



一种面向高性能计算的自主众核处理器结构

郑方^{①*}, 许勇^②, 李宏亮^①, 谢向辉^①, 陈左宁^②

① 数学工程与先进计算国家重点实验室, 无锡 214125

② 国家并行计算机工程技术研究中心, 北京 100190

* 通信作者. E-mail: zheng.fang@meac-skl.cn

收稿日期: 2014-10-09; 接受日期: 2014-11-10

国家高技术研究发展计划 (863 计划) (批准号: 2014AA01A300) 和国家科技重大专项 “核高基” (批准号: 2013ZX0102-8001-001-001) 资助

摘要 随着半导体技术进步, 众核处理器已广泛应用于高性能计算领域. 近年来, 在国家 “863” 计划、“核高基” 重大专项等项目的支持下, 我国高性能众核 CPU 的研发水平也取得了长足进步. 本文介绍一种面向高性能计算的国产片上异构众核处理器结构, 通过统一指令系统、统一执行模型和支持一致性的主存共享, 实现异构核心的深度融合. 本文主要介绍了该处理器面向 “存储墙”、“功耗墙” 和 “可靠性墙” 的优化技术体系. 该处理器已完成集成了 256 个运算核心和 4 个管理核心的原型芯片设计, 峰值性能超过 1 TFlops.

关键词 众核处理器 异构芯片 访存性能 能效 可靠性

1 引言

随着半导体技术的进步, 处理器一直遵循摩尔定律发展, 近十年来, 通过片上集成多个核心提升聚合性能的方式延续了这一规律. 目前处理器的核心数目增长很快, 尤其是面向高性能计算的处理器正在快步迈向众核时代. 相比商用主流的通用多核微处理器, 众核处理器在片上集成了超出数量级的简化核心, 可以提供更高的计算能力、计算密度和性能功耗比, 被认为是构成未来高性能计算系统的核心器件^[1].

近年来, 在国家 “863” 计划、“核高基” 重大专项等项目的支持下, 我国在超级计算机 CPU、服务器及桌面计算机 CPU、嵌入式 CPU 和专用 DSP 领域取得了一系列成果, 提升了我国在 CPU 和 DSP 领域的研发能力和支撑电子信息产业发展的能力. 本文介绍的面向高性能计算的众核处理器结构, 是其中面向超级计算机 CPU 的一项研究成果, 该结构已经完成了原型芯片的设计, 双精度浮点峰值运算性能超过 1 TFlops.

一种可实现的众核处理器架构的设计过程中, 需要进行面积、性能、功耗、可靠性等多方面因素的平衡考虑. 在多目标约束下, 核心组织、存储层次、通信机制、可靠性和能效技术体系等方面都与传统多核处理器设计思路有较大差异^[2]. 为了获得高性能、高可靠、高能效的众核处理器, 学术界和工业界有多种不同的结构设计思路.

引用格式: 郑方, 许勇, 李宏亮, 等. 一种面向高性能计算的自主众核处理器结构. 中国科学: 信息科学, 2015, 45: 523–534, doi: 10.1360/N112014-00299

目前, 商用高性能众核处理器一般采用同构加速器的方式, 如 Intel 的 MIC/Xeon Phi 处理器^[3,4]和 NVIDIA 公司的 GPGPU 产品^[5], 这些众核处理器主要采用加速器或协处理器的方式与通用 CPU 一起构成高性能计算系统, 专门针对应用的计算密集部分进行加速. 使用加速芯片的一个问题是 CPU 与众核处理器的松耦合关系, 存在较大的程序加速部分卸载或片间交互的开销, 带来了额外的数据传递功耗, 混合芯片的系统构建方式也降低了高性能计算系统的组装密度和性能功耗比^[6,7]. 另一方面, 使用同构型众核处理器独立构成系统, 受单核心能力的限制, 一般只能应用于特定的领域, 如 TILERA 公司的 Tile 系列处理器主要用于嵌入式和网络应用^[8]. 而异构众核处理器能够弥补上述不足, 通过在同一芯片内集成具备较强控制和管理功能、能够充分挖掘 ILP 并行的“重”核心和控制结构简单、主要用于 TLP 级并行的“轻”核心, 兼顾应用的好用性和性能, 实现较优的性能功耗比和较高的计算密度.

首先在高性能计算领域采用异构结构的是 Sony, Toshiba 和 IBM 设计的 CELL 处理器, 片上包含 1 个通用处理器 PPE 和 8 个顺序执行的协处理器 SPE^[9]. 目前各主要处理器厂商已经开始异构众核处理器的开发或研究工作, 如 AMD 公司的 APU^[10]、NVIDIA 公司牵头的 Echelon^[11] 项目、Intel 公司牵头的 Runnemedede 项目^[12] 等. 这些众核处理器将功能强大的复杂处理核心和用于线程加速的计算核心集成在一个芯片内, 在大幅提升芯片性能和能效的同时, 保证了芯片处理不同工作集合的适应性.

