

文章编号:1007-130X(2011)12-0027-10

# 异构多核处理器体系结构设计研究<sup>\*</sup>

## Research of the Heterogeneous Multi-Core Processor Architecture Design

陈芳园,张冬松,王志英

CHEN Fang-yuan, ZHANG Dong-song, WANG Zhi-ying

(国防科学技术大学计算机学院,湖南 长沙 410073)

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

**摘要:**多核技术成为当今处理器发展的重要方向,异构多核处理器由于可将不同类型的计算任务分配到不同类型的处理器核上并行处理,从而为不同需求的应用提供更加灵活、高效的处理机制而成为当今研究的热点。本文从体系结构的角度探讨了异构多核处理器设计中的关键点,从内核结构、互连方式、存储系统、操作系统支持、测试与验证、动态电压调节等方面分析了异构多核处理器对体系结构设计带来的挑战。最后本文针对高性能应用和嵌入式实时应用分析了异构多核在这两种应用中的设计关键点,指出了高性能异构多核在性能提升、内核数量以及嵌入式异构多核在实时性、低能耗需求等方面的设计难点和研究方向。

**Abstract:** Nowadays processors advance towards multi-core architecture. In heterogeneous multi-core processors, different kinds of computing tasks can be assigned to different processor kernels in order to perform parallel execution, which can provide more flexible and more effective principles. Nowadays the heterogeneous multi-core architecture becomes the research hot. This paper introduces the design of heterogeneous multi-core processors from architecture, and analyzes the challenges in terms of some important aspects: architecture of cores, interconnect, storage system, operating system support, test and validation, DVFS and so on. The challenges for the design of heterogeneous multi-core processors are discussed. Finally we propose the design key points for high performance and embedded real-time applications: the performance and the number of cores for high performance heterogeneous multi-cores, real-time and lower power for embedded real-time multi-cores.

**关键词:**多核处理器;异构多核处理器;体系结构;嵌入式实时系统

**Key words:** multi-core processor; heterogeneous multi-core processor; architecture; embedded real-time system

**doi:**10.3969/j.issn.1007-130X.2011.12.006

中图分类号:TP332

文献标识码:A

<sup>\*</sup> 收稿日期:2010-05-12;修订日期:2010-08-30  
基金项目:国家自然科学基金资助项目(60773024,60873015,60903039)  
通讯地址:410073 湖南省长沙市国防科学技术大学计算机学院博士生队  
Address: Doctoral Brigade, School of Computer Science, National University of Defense Technology, Changsha, Hunan 410073, P. R. China

# 1 引言

随着半导体工艺的进步,以及单处理器所碰到的物理极限和功耗等无法逾越的障碍,体系结构技术发生了深刻的变革<sup>[1]</sup>。经过最近几年的不断研究与发展,以多核处理器为代表的先进体系结构已经逐渐取代单核处理器成为提高处理器性能的主要途径。多核处理器在一个芯片中集成多个处理器核,这些处理器核具有相同或者不同的功能和结构,以一种有效的方式集成在同一个芯片中,并以有效的划分方式将应用程序分配给不同的微处理器核进行并行处理,从而提高微处理器系统的性能。多核体系结构强调结构的层次化、功能的模块化与分布化,它让每个功能部件都尽量简单,部件内部尽可能保持连线的局部性。较之单核处理器,多核处理器具有很多天然的优势。多核处理器可以在较低的时钟频率下提供单核处理器需要很高时钟频率才能达到的计算性能,而较低的时钟频率也可以很好地满足功耗、散热等方面的限制。此外,多核处理器的本质是采用相对较简单的多个计算内核并行工作,以提供较高的计算能力。因此,设计者在设计多核处理器时,只需设计相对较简单的计算内核,并通过一定的互连方式将其连接起来。与传统的设计单个内核的复杂处理器相比,这大大降低了设计的难度和成本,提高了设计的效率。正是由于多核的这些优势使它一出世就成为业界和学术界关注的焦点。现在几乎所有的微处理器厂商都投身到多核处理器的研制队伍中。

从所包含的处理器核结构的角度来看,多核处理器分为同构多核处理器和异构多核处理器。同构与异构是多核处理器主要的两种结构形态。同构多核处理器中处理器芯片内部的所有内核结构完全相同,各个内核具有等同的地位。异构多核处理器中异构多核处理器芯片内部采用多种功能不同的内核,一般是由负责管理调度的主核和负责计算的从核构成,或者由承担定点、浮点、特殊计算等不同计算功能的多种内核组成。从应用的角度来说,目前的同构多核处理器大多数由通用的处理器核组成,每个处理器核可以独立地执行任务,与通用单核处理器结构相近。异构多核处理器通常同时集成通用处理器、DSP、媒体处理器、网络处理器等多种类型的处理器内核,各个内核针对不同的需求,从而提高应用的计算性能。其中,通用处理器核常作为主核控制处理器以进行通用计算,而其他

处理器核用于加速特定的应用。

异构多核处理器具有多个不同结构的处理器核,在同一块芯片上集成了两种甚至两种以上的计算内核,例如 SONY 的 CELL 芯片。这类处理器通常具有很强的针对性,多是针对某一特定的应用设计而成的,所选用的处理器内核一般包括一个或多个通用处理器内核和多个特定高性能处理内核。在异构多核处理器中,可将不同类型的计算任务分配到不同类型的处理器核上并行处理,从而为不同需求的应用提供更加灵活、高效的处理机制。异构多核处理器在晶体管的设置上可以进行针对性的优化,相对于同构多核处理器来说可以采用较少的晶体管获得等高的性能。

