

# 多核、多线程处理器的低功耗设计技术研究<sup>\*</sup>

张 骏<sup>1</sup> 樊晓桢<sup>1</sup> 刘松鹤<sup>2</sup>

(西北工业大学计算机学院 西安 710072)<sup>1</sup> (长安大学信息工程学院 西安 710064)<sup>2</sup>

**摘 要** 随着微处理器设计技术和半导体制造工艺的进步,芯片的规模和复杂度急剧增大,超高的功耗密度对系统稳定性造成很大影响,功耗壁垒已经成为提升微处理器性能的最大障碍。本文介绍了低功耗设计的基本原理、研究内容、设计方法,分析了 CMP 和 SMT 体系结构的功耗需求和特性,讨论了不同的功耗优化策略在两种体系结构下的适用程度以及对性能造成的影响。针对多核、多线程体系结构,着重从系统级、结构级和电路级等不同抽象层次对典型的功耗优化技术做了讨论。最后,展望了未来微处理器低功耗设计技术的发展趋势。

**关键词** 多核,多线程,微处理器,低功耗

## Research of Low Power Design Techniques for Multi-core and Multithreading Microprocessor

ZHANG Jun<sup>1</sup> FAN Xiao-Ya<sup>1</sup> LIU Song-He<sup>2</sup>

(College of Computer Science, Northwestern Polytechnical University, Xi 'An 710072)<sup>1</sup>

(Department of Information Engineering, Changan University, Xi' An 710064)<sup>2</sup>

**Abstract** With the development of microprocessor design techniques and semiconductor fabrication, scale and complication of a single chip increase rapidly. The ultra high power density impacts the system stability seriously. Power wall is going to become the biggest obstruction for upgrading the performance of microprocessor. This paper presents the principle, content and techniques for low power design, analyses the power demand and characteristic of CMP and SMT architecture, and discusses the applicable degree and influence to performance when using different low power design techniques for these two architectures, and also discusses several typical power optimization techniques in different abstract level. At last, the development trend of the low power design of multi-core and multithreading microprocessor in the future is prospected.

**Keywords** Multi-core, Multithreading, Microprocessor, Low power

## 1 引言

半导体工艺的迅速发展使微处理器的集成度越来越高,同时处理器表面温度也变得越来越高并呈指数增长,每 3 年处理器的功耗密度就能翻一番。功耗过高是微处理器设计中所遇到的最大问题,而目前多核、多线程的处理器体系结构更是加剧了解决功耗问题的紧迫性。Intel 和 AMD 主流的 Pentium 4 和 Opteron 处理器的功耗均已超过 100W, Pentium 4 600 系列和 Itanium 系列处理器更是达到 130W 的惊人水平。这不仅仅意味着大量的能源消耗,而且热堆积和不断增加的功耗密度将造成系统稳定性问题。过高功耗将限制处理器的性能的提升,如果要进一步提高频率或增大缓存的容量,会使处理器的功耗继续向上攀升,进而走入一个恶性循环。更糟糕的是,处理器对这些功耗的有效利用率只有 0.1%,其余功耗要么浪费在闲置的电路,要么直接转化为热量。面对多核、多线程处理器的高功耗压力,低功耗设计已经成为未来微处理器设计中的核心问题。

## 2 低功耗设计技术概述

CMOS 数字集成电路的功耗主要来自动态功耗、短路功

耗、静态功耗和漏电流功耗<sup>[1]</sup>。动态功耗是整个电路功耗的主要组成部分,但随着多核、多线程处理器集成晶体管数目的不断增加,降低短路功耗、静态功耗和漏电流功耗也成为微处理器低功耗设计中不可缺少的组成部分。集成电路的低功耗设计可以从降低节点电容、减小开关活动的次数、降低工作电压和工作频率这几个方面进行考虑。其中,功耗与 VDD 的平方和工作频率成正比,所以降低 VDD 和工作频率是降低电路功耗最有效的方法。在微处理器设计中,时钟逻辑的功耗最大,它包括时钟发生器、时钟驱动器、时钟树和时钟负载,其中时钟负载的功耗最大;数据通路的功耗位居其次,它包括复杂的执行部件、高频度使用的寄存器文件和总线,这部分电路常常需要使用动态电路和全定制设计方法。在多核多线程处理器中,片内存储器往往占据 50% 以上的芯片面积,数量众多的晶体管使它的功耗也相对较大;其它的控制逻辑和接口的功耗相对较小。

