

纳米级工艺下多处理器功耗评估与优化技术

张戈 张量 杨荣秋

摘要: 随着处理器设计进入纳米级工艺, 功耗不可避免地成为阻碍摩尔定律继续快速前进的主要因素之一。与此同时片上多核处理器 (Chip Multiple Processors--CMP) 已成为当今处理器设计的主流。本文主要从体系结构设计角度, 对纳米级工艺下片上多核处理器的功耗评估方法及不同构件的低功耗技术进行概括性介绍, 为目前片上多核处理器的结构设计提供参考。

关键词: 片上多核处理器; 低功耗

1 引言

功耗已经成为了目前处理器设计中第一位的限制因素, 但是处理器设计者追求高性能的脚步并不会因此而停止。目前越来越多的处理器朝着片上多核的方向发展, 而有效的功耗结构设计是决定这些处理器是否具有可扩展潜力以及好的性能功耗比的重要因素之一, 因此越来越多的设计者开始关注多核处理器的功耗评估和功耗优化方法。

片内多核处理器结构的主要设计思想是通过简化超标量结构, 将多个相对简单的超标量处理器核集成到一个芯片上, 这样可以充分利用工艺进步带来的芯片面积上数以亿计的晶体管的增加, 同时避免连线延迟的影响, 并充分开发线程级并行性 (Thread-Level Parallelism, TLP), 提高处理器吞吐量。出于上述原因, 片内多核已经势不可当地成为了当今通用处理器的主流发展方向。但是, 功耗问题在多核处理器设计中并没有得到很好的解决, 尤其在纳米级工艺条件下 (90 纳米以下), 一些新的功耗问题, 如漏电功耗等使得片上多核的功耗问题更加突出。因此多核处理器更加需要良好的功耗解决方案, 低功耗设计也成为了贯穿多核处理器各个层次设计的主要考虑因素之一。

从体系结构角度看, 多核处理器主要包括处理器核、核间互连以及片上存储体 (Cache) 三个主要部分。片上多核处理器的低功耗研究可以围绕功耗评估, 处理器核功耗优化, 片上网络功耗优化以及片上缓存功耗优化这四个方面对各部分展开, 其中功耗评估是处理器低功耗设计的基础。本文下面对这几个方面做概括性综述。

2 功耗评估

CMOS^[1] 电路产生的功耗主要分为动态功耗和静态功耗。动态功耗与电路工作时的翻转率有关, 而静态功耗主要由漏电流引起, 不管电路是否工作, 都一直存在。动态功耗与供电电压和工作频率关系密切, 静态功耗则与晶体管阈值电压关系很大。随着线宽变窄及阈值电压的降低, 静态功耗在总功耗中的比重正在不断增大。例如, 由于不能很好地控制 90 纳米级工艺条件下不断增大的漏电流, Intel 就曾多次推迟其 90 纳米芯片的开发计划。

功耗评估研究的两个焦点问题是精度和效率问题。通常来讲, 功耗评估和优化工作的抽象层次越高, 其分析精度就越差, 但其分析效率和功耗优化技术的成效就越好; 反之, 抽象层次越低, 其分析精度就越高, 但其分析效率和功耗优化技术的成效就越差。具体的功耗分析方法根据芯片的不同设计阶段可以分为以下几种: (1) 结构级的功耗分析方法, 在性能模拟器上进行功耗分析; (2) 逻辑级的功耗分析方法, 在 RTL^[2] 代码设计完成之后进行功耗分析; (3) 门级 (网表级) 的功耗分析方法, 使用逻辑综合之后的门级网表和互连线延时模

型，或者布局、布线完成之后的网表和版图寄生参数进行功耗分析；(4) 晶体管级的功耗分析方法。其中前两个阶段的功耗分析方法由于不需要任何真实电路的实际信息，因此属于较高层次的功耗分析方式，一般在逻辑综合之前进行。门级和晶体管级的功耗分析方法必须在物理设计阶段进行。