面向高性能计算的众核处理器面临严重的“存储墙”挑战. 一方面, 众核处理器计算性能高, 但访存计算比较低, 访存带宽成为了众核处理器的宝贵资源, 需要充分提高带宽利用率, 并实现访存和计算的充分隐藏, 以发挥处理器的强大计算能力. GPGPU 处理器核心中使用了 Share Memory 的结构, 片上存储由软件显式精确使用, 减少因为 Cache 冲突和颠簸带来的额外访存流量, 以避免带宽浪费, 并采用大量线程调度的 SIMT 执行机制, 实现访存延迟和计算的充分隐藏^[5]. Cell 处理器也采用了非 Cache 的存储结构, 并设计了异步 DMA 机制, 并通过软件多缓冲有效实现计算与访存的隐藏^[13]. 另一方面, 受芯片资源总量限制, 众核处理器用于发掘存储局部性的片上缓存往往被削弱了, 片上数据的重用机制是众核处理器体系结构研究最为关键的问题之一, 片上数据重用机制包括共享片上存储和片上通信两种方式. 例如, Intel MIC 采用了共享 L2 Cache 的数据复用机制. Tile64, Intel SCC 等处理器都设计了较为强大的片上 2D Mesh 通信网络^[14~16].

面向高性能计算的众核处理器面临严峻的“功耗墙”挑战. 处理器是系统能量的主要消耗者, 其能耗控制是实现未来超级计算系统的核心技术. 根据对集成电路工艺发展路线和国际主流商用处理器规划的分析, 尽管晶体管的密度仍然会随工艺进步不断提升, 但是每个晶体管的能量优化的速率在快速降低, 面向高性能计算的处理器必须系统目标和关键应用需求进行针对性的低功耗设计^[17~19]. 例如, 美国面向 E 级计算的泛在高性能计算 (ubiquitous high performance computing, UHPC) 计划, 主要目标是通过软硬件核心技术突破来大幅提升系统能效. 该计划中面向处理器的研究包括 Intel 公司牵头研究的 Runnemedede 处理器、由 NVIDIA 公司牵头研究的 Echelon 处理和 Sandia 国家实验室牵头的 X-Caliber 处理器. 上述新型处理器的结构具有片上异构、增强的存储和片上网络结构、简化一致性管理等显著共同特点, 通过充分降低硬件复杂性、减少数据移动等方式降低芯片功耗.

面向高性能计算的众核处理器面临严峻的“可靠性墙”挑战^[18]. 未来高性能计算机将集成 10 万以上处理器, 系统核心总数达到亿量级, 必须在处理器中综合采用多种可靠性技术保证系统基本的可靠运行能力. 众核处理器的可靠性设计, 必须权衡开销与收益, 并通过硬件—软件协同的方式, 构建完整的错误校验、纠正、隔离、恢复体系, 以达到系统的可靠性设计要求.

面向上述三大挑战, 本团队设计了一种面向高性能计算的异构众核处理器结构. 该项目在国家多

项重大科技专项的支持下, 已完成了 FPGA 验证和原型芯片设计.

后续章节组织如下: 第 2 节介绍了该众核处理器的体系结构; 第 3 节介绍了该处理器面向“存储墙”的优化技术; 第 4 节介绍了该处理器面向“功耗墙”的技术体系; 第 5 节介绍了该处理器面向“可靠性墙”的技术体系; 第 6 节总结了全文.

2 体系结构

2.1 总体结构

在高性能计算领域, 大量应用对芯片计算能力有很高要求, 同时核心段具有良好的并行可分特征, 这就使众核处理器具有很好的应用前景. 但是完整的应用都具有一定的串行部分, 根据 Amdahl 定律, 这部分将严重影响应用在众核处理器上运行的加速比. 本文设计的众核处理器将不同类型核心集成在一个芯片内, 分别处理程序中不同特征的代码段. 集成少量强大的管理核心有效发掘指令级并行同时完成芯片的管理, 集成众多面向计算开发的精简运算核心高效处理线程级并行、大幅提高芯片性能.

本文设计的异构众核处理器如图 1 所示. 芯片主要由多个管理核心 (management processing elements, MPE)、运算核心簇 (computing processing elements clusters, CPE cluster)、存储控制器 (memory controller, MC) 和系统接口 (system interface, SI) 组成, 一个 MPE、一个 CPE cluster 和一个 MC 为一个核组 (group), 核组间由片上网络连接起来. 系统接口用于与片外系统进行连接, 可以采用标准的 PCIe 接口或自定义接口协议. 核组的数量和运算核心簇内的核心规模可根据需求和物理实现难度灵活调节, 目前原型芯片的设计中包括 4 个核组, 每个运算核心簇包含 64 个运算核心.

自主众核处理器的管理核心是一个字长 64 bit 的全功能 RISC 核心, 可运行在用户态和系统态, 支持完整的中断处理和空间管理, 并支持超标量、乱序发射、乱序执行和推测执行, 可以高效处理应用的串行段, 并向用户提供标准服务和运行环境.