低功耗设计可以从数字系统设计的各个阶段入手,不同的阶段可以降低功耗的因素有所不同,在设计时考虑的重点也不同。在系统级设计阶段,可以改变系统的软硬件分工、在算法或结构级设计阶段改变编码方式、在 RTL 设计阶段使用门控时钟、在逻辑设计阶段改变具体门电路的逻辑安排等。

<sup>\*</sup> 国家自然科学基金项目(60573107)、国家自然科学基金项目(60573143)资助。张 骏 博士研究生,主要研究方向:专用集成电路、计算机系统结构;樊晓桢 教授,博士生导师,主要研究方向:专用集成电路、计算机系统结构;刘松鹤 硕士研究生,主要研究方向:计算机软件、计算机网络。

这些方法最终都可以降低电路的平均翻转率,不同措施的效果也不同。通过软硬件分工有可能降低电路 30% 的翻转次数,而通过逻辑的重新安排却只能降低 5% 的翻转次数。根据介入系统设计阶段的不同,数字系统的低功耗设计方法可以分成若干层次。

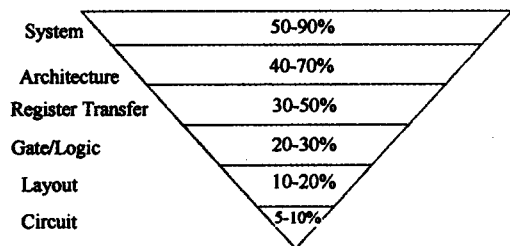


图 1 不同抽象层次功耗优化的概率

图 1 是不同抽象层次功耗优化的概率。按照抽象层次的不同,可以分为:系统级、结构级、寄存器传输级、逻辑门级、版图级和电路级。每个级别可以达到的低功耗设计效果也完全不同,抽象层次越高表明在数字系统的设计中越早的进行低功耗设计,因此在较高抽象层次进行低功耗设计效果较为明显<sup>[2]</sup>。

系统级低功耗设计主要考虑算法、软硬件分工、动态电压管理等方面,它研究的重点是数字系统如何操作和控制芯片工作,从而达到降低功耗的目的。在进行任务调度时,操作系统可以根据软件负载情况以及处理器功耗负载,动态地启动或关闭一定数量的处理器内核,进而降低整体系统的功耗。比如说,当操作系统监测到某个处理器核心温度过高,则可以将正在该核心上运行的任务迁移到其他处理器核心上执行,并暂时关闭该处理器核心。结构级主要考虑调度、硬件资源分配、操作数交换等方面。结构级低功耗设计可以应用动态电源管理、热能分布、热敏感的核间通信路径动态调度等策略来避免集中通信引起的芯片局部过热等问题。有些分类方法在系统级和结构级之间还加入了一层算法级,它主要考虑硬件的算法以及调度问题,从这一级开始的方法大多数都可以应用在芯片的设计上。寄存器传输级常采用的低功耗设计方法有:门控时钟、操作数隔离、动态操作数交换、有限状态机编码以及总线编码等。逻辑门级低功耗设计的主要手段有:工艺映射、改变晶体管尺寸缓冲器插入、调整相位、管脚置换、因式化简等。在版图级,设计者可以从以下方面入手降低电路功耗:互连线电容的降低、时钟树的综合等。电路级是可以进行低功耗考虑的最低层次,也就是在具体电路实现采用某些措施来降低电路的功耗,在这个层次主要可以考虑:逻辑类型的选择、优化工艺降低电容、电压缩小等方面。这里的电压缩小是指在具体电路实现时如何通过降低电路的供电电压来达到降低功耗的目的,是系统级的动态电压管理的具体实现。目前,多核、多线程处理器的低功耗设计技术主要是在系统级、结构级和电路级上进行的。

### 3 多核、多线程处理器的功耗特性分析

片上多处理器(CMP)和同时多线程(SMT)是多核、多线程处理器微体系结构成功典范。SMT 通过从多个线程中取指令并在同一条流水线中执行来提高性能,从而在较小的面积代价下来达到较高的吞吐量。有文<sup>[3,4]</sup>指出,SMT 结构能有效地节省功耗,但是由于 SMT 体系结构对面积的增加相对较少,关键资源的竞争更为激烈,在每个时钟周期发射大