在片上多核处理器的设计中，由于处理器各个层次的设计都受到功耗因素的制约，不同构件间又存在相互影响，能在设计早期就对功耗进行完整和准确的测量，并对不同的设计方案做出评价，具有非常重要的意义。在针对单核处理器方面，应用较为广泛的结构级功耗模拟器包括 Wattch^[1]、SimplePower^[2]和 PowerTimer^[3]等，这些模拟器的基本做法是按照功能把处理器分成不同的模块，再使用结构级建模方法对每个模块赋予相应的功耗代价，并采用性能模拟器驱动方式，在该模块被访问时，计算出相应的功耗。另外，随着片上网络结构在多核处理器上的流行，针对片上网络功耗估计的高层模拟器也应运而生，如 Orion^[4]，就是针对片上互连部件如路由器（Router）等进行功耗建模的。

结构级功耗模拟的主要问题在于无法对处理器模块的实际电路特性（如晶体管尺寸、物理连线、实际工艺参数，以及电路形式等）进行准确的刻画，因此其功耗模拟误差较大。以目前应用最广泛的 Wattch 为例，它把处理器中的大部分结构归结为 RAM^[3]/CAM^[4]两类结构，这种刻画方式过于抽象，不能很好地反应真实处理器设计中的情况；其次，对于处理器中普遍存在的大量随机控制逻辑，由于其实现形式不规则，也无法通过一个简单的抽象模型进行刻画。除此之外，面对新工艺与新结构，片上多核处理器的功耗建模还存在以下几个新的挑战：

1. 纳米级工艺条件下的功耗新问题。

工艺进步使得片上多核处理器可以集成更多的晶体管，目前已经有越来越多的处理器采用 65nm 和 45nm 设计，如正在研制中的龙芯 3 号多核处理器就采用了国际上较为先进的 65nm 工艺。在这种工艺条件下，工艺偏差对性能和功耗的影响加大，芯片的电压降以及在片波动（On-Chip Variation）问题严重，片上连线的互连宽度和延迟都显著增加，同时漏电功耗的比重越来越大，在工艺特征尺寸小于 90 纳米条件下广泛采用了多阈值电压工艺，因此必须慎重评估上述各个方面对多核处理器设计的影响，结合准确的电路与工艺参数进行功耗计算；

2. 多核处理器中结构变化对功耗的影响。

多核处理器所集成的单个处理器核必须保持良好的低功耗特性，因此不再像传统单处理器那样，为了一味地追求性能而采用十分复杂的结构，转而在设计中尽量使结构简洁有效，更加倾向于采用以往的嵌入式处理器内核的结构，因此以往的高性能通用处理器的功耗评估模型会显得不太适应。另外，对多核互连部件如交叉开关（crossbar）、路由器（router）、缓冲器（buffer）等都需要进行单独的功耗建模。尤其是对于核间长距离的互连线，需要进行细致的功耗刻画；

3. 物理实现方法对功耗的影响。

对于同一种结构，其不同的物理实现方法会造成完全不同的功耗结果，如龙芯 2F 处理器核与龙芯 2E 处理器核采用了完全相同的体系结构，但通过物理设计中的一些优化手段，使得龙芯 2F 比龙芯 2E 处理器的平均功耗降低了 30%~40%，因此必须考虑不同的物理实现方法对处理器功耗的影响；另外，传统的功耗模拟器都是针对以动态电路为主实现的高性能处理器进行建模，而在多核处理器中，单处理器核为了保持良好的低功耗特性，很多设计都