自主众核处理器的运算核心也是字长 64 bit 的 RISC 核心, 只能运行在用户态, 采用乱序发射、乱序执行、静态转移预测的执行机制, 不支持中断处理, 通过段表的方式实现虚实地址代换和保护. 在保证强大聚合运算效率的前提下, 最大程度简化微结构设计.

运算核心采用紧耦合簇的方式进行管理, 簇内的运算核心之间采用拓扑为 $N \times N$ Mesh (本文中为 8×8) 的簇通信网络进行连接. 簇通信网络采用基于信用的虫孔路由, 用于核心间低延迟的轻量级寄存器级数据通信. 簇中还集成了用于多种功能管理的簇控制器, 主要功能如下:

- 流传输引擎. 用于解析和管理来自运算核心的数据流传输命令;
- 中断控制. 用于处理运算核心发往外部系统的中断信号;
- 同步控制. 支持对簇内的运算核心进行快速硬件同步;
- L2 ICache. 运算核心簇内核心共享的二级指令 Cache;
- 一致性处理部件. 保证运算核心访存能够获得管理核心 Cache 中的最新副本.

2.2 存储层次

自主众核处理器的管理核心采用一级数据和指令 Cache 分离、二级指令数据共享的两级片上存储层次. 运算核心指令系统采用了每个运算核心私有的 L1 指令 Cache 和一个运算核心簇共享的 L2 指令 Cache 设计; 运算核心的数据存储以草稿本 (Scratch Pad Memory, SPM) 的方式组织, 直接开放给软件管理, 这种方式简化了传统 Cache 实现的控制开销, 可以实现空间精确使用, 减少 Cache 引起

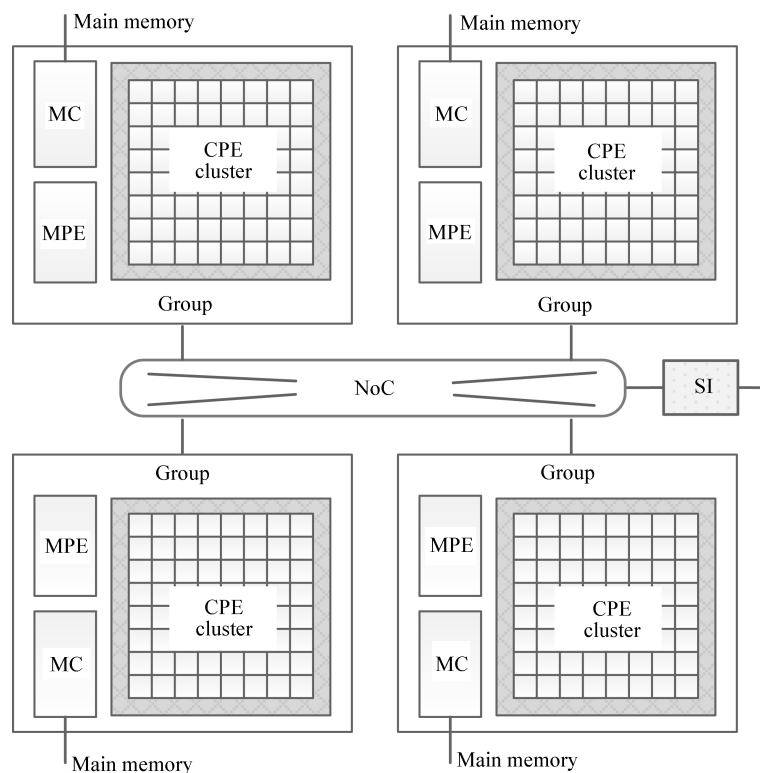


图 1 自主众核处理器架构

Figure 1 Architecture of the homegrown many-core processor

的脱靶流量, 并避免了众多运算核心间一致性处理带来的设计复杂性和性能降级. SPM 也可以配置成一级数据 Cache, 但运算核心间的一致性由软件进行管理. SPM 与主存间支持异步数据流传输, 可通过多缓冲机制实现访存隐藏.

2.3 异构核心融合

自主众核处理器结构的一个重要特点是, 通过采用统一指令系统、统一执行模型和支持一致性的主存共享, 实现片上异构核心的深度融合. 向程序员提供管理核心、运算核心一致的编程环境, 并最大程度减少异构核心协同处理过程中的主存拷贝.

- 指令系统: 管理核心和运算核心的基本指令集兼容, 均为 64 bit 字长的 RISC 结构, 运算核心增加了部分与运算核心簇微结构特征相关的寄存器级通信、数据流传输和硬件同步等指令;

- 执行模型: 管理核心运行标准 Linux 系统, 运算核心运行轻量级操作系统. 管理核心、运算核心均可独立运行, 且执行环境和执行过程一致;

- 存储空间: 各种形态的异构计算的存储空间管理有 3 种不同的模式 (图 2). 目前商用协处理器一般采用前两种方式, 而自主众核处理器采用管理核心和运算核心共享主存的方式. 异构核心的虚空间管理一致, 可以有效减少异构核心协同处理过程中的主存拷贝. 另外, 自主众核处理器的管理核心之间支持 Cache 一致性, 运算核心间不维持 SPM/Cache 的一致性, 但运算核心访存可以获取管理核心 Cache 中的最新副本.