量的指令将会增加功耗的浪费和功耗的密度。CMP 则在同一块芯片上实例化了多个处理器内核。每个内核都有它自己私有的分支预测器和一级 CACHE,但是多个内核共享二级 CACHE。对于多线程或者多道程序的负载,CMP 可以通过多个内核执行并共享二级 CACHE 来增加吞吐量。在给定的芯片面积下,单内核的 SMT 处理器相对于多内核的 CMP 处理器来说能支持更大的二级 CACHE 容量。有文<sup>[5,6,8]</sup>指出 CMP 处理器由于在多个线程执行时很少存在线程间对执行资源的竞争而拥有比 SMT 处理器更大的吞吐量。但是另一个负面影响是 CMP 芯片上每一个额外的处理器将会增加功耗。

对于多线程和多道程序的负载,CMP 显然对性能提高有很大的帮助。如果对二级 CACHE 的竞争不是问题的话,性能的提高基本上和内核的数量是一个线性关系。另外一些研究者将 SMT 和 CMP 做了比较,这些研究者都发现 CMP 在有足够的无关线程在每个内核上运行时能提供较好的性能提升,即使在 CMP 的内核相对于 SMT 内核较简单的情况下也是如此,因为假设有足够的线程级并行性来充分的利用 CMP 的多个内核。也有研究者对 SMT 和 CMP 的功耗行为做了研究,并寻找了一系列方法来使用冗余的资源来对单一线程的计算进行转移从而控制功耗密度的分散和转移,比如多个处理器内核。文<sup>[7]</sup>对 SMT 和 CMP 进行对比,发现 SMT 相对于 CMP 来说会产生较大的功耗。但是分析是建立在 CMP 处理器的每个内核都相对于单线程的 SMT 处理器内核有较低的复杂度和较低的带宽的基础上的。

CMP 和 SMT 在现有的工艺条件下有相似的功耗特性,但是两者的消耗能量的行为是不同的。SMT 主要是因为一些关键的局部部件,比如寄存器文件被充分地竞争使用而增大了能量的消耗。CMP 的功耗增大主要是因为实例化了多个处理器内核。在未来的工艺条件下,渗漏功耗将占整个芯片功耗的很大比例,所以 CMP 比 SMT 将有更大的功耗。因为对于 SMT 体系结构,很高的资源利用率隐藏了渗漏功耗所造成的影响。对于 CMP 处理器来说,将较大的二级 CACHE 换成了另一个处理器内核增加了额外的渗漏功耗。CMP 和 SMT 在采取了适当的动态线程分派和转移策略时能达到较好的性能。总的来说,在面向性能的系统,局部的动态线程分派和转移策略对于 SMT 结构更加的有效,全局的动态线程分派和转移策略对 CMP 结构更加的有效。对于面向功耗的系统中,全局的动态电压缩放功耗管理策略能节省更多的功耗。然而,局部的动态线程分派和转移策略对性能所带来的贡献使得这些策略技术对于面向功耗的 SMT 系统来说更具竞争力。

### 4 多核、多线程结构的低功耗设计方法

在多核、多线程处理器中,单个晶体管的计算能力大为提高,资源竞争更为激烈,功耗和热量集中在一些高频率活动的处理器资源上,比如 ALU 和寄存器文件。这些局部的部件很快就达到了所能承受的温度极限,甚至导致电路的功能错误,极大的影响了电路的可靠性。随着线宽不断地减小,电流的不断增大,电路工作速度的大幅增加,功耗密度不断提升。通过降低工作电压和封装散热的方法已经不能有效地去散热量。目前,主要有两类方法可以应用在多核、多线程处理器的低功耗设计中:一类是时间上的,一类是空间上的。时间上的解决方法是通过降低工作频率和工作电压、在一段时间内停

止处理器整体或者部分功能部件工作等方式来散发掉已经产生的热量,然后再全速工作,这种方法是以性能为代价的,比如电压和频率的动态调整技术和关键资源的动态功耗管理技术。空间上的解决办法是将发热量较高的资源上所正在进行的运算转移到一个可替换的运算单元上去,从而使得功耗密度能较均匀地分布在整个处理器上,这种策略要求有冗余的资源等待替换,但是性能上几乎没有损失,比如动态线程分派与转移技术。另外,一些新的体系结构和电路级动态管理技术也对降低电路功耗起到重要作用,比如资源共享的低功耗多核技术和泄漏电流和切换功耗的动态管理技术。