开始由全定制动态电路设计转向静态电路设计,这样会使传统的基于动态电路刻画的处理器的功耗模型不再适用。

文[5]提出一种基于物理反馈的结构级功耗评估方法和平台。如图 1 所示,该功耗模拟平台包括高层模拟和底层实现两个部分,二者通过基于物理反馈的功耗模型库进行接口调用,能够灵活快速地对一个多核处理器的不同微体系结构进行功耗评估。其中底层实现部分包括对处理器各种基本块的电路建模与参数化 RTL 的描述,通过自动综合与功耗计算 EDA^[5]平台,能够对各个电路基本块的功耗数据进行自动计算,并反标至功耗模型库。高层模拟部分主要包括一个多核性能模拟器和一个多核功耗模拟器,功耗模拟器通过接收性能模拟器的微体系参数、模块翻转信息,并调用功耗模型库的功耗建模数据,完成对处理器运行程序过程中的实时功耗计算。这种基于物理反馈的功耗建模与评估流程,可以有效地利用上层结构模拟的灵活性和底层功耗建模的准确性,一定程度上解决了多核功耗模拟的上述问题。

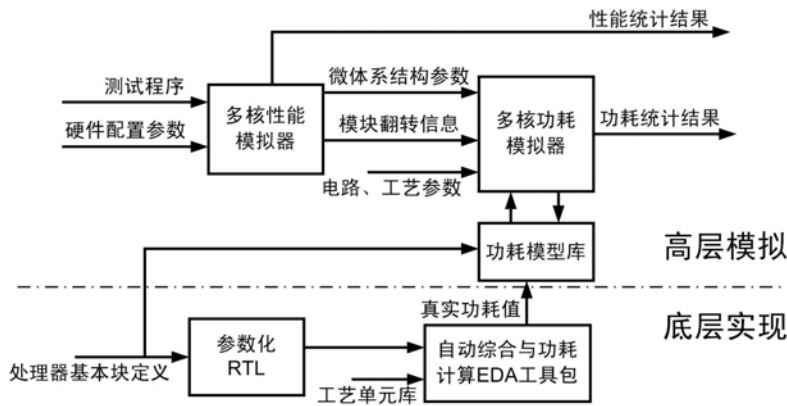


图 1 基于物理反馈的功耗评估流程

3 核内功耗

单处理器核的多数技术都可以继续应用于片上多核处理器。但与传统高性能单处理器设计中所采用的高复杂度、高功耗处理器核所不同的是,片上多核处理器通过在片上集成更多的复杂性较低的处理核,以利于通过开发线程级并行性来提升性能,降低功耗。

在片上多核处理器中,功耗管理变得更加重要。由于片上多核的每个处理器核是相对独立并通过片上网络松散耦合的,每个处理器核运行完全独立的指令流(multi-programming)或者运行通过同步操作协同运行的指令流(parallel programming)。在这种情况下,程序本身在不同时段的运行特性(指令级并行度、访存操作密度、二级缓存失效率等)不同,或者不同处理器核到达同步点的时间先后不同,各个处理器核在同一时刻的繁忙状况也不同。利用程序本身不同阶段执行特性的差别或者并行程序各线程不精确同步的特点,可以单独对各个处理器核进行变压变频控制,降低功耗。

文[6]基于栅障^[6](Barrier)机制实现高级配置与电源接口(ACPI, Advanced Configuration and Power Interface)电源管理规范^[7],提高并行程序的功耗有效性。其基本思想是当某个处理器核过早地到达栅障时,可以让其进入睡眠状态以节省无谓的功耗开销。由于进出睡眠状态的切换操作功耗开销也较大,在处理器核到达栅障之前降低其工作电压和频率,使与其他处理器核能同时到达栅障,可能会更好地节省功耗。

文[8]提出一种基于投票选择的动态电压/频率调节算法。其基本思想是通过通过对每个处理器核的 IPC^[7]、定点队列、浮点队列以及访存失效队列利用率的变化趋势的研究,按照投票策略组织这些信息来动态调节各处理器核的电压和频率,从而降低多核处理器运行时的能量消耗并提高其功耗效率。