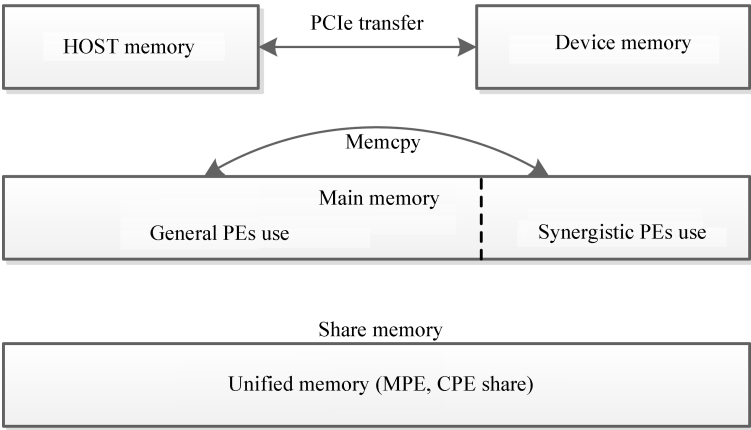


图 2 异构计算的存储空间
Figure 2 Memory space for heterogeneous computing

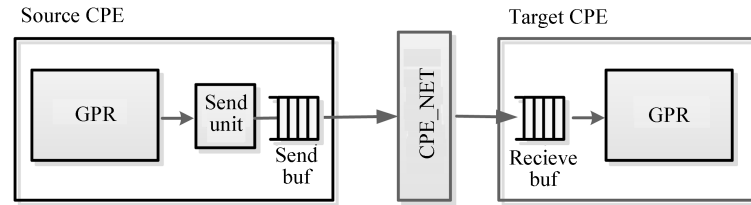


图 3 寄存器级通信
Figure 3 Register level communication

3 面向“存储墙”的优化技术

自主众核处理器面向“存储墙”的主要优化技术包括: 寄存器级通信技术、多模式数据流传输和快速硬件同步, 下文将分别予以介绍.

3.1 寄存器级通信

在高性能计算领域中, 有很多的计算问题虽然有良好的可分性, 但是分割后的子任务具有强相关性, 子任务间的交互可能成为并行化后新的性能瓶颈. 众核处理器本身计算访存比较低、每个核心片上存储空间受限, 每个核心能够独立处理的工作集一般较小, 必须通过有效的多个核心片上数据重用机制, 扩大片上工作集的规模, 最大限度保证处理器计算能力的发挥.

本文设计的众核处理器中采用了一种基于寄存器级轻量级通信的数据重用机制, 通过很低的硬件开销实现了核心间数据细粒度、低延迟移动, 并支持了多播等集合通信功能. 寄存器通信功能通过包装, 可以实现轻量级 MPI 通信原语供程序员调用. 众核处理器寄存器通信机制, 支持数据直接从流水线送入簇通信网络, 通信流程如图 3 所示. 寄存器级通信功能使用双边的生产者 — 消费者协议, 硬件发送接收逻辑简单, 并且不需要设计虚通道, 生产者 — 消费者协议也避免了为建立通信进行的握手或同步协议. 根据实现条件, 自主众核处理器的簇通信网络采用了 Mesh 结构, 通过信用方式进行流控.

与传统的数据共享或通信的机制相比, 寄存器通信完成一次数据传递不需要经过额外的存储层次和握手协议, 并且具有很低的路由器穿透延迟, 对大量片上通信较为规则的课题具有很好的适应性.

表 1 数据流传输模式
Table 1 Patterns of data stream transfers

Pattern	Direction	Description
Single CPE	$MM. \Leftrightarrow SPM$	A CPE transfer data from/to memory
Broadcast	$MM. \Rightarrow SPM$	This pattern broadcasts data to all of the CPEs in a CPE cluster while accessing the main memory
Row	$MM. \Leftrightarrow SPM$	This pattern can circularly arrange the data blocks in one row CPEs
Broadcast row	$MM. \Rightarrow SPM$	This pattern can circularly arrange the data blocks via row dimension and broadcast to multi row CPEs
Column	$MM. \Leftrightarrow SPM$	This pattern can circularly arrange the data blocks in one column CPEs
Broadcast column	$MM. \Rightarrow SPM$	This pattern can circularly arrange the data blocks via row dimension and broadcast to multi column CPEs
Array	$MM. \Leftrightarrow SPM$	This pattern can circularly arrange the data blocks in a 2-dimensional (2D) array

3.2 多模式数据流传输

运算核心支持由软件控制的 SPM 和主存之间的数据流传输, 以精确高效的进行片上存储空间使用. 数据流传输与运算核心流水线异步, 便于预取数据并有效支持访存延迟隐藏.