#### 4.1 动态线程分派与转移技术

随着工艺技术不断的进步,当今高性能处理器的工作电流,时钟频率大幅提高,功耗密度也急剧增大,所产生热量的速度已经超过了散热的速度。功耗密度的不断增大有可能导致某些部件的温度过高而发生错误。多核、多线程处理器在增加了吞吐量的同时也由于容纳了多个处理器内核和较高的资源利用率而增加了芯片所产生的热量,但是 SMT 的多线程协同调度能力和 CMP 与生俱来的多内核处理能力也为低功耗设计提供了新的手段。动态线程分派与转移技术可以根据当前的多个内核的功耗密度情况动态的进行线程的分配,并可以在一个处理器内核功耗密度过大的情况下将这个内核上的负载转移到另外相对闲置的内核上去,从而在保证吞吐量不会降低的前提下使功耗密度较大的内核减小负载,进而冷却下来。

动态线程分派策略通过协同调度有不同资源需求的线程,使之在 CMP 的多个 SMT 内核上运行,从而更合理地利用一个 SMT 内核上的不同类型资源,避免了由于多个线程竞争使用同一类型资源而导致的单一资源过热现象,并使产生的总体功耗合理均匀的分布在所有可用内核上,而不至于某个区域发热量过高,从而增加了每个内核的关键资源的冷却时间,进而加快了整个芯片的冷却速度。动态线程分派策略主要包括以下四点关键技术:

第一,协同调度整数和浮点线程,这将会更加高效利用指令发射队列,寄存器文件和执行资源。但是这还不够,因为很有可能没有足够的整数或者浮点资源用来协同调度。另外,将 IPC 很高的整数和浮点线程进行协同调度会增大共享资源的压力,比如说数据 CACHE。

第二,将有高 IPC 的线程和有低 IPC 的线程协同调度。低 IPC 的线程大部分是由于 CACHE 缺失造成的,这些线程会给数据 CACHE 造成很大的压力,但是不会给执行资源造成太大压力。将这两类线程协同调度能在发热率和高 IPC 线程单独运行时接近的前提下保证较高的吞吐量。

第三,多个内核具有差异较大的 IPC 将导致某些内核的空闲周期相对较少,温度上升过快。动态线程分派策略能使整个 CMP 处理器多个内核的 IPC 尽量平衡,从而避免了某个内核比其他内核更快地达到温度极限。

第四,根据程序对具体资源的需求情况进行协同调度。协同调度的程序不应该对同一种资源同时产生较大压力。这种策略在将两个高 IPC 或者两个低 IPC 的资源要被协同调度的时候很有必要,仅仅根据定/浮点来对程序进行划分是不充分的。比如说,一个高 IPC 的程序对数据 CACHE 和乘法器的使用很频繁,另一个高 IPC 的程序对 ALU 的使用很频繁。协同调度这两个程序是很有效的。

动态线程转移策略挖掘了由多个 SMT 内核构成的 CMP

处理器空间上的性能提升的可能性。通过将已经过热内核上运行的线程转移到另外温度相对较低但有空闲资源的内核上运行,过热内核能缓解由于某些关键资源竞争过于激烈所造成的升温过快。以多个处理器内核作为粒度是 CMP 处理器与生俱来的能力,线程在内核间进行转移无需停止内核的工作,这对性能几乎没有影响。而以内核中的关键资源作为粒度来动态控制热均衡需要暂停内核的运行,这是以降低性能为代价的。将线程转移到距离过热内核较远的内核上执行不会在温度上受到影响。

这种动态线程转移策略通常由操作系统完成,并且尽量要使得整个转移过程的时间相对于转移目标内核的处理时间最小,从而最小化性能的损失。由于线程的转移需要转移时间和操作系统的介入,所以线程转移的频度越小越好。温度的侦测由计数器和硬件温度传感器来完成,这些部件将指出处理器是否过热并且需要冷却。硬件温度传感器被放置在一些关键的流水线部件和功能部件上,比如寄存器文件和 ALU。当硬件温度传感器侦测到极限温度时就会在几个微秒内发生一个自陷异常。然后由操作系统使用动态线程转移策略来决定被转移线程的目标内核,并且将寄存器状态和程序计数器复制到目标内核的一个空闲上下文,从而完成一次线程的转移过程。