文[9]提出使用相同指令集异构多核的方法降低功耗。由于隶属同一系列的不同代的处理器核,其复杂性、性能和功耗上的开销都有很大差别,因此根据程序的当前运行特性,动态地在这些不同复杂性的处理器核间进行切换,可以达到提高性能并降低功耗的目的。但该方法限定任意时刻只能有一个处理器核处于活跃状态,因此这种结构仅适用于串行程序。

另外,时钟网络是处理器的重要组成部分,以前的时钟网络设计基本上都是全局同步的设计,因为同步的设计容易实现而且信号在模块之间通信没有异步开销。但是由于硅片的增大和片上晶体管数量指数型增加,使得要设计出满足低扭斜和低抖动要求的全局同步的时钟网络已经越来越困难和昂贵。因此,在片上多核处理器中有必要将目光转向全局异步、局部同步(Globally Asynchronous Locally Synchronous-GALS)的设计^[10],即处理器在模块级保持时钟的同步,在模块间则可以异步。其优点是:

1. 不需要设计满足全局同步要求,约束极为苛刻的低扭斜和低抖动全局时钟网络,各模块的时钟设计变得简单,因此 GALS 的设计相对于全局时钟同步的设计更容易提高主频并降低时钟功耗,处理器的峰值性能也可以更高;
2. 各模块运行于独立的频率,可以十分方便地根据程序行为针对各模块动态地调节电压和频率,而动态电压/频率调节(Dynamic Voltage and Frequency Scaling-DVFS)是降低处理器功耗和提高功耗有效性的重要方法。

4 片上网络功耗

在纳米级工艺下的多核处理器设计中,片上互连线的数目、宽度和延迟都显著增加,信号跨越整个芯片需要多个周期,造成片上通信和存储体访问时延的不均匀性。片上网络(Networks on Chip-NoC)代替总线连接处理器核和片上缓存,具有较高的吞吐率和较好的可扩展性,已经成为片上多核处理器互连结构的事实标准^[11]。但片上网络同时也引入了功耗开销。影响片上网络功耗的因素很多,主要可以分为两个方面:微体系结构和注入流量。微体系结构主要决定单个消息在片上网络中传递的功耗,而注入流量则决定片上网络中消息的密度和总量。

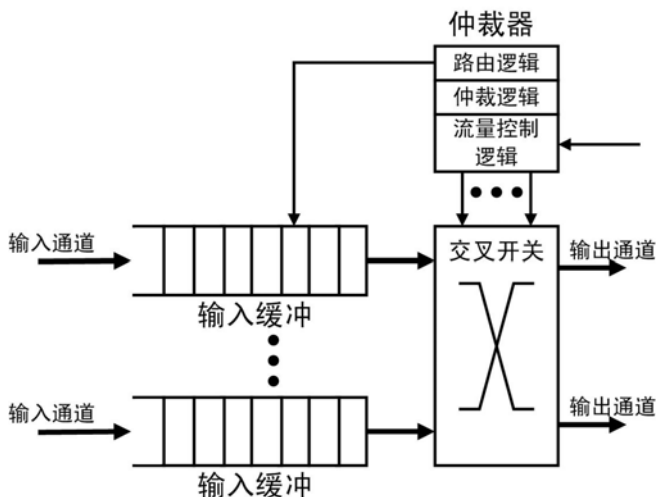


图 2 片上网络结构示意图

片上网络的结构如图 2 所示，主要包括 4 个组件：输入缓冲（Input Buffer）、仲裁器（Arbiter）、交叉开关（Crossbar）和链接（Link）。图中的仲裁器由路由电路（Routing Logic）、仲裁电路（Arbitration Logic）和流控制电路（Flow Control Logic）组成，流控制电路还接收下行网络传送过来的令牌（Credit）信息；图中的队列为输入缓冲，片上网络有多少方向的输入通道，就有多少队列项；交叉开关逻辑将输入缓冲中的数据包转发给输出通道；而链接则是连接不同路由电路之间的数据线，即输入通道和输出通道。