运算核心通过专门的指令, 完成流传输的发起和配置. 数据流传输的功能主要由集成在簇控制器中的流传输引擎完成, 数据流传输引擎内支持多个传输的并发处理. 为了提高访存效率、保证运算核心服务质量和公平, 数据流传输引擎分别进行了流级、请求 Flit 级和响应 Flit 级三级调度.

数据流传输的一个主要特点是根据结构特征和应用需求, 支持多种流传输模式. 这些传输模式可以使数据以多维方式在运算核心簇中进行分布, 有效提高数据局部化效率, 节约访存带宽. DFMA 的运算核心簇支持单运算核心模式 (Single CPE)、广播模式 (Broadcast)、行模式 (Row)、广播行模式 (Broadcast row)、列模式 (Column)、广播列模式 (Broadcast column)、阵列模式 (Array) 等 7 种. 各种模式的描述见表 1.

3.3 快速硬件同步

自主众核处理器运算核心采用了紧耦合的方式, 运算核心间的同步频繁, 为了提高同步效率减少访存操作数量, 提供了效率较高的硬件方式.

运算核心可以通过指令发出一类特殊的同步包, 包中携带了各个运算核心的同步信息. 同步包通过运算核心网络进入簇控制器中的同步控制模块, 同步控制模块收到同一个同步事件中所有需要同步的核心同步包后, 向这组返回同步完成信号.

4 面向“功耗墙”的技术体系

“功耗墙”是面向高性能计算的自主众核处理器的主要挑战之一. 面向众核处理器的“功耗墙”的技术体系包括: 功耗预评估、多层次低功耗设计、功耗管理、功耗平滑技术等, 其中多层次低功耗设计和功耗管理最为重要.

4.1 多层次低功耗设计

低功耗设计体现在自主众核处理器的结构、电路和工艺等多个层次.

(1) 结构级

主要技术途径是根据高性能计算领域目标, 在保证性能的基础上进行信号翻转率控制、结构精简和局部性开发. 具体采用的技术包括:

- 精简流水线结构. 运算核心的流水线结构 (包括取指译码宽度、发射宽度、执行顺序、推测机制、算子设置、SIMD 宽度等) 都是根据高能效要求, 在复杂的设计空间中选择出的优化组合;
- 0 级指令缓冲. 根据课题指令流的局部性特征, 在一级指令 Cache 和流水线之间增加了容量小、访问能耗低的 0 级指令缓冲, 同时为节省译码能耗, 该缓冲的位置设置在译码站台之后;
- 操作数锁存技术. 为降低通用寄存器文件访问的功耗, 运算核心中使用了操作数锁存技术. 硬件探测连续执行指令的相同操作数对同一寄存器文件条目的读操作, 硬件自动作废对寄存器文件该条目的读访问, 直接使用锁存的操作数; 硬件探测到某个操作数可以从旁路网络获取时, 作废对应寄存器文件的访问, 数据从旁路网络获得; 硬件探测到某个目标操作数被软件标记为“临时变量”(该数据只被使用一次), 当该目标操作数旁路成功后, 不写回寄存器文件;
- 能效优先的部件设计方法. 通过调整部件内部站台的划分和频率约束, 寻找功耗、性能、面积的最佳平衡点.

(2) 电路级

主要针对确定功能的部件, 选择能效最有利的电路实现. 在自主众核处理器中具体采用的技术包括:

- 采用门控时钟. 根据具体部件功能和忙闲特征, 对整个芯片全面采用多种粒度的门控时钟设计. 对于浮点部件等功耗消耗较大的部件, 采用以站台和数据特征为单位的细粒度门控设计. 全片门控率在 80% 以上, 运算核心的门控率在 90% 以上.
- 操作数隔离技术. 运算核心中的多个运算部件之间, 采用操作数隔离技术, 精细控制流水线站台和缓冲中各种触发器的高低电平翻转, 避免电路因无效翻转导致的功耗浪费.

(3) 工艺级

需要密切结合工艺情况, 采用合适的晶体管和逻辑器件, 优化后端设计流程. 并在自主众核处理器根据性能和功耗目标的综合情况, 混合使用多种不同阈值电压的晶体管单元库, 保证频率性能的同时有效控制静态功耗.

4.2 功耗管理技术

功耗管理主要是根据处理器运行情况, 进行芯片的频率、电压、运行速度等的调节. 自主众核处理器中主要采用了多种睡眠机制, 有效降低能量消耗.

(1) 运算核心浅睡眠

运算核心没有收到启动运行信号, 或者运算核心执行“待机”指令后, 流水线将停止取指. 在这两种状态下, 运算核心均处于一种浅睡眠状态, 不执行任何指令不发出任何操作, 寄存器、SPM 和 Cache 中的数据不会丢失. 虽然浅睡眠状态时钟仍然运行在高频状态, 但配合硬件上实现的门控时钟、操作数隔离技术, 仍然可大大降低浅睡眠期间的运行功耗. 运算核心浅睡眠和唤醒的过程很快, 一般用于运算核心运行过程中的短期流水线停顿, 睡眠的基本单位为一个运算核心.