异构多核处理器的每个内核都具有各自独特的结构,因而每个处理器都具有自己特有的指令集,由特定的指令集决定了每个内核的特定应用,即每个内核都有自己擅长处理的一类程序。而对于同构多核处理器而言,每个内核的结构相同,指令集相同,所支持的功能也相同。在处理线程间差异小的应用程序时,同构多核确实具有优势,因为它结构简单、核间无差别、指令集统一,这也正是现在市场上同构多核产品具有绝对性优势的原因。而对于有特定要求的多线程应用程序来说,异构则是更好的选择<sup>[2]</sup>。在现今的应用程序中,有很大一部分应用是多线程,所包含的线程各不相同,而且线程间存在很大的差别,其要求也不同(严格的时效性、高速的数据处理能力)。这样的应用程序如果放在同构多核上运行则无法满足不同线程的要求;用同样的处理器处理差别很大的不同线程,那么其执行结果的瓶颈必然由最耗时的那个线程所决定<sup>[3]</sup>。而异构多核处理器内核的异构性可以同时执行具有不同要求的应用程序,满足不同的要求。在异构系统中,可以通过操作系统把该应用程序的线程依照自己的特点分配到不同的处理器上(比如把包含大量运算的线程分配到善于处理运算指令的处理器上,把包含大量 load/store 指令的线程分配到善于处理访存指令的处理器上),让每个处理器都发挥出自己的特点,这样就可以全面提升程序的运行效率,而且可以在一定程度上消除不同线程之间由于线程本身特点而导致的运行时间上的差别<sup>[4,5]</sup>。

综上所述,随着应用的变化、需求的增长以及技术的发展,异构多核必将成为未来多核的发展主流。异构多核的内核的功能各不相同,其结构也不尽一样,内核的指令集不同,内核的连接也有多种

形式。如异构多核处理器中可以包含定点、浮点运算核,标量、向量运算内核;各个内核结构中的局部存储、多级 Cache 由各个内核的功能和需求决定;内核间可以有 Cache 共享、共享的存储器;内核间的互连方式也不尽相同。因此,面对不同的应用研究核结构的实现对异构多核处理器的性能至关重要。

本文对异构多核处理器体系结构的设计进行了探讨,以促进进一步的研究。从体系结构的角度分析了异构多核处理器的设计与研究;考虑了高性能应用和嵌入式应用对异构多核设计的影响,以及异构多核处理器在这两种应用中面临的挑战,指出了异构多核针对二者的设计难点和关键因素。

## 2 异构多核处理器的体系结构设计

多核处理器并未规定核的结构是否相同。通常来说,同构多核处理器的设计比较简单,其各个核结构相同,所以一般来说各个核均为通用处理单元,其设计原则是优化各种程序的平均执行性能,设计的重点在于内核的互连和通讯方式(内部互连总线、共享 Cache、共享存储等)、共享资源的调度策略与方法等。与同构多核处理器相比,异构多核处理器的设计比较复杂,不仅需要考虑核间的互连通信和存储结构,更要根据各个内核所支持和执行的应用程序来设计每个单处理器。此外,操作系统对异构多核处理器的底层架构也要有很好的支持,否则无法显示出异构多核处理器的特性和优势。操作系统在进行任务分配时,需要将核特性作为任务分配的一项参考因素,把具有不同需求的任务分配给擅长不同类型的计算内核。

异构多核处理器体系结构的设计方法已经从单纯的多处理器核的统一构建发展为根据需求进行的自动化构建,逐渐从通用的设计目标转化为面向某一特定领域专用的设计目标<sup>[6~9]</sup>。随着工艺的发展和应用程序需求的提高,使用可配置的、具有基本加速功能单元的处理器核作为多核处理器的节点已经不足以满足性能需求。而使用定制处理器核作为异构多核处理器的节点进行更加灵活高效的多处理器核配置,是异构多核处理器体系结构发展的重要趋势<sup>[10,11]</sup>。同时,异构多核处理器体系结构的设计需要在不同体系结构的处理器核之间进行任务的映射以及任务间的通信。如何构建高效的异构多核处理器的设计开发平台,进行有效的软硬件任务划分后进行任务映射,都是当前该领域

的重要研究内容<sup>[12,13]</sup>。

总体而言,在异构多核架构设计中存在着诸多的设计难点。目前,多核结构已经取得了理论和实践上的巨大成功,但是针对异构多核的设计仍然面临着从体系结构到软件开发等各方面的若干挑战。这些挑战同时也是机遇,它们的成功解决决定着异构多核处理器未来的发展方向<sup>[14]</sup>。

### 2.1 内核结构及核间关系的设计

相对于同构多核处理器的同构性,异构多核处理器具有多个不同架构的内核,确定每一个内核自身的结构和功能特性是构建异构多核结构的基础。一般来说,异构结构所包含的每个单处理器都应该是针对某种特定应用的特点进行设计和选择,在设计的过程中设计者需要考虑内核间的功能重复性。处理器内核的功能重复会导致结构上的相似性,甚至是某部分完全相同,而重复的结构或功能在一定程度上意味着结构中存在冗余设计。在实际应用中,某些异构系统由于应用的需求而需要处理器内核具有某些重复甚至相同的功能,这是由特定应用的特殊需求造成的<sup>[15]</sup>。所以,设计者要考虑这种重复的必要性,这个问题也是异构结构设计过程中需要充分考虑的一个难点。

### 2.2 多核之间的互连方式

核间互连技术是多核处理器设计的关键问题,核间互连方式和通信协议影响着多核的性能。多核处理器的出现使微处理器的发展从传统的单纯追求主频变为追求更多的处理器核。此时,整个芯片的主要性能瓶颈之一是大量处理器核之间的通信问题。几十甚至上百个处理器核的协同工作必然需要彼此之间大量的信息交换。只有在高效的互连结构的支持下,如此多的处理器核才有可能有效地进行信息交换,为处理器核间的并行工作提供有效的保障。异构多核由于内核的异构性以及内核间性能的不同使得异构多核中的互连设计更复杂。能否有效地解决核间互连的问题将直接影响异构多核的设计以及应用。

