）是位于 CPU 与内存之间的临时存储器，它的容量比内存小的多但是交换速度却比内存要快得多。缓存的出现主要是为了解决 CPU 运算速度与内存读写速度不匹配的矛盾，因为 CPU 运算速度要比内存读写速度快很多，这样会使 CPU 花费很长时间等待数据到来或把数据写入内存。下面我们来详细说说 CPU 缓存对 CPU 性能的影响。

缓存的工作原理是当 CPU 要读取一个数据时，首先从缓存中查找，如果找到就立即读取并送给 CPU 处理；如果没有找到，就用相对慢的速度从内存中读取并送给 CPU 处理，同时把这个数据所在的数据块调入缓存中，可以使得以后对整块数据的读取都从缓存中进行，不必再调用内存。

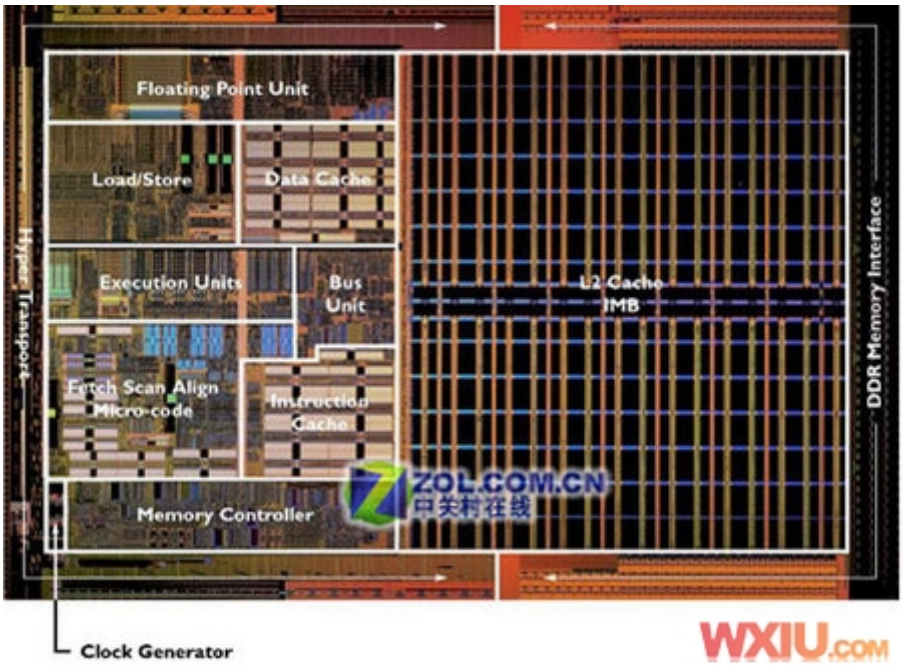


处理器缓存工作原理

正是这样的读取机制使 CPU 读取缓存的命中率非常高（大多数 CPU 可达 90%左右），也就是说 CPU 下一次要读取的数据 90%都在缓存中，只有大约 10%需要从内存读取。这大大节省了 CPU 直接读取内存的时间，也使 CPU 读取数据时基本无需等待。总的来说，CPU 读取数据的顺序是先缓存后内存。


CLOCK DRIVER
CODE CACHE
INSTRUCTION FETCH
BRANCH PREDICTION LOGIC
CODE TLB
INSTRUCTION DECODE
COMPLEX INSTRUCTION SUPPORT
BUS INTERFACE LOGIC
SUPERSCALAR INTEGER EXECUTION UNITS
DATA TLB
DATA CACHE
PIPELINED FLOATING POINT
MP LOGIC
ZOL.COM.CN
中关村在线
WXIU.com

处理器缓存构造


Floating Point Unit
Load/Store
Data Cache
Execution Units
Bus Unit
L2 Cache IMB
Fetch Scan Align Micro-code
Instruction Cache
Memory Controller
HyperTransport
DDR Memory Interface
Clock Generator
ZOL.COM.CN
中关村在线
WXIU.com

L2 级缓存

缓存大小是 CPU 的重要指标之一，而且缓存的结构和大小对 CPU 速度的影响非常大，CPU 内缓存的运行频率极高，一般是和处理器同频运作，工作效率远远大于系统内存和硬盘。实际工作时，CPU 往往需要重复读取同样的数据块，而缓存容量的增大，可以大幅度提升 CPU 内部读取数据的命中率，而不用再到内存或者硬盘上寻找，以此提高系统性能。但是由于 CPU 芯片面积和成本的因素来考虑，缓存都很小。

L1 Cache（一级缓存）是 CPU 第一层高速缓存，分为数据缓存和指令缓存。内置的 L1 高速缓存的容量和结构对 CPU 的性能影响较大，不过高速缓冲存储器均由静态 RAM 组成，结构较复杂，在 CPU 管芯面积不能太大的情况下，L1 级高速缓存的容量不可能做得太大。一般服务器 CPU 的 L1 缓存的容量通常在 32—256KB。

L2 Cache（二级缓存）是 CPU 的第二层高速缓存，分内部和外部两种芯片。内部的芯片二级缓存运行速度与主频相同，而外部的二级缓存则只有主频的一半。L2 高速缓存容量也会影响 CPU 的性能，原则是越大越好，现在家庭用 CPU 容量最大的是 4MB，而服务器和工作站上用 CPU 的 L2 高速缓存更高达 2MB—4MB，有的高达 8MB 或者 19MB。