#### 4.2 电压和频率的动态调整技术

对处理器性能提升的不断要求直接造成了处理器内核中晶体管数量和工作频率的急速提升。在芯片的速度和面积变大的同时,设计者们面临着全局时钟分布和散热问题的考验。单一时钟树所造成的时钟偏斜和驱动能力的下降将给物理实现造成难题,而多时钟域的设计正是解决这一问题的方案。多时钟域所具备的在不同时钟域中独立调节电压和频率的能力将为低功耗设计提供有力手段。

多时钟域体系结构使用了一个全局异步但是局部同步的时钟方案,从而允许在未来更苛刻的高时钟频率下保持同步设计的方法。在多时钟域方案中,每个处理器内核为一个时钟域,其内部是同步的,但是多个处理器内核之间是异步的。设计者在内核设计时仍然应用同步设计策略,但是在全局的时钟偏斜将不再是设计的约束条件。除此之外,每个时钟域可以有独立的工作电压和频率控制逻辑,从而允许在处理器内核,也就是时钟域这一级来动态地调节电压和频率。全局的动态电压调节已经出现在许多系统中,从而可以在一些空闲的状态下减少功耗的损失。多时钟域结构能在多道程序在多个内核上运行的同时降低非关键的内核的工作速度,从而降低功耗。由此所带来的问题是需要多个不同时钟域间进行同步,这将会使性能有不到 2% 的损失。如果不同时钟域上的频率和电压在合理的时间段内有合理的值,多时钟域的体系结构能以极小的性能损失为代价实现较大的功耗优势。设计者完全可以通过瞬时信息用硬件来实现每个时钟域中的控制逻辑和多个时钟域之间的同步逻辑,从而使得对于使用者和系统软件的完全透明。实验证明,在不同的控制策略下,性能平均下降 7%,功耗有 17%~27% 的降低。

#### 4.3 基于资源共享的低功耗多核技术

基于资源共享的低功耗多核技术的设计思想出发点在于对当前两种微处理设计方法的思考。CMP 采用复制多份单处理器的方法来得到性能提高,是一种资源重复的设计思路。而 SMT 则是采用相对复杂的机制,采用多个现场来更加有效地利用当前的系统资源。它们两者在资源共享方面走向两

个极端,可能都不是最有效的设计方法。可以考虑一种介于两者之间的体系结构:复制多个内核。但是在拓扑上相邻的内核共享某些部件,这样既能避免重复的功能部件利用率不高而白白浪费功耗,又不会明显的降低处理器的性能。如果仅从性能方面考虑,这种方案是不可行的。因为共享了某些部件后,一定是要在该部件未被其它核占用时才可用。但是如果某些部件本身利用率就不高,或者是在这些部件的性能相对降低的情况下功耗或者面积可以有极大地下降,那么对这种部件的共享就是有意义的。对共享后的面积、功耗、连线资源、通信开销和拓扑结构可行性等几个因素进行分析,最终发现四个候选的可共享部件:FPU,一级数据 CACHE,一级指令 CACHE 以及交叉开关端口。在对这些共享部件的分配使用上可以采用多种不同分享机制,最简单的机制就是时间片轮转的方法。此外还有一些更加智能的机制,比如使用查询方式等。

该种设计方法最大优点就是每个核的功耗和面积可以极大地减少,而仅带来很小的性能下降。这样,单位面积上的计算能力和功耗利用率都有很大提升。片上面积也有所下降,在片上面积固定的情况下,可以集成更多的核。由于相同的计算能力所需的晶体管数目减少,漏电功耗也相对下降。