当内核数目较少时,最简单有效的互连方式是基于共享存储器的总线互连。可以采用片上系统的分时共享总线的方法实现内核之间的互连。常用的分时共享总线包括 ARM 公司推出的 AMBA 和 OpenCore 组织所支持的 WISHBONE。采用总线技术进行内核间的互连存在可扩展性不强、通信性能不高、同步困难、功耗过大和可靠性不高等问题。

还有一些其他的互连方式,如交叉开关、点到点通信等。这些互连方式对于多处理器的并行计算应用来说采用交叉开关进行通信,其效率是最高的,相比于总线的每次只能处理一个请求,交叉总线或者点到点通信都可以实现并行通信,但其实现的代价较大。

集成电路技术的高速发展,为未来将数百甚至数千个内核集成到单一芯片中提供了可能性。当处理器核个数越来越多时,亟需一种高带宽、低延迟、低代价的互连结构。为了提高处理器核之间的通信带宽和效率,需要设计复杂高效的片内多核之间的互连网络结构,如片上网络 NoC(Network-on-Chip,简称 NoC)<sup>[16]</sup>。片上网络是芯片设计工程师们借鉴网络领域的相关概念提出的一种新的通信技术,它借鉴计算机间网络的特点,将其下移到处理器内部,以实现多核之间的通信。截止目前,典型的片上网络研究包括 RAW<sup>[17]</sup>、TRIPS<sup>[18]</sup>、SPIN<sup>[19]</sup>、Aethereal<sup>[20]</sup>与 MANGO<sup>[21]</sup>等。可以预测:在未来几年内,随着处理器核数量的增加,片上网络结构将以其性能、功耗、成本等方面的优势完全取代总线式、交叉开关式结构。图 1 是一个典型的片上网络结构。

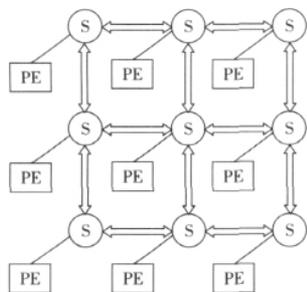


图 1 片上网络

与传统的总线结构相比,片上网络结构采用模块化设计,可扩展性强,集成的多核数量理论上不受限制,有效控制了线延迟,有利于提高工作主频。此外,片上网络结构提供了更高带宽,更容易满足多核并行通信需求,很好地解决了总线结构中存在的主要问题。

### 2.3 存储系统的设计

异构多核结构中片上通信架构的功能不仅包括各个内核之间的数据传输,也包括为各个内核提供高效的内存数据存取性能。内核如何合理地存取数据成为多核结构设计中的一个难点。

大多数的多核处理器采用层次存储结构,包括寄存器、Cache/Scratchpad Memory、主存及磁盘。当前存储系统的研究工作也大都基于多级存储层

次结构,在不同的存储层次上进行各种访问带宽和延迟的优化,如更丰富的存储资源类型、动态调整访存序列、使用更宽的总线以及 eDRAM 技术等。

在多核结构中,有多种 Cache 结构。一般来说,每个内核都有一个私有的一级指令 Cache 和数据 Cache。内核也可以有各自私有的二级 Cache,如图 2a 所示;也可以共享二级 Cache,如图 2b 所示;还可以在私有的二级 Cache 基础上共享三级 Cache,如图 2c 所示。

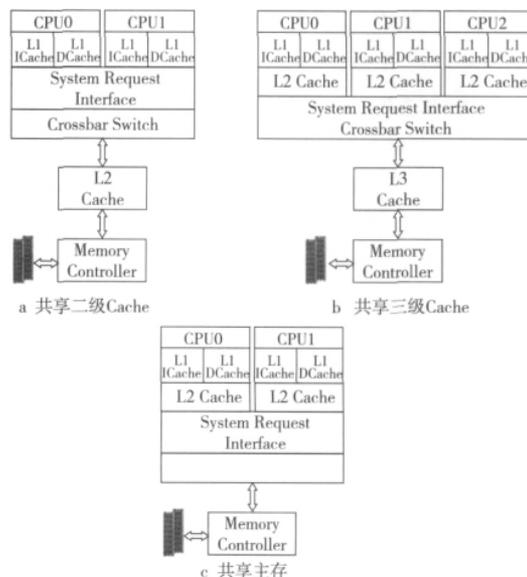


图 2 多种存储层次

存储系统的设计一直是处理器设计面临的一个难题。异构多核处理器中的存储系统设计也是其设计的难点和关键点。异构多核处理器中存储模型使用硬件支持的一致性协议或者添加同步指令来解决共享存储器的一致性,采用不同的方式掩盖存储器访问延迟或者优化存储器组织结构来加速存储器访问速度,都是重点研究方向<sup>[22,23]</sup>。

常用的提高存储带宽的方法就是提高存储器时钟频率或者增加总线宽度,但对于传统 DRAM 器件来说,这些方法已经接近其物理极限。因此,研究者开始研究将存储器实现工艺和逻辑实现工艺有效融合的技术,设计实现嵌入式 DRAM,将处理器与存储器集成在同一块芯片上,使处理器逻辑充分利用 DRAM 内部结构本身的巨大带宽(可达 TB/s 量级),从而彻底消除处理器和存储器之间的带宽鸿沟。另外一种可能的途径是三维内存集成技术,它能够同时实现高密度与高带宽,是未来解决异构多核处理器存储瓶颈的有效方法。

随着微电子技术的发展,可以使用更大容量的片上 Cache 来缓解 Cache 失效问题,如 Sun 的 Ni-

agora-2 处理器使用了 4MB 的 L2 Cache, Intel 的 Xeon5300 系列处理器使用了 8MB 的 L2 Cache。虽然片内 Cache 可以有效地捕捉局部性较好的数据重用,但随着应用程序的多样化,很多程序访存行为并不符合 Cache 特性。同时,Cache 结构存在访问失效开销不易预知的特点,不利于多核处理器性能发挥。新的片上存储技术例如软件可管理的片上 Scratchpad 存储器<sup>[24]</sup>应运而生,它使用更精细的方法使用宝贵的片上存储器和片上带宽,减小了对片外存储的依赖。