从功耗角度分析，仲裁器由于自身所占用的晶体管资源很少，功耗很小，几乎可以忽略不计；而其他组件都占用大量晶体管资源和金属线资源，且翻转率很高，因此功耗也是相当可观的。

片上多核处理器一般遵循共享存储编程模型，且共享片上二级缓存也已成为设计主流^[12]。每个处理器核一级缓存失效引发到共享二级缓存的请求与应答以及缓存一致性协议要求的核间数据传输和无效请求与应答形成片上网络中的主要流量；在分片式片上多处理器（tiled CMP）结构中^[13]，处理器核发出的访存请求映射到二级缓存块的方式也会导致消息在片上网络里传输的跳数及拥塞程度不同，形成不同的功耗。

可以从不同角度降低片上网络功耗，如微体系结构与互连电路的选择，缓存一致性协议的优化，以及访存地址到片上网络节点的映射方法等。文[14]针对片上网络路由电路提出了 3 个低功耗技术：分段交叉开关（Segment Crossbar）、捷径交叉开关（Cut-through Crossbar）和直写^[8]缓冲（Write-through Buffer）。

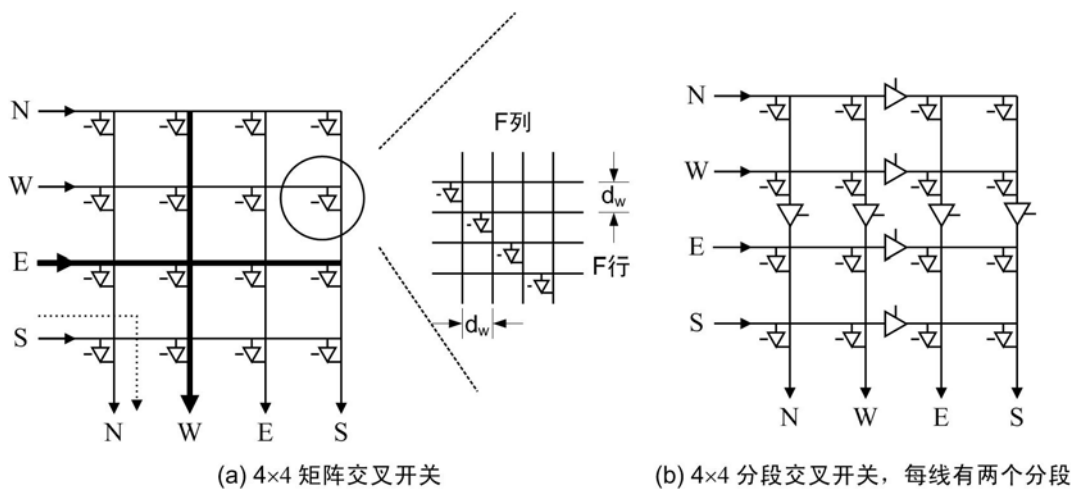


图 3 分段交叉开关

图 3 是分段交叉开关，(a) 是矩阵式 4x4 交叉开关，可见，当从 E 输入到 W 输出时，整个输入线和输出线都发生翻转，采用 (b) 中结构，通过三态门把输入/出线均匀分割，可以将发生翻转的连线分段限制在最小的范围内。

“捷径”交叉开关（如图 4 所示）技术主要利用了片上网络中消息的路由特性，比如采用 X-Y 寻径方式，消息将先沿 X 维再沿 Y 维传输，中途只发生一次方向变更。在“捷径”交叉开关中，相反方向通过互连线直连，只有发生方向变更时，才会通过交叉点的三态门。