#### 4.4 基于关键资源的动态功耗管理技术

通过多种策略限制多核多线程处理器每个内核可以使用资源的数量,可以使某个处理器内核的运行慢下来,从而消耗较少的功耗。第一种技术是指令选取的阻塞技术。这种技术限制了取指令阶段处理器的频度,从而减少了引起流水线高频率活动的因素进而达到低功耗的目的。第二种技术是重命名阻塞技术。这种技术限制了每个时钟周期重命名的指令的数目。这是根据先前的采样区间内发现的某个寄存器文件使用的频度来决定的,浮点寄存器文件和定点寄存器文件的重命名都是可以阻塞的。这种策略可以限制某一个线程被分配到一些已经过热使用的寄存器文件资源从而降低功耗,并且相对于指令选取阻塞技术来说更具有局部性。但是这种策略的阻塞达到足够严重的时候,有可能会影响到引起阻塞的线程的正常运行。第三种技术是基于寄存器文件占有率的阻塞策略。我们发现寄存器文件是整个芯片上功耗最大的部件,它的功耗和它的占用率是成比例的。因此,一个降低功耗的方法就是限制寄存器文件可以使用的数量。为了分散寄存器文件的功耗密度,有研究者提出使用交替开关寄存器文件来使整个功耗能均匀地分布在整个寄存器文件中。在这三种策略中,指令选取阻塞策略通过在前端进行控制从而在整个芯片的级别上对全局有较大的影响。基于寄存器文件占有率的阻塞策略目标是直接控制功耗大的部件,比如整数和浮点的寄存器文件,所以这种策略的影响是很局部的。虽然这种策略有较少的性能损失,但是对功耗的降低也是很有限制的。重命名阻塞策略则显得更加的局部化。

多核、多线程处理器的高速缓存往往占据 50% 甚至更大的芯片面积,为了达到提高性能的目的,处理器的高速缓存被做得越来越大。而高速缓存所采用的高速六管 SRAM 存储密度极低,2MB 高速缓存所需的晶体管总数高达 1 亿。因此,高速缓存是处理器中耗能较大的功能部件,有效降低高速缓存的功耗将使处理器的整体功耗大幅降低。目前,常用两种方法来降低高速缓存的功耗:一种是直接削减高速缓存容量,这显然是以牺牲性能为代价的。另一种方法将  $n$  路组相连的高速缓存中的每一路又分为  $m$  个象限,这样整个缓存就

被分为  $[n \times m]$  个象限区域,工作时并非所有的缓存都处于启动状态,处理器自身可以动态地进行逻辑选择,每次操作只激活一个象限,其余的  $[n \times m] - 1$  个象限区域则被系统关闭,处于暂时睡眠状态。这样一来,高速缓存的实际能耗只有常规做法的  $1/[n \times m]$ ,在保持高性能的条件下将功耗大幅度降低。

#### 4.5 渗漏电流和切换功耗的动态管理技术

众所周知,半导体晶体管一直存在电流泄漏现象,即便是在关闭状态下,电流泄漏依然发生,由此产生的功耗浪费越来越引起研究人员的关注。随着半导体制造工艺进步,晶体管线宽变得越来越小,各逻辑门间的障壁也越来越薄,漏电流现象变得越发难以控制。减少渗漏电流的最直接方法是尽量少地使用低电压阈值的晶体管,在阵列逻辑和一些关键部件中尽量使用高电压阈值的晶体管。多核多线程处理器通过在片上集成多个处理器内核来提高性能,这直接导致了晶体管总数的激增,并且关键的共享部件竞争更为激烈。随着工作电压的不断降低,高电压阈值晶体管的使用将在减少多核、多线程处理器的渗漏电流方面起到更加重要的作用。另一种方法通过软硬件协同设计来减少渗漏电流。硬件部分为芯片内的一组电路逻辑,它的作用是对芯片上不同的区域进行监控,根据使用频度、区域负载状况和热分布状况对本区域晶体管的阈值电压动态的进行调整,软件部分则在更高层次结合线程和任务的调度状况对这些监控逻辑动态的进行控制。研究表明,这种动态的渗漏电流动态管理技术可以使处理器在休眠状态的功耗降低到原先的  $1/70$ ,即使在 65nm 或者 45nm 工艺条件下,该方法也十分有效。

除了渗漏电流外,切换功耗由于芯片面积的不断增大和持续提高的指令执行速度也在持续增加。为了减少切换功耗,可以考虑在设计中使用细粒度的动态的门控时钟策略。细粒度的动态门控时钟策略允许一个局部的时钟缓冲器在下一个时钟周期不会被使用到的情况下被关闭,这将在不影响性能的前提下充分的节省功耗。在每个时钟周期内,动态的门控时钟策略决定在下一个时钟周期内每一个局部的时钟缓冲器是否应该被门控。当一个局部的时钟缓冲器被门控以后,它所驱动的寄存器还是一样能被读取,但是不能被写入。与此同时,要保证不能由于门控时钟而造成性能上的影响,动态的门控时钟逻辑不能成为关键路径,所以相关逻辑必须保证简单和足够小。