#### 2.4 软件开发、操作系统及应用程序的移植和改造

异构多核设计中的一个重要研究方向是基于异构多核处理器平台的软件研究<sup>[25,26]</sup>。与传统的单核微处理器平台上的软件开发不同,异构多核在不同的处理器核上运行不同的操作系统、不同的编译器以及不同的编程模式,而不同平台上的软件需要定义合适的接口来有效支持多个处理器核之间的信息交互。当前,针对异构多核处理器的编程工作十分复杂,软件工程师必须了解异构多核处理器的底层结构,决定硬件和软件的任务分工。此外,还必须决定各个处理器核的任务分配以及多个处理器核之间的协同交互。有时还必须兼顾实时性与低功耗的需求。

对于一个新设计的体系结构来说,选取的操作系统以及如何把它移植到新结构上是一个很棘手的问题。首先,设计者需要根据新结构的指令集系统选取合适的汇编器以及编译器,然后根据编译器的需求来选择操作系统的类型。另外,对于异构处理器来说,很大程度上是为了特定应用而设计的,因此在选择操作系统时还要考虑到对特定应用程序的支持,比如多线程、可调度性等。

#### 2.5 异构多核处理器的测试与验证

异构多核处理器的异构性给处理器的测试和验证带来了多样性和复杂性。如何对异构多核处理器进行测试和验证是当前的研究热点之一,其研究内容主要集中在如何定义不同体系结构处理器内核间的接口,对多个处理器内核进行协同测试。

Richter K 等人提出了一种事件模型接口和事件流机制,将传统的用于实时系统测试的方法扩展到异构多核处理器的测试和验证中<sup>[27]</sup>;Yoo S 等人提出了一种基于软硬件接口的混合层的模拟方法,对异构多核处理器进行软硬件协同测试与验证<sup>[28]</sup>。PAUL J M 等人提出了一种系统级的建模和模拟方法,对可编程的异构多核处理器进行系统

级的建模与测试验证<sup>[29]</sup>。但是,这些研究大部分停留在理论阶段,真正能用于实践的测试验证开发环境目前还很不成熟。

#### 2.6 异构多核处理器的 DVFS 设计

目前,多核处理器是提高性能的主要途径,但多核处理器未在嵌入式领域大量使用,首要因素是能耗问题。处理器能耗往往占整个计算机系统能耗的 50%以上,降低处理器能耗已经成为多核处理器系统实时节能设计的关注焦点。为了降低能量消耗,一些硬件节能技术,如关闭闲置的电路或者动态降低电压和频率(Dynamic Voltage Frequency Scaling,简称 DVFS)已被广泛应用。近年来随着片上稳压器的有效使用,片上 DVFS 的效能得到了显著提高,并获得更多的多核系统能量节余,从而使得在处理器核内进行更细粒度的能量管理已成为现实。

在 DVFS 设计中,电压转换器不具有在短时间内迅速调节到不同电压的能力,因此 DVFS 技术在早期体系结构设计中的使用一直受到限制。如今,发展片上多核处理器系统结构已成为在节能限制条件下保证性能提高的重要方法,由此产生了对实现快速核内 DVFS 控制机制的需求。电压稳压器可以集成到同一块芯片上,此时可以得到纳秒级电压切换和核内电压控制的优势。现有研究已经从体系结构设计的角度得出结论:片上稳压器能够显著地提高 DVFS 效能,并能得到整个片上多核系统的能量节余,但必须在设计系统和算法时考虑开销和成本。

文献[30]以 4 核的片上多处理器(CMP)系统为例,主要研究了三种 DVFS 配置,如图 3 所示。图 3a 表示为传统的片外稳压器设计,实现了对整个处理器从 3.7V 直接到 0.6V~1V 的电压调节;图 3b 表示两步电压转换配置,先经过片外稳压器调节电压,再通过一个片上稳压器为所有 4 个核提供片上全局 DVFS 支持;图 3c 表示通过 4 个独立的片上稳压器提供片上每个核 DVFS 支持。在多核系统中实时节能调度研究中可以基于片上全局 DVFS 技术,针对所有正在执行的任务来设计统一电压频率调节策略;也可以基于片上每个核 DVFS 技术,针对每个调度任务来设计各自不同的电压频率调节策略。

一般来说,在异构多核系统中主要有三种片上 DVFS 技术:全局 DVFS、局部 DVFS 和混合 DVFS。全局 DVFS 对整个异构多核系统进行统一的电压频率调节,所有的处理器核由一个全局的

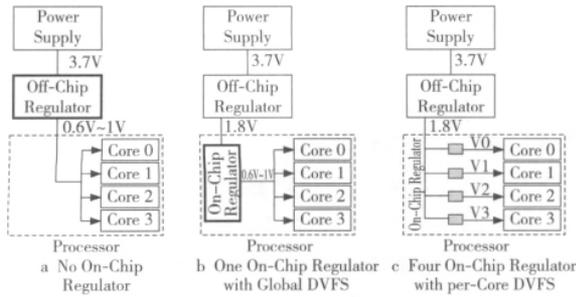


图3 4核CMP的三种DVFS配置<sup>[30]</sup>

DVFS控制单元控制,不考虑各个处理器核性能的异构性,这种全局DVFS可以是片外DVFS也可以是片内DVFS,即图3a和图3b;局部DVFS则将DVFS控制单元放到每个处理器核内部,允许对每个处理器核独立进行电压频率调节,即图3c中的配置方式;而混合DVFS可根据异构多核系统的异构性将其分为多个核域,每个核域包含同种类型的处理器核,属于同一核域的处理器核具有相同的DVFS控制单元。