(2) 运算核心深睡眠

当程序进入串行段, 管理核心暂不需要运算核心进行加速时, 可设置运算核心进入更深的睡眠等级, 进一步节省功耗. 运算核心进入深睡眠, 首先需要将其设置于“停工”状态, 当处于该状态的运算核心满足一定的集合要求时 (与时钟树设计有关), 硬件会自动将这些运算核心置于深睡眠状态, 这部分运算核心处于很低的工作频率. 深睡眠唤醒也是以一定集合为单位, 唤醒后运算核心的指令 Cache 和状态寄存器被复位, 可提交新任务.

(3) 管理核心深睡眠

当作业完成, 等待新任务的过程中, 管理核心也可以进入深睡眠状态, 此时管理核心保持很低的工作频率. 管理核心的睡眠和唤醒都是通过中断的方式. 唤醒后的管理核心可以执行新的作业.

5 面向“可靠性墙”的技术体系

众核处理器核心众多、规模很大, 面临严峻的可靠性挑战. 芯片内的故障主要分成固定故障和瞬时故障. 固定故障是指, 芯片制造缺陷或者运行过程中功能单元出现的永久失效, 通过加关电和复位措施不能消除; 瞬时故障是指芯片内部电源/信号噪声或者外部放射性粒子引起的瞬态故障.

自主众核处理器应对固定故障主要技术是冗余和裁剪技术, 以及配套的可测性技术. 本章主要介绍用于瞬态故障的校验体系, 并对其中的纠正与预警、隔离与恢复两类技术进行说明.

5.1 全程一体化校验体系

为保证芯片的稳定可靠, 芯片的校验体系采用了自顶向下的体系化设计, 保证数据和控制通路、存储器与缓冲、芯片外接口、软硬界面校验的连续全覆盖. 并能够较为准确地定位发生错误的位置, 保留重要现场.

进行全程一体化校验设计时, 需要把片上所有的通路、接口和存储器按照出错率、影响严重程度、错误恢复方式等要素进行分类, 以确定采用的校验方式, 除个别低频时钟域的信号外, 实现全片校验覆盖; 需要把重要通路根据错误定位需要, 合理分布校验点; 为加快错误定位和恢复过程, 需要根据错误影响范围、严重程度, 进行底层硬件错误自动分析, 硬件对错误进行分类登记和层次化报送, 为进一步处理提供依据.

自主众核处理器的全程一体化校验体系由校验机制和硬件自动错误分析组成, 其中硬件自动错误分析又包括严重程度分析、影响范围分析和报送登记机制几部分, 下面对各项技术进行简要介绍.

(1) 校验机制

- 片上存储和缓冲主要的校验手段是 ECC 校验和偶校验. 片上主要的 SRAM、Cache、寄存器文件和缓冲都在保护范围;
- 数据通路主要的校验手段是 ECC 校验和偶校验;
- 片上控制通路主要进行偶校验和协议检查. 协议检查包括队列和缓冲状态检查、状态机状态检查、接口协议检查、控制状态一致性检查等;
- 主存接口采用 ECC 校验、存储器清洗和 ChipKill 技术; PCIe 接口采用 CRC 和链路重传; 以太网接口采用 CRC 校验; 维护接口采用偶校验.

(2) 严重程度分析

当芯片发生错误时, 硬件根据错误严重程度从低到高自动分为 5 类.

- 可纠正错误. 对任何可以硬件自动纠正的错误都属于可纠正错误. 可纠正错误不直接对处理器运行产生干扰, 但可能累积引发警告;

- 警告. 警告表示处理器可能出现潜在的运行风险, 该状态将报送上层软件, 由软件决定是否进行应用迁移、芯片运行降级或暂不处理;

- 异常. 硬件检查出的软硬件接口界面发生的协议错误. 主要是由软件程序引发, 也存在硬件发生瞬态错误且未被检出的可能. 该类错误一般交由上层软件进行异常处理 (根据类型不同, 可能交给操作系统或用户处理);

- 不可纠正错误. 硬件对进行错误检查时发生无法硬件纠正或软硬协同纠正的错误, 且该类错误不会导致芯片运行状态混乱. 上层管理软件对该类错误可能采用回卷等容错手段;

- 硬件故障. 片上控制通路发生的严重错误, 该类错误必须通过局部或全片复位进行处理.

(3) 影响范围分析

硬件还根据错误的影响范围进行了分类, 以便于硬件或软件进行后续处理, 尽量通过局部处理的方式减轻错误带来的损失. 其中, 可纠正错误和警告由于严重程度较轻、异常一般是软件程序引起的, 这 3 类不进行影响范围分类. 对于不可纠正错误和硬件故障, 按照影响范围分成运算核心内、运算核心簇内、管理核心内和全芯片 4 类.

需要说明的是, 上述 4 种错误影响范围分类方式, 只是硬件的最小影响范围, 实际影响范围还需要结合正在运行的课题特征进行分析; 另外, 影响范围并不完全与错误发生位置一致, 例如, 访存过程中发生的主存校验错误, 错误发生点是主存, 但是影响范围可能是某个核心.