## 5 未来的发展趋势

从设计角度来看,在未来的多核多线程微处理器设计中首先应该最早的对性能、面积和功耗作合理的折衷。把低功耗和热优化设计方法和设计目标约束有机的结合起来,使低功耗和热优化设计的理念和方法贯穿整个设计,从而在不同的设计层次上进行优化。其次,增强对功耗的跟踪和管理。面向功耗的设计中必须能够实时地捕获到所有器件的功耗信息并能够对其进行管理,这些信息是所有动态低功耗优化策略的依据,其实时性和准确性直接影响到处理器的整体性能和对功耗优化的效果。再次,合理地利用低功耗和热优化技术。在通过降低工作电压和工作频率来降低功耗的同时,应该考虑到其对性能的影响。尽量使用一些合理的结构和管理策略来折衷和弥补功耗优化对性能所造成的影响。最后,加强低功耗器件单元库和新的低功耗工作模式的开发。随着多核多线程处理器设计的不断优化,单个晶体管的计算能力和

功耗利用率显著增加,新型的低功耗器件将在减少渗漏功耗和切换功耗方面发挥更重要的作用。另外,多核、多线程体系结构需要新的有针对性的低功耗设计方法,但这些新的体系结构同时也为研究和开发新的低功耗工作模式提供了更大的空间。

**结论** 随着微处理器设计技术和半导体工艺的发展和进步,低功耗设计已经成为深亚微米设计领域中的严峻考验。本文针对多核、多线程处理器介绍了低功耗设计技术的基本原理、研究内容、设计方法。分析了 CMP 和 SMT 体系结构的功耗需求和特性,讨论了不同的功耗优化策略在两种体系结构下的适用程度以及对性能造成的影响。着重从不同抽象层次对动态线程分派和转移、电压和频率的动态调整、共享关键资源的多核体系结构等典型的功耗优化技术做了讨论,并展望了未来微处理器低功耗设计的发展趋势。

(上接第 300 页)

制、幅度调制等)要求而设计的,这一点也是利用了 FPGA 的灵活性。其设计框图如图 5。

### 3.2.3 PDSP 控制模块设计

PDSP 是为查找表而设计,这一点也是利用了 FPGA 的灵活性。其设计框图如图 6。

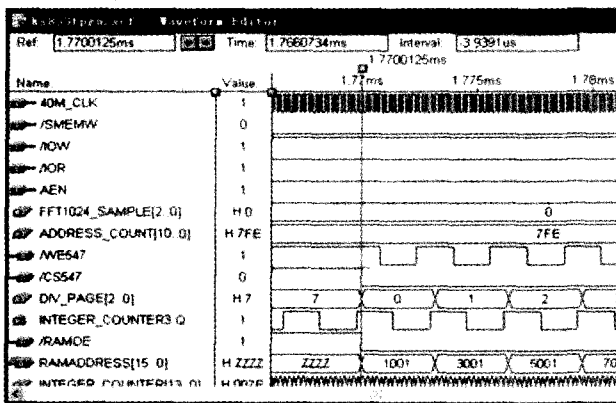


图 4 PDSP 实现过程

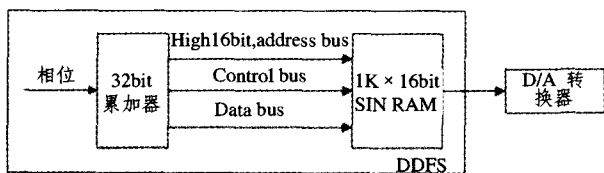


图 5 DDFS 设计框图

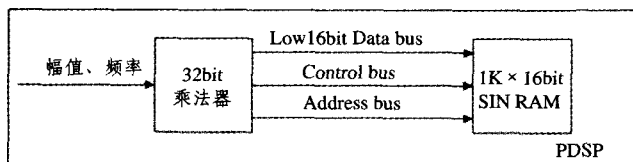


图 6 PDSP 设计框图

### 3.2.4 测试结果对比

本设计与传统设计测试结果对比如表 1,表 2,表 3 所示,分别对幅值精度,相位精度和频率精度进行比较。从表中可以看出采用本设计方案的精度对比传统设计方案有较大的提高,设计完全达到预期效果。