### 3 高性能异构多核处理器设计

在计算机发展初期,微处理器主要用于科学及工程计算,重视浮点运算能力。目前,虽然科学计算已不是市场主流,但人类对科学计算的需求永无止境。高性能计算技术在航空航天、石油勘探和开发、大范围气象预报等领域仍起着重要作用。因此,未来微处理器的体系结构仍将沿着高性能的路线不断发展。多核处理器体系结构首先被应用于高性能计算领域,也是当今高性能计算系统核心计算芯片的主流结构。而异构多核处理器由于其可将不同类型的计算任务分配到不同类型的处理器核上并行处理,可以为不同需求的应用提供更加灵活、高效的处理机制,从而成为未来高性能处理器发展的必然选择和趋势<sup>[31,32]</sup>。随着集成电路工艺技术的发展和应用需求的不断提高,以及异构多核内在的异构性带来的灵活、高效的处理器机制,可以预见异构多核处理器必将应用于未来高性能处理器系统中。

异构多核在高性能计算系统的应用面临着诸多挑战:实际应用的持续性能低(存储墙问题,系统平衡性差,资源管理复杂度高可扩展性差)、可靠性和可用性低、功耗问题严重等。总的来说,学术界和工业界针对高性能应用的多核处理器的研究思路可以归为三类:

(1)小量大核。小量大核研究思路侧重于将多

个已有复杂单核处理器资源集成于一个芯片上,并进行存储系统和多核互连的优化设计,以达到提高处理器性能的目的。由于所集成的处理器核比较复杂,消耗的资源比较多,片上只能集成数量较少的核。这种多核体系结构广泛应用于通用多核处理器的研制,如Intel与AMD的通用多核处理器系列芯片。不过随着集成电路技术的发展,这类结构也将会集成越来越多的处理器核。

(2)大量小核。这类多核体系结构侧重于探索未来高性能处理器发展思路。大量小核可以提高多核处理器的效能,并且芯片的层次性与局部性较强。采用了该种多核并行结构的研究项目有Raw体系结构<sup>[17]</sup>、TRIPS体系结构<sup>[18]</sup>和Intel 80核等。也有越来越多的该种多核处理器已经进入实用阶段,如80核Cyclops处理器和Tilera公司的64核处理器。

(3)专用处理器。这类多核体系结构针对特定应用进行加速而提出专用的并行结构,主要包括SIMD多核、大规模多线程多核以及异构多核。其中SIMD多核广泛应用于高性能计算加速芯片设计以及GPU芯片设计,如GRAPE-DR、Clearspeed和GPU系列芯片。而针对存储墙,以及其它CPU长延迟操作所带来的性能消耗,Threadstorm处理器提供了128个硬件线程资源,可以在每个周期进行切换,从而保证处理器时刻处于运行状态。另外,异构多核方式在芯片中除了有通用的主处理器,还有用于加速应用的协处理器,从而能够更加高效地完成计算任务,如CELL处理器和Imagine等。不过可以看到,这些加速应用的协处理器也是基于SIMD执行模型的。

异构多核处理器能否达到千亿、万亿次级别的高性能计算能力,以及能否具有良好的实际应用性能,是异构多核处理器在高性能应用中面临的首要问题。在进行异构多核设计时研究者要考虑是采用数量较少的高性能计算内核,还是采用数量较多的适度性能计算内核。另外,在片内单核并行结构方面还必须深入研究:如何设计结构简单、芯片资源利用率高、并行计算能力强、可扩展性好的处理器计算内核。如何更有效地利用硬件资源来实现大规模运算单元集合,充分挖掘各层次并行性并降低编程开销。

### 4 嵌入式实时异构多核处理器设计

随着互联网和媒体技术的迅猛发展,网络服务

和移动计算逐渐成为一种重要的计算模式,它需要处理器具有响应实时性、处理流式数据的能力,支持数据级以及线程级并行,同时还要求处理器具备较短的上市周期。因此,未来处理器的体系结构在充分发挥高性能的同时也会朝着嵌入式领域逐步推进。异构多核处理器由于其内在的异构性带来的灵活、高效的处理机制成为嵌入式系统的必然选择。对于嵌入式实时系统而言,最重要的两个指标是实时和低功耗。因此,在进行嵌入式异构多核处理器设计时,必须考虑嵌入式实时系统的实时性和低功耗的需求<sup>[33,34]</sup>。

#### 4.1 异构多核系统中的实时性

目前多核处理器很难甚至不能应用于嵌入式实时系统中,这是因为实时系统的正确性不仅体现在程序的正确执行,还体现在任务的执行时间不能超过截止期的要求。大部分嵌入式应用的实时性要求处理器体系结构设计给出确定性的实时性预估结果。任务的执行如果不能满足对截止时间的要求,会降低系统服务的级别,甚至造成系统的崩溃。因此,获得任务安全、准确的最坏情况执行时间(Worst Case Execution Time, 简称 WCET)是至关重要的。分析实时程序最坏情况执行时间的目的是在程序或者程序片段执行之前获得其最坏情况的执行时间估值。事先获知任务的 WCET 估值是实时系统调度及可调度性分析的前提,也是检查实时系统的性能是否满足要求的依据。这意味着在进行多核微处理器体系结构设计时,对应用程序的实时性指标满足情况应使其具备一定的可确定性,便于对异构多核处理器的设计是否满足应用需求具有一定的掌控性。即在异构多核系统中对任务最坏情况执行时间的确定和任务实时调度是满足嵌入式系统实时性的重要因素。

相对于单核处理器而言,多核处理器很难分析任务在最坏情况下的执行时间。而异构多核的异构性使得这种不确定性变得更加复杂。在多核处理器中,任务在访问硬件共享资源(例如 Cache、片上互连、存储等)时会产生干扰,一个任务的执行时间可能会随着其它正在同时运行的任务的改变而改变,这使得任务的最坏情况执行时间无法确定,时间分析变得比较复杂。先进结构带来的先进特性也导致 WCET 的分析方法变得很保守,从而影响调度分析,使得系统效率和资源利用率降低,甚至影响可调度性分析,使得原本可调度系统变得不可调度。体系结构影响了 WCET 分析方法的有效性、可行性以及精确性。较之于简单的单核体系结