L3 Cache（三级缓存），分为两种，早期的是外置，现在的都是内置的。而它的实际作用即是，L3 缓存的应用可以进一步降低内存延迟，同时提升大数据量计算时处理器的性能。降低内存延迟和提升大数据量计算能力对游戏都很有帮助。而在服务器领域增加 L3 缓存在性能方面仍然有显著的提升。比方具有较大 L3 缓存的配置利用物理内存会更有效，故它比较慢的磁盘 I/O 子系统可以处理更多的数据请求。具有较大 L3 缓存的处理器提供更有效的文件系统缓存行为及较短消息和处理器队列长度。

其实最早的 L3 缓存被应用在 AMD 发布的 K6-III 处理器上，当时的 L3 缓存受限于制造工艺，并没有被集成进芯片内部，而是集成在主板上。在只能够和系统总线频率同步的 L3 缓存同主内存其实差不了多少。后来使用 L3 缓存的是英特尔为服务器市场所推出的 Itanium 处理器。接着就是 P4EE 和至强 MP。Intel 还打算推出一款 9MB L3 缓存的 Itanium2 处理器，和以后 24MB L3 缓存的双核心 Itanium2 处理器。

但基本上 L3 缓存对处理器的性能提高显得不是很重要，比方配备 1MB L3 缓存的 Xeon MP 处理器却仍然不是 Opteron 的对手，由此可见前端总线的增加，要比缓存增加带来更有效的性能提升。

CPU 高速缓存的工作原理

1、读取顺序

CPU 要读取一个数据时，首先从 Cache 中查找，如果找到就立即读取并送给 CPU 处理；如果没有找到，就用相对慢的速度从内存中读取并送给 CPU 处理，同时把这个数据所在的数据块调入 Cache 中，可以使得以后对整块数据的读取都从 Cache 中进行，不必再调用内存。

正是这样的读取机制使 CPU 读取 Cache 的命中率非常高（大多数 CPU 可达 90%左右），也就是说 CPU 下一次要读取的数据 90%都在 Cache 中，只有大约 10%需要从内存读取。这大大节省了 CPU 直接读取内存的时间，也使 CPU 读取数据时基本无需等待。总的来说，CPU 读取数据的顺序是先 Cache 后内存。

2、缓存分类

前面是把 Cache 作为一个整体来考虑的，现在要分类分析了。Intel 从 Pentium 开始将 Cache 分开，通常分为一级高速缓存 L1 和二级高速缓存 L2。在以往的观念中，L1 Cache 是集成在 CPU 中的，被称为片内 Cache。在 L1 中还分数据 Cache（D-Cache）和指令 Cache（I-Cache）。它们分别用来存放数据和执行这些数据的指令，而且两个 Cache 可以同时被 CPU 访问，减少了争用 Cache 所造成的冲突，提高了处理器效能。

在 P4 处理器中使用了一种先进的一级指令 Cache——动态跟踪缓存。它直接和执行单元及动态跟踪引擎相连，通过动态跟踪引擎可以很快地找到所执行的指令，并且将指令的顺序存储在追踪缓存里，这样就减少了主执行循环的解码周期，提高了处理器的运算效率。

以前的 L2 Cache 没集成在 CPU 中，而在主板上或与 CPU 集成在同一块电路板上，因此也被称为片外 Cache。但从 PIII开始，由于工艺的提高 L2 Cache 被集成在 CPU 内核中，以相同于主频的速度工作，结束了 L2 Cache 与 CPU 大差距分频的历史，使 L2 Cache 与 L1 Cache 在性能上平等，得到更高的传输速度。L2Cache 只存储数据，因此不分数据 Cache 和指令 Cache。在 CPU 核心不变化的情况下，增加 L2 Cache 的容量能使性能提升，同一核心的 CPU 高低端之分往往也是在 L2 Cache 上做手脚，可见 L2 Cache 的重要性。现在 CPU 的 L1 Cache 与 L2 Cache 惟一区别在于读取顺序。

3、读取命中率

CPU 在 Cache 中找到有用的数据被称为命中，当 Cache 中没有 CPU 所需的数据时（这时称为未命中），CPU 才访问内存。从理论上讲，在一颗拥有 2 级 Cache 的 CPU 中，读取 L1 Cache 的命中率为 80%。也就是说 CPU 从 L1 Cache 中找到的有用数据占数据总量的 80%，剩下的 20%从 L2 Cache 读取。由于不能准确预测将要执行的数据，读取 L2 的命中率也在 80%左右（从 L2 读到有用的数据占总数据的 16%）。那么还有的数据就不得不从内存调用，但这已经是一个相当小的比例了。在一些高端领域的 CPU（像 Intel 的 Itanium）中，我们常听到 L3 Cache，它是为读取 L2 Cache 后未命中的数据设计的—种 Cache，在拥有 L3 Cache 的 CPU 中，只有约 5%的数据需要从内存中调用，这进一步提高了 CPU 的效率。

为了保证 CPU 访问时有较高的命中率，Cache 中的内容应该按一定的算法替换。一种较常用的算法是“最近最少使用算法”（LRU 算法），它是将最近一段时间内最少被访问过的行淘汰出局。因此需要为每行设置一个计数器，LRU 算法是把

命中行的计数器清零，其他各行计数器加 1。当需要替换时淘汰行计数器计数值最大的数据行出局。这是一种高效、科学的算法，其计数器清零过程可以把一些频繁调用后再不需要的数据淘汰出 Cache，提高 Cache 的利用率。 缓存技术的发展

总之，在传输速度有较大差异的设备间都可以利用 Cache 作为匹配来调节差距，或者说是这些设备的传输通道。在显示系统、硬盘和光驱，以及网络通讯中，都需要使用 Cache 技术。但 Cache 均由静态 RAM 组成，结构复杂，成本不菲，使用现有工艺在有限的面积内不可能做得很大，不过，这也正是技术前进的源动力，有需要才有进步！