

# SAM: 一种容错的末级缓存可扩展地址映射方法

李崇民 汪东升 王海霞 薛一波

(清华信息科学技术国家实验室(筹) 北京 100084)

**摘 要** 随着半导体工艺进步,多核处理器超过 60% 的面积由片上缓存占据. 由于特征尺寸缩小及供电电压下降,片上缓存较以往更容易发生错误. 缓存错误包括可恢复的软错误(soft error)及不可恢复的不稳定位(erratic bit)失效. 传统容错技术主要研究针对单个缓存模块的保护. 当缓存中包含成百上千个模块时,即使单个缓存模块出错的概率很低,系统中有一个或多个缓存出错的概率也相对较高. 文中提出可扩展地址映射(SAM)方法,支持对可缓存地址空间灵活高效的映射,提高末级缓存的可靠性. 通过对末级缓存地址空间进行重构,只要有末级缓存模块可以工作,SAM 就能够保证系统正确运行. SAM 可应用于共享或集群缓存组织方式. 文中提出的算法能根据末级缓存中出错缓存模块的数目变化,动态调整集群缓存组织方式下的集群大小. 实验结果表明,SAM 方法可在多种出错环境下保证系统功能正确,且性能平滑下降.

**关键词** 多核处理器;高速缓存;地址映射;容错;可靠性

**中图法分类号** TP310 **DOI 号**: 10.3724/SP.J.1016.2012.02080

## SAM: A Fault-Tolerant Scalable Address Mapping Method in Last-Level Cache

LI Chong-Min WANG Dong-Sheng WANG Hai-Xia XUE Yi-Bo

(Tsinghua National Laboratory for Information and Technology, Beijing 100084)

**Abstract** With the advance of semiconductor technology, more than 60% area of chip multiprocessor may be dedicated to on-chip cache memory. Due to shrinking feature sizes and decreasing voltage, on-chip caches are more vulnerable to faults including recoverable soft errors and unrecoverable erratic bit failures. Conventional fault tolerance techniques mainly focus on the protection of single cache module. For a cache hierarchy with hundreds of modules, there is a considerable probability that one or more cache modules fail even if the fault probability of single module is relatively low. In this paper we propose Scalable Address Mapping (SAM) method which supports an efficient and flexible mapping mechanism for the cacheable address space and improves the LLC reliability. SAM can reconstruct the LLC memory hierarchy. If there is at least one functional LLC module left on-chip, SAM can maintain the system operating correctly. SAM can be used in both shared and cluster cache organizations. A reorganization algorithm is proposed which can adaptively change the region size for cluster organizations. When the number of fault LLC modules increase, SAM adjusts the region numbers in LLC to alleviate capacity stresses. Simulation results show that SAM can successfully keep system functional under various cache fault situations, while sustaining graceful average performance degradation.

**Keywords** multiprocessors; cache; address mapping; fault tolerance; reliability

收稿日期:2012-06-30;最终修改稿收到日期:2012-08-16. 本课题得到国家自然科学基金(60833004,60970002)、国家“八六三”高技术研究发展计划项目基金(2012AA010905)资助. 李崇民,男,1973年生,博士,助理研究员,主要研究方向为多核处理器体系结构. E-mail: liismn@tsinghua.edu.cn. 汪东升(通信作者),男,1966年生,博士,教授,博士生导师,主要研究领域为多核处理器体系结构、高性能计算、网络存储与安全等. E-mail: wds@tsinghua.edu.cn. 王海霞,女,1977年生,博士,副研究员,主要研究方向为多核处理器体系结构、形式化验证等. 薛一波,男,1967年生,博士,研究员,主要研究领域为计算机网络、信息安全、并行处理、分布式系统等.

## 1 引言

随着半导体工艺不断进步,每一代产品的集成度都在增加,在一个芯片上可以集成多个处理器核。在不久的将来,片上多处理器中会包含有成百上千个处理器核<sup>[1]</sup>。片上存储资源也将占据超过 60% 的芯片面积,缓存的可靠性对芯片的可靠性有较大影响。导致片上缓存错误的原因主要有粒子撞击(软错误)或不稳定位(erratic bit)失效<sup>[2]</sup>。软错误是瞬时的、可恢复的,而不稳定位失效是不可恢复的。这两种错误都与工艺参数密切相关。随着特征尺寸减小,出错概率呈指数式增加。

许多研究关注高速缓存的可靠性,如 CPPC<sup>[3]</sup>、ZerehCache<sup>[4-5]</sup>等,这些技术为单个缓存模块(module)提供容错机制。目前片上多处理器的缓存普遍采用非均一性缓存结构<sup>[6]</sup>(Non-Uniform Cache Architecture, NUCA),末级缓存由多个缓存模块组成,目的是减少数据访问的延迟与竞争<sup>①</sup>。即使单个缓存模块出错的概率很低,片上所有缓存模块中有至少一个模块出错的概率还是相当可观的。

NUCA 结构中,末级缓存的可缓存地址空间由所有缓存模块组成。当一个模块出现不可恢复错误时,整个地址空间的完整性将被破坏,影响到整个缓存系统的正确性。这比某个一级缓存出现不可恢复错误更加严重,因为系统可以关闭出错的一级缓存模块,在损失部分性能后系统仍然能够正确运行。目前的相关研究主要关注单个缓存模块的可靠性,考虑末级缓存系统整体可靠性的研究较少。