构而言,异构多核体系结构对嵌入式实时系统的 WCET 分析提出了更多的挑战和难题。

(1)共享 Cache:Cache 通过存储访问的局部性来隐藏 CPU 和存储之间的延迟。在单核体系结构中,Cache 结构以及替换策略都已经研究成熟,这些技术也都已经运用到了 WCET 分析工具中。然而,单核处理器中的这些技术在多核中不再有效。多核体系结构支持共享二级 Cache,运行在某个核上的任务可能会破坏另一个运行在其它核上的任务的二级 Cache 中的数据。因此,任务的 WCET 不可再像单核中那样进行独立的分析。在具有共享 Cache 的多核体系结构中,设计者需要模拟和预测并行程序的 Cache 行为,分析任务由共享 Cache 干扰而引起的执行时间的改变以及对 WCET 的影响。

(2)片上互连:片上互连对 WCET 分析有着很大的影响。在传统的单核体系结构中,只有一个 CPU 请求总线,此时的时序行为是可确定的。而在多核结构中,由于同时有多个核在运行程序,片上互连可能同时接到多个核的请求,片上互连时序行为随着请求的改变而变化,则每个任务的执行行为也存在着很大的不确定性,实时任务的 WCET 也很难确定。如果片上的核较多,一般采用片上网络 NoC 进行互连,片上网络可以同时处理多个请求,在这种多核结构下由 NoC 引起的干扰相对总线而言复杂性更大,WCET 的可分析性更难。

(3)共享存储:Cache 和片上互连都是片上的共享资源,对于片外的共享存储而言,在多核结构中由片外共享存储系统引起的干扰对任务的执行时间和 WCET 估计也有着非常重要的影响。多核中同时运行的任务会同时发出访存请求,则任务间会存在由存储引起的存储干扰,由此会影响实时任务的执行时间和 WCET 分析。

由以上分析可知,嵌入式实时系统的 WCET 分析在多核处理器中还面临着软、硬件系统的众多理论和技术难题,国内目前还没有针对 WCET 在多核中的研究,国外的相关研究也才刚刚起步。设计者需要针对多核处理器在嵌入式实时系统的 WCET 分析中的这些难题,提出多核中的 WCET 分析方法。可以借鉴成熟的、单核中的 WCET 分析方法,为分析多核中 WCET 分析提供有力的支持。一些研究者从多核体系结构中的共享硬件资源出发,分析构成任务干扰的硬件共享资源;从片上共享二级 Cache、总线/片上网络以及片外共享存储出发,解决多核在嵌入式实时系统中 WCET

分析所面临的关键技术。还可以从体系结构出发,提出可预测的硬件支持保证实时任务的时间可分析性,满足任务的时限,为阻碍多核处理器在嵌入式实时系统中的应用探索新的道路。

对于异构多核的实时调度,多核处理器的高性能正是通过多线程并行化来获得的,而并行处理的高性能又极大依赖于并行任务的调度。因此,多核系统中实时调度应该主要关注并行任务模型的实时调度,重点考虑如何将多个线程映射到多个处理器核或者功能单元,目标是既满足任务的实时性要求又获得好的整体性能。与同构多核系统相比,异构与之有许多不同特性,核与核之间具有不同的指令集,存在差异性,不允许任意任务都可以在任意处理器核上运行。这导致两者所基于的任务分配方法不同。在同构多核系统中一般采用全局法的实时调度,而异构多核系统的异构性导致处理器核分配的局限性,使得异构多核系统多采用划分法的实时调度算法。一般来说,划分法调度相比全局法调度有如下优点<sup>[35]</sup>:

(1)如果任务超过最坏执行时间而完成,那么它只能影响到相同处理器核上的其他任务。

(2)当每个任务只运行在单个处理器核上时,那么不会有根据迁移代价带来的惩罚。例如,一个任务实例在一个处理器核上执行时,被强占后又在另一个处理器核上恢复执行,则必须在第二个处理器核上保存其上下文。这会导致额外的通信开销和 Cache 丢失,但在非迁移的划分法调度情况中则不会发生。

(3)划分法方法为每个处理器核使用单独的运行队列,而不是一个全局队列。对于大型系统,操作单个全局队列的开销可能会很大。

使用划分法调度的主要优势在于一旦任务确定了分配到处理器核上,那么以前单处理器系统中大量的实时节能调度技术和分析结果都可以被使用。但是,划分法调度的主要缺点是任务分配问题类似于装箱问题,这已知是 NP 完全问题。目前,多核系统中实时调度研究大多假设单个线程的串行任务模型,并行任务模型的实时调度研究才刚刚起步,如 2009 年 Kato 等<sup>[36]</sup>提出了一种基于全局 EDF 的组(Gang)调度算法,在保证同一个任务的多个线程总是同时执行的条件下,实现并行任务集的硬实时调度。

#### 4.2 异构多核系统中的低能耗

在很多嵌入式实时领域,特别是无线移动和便携式计算领域,能耗是首要考虑的因素,这也是多

核处理器目前并未全面占领嵌入式领域的首要因素。首先,大多数嵌入式设备往往都是依靠电池供电,电池的供电时间是衡量系统的重要参数。其次,能耗的急剧增加会产生更多的热量并导致系统可靠性的下降,因而需要更好的冷却技术,特别是在多核系统中。此外,许多实时系统为了满足严格的实时约束,往往采用保守设计策略,即必须满足系统最大资源需求,而这种策略也会引起较高的系统能耗。因此,实时嵌入式市场的需求,尤其是降低系统能耗的需求,导致了对多核系统中节能设计需求的不断提升。