(4) 报送登记机制

为保证硬件进行错误分析的准确性, 众核处理器设计了相应的错误报送登记原则:

- 已纠正错、警告和异常在检测出的位置进行登记和报送;
- 不可纠正错除了在发现错误的部件进行登记和报错外, 还要携带错误标志通过正常流程返回请求源进行登记 (便于错误关联定位, 也避免将错误的请求源挂死);
- 硬件故障在发现错误的位置进行登记和报错, 并一般采用丢弃操作处理, 防止错误扩散;
- 报错信息分别通知软件系统和维护系统, 便于错误后续处理;
- 错误登记同时登记主要的现场信息, 便于错误后续处理.

5.2 错误纠正与预警

可纠正错和警告是自主众核处理器中较为轻微的错误, 其恢复过程对用户透明, 这里对出现这两类错误后的处理方式说明.

(1) 错误纠正

对于检测出的可纠正错, 在对用户程序透明的前提下, 由硬件完成的错误纠正. 主要的纠正方式包括纠错编码、重试机制和作废.

- 纠错编码. 芯片内广泛使用的 ECC 编码, 具备单错纠正的能力;
- 重试机制. 芯片内采用了重试机制, 例如, PCIe 接口采用的链路重传技术、发生运算部件校验错误后运算核心流水线采用的运算重试技术、取指通路发生错误后采用的取指重试技术等;
- 作废. 多级 Cache 类结构中发生错误, 可通过更高层次获得正确数据的, 对发生错误的 Cache 行进行作废.

(2) 预警

警告是众核处理器包含的一类比较特殊类型的错误状态, 发生警告时处理器的软硬件运行仍是正常的, 但是预示着可能存在很大几率的潜在运行风险, 由上层软件对警告的处理进行决策. 警告的类型主要分为:

- 处理器状态 (如温度、电压) 偏离阈值较大;
- 某特定可纠正错误在一定时间段内频繁发生;
- 硬件发现正常情况下不会使用且可能对处理器造成危害的配置或事件;
- 软件发现存在安全风险的事件.

5.3 错误隔离与恢复

众核处理器片上逻辑规模巨大, 传统的以单个处理器为粒度进行错误处理的做法, 是对计算资源的极大浪费. 对于不可纠正错和硬件故障, 众核处理器支持以多个层次进行独立的错误隔离与恢复.

(1) 错误隔离

错误隔离的主要目标是防止错误扩散, 保证片内分区降级或错误恢复成为可能. 错误隔离的层次分成运算核心、运算核心簇、管理核心、全片 4 个层次. 错误隔离的主要技术是上述 4 个层次的硬件错误追赶技术, 在可能引起错误扩散的错误发生后, 硬件支持以最快速度分析完成故障影响范围, 将该范围内的所有外部接口升起隔离墙, 并保证可能发生扩散的所有信息尚未离开隔离区. 被隔离的部件内部任何请求无法发出, 隔离区外只有维护访问可以进入 (便于错误定位和处理), 其他操作都携带特殊标记返回源方.

(2) 错误恢复

对于已经发生错误的隔离区, 可以采用降级使用或动态恢复的处理策略. 众核处理器错误恢复主要依靠复位技术, 为了保证在复位过程中不影响片上其他正在运行的部分, 众核处理器设计与多层次隔离相对应的复位技术, 包括运算核心、运算核心簇、管理核心 3 种局部复位和全片芯片复位.

6 小结

本文全面总结了本团队多年来设计的一种面向高性能计算的众核处理器结构. 该处理器具有片上异构和深度耦合的结构特征, 同时根据其应用目标, 进行了大量面向“存储墙”、“功耗墙”和“可靠性墙”的关键技术研究工作. 在多项国家重大科技专项支持下, 已完成了基于该结构的 FPGA 验证和原型芯片设计. 原型芯片包括 4 个管理核心和 256 个运算核心, 峰值性能超过 1 TFlops. 后续本团队将继续深入开展面向 E 级计算的众核处理器结构的软硬协同优化研究工作.

参考文献

- 1 Manferdelli J L, Govindaraju N K, Crall C. Challenges and opportunities in many-core computing. *Proc IEEE*, 2008, 96:808–815
- 2 Jung H, Ju M, Che H A. A theoretical framework for design space exploration of manycore processors. In: *Proceedings of the 19th Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Singapore, 2011. 117–125
- 3 Duran A, Klemm M. The intel many integrated core architecture. In: *Proceedings of International Conference on High Performance Computing and Simulation (HPCS)*, Madrid, 2012. 365–366