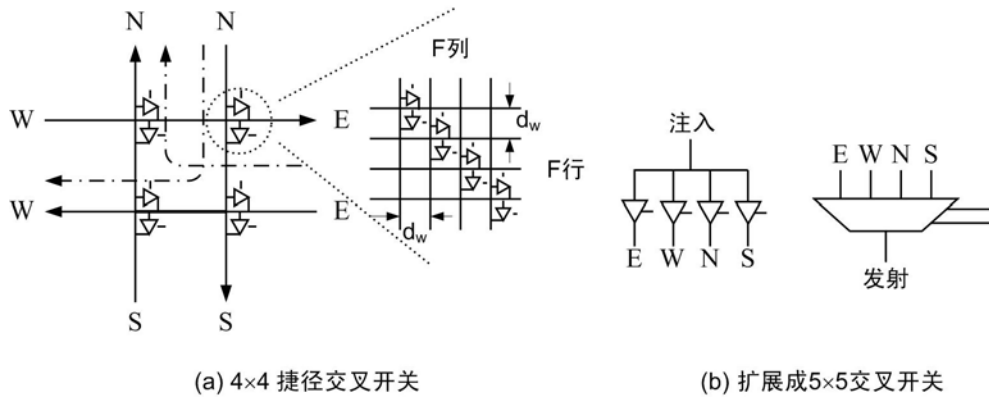


图4 捷径交叉开关

直写缓冲的主要思想是当消息进入输入缓冲时，若输入缓冲为空，则没有先到消息等待，消息可直接送到交叉开关传输，减少输入缓冲读写，从而降低功耗。

文[15]基于片上网络数据传输宽度不断增加而控制传输宽度保持不变的假设，指出控制传输将造成绝大部分传输线的无谓翻转，也会造成先进先出缓存（FIFO Buffer）读操作对非有效存储位的无谓预充电。该文提出把先进先出缓存从物理上划分为 Cbits（控制传输宽度）和（m-C） bits（m 为先进先出存储宽度）两部分，并根据消息的性质决定是否对（m-C）部分进行预充电，从而降低功耗。

文[16]将动态电压调整（dynamic voltage scaling, DVS）技术应用于片上网络的互连线，采用基于历史信息的动态电压调整策略，根据连线的占用状况调整连线的电压和频率。文[8]对于片上网络传输信号的特点进行分析，发现互连网络传输中信号为 0 的可能性大大高于信号为 1 的可能性，由此提出“零有效”的片上互连网络微体系结构及其电路实现，能够显著提高互连网络的功耗效率。

文[17]基于分片式片上多处理器和共享二级缓存，指出若一级缓存失效引起的片上网络请求由一个远程二级缓存块应答，消息的延迟和引起的功耗都将增大。该文提出受损复原（Victim Replication）技术和受损迁移（Victim Migration）技术，有选择地把一级缓存的替换块备份在本地二级缓存块中，从而减少远程访问带来的延迟和功耗开销。

文[18]提出了一种新颖的从访存请求到分散共享的二级缓存块的映射方法，传统的映射方法通过物理块地址的某几位将访存请求均匀地散列到各个二级缓存块上，完全基于硬件，实现简单，但没有考虑处理器核到应答二级缓存块在片上网络拓扑中的相对距离。

该文提出将从访存请求到二级缓存块的映射与页映射绑定的方法，在页映射转换出物理页号的同时，也转换出该物理页对应的二级缓存块号，这样，一个物理页将映射到哪个二级缓存块可以由操作系统在缺页异常处理时，灵活地根据处理器核所在位置来决定。通过合理的映射，可以使得访存请求尽可能多地由靠近处理器核的二级缓存块应答，降低片上网络的延迟和功耗。

5 片上缓存功耗

工艺的进步使得片上集成更多的晶体管成为可能，这对于多核处理器设计尤为重要。在纳米级工艺条件下，可以利用丰富的片上资源，把更多的存储单元（cache）集成到片内，显著提高存储访问的带宽。由于在片上多核处理器中，多个进程/线程同时运行需要更大的片上缓存容纳所有进程/线程的工作集，且会造成更为严重的资源冲突，极大地增加了片上缓存的访问压力，而片外存储体的访问延迟和功耗都非常大（文[19]指出，片外存储体访问