针对嵌入式领域的异构多核处理器对低功耗的需求,当前最先进的低功耗设计技术也被广泛地应用到异构多核处理器中,例如动态电压调节技术、门控时钟技术等<sup>[37]</sup>。值得注意的是,将传统通用高性能处理器中的低功耗技术应用在异构多核处理器系统设计中时,需要从更多更细的层面上进行综合考虑。为了满足实时应用需求的进一步发展,需要从操作系统、编译、体系结构和电路实现等多个层面解决多核系统中的能耗问题。由于处理器能耗往往占整个计算机系统能耗的 50%,所以降低处理器能耗已经成为多核处理器系统实时节能设计的关注焦点<sup>[38]</sup>。虽然多核系统中硬件节能设计技术已经日趋成熟,但是硬件自适应的能耗管理策略无法保证截止期约束,只有软件层面的节能调度技术才能从根本上决定能否在满足实时性约束的同时有效地降低处理器的能耗。为了降低能量消耗,一些硬件节能技术,如关闭闲置的电路或者动态降低供应电压和执行频率(Dynamic Voltage Frequency Scaling,简称 DVFS)<sup>[39,40]</sup>已被广泛应用。近年来,随着片上稳压器的有效使用,片上 DVFS 的效能得到了显著提高<sup>[41]</sup>,并获得更多的多核系统能量节余,从而使得在处理器核内进行更细粒度的能量管理已成为现实。然而,目前基于 DVFS 的一些软件实时节能调度技术大多针对单处理器系统和多处理器系统<sup>[31]</sup>,在多核系统实时应用中,基于 DVFS 的节能调度技术讨论得较少。

## 5 结束语

随着工艺的发展,多核技术成为当今处理器技术发展的重要方向,异构多核处理器由于可将不同类型的计算任务分配到不同类型的处理器核上并行处理,从而为不同需求的应用提供更加灵活、高效的处理机制而成为当今研究的热点。异构多核

技术在高性能系统和嵌入式系统中的应用使处理器性能的发展面临更多的挑战。对计算机系统结构的设计者来说,原本基于单核的计算机系统系统中的系统级的一些问题也随之引入到了处理器芯片内部,产生了诸如系统的多级并行问题、芯片内的核间通信问题、多 Cache 一致性问题、芯片间互连问题等,更加剧了高性能问题、系统功耗问题、系统实时性问题等。这些新技术面临的挑战是新一轮发展必须的,新问题的解决依赖于新的系统结构技术的突破,更依赖于相关的操作系统、编译器等软件技术的进步,围绕着多核的新的系统技术平衡将逐步建立起来。

总之,有关异构多核结构的设计包含着很多方面,要成功设计、实现并验证一个异构结构,需要从底层硬件、仿真平台、操作系统和上层多线程软件等多个方面来共同开发完成。

#### 参考文献:

- [1] Burger D, Goodman J R. Billion-Transistor Architectures: There and Back Again [J]. IEEE Computer, 2004, 37(3): 22-28.
- [2] Balakrishnan S, Rajwar R, Upton M, et al. The Impact of Performance Asymmetry in Emerging Multicore Architectures[J]. IEEE Computer Society, 2005,33(2):506-517.
- [3] Kumar R, Tullsen D M, Ranganathan P, et al. Single-ISA Heterogeneous Multi-Core Architectures for Multithreaded Workload Performance[C]//Proc of the 31st International Symposium on Computer Architecture, 2004:64-75.
- [4] Sun Fei, Ravi S, Raghunathan A. Application-Specific Heterogeneous Multiprocessor Synthesis Using Extensible Processors[J]. IEEE Transactions on CAD of Integrated Circuits and Systems, 2006, 25(9):1589-1602.
- [5] Chen Guobing, Chen Tianzhou, Yan Like. On-chip Communication Framework Design for Embedded Heterogeneous [D]. Zhejiang University, 2007.
- [6] Gauthier L, Yoo S, Jerraya A. Automatic Generation of Application Specific Architectures for Heterogeneous Multiprocessor System-on-Chip[C]//Proc of the Design Automation Conference, 2001:518-523.
- [7] Craven S, Patterson C, Athanas P. A Methodology for Generating Application-Specific Heterogeneous Processor Arrays [C]//Proc of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), 2006:251a.
- [8] Wolf W. Multiprocessor Systems-on-Chips [C] // Proc of IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06), 2006.
- [9] Rutten M J. Eclipse: A Heterogeneous Multiprocessor Architecture for Flexible Media Processing[C]//Proc of Workshop on Parallel and Distributed Computing in Image Processing, Video Processing, and Multimedia (PDIVM), 2002:39-50.
- [10] Sun Fei, Ravi S, Raghunathan A, et al. Synthesis of Application-Specific Heterogeneous Multiprocessor Architectures Using Extensible Processors[C]//Proc of the 18th International Conference on VLSI Design Held Jointly with 4th International Conference on Embedded Systems Design (VLSID'05), 2005:551-556.
- [11] Prakash S, Parker A C. SOS: Synthesis of Application-Specific Heterogeneous Multiprocessor Systems[J]. Journal of Parallel and Distributed Computing, 1992,16(4):38-51.
- [12] Sarkar A, Chakrabarti P P, Kumar R. Frame Based Fair Multiprocessor Scheduler: A Fast Fair Algorithm for Real-Time Embedded Systems[C]//Proc of the 19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design (VLSID'06), 2006:677-682.
- [13] Fisher N, Anderson J H, Baruah S. Task Partitioning upon Memory-Constrained Multiprocessors[C]//Proc of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05), 2005: 416-421.
- [14] Krste A. The Landscape of Parallel Computing Research: A View from Berkeley[R]. Technical Report No. UCB/EECS-2006-183, University of California, 2006.
- [15] Wang D T. The CELL Microprocessor[EB/OL]. [2005-02-10]. <http://www.realwordtech.com>.
- [16] Micheli G D, Benini L. Networks-on-Chips: A New SoC Paradigm [J]. Computer, 2002, 35(1):70-78.
- [17] Taylor M B, Lee W. Evaluation of the Raw Microprocessor: An Exposed-Wire-Delay Architecture for ILP and Streams[C]//Proc of the 31th Annual International Symposium on Computer Architecture, 2004:2-13.
- [18] Sankaralingam K, Nagarajan R. Exploiting ILP, TLP, and DLP with the Polymorphous TRIPS architecture[C]//Proc of the 30th Annual International Symposium on Computer Architecture, 2003:422-433.
- [19] Guerrier P, Greiner A. A Generic Architecture for On-Chip Packet-Switched Interconnections[C]//Proc of Design, Automation and Test in Europe (DATE), 2000:250-256.
- [20] Rijpkema E. Trade Offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Network on Chip [J]. Proc of Computers and Digital Techniques, 2003:125-139.
- [21] Bjerregaard T, Mahadevan S, Olsen R G, et al. An OCP Compliant Network Adapter for GALS-Based SoC Design Using the MANGO Network-on-Chip[C]//Proc of Symposium on System-on-Chip (SoC), 2005:171-174.
- [22] Ozturk O, Kandemir M, Irwin M J, et al. On-Chip Memory Management for Embedded MpSoC Architectures Based on Data Compression[C]//Proc of IEEE International SOC Conference (SOCC 2005), 2005:175-178.