## 参考文献

- 1 梁宇,韩奇,魏同力. 低功耗数字系统设计方法. 东南大学学报自然科学版, 2000,09,30
- 2 Yano K, et al. A 3, 8ns CMOS 16x16 Multiplier Using Complementary Pass Transistor Logic. IEEE Journal of Solid State Circuits, April 1990, 388~395
- 3 Li Y, Brooks D, Hu Z, et al. Understanding the energy efficiency of simultaneous multithreading. In: Proc ISLPED'04, Aug, 2004
- 4 Seng J, Tullsen D, Cai G. Power sensitive multithreaded architecture. In: Proc. ICCD 2000, 2000, 199~208
- 5 Burns J, Gaudiot J L. Area and system clock effects on SMT/ CMP processors. In: Proc. PACT Sep, 2001, 211~18
- 6 Hammand L, Nayfeh B A, Olukotun K. A single-chip multiprocessor. IEEE Computer, 1997, 30(9):79~85
- 7 Donald J, Martonosi M. Temperature aware design issues for smt and cmp architectures. In: Proceedings of the 2004 Workshop on Complexity-Effective Design, June 2004
- 8 Sasanka R, Adve S V, Chen Y K, et al. Debes. The energy efficiency of cmp vs. smt for multimedia workloads. In: Proc. 18th ICS, June 2004

表 1 幅值信号改进方案和传统方案测试结果对比

| 幅值(V) | 改进方案(V) | 误差(%) | 传统方案(V) | 误差(%) |
|-------|---------|-------|---------|-------|
| 0.1   | 0.10002 | 0.02  | 0.10050 | 0.50  |
| 0.5   | 0.50008 | 0.01  | 0.50200 | 0.40  |
| 1     | 1.00010 | 0.01  | 1.00350 | 0.35  |
| 2     | 2.00025 | 0.01  | 2.00640 | 0.32  |
| 3     | 3.00036 | 0.01  | 3.00991 | 0.33  |
| 5     | 5.00700 | 0.01  | 5.01503 | 0.30  |

表 2 相位信号改进方案和传统方案测试结果对比

| 幅值(°) | 改进方案(°)  | 误差(°)  | 传统方案(°)  | 误差(°) |
|-------|----------|--------|----------|-------|
| 30    | 30.0016  | 0.001  | 30.015   | 0.015 |
| 90    | 89.9993  | -0.001 | 90.0027  | 0.002 |
| 120   | 120.0008 | 0.001  | 119.9978 | 0.002 |
| 240   | 240.0005 | 0.001  | 240.0033 | 0.003 |
| 270   | 270.0007 | 0.001  | 270.0022 | 0.002 |

表 3 频率信号改进方案和传统方案测试结果对比

| 频率(Hz) | 改进方案(Hz)  | 误差(Hz) | 传统方案(Hz)  | 误差(Hz) |
|--------|-----------|--------|-----------|--------|
| 1      | 1.00019   | 0.0002 | 0.89154   | 0.1085 |
| 10     | 10.00011  | 0.0001 | 9.76552   | 0.2354 |
| 50     | 50.00008  | 0.0001 | 50.39732  | 0.6027 |
| 100    | 100.00012 | 0.0001 | 100.25752 | 0.7425 |
| 400    | 400.00015 | 0.0001 | 400.46903 | 0.4310 |
| 700    | 700.00016 | 0.0001 | 700.67723 | 0.3228 |

**结论** 文章介绍一种基于高集成化的直接数字频率合成技术的程控信号发生器的设计方案,用于模拟微硬盘读写通道的伺服信号,测试结果证明,信号的质量比传统的设计方案有较大的提高,该方案的有效和重要性将逐渐被业界所认识。

## 参考文献

- 1 Best E. Phase-Locked Loops, 2003
- 2 Ren XiaoDong, Wen Bo. CPLD/FPGA Advanced Application Design Guide, 2003
- 3 Zhu ZhengYong, Weng MuYun. FPGA DESIGNED AND APPLICATION, 2002
- 4 Zhang YouZheng, Cheng ShangQing, Zhou ZhengZhong. Frequency Synthesis Technology, 1984
- 5 Wang ChunMei, Shun HongBo, Ren ZhiGang. TMS320C5000DSP System Design and Research Example, 2004
- 6 Wang LianXu. DSP BASE and Application System Design, 2002