本文提出可扩展地址映射(SAM)方法,可以在大规模缓存系统中进行灵活的地址映射。当末级缓存中有缓存模块出现不可恢复错误时,SAM 将缺失的那部分地址空间重新分配到其他功能正常的缓存模块来重构可缓存地址空间。在重新配置之后,可缓存地址空间重新变得完整,系统可以继续运行,因此能够提高系统的可靠性。极端情况下,SAM 可以保证系统在所有末级缓存全部失效之前,系统仍然能够正确运行。SAM 可应用于共享与集群缓存组织方式。在集群组织方式(cluster organization)下,当出错缓存模块数目过多时,SAM 可以动态将集群规模调整到一个适当大小并保证性能的平滑下降。SAM 可与其它末级缓存容错技术共存。

本文对 64 核瓦片(TILE)结构的片上多处理器进行模拟实验,结果表明,SAM 可以在不同出错情

况下保证系统性能的平稳下降。本文主要工作包括:(1)指出在末级缓存中,即使单个缓存模块出错的概率很低,但整个系统中存在出错缓存模块的概率相对较高。(2)提出可扩展地址映射方法,可以以较小硬件代价完成可缓存地址空间的重新构建。(3)在 SAM 对地址空间重构过程中,能够动态调整集群组织方式下每个集群规模的大小,以保证缓存性能的平滑下降。

## 2 背景与动机

### 2.1 背景

在当前多数片上多处理器中,SRAM 缓存占据了一半以上的芯片面积。缓存出错通常由缓存位失效引发,缓存位失效可以分为永久失效与非永久失效两大类。永久性失效主要是由随机掺杂波动<sup>[7]</sup>(Random Dopant Fluctuation, RDF)引起的。Kulkarni 等人<sup>[8]</sup>指出缓存位失效的概率随着供电电压的下降成指数上升。永久性失效对芯片的产出率有较大影响,但在对多处理器缓存可靠性的影响中所占比例不大。

非永久性失效包括软错误和不稳定位失效<sup>[1]</sup>。其中软错误由宇宙射线/粒子撞击导致,软错误已对微电子工业产生影响,有厂商称软错误已经成为客户站点中断的主要原因<sup>[9]</sup>。软错误属于瞬时错误,可以通过如 ECC<sup>[10]</sup>纠错码等方式纠正。不稳定位失效会导致 SRAM 单元  $V_{cc,min}$  的波动<sup>[11]</sup>,不过很难得到关于不稳定位失效的详细信息与物理模型。在低供电电压情况下,不稳定位失效会直接导致一个缓存单元的故障率升高,需要引起重视。单个缓存单元的可靠性通常用 FIT(Failures In Time)表示,即一个缓存单元在  $10^9$  个小时内的故障次数。

非永久性失效发生的概率随着工艺的进步而不断上升,研究表明,每一代工艺进步,单个缓存单元发生软错误的概率增加约 8%<sup>[12]</sup>。根据摩尔定律,每一代产品芯片上的晶体管数目约增加一倍,因此芯片上缓存出错的概率成指数式增加。例如将来 16 nm 工艺下芯片的失效概率是 180 nm 工艺下的 100 倍左右<sup>[12]</sup>。老化现象也对晶体管性能有较大影响,导致晶体管的饱和电流逐渐降低。研究认为当工艺达到 32 nm 后老化现象会更加明显。在 45 nm 工

① 为简单起见且不失普遍性,本文假定片上共有两级存储,二级缓存也就是末级缓存。文中提到的故障率/出错概率为系统工作 1 年,出现故障的概率。

艺下,缓存单元的临界电压(threshold voltage)会有 30 mV 的标准差,缓存的出错概率会达到  $10^{-3}$  [13]. 当特征尺寸减小时,缓存的出错概率会更高. 图 1 给出了元器件在不同工艺中,时间与电压的可靠性变化趋势. 除了缓存单元出错之外,一个缓存模块还会受其它故障的影响,如控制逻辑出错,数据链路断开,缓存中的标志或目录出错等 [14],实际上缓存模块的出错概率还会更高.

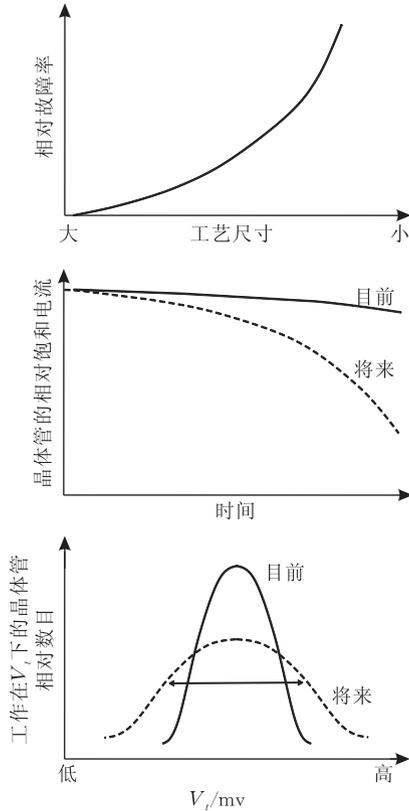


图 1 元器件可靠性下降趋势

## 2.2 动机

对采用 NUCA 结构的片上多处理器,如果一个末级缓存模块出错,则整个末级缓存将无法工作. 假定末级缓存中共有  $n$  个缓存模块,每个缓存模块出错的概率是  $p$ ,且各缓存模块之间相互独立,则整个末级缓存出错的概率为

$$P_{f\_LLC} = 