- 4 Seiler L, Carmean D, Sprangle E, et al. Larrabee: A many-core x86 architecture for visual computing. *IEEE Micro*, 2009, 29: 10–21
- 5 Wittenbrink C M, Kilgariff E, Prabhu A. Fermi GF100 GPU architecture. *IEEE Micro*, 2011, 31: 50–59
- 6 Lee V W, Kim C, Chhugani J, et al. Debunking the 100X GPU vs. CPU myth: An evaluation of throughput computing on CPU and GPU. In: *Proceedings of the 37th annual International Symposium on Computer Architecture(ISCA)*, Saint-Malo, 2010. 451–460
- 7 Daga M, Aji A M, Feng W. On the efficacy of a fused CPU+GPU processor (or APU) for parallel computing. In: *Proceedings of Symposium on Application Accelerators in High-Performance Computing (SAAHPC11)*, Knoxville, 2011. 141–149
- 8 Bell S, Edwards B, Amann J, et al. TILE64 processor: A 64-core SoC with mesh interconnect. In: *Proceedings of the Solid-State Circuits Conference (ISSCC)*, San Francisco, 2008. 88–90
- 9 Maeurer T R, Shippy D. Introduction to the Cell multiprocessor. *IBM J Res Dev*, 2005, 49: 589–604
- 10 Yang Y, Xiang P, Mantor M, et al. CPU-assisted GPGPU on fused CPU-GPU architectures. In: *Proceedings of IEEE 18th International Symposium on High Performance Computer Architecture (HPCA)*, New Orleans, 2012. 1–12
- 11 Keckler S W, Dally W J, Khailany B, et al. GPUs and the future of parallel computing. *IEEE Micro*, 2011, 31: 7–17
- 12 Carter N P, Agrawal A, Borkar S, et al. Runnemed: an architecture for ubiquitous high-performance computing. In: *Proceedings of IEEE International Symposium On High Performance Computer Architecture (HPCA)*. Shenzhen, 2013. 198–209
- 13 Khunjush F, Dimopoulos N J. Extended characterization of DMA transfers on the Cell BE processor. In: *Proceedings of IEEE International Symposium on Parallel and Distributed Processing*, Miami, 2008. 1–8
- 14 Wentzlaff D, Griffin P, Hoffmann H, et al. On-chip interconnection architecture of the Tile Processor. *IEEE Micro*, 2007, 27: 15–31
- 15 Hoskote Y, Vangal S, Singh A, et al. A 5-GHz mesh interconnect for a teraFLOPS processor. *IEEE Micro*, 2007, 27: 51–61
- 16 Gries M, Hoffmann U, Konow M, et al. SCC: a flexible architecture for many-core platform research. *Comput Sci Eng*, 2011, 13: 79–83
- 17 Esmaeilzadeh H, Blem E, St Amant R, et al. Dark silicon and the end of multicore scaling. In: *Proceedings of the 38th annual international symposium on Computer architecture (ISCA)*, San Jose, 2011. 365–376
- 18 Bergman K, Borkar S, Campbell D, et al. Exascale Computing Study: Technology Challenges in Achieving Exascale Systems. Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Technical Report: TR-2008-13. 2008
- 19 Kaxiras S, Martonosi M. Computer architecture techniques for power-efficiency. *Synthesis Lectures on Computer Architecture*. San Rafael: Morgan & Claypool publishers, 2008

A homegrown many-core processor architecture for high-performance computing

ZHENG Fang^{1*}, XU Yong², LI HongLiang¹, XIE XiangHui¹ & CHEN ZuoNing²

¹ State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi 214125, China;

² National Research Center of Parallel Computer Engineering and Technology, Beijing 100190, China

*E-mail: zheng.fang@meac-skl.cn

Abstract Owing to advances in semiconductor technology, many-core processors have been widely used for high-performance computing. Through the support of the national 863 program and the national major project of HeGaoJi, research and development in many-core processors has significantly progressed in China. In this paper, we present a homegrown, many-core processor architecture for high-performance computing systems. The processor is heterogeneous and supports different application features with a unified ISA architecture, a unified execution model, and a share-memory that supports cache coherence. This paper illustrates techniques for solving

“memory wall”, “power well”, and “reliability wall” problems in the processor. The prototype chip achieved a peak performance of more than 1 TFlops with four MPEs and 256 CPEs during experimental tests.

Keywords many-core processor, heterogeneous chip, memory performance, power efficiency, reliability



ZHENG Fang was born in 1984. He received his M.S. degree in computer science from the National Research Center of Parallel Computer Engineering and Technology, Beijing. Currently, he is a Ph.D. candidate in computer science at the State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi. His research interests include high-performance computing and processor architecture.



XU Yong was born in 1974. He received her M.S. degree in computer science from the National Research Center of Parallel Computer Engineering and Technology, Beijing. Currently, he is a senior engineer at the National Research Center of Parallel Computer Engineering and Technology. His research interests include high-performance computing and processor architecture.



LI HongLiang was born in 1975. He received his Ph.D. degree in computer science from the National University of Defense Technology, Changsha. He is an associate professor at the State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi. His research interests include computer architecture and processor architecture.



XIE XiangHui was born in 1958. He received his Ph.D. degree in computer science from the Institute of Computer Technology of the Chinese Academy of Sciences, Beijing. He is a professor at the State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi. His research interests include computer architecture and parallel computing.