功耗可达片上缓存访问功耗的 35 倍)。因此如何更合理地在多进程/线程间分配片上缓存, 减少片外存储体访问, 也是片上多核处理器当前研究的热点。

文[12]比较了私有二级缓存和基于地址散列的分散共享二级缓存, 指出共享二级缓存可以缓解处理器核间二级缓存访问的不平衡, 而共享二级缓存又可能导致较为严重的资源冲突。该文提出了“基于处理器的共享二级缓存分组”(Shared Processor-Based Split L2) 技术, 每个处理器核都根据动态运行特性, 分配到若干二级缓存分组(L2 Cache Split), 处理器核(ID)到相应二级缓存分组的映射关系由硬件表格维护, 当在分配的二级缓存分组里访存失效时, 才去查找其他的二级缓存分组。“基于处理器的共享二级缓存分组”技术介于私有二级缓存和基于地址散列的分散共享二级缓存之间, 但无论处理器核分配到的二级缓存分组的查询还是其他二级缓存分组的查询都需要多个分组同时进行, 这会产生一定的开销。

文[20]指出, 在互相竞争的进程的缓存失效率的导数相等时, 这些进程的总缓存失效率数将达到最小, 并采用较为简单的硬件跟踪缓存失效率导数的 1 阶近似, 在不同进程间合理分配二级缓存份额, 减少高功耗的片外存储体访问操作。

文[21]指出传统的 LRU^[9]替换策略会使得局部性差的进程侵吞其他进程的共享二级缓存份额, 增加二级缓存上的冲突, 因此采用改进的 LRU 策略, 限制局部性差的进程的共享二级缓存份额, 可以减少二级缓存的失效次数。该文通过在体系结构级提供二级缓存份额寄存器, 并由操作系统根据进程实时运行信息配置各进程的二级缓存份额, 达到硬件简单和操作系统宏观控制的设计目标。

6 总结

功耗是导致片上多核处理器出现的重要诱因, 也是片上多核处理器设计的重要制约因素。对于片上多核处理器的不同设计模块和设计层次, 都存在行之有效的降低功耗的方法, 而这些方法又可能是互相牵制, 互相影响的。因此多核处理器设计需要贯穿系统结构到电路工艺的各方面的丰富知识, 才能在设计早期做出正确的多核架构的选择。

注释

【1】Complementary Metal Oxide Semiconductor, 互补金属氧化物半导体。

【2】Register Transfer Level, 暂存器转换层。硬件描述语言的一种描述层次。

【3】Random-Access Memory 随机存取存储器。

【4】Content Addressable Memory, 内容可寻址存储器。

【5】Electronic Design Automatic, 电子设计自动化。

【6】栅障(barrier)机制作为一种同步手段, 用于保持线程集合在其操作控制流的某个逻辑计算点上的协调。采用这种手段, 操作线程集内的一个线程必须等待该集合中所有其他线程都执行完某个操作后, 才能继续向下执行。栅障保证在所有的线程全部到达某逻辑执行点前, 任何一个线程都不能越过该逻辑点继续执行。

【7】Instructions Per Clock Cycle, 每时钟周期执行的指令数。

【8】也有译成“通写”缓冲, 即在 CPU 向缓存写入的同时, 也把数据写入相应内存单元, 保证缓存和内存中相应单元数据的一致性。

【9】Least Recently Used, “最近最少使用页面置换”。

参考文献:

- [1] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations, In Proceedings of the 27th Annual International Symposium on Computer Architecture, pp.83-94, June 2000
- [2] W. Ye, N. Vijaykrishnan, M. Kandemir and M. Irwin. The design and use of simplepower: a cycle-accurate energy estimation tool, Proceedings of 38th Design Automation Conference, pp.340-345, June 2000
- [3] D Brooks, P Bose, V Srinivasan, M K Gschwind, P G Emma, and M G Rosenfield. New methodology for early-stage, microarchitecture-level power-performance analysis of microprocessors, IBM Journal of Research & Development, September/November 2003, Vol. 47 No.5/6, pp.653-670
- [4] Hang-Sheng Wang; Xiping Zhu; Li-Shiuan Peh; Malik, S. Orion: a power-performance simulator for interconnection networks. 35th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-35) , Nov. 2002
- [5] 黄琨, 张戈, 王君. 基于物理反标的高性能处理器功耗建模与评估方法. 计算机辅助设计与图形学学报, 2007. 11. 19 (11) :1471-1475
- [6] J. Li, J.Martinez, and M. Huang. The thrifty barrier: Energy efficient synchronization in shared-memory multiprocessors. In Proceedings of HPCA, 2004, pp.14-23
- [7] The official ACPI specification, Revision 3.0b, October 10, 2006, <http://www.acpi.info/>
- [8] 黄琨. 片上多核处理器的功耗有效性技术研究, 博士学位论文, 中国科学院计算技术研究所, 2008年1月
- [9] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D.M. Tullsen. Single-ISA heterogeneous multi-core architectures: The potential for processor power reduction. In Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003 , pp.81-92
- [10] Anoop Iyer, Diana Marculescu. Power and Performance of Globally Asynchronous Locally Synchronous Processors. In Proceedings of ISCA, 2002 , pp.158-168
- [11] W. J. Dally, B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in Proceedings of the 38th Design Automation Conference, Las Vegas, NV, June 2001, pp.684-689
- [12] C. Liu, A. Sivasubramaniam, and M. Kandemir, Organizing the Last Line of Defense before Hitting the Memory Wall for CMPs, 10th IEEE Symposium on High-Performance Computer Architecture, Feb. 2004 , pp. 176-185
- [13] M. Zhang and K. Asanović. Victim Replication: Maximizing Capacity while Hiding Wire Delay in Tiled Chip Multiprocessors, Proc. Int'l Symp. Computer Architecture, June 2005, pp. 336-345
- [14] H-S. Wang, L-S. Peh, N. Jha, Power-driven design of router microarchitectures in on-chip networks, International Symposium On Microarchitecture (MICRO'03) . Nov. 2003 , pp. 105-116
- [15] S. Banerjee and N. Dutt, FIFO Power Optimization for On-Chip Networks, TR 03-40, December 19, 2003 , pp. 187-191
- [16] Shang L, Peh L S, Jha N K. Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks. In: Proceedings of HPCA'03, Anaheim, CA. 2003, pp. 91-102
- [17] Michael Zhang. Latency Reduction Techniques in Chip Multiprocessor Cache Systems. PhD Thesis at the Massachusetts Institute of Technology, January 2006
- [18] L. Jin, H. Lee, and S. Cho. A Flexible Data to L2 Cache Mapping Approach for Future Multicore Processors, Proc. Workshop Memory Systems Performance and Correctness (MSPC) ,Oct. 2006., pp. 92-101

- [19] T. Moreshet, R. I. Bahar, and M. Herlihy. Energy-aware microprocessor synchronization: Transactional memory vs. locks. In WMPI, 2006 , pp. 47-54
- [20] G. E. Suh, L. Rudolph, and S. Devadas. Dynamic partitioning of shared cache memory. Journal of Supercomputing, 28 (1) :7-26, 2004
- [21] Michelle J. Moravan, Jayaram Bobba, Kevin E. Moore, Luke Yen, Mark D. Hill, Ben Liblit, Michael M. Swift and David A. Wood, Supporting Nested Transactional Memory in LogTM, International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) , October 2006 , pp. 359-370

作者简介:

张 戈, 中国科学院计算技术研究所、中国科学院计算机系统结构重点实验室, 助理研究员;

张 量, 中国科学院计算技术研究所博士研究生。

杨荣秋, 中国科学院计算技术研究所硕士研究生。