- [23] Han Sang-Il, Baghdadi A, Bonaciu M, et al. An Efficient Scalable and Flexible Data Transfer Architecture for Multiprocessor SoC with Massive Distributed Memory[C]//Proc of the 41st Conference on Design Automation, 2004; 250-255.
- [24] Kandemir M, Choudhary A. Compiler-Directed Scratch pad Memory Hierarchy Design and Management[C]//Proc of the 39th Design Automation Conference, 2002;628-633.
- [25] McKee S A, Aylor J H, Salinas M H. Dynamic Access Ordering for Streamed Computations[J]. IEEE Transactions on Computers, 2000, 49(11):1255-1271.
- [26] Wolf W. The Future of Multiprocessor Systems-on-Chips [C]//Proc of the 41st Conference on Design Automation (DAC'04), 2004;681-685.
- [27] van der Wolf P, de Kock E, Henriksson T, et al. Design and Programming of Embedded Multiprocessors: An Interface-Centric Approach[C]//Proc of the 2th CODES+ISSS, 2004;206-217.
- [28] Glass R. Formal Methods are a Surrogate for a More Serious Software Concern [J]. IEEE Computer, 1996, 29(4): 19.
- [29] Yoo S, Jerraya A A, Yoo S. Hardware/Software Cosimulation from Interface Perspective[J]. Computers and Digital Techniques, 2005,152(3):369-379.
- [30] Paul J M, Thomas D E, Cassidy A S. High-Level Modeling and Simulation of Single-Chip Programmable Heterogeneous Multiprocessors[J]. ACM Transation of Design Automation Electric System, 2005,10(3):431-461.
- [31] Kim W, Gupta M S, Wei Gu-Yeon, et al. System Level Analysis of Fast, Per-Core DVFS Using On-Chip Switching Regulators[C]//Proc of HPCA'08, 2008;123-134.
- [32] Jerraya A A, Wolf W. Multiprocessor Systems-on-Chips [J]. San Francisco:Morgan Kaufman, 2004.
- [33] Pedretti K. Summary of Multi-Core Hardware and Programming Model Investigations[R]. Sandia Report, 2008.
- [34] Wolf W. The Future of Multiprocessor Systems-on-Chips [C]//Proc of the 41st Conference on Design Automation (DAC'04), 2004;681-685.
- [35] Shalan M, Mooney V. Hardware Support for Real-Time Embedded Multiprocessor System-on-a-Chip Memory Management[C]//Proc of the 10th International Symposium on Hardware/Software Codesign (CODES'02), 2002;79-84.
- [36] Davis R I, Burns A. A Survey of Hard Real-Time Scheduling Algorithms and Schedulability Analysis Techniques for Multiprocessor Systems[R]. Technical Report YCS-2009-443, Department of Computer Science, University of York, 2009.
- [37] Davari S, Dhall S K. On a Periodic Real Time Task Allocation Problem[C]//Proc of Annual International Conference on System Sciences, 1986.
- [38] Rosing T S. Optimization of Reliability and Power Consumption in MPSoCs[EB/OL]. [2009-11-16]. <http://tima.imag.fr/MPSOC/>.
- [39] 谢向辉, 胡苏太, 李宏亮. 多核处理器及其对系统结构设计的影响[J]. 计算机科学与探索, 2008, 2(6):641-650.
- [40] Burd T D, Brodersen R W. Energy Efficient CMOS Micro-Processor Design[C]//Proc of Hawaii Int Conf on System Science, 1995;288-297.
- [41] Chandrakasan A, Sheng S, Brodersen R. Low-Power CMOS Digital Design [J]. IEEE Journal of Solid-State Circuit, 1992, 27(4):473-484.



陈芳园(1982-),女,湖北钟祥人,博士生,研究方向为计算机体系结构和嵌入式实时系统。E-mail: cfysky@gmail.com

**CHEN Fang-yuan**, born in 1982, PhD candidate, her research interests include computer architecture, and embedded real-time system.



张冬松(1980-),男,河南信阳人,博士生,CCF会员(E200015526G),研究方向为嵌入式实时系统。E-mail: dsongzhang@gmail.com

**ZHANG Dong-song**, born in 1980, PhD candidate, CCF member(E200015526G), his research interest includes embedded real-time system.



王志英(1956-),男,山西长治人,博士,教授,研究方向为计算机体系结构。E-mail: zywang@nudt.edu.cn

**WANG Zhi-ying**, born in 1956, PhD, professor, his research interest includes computer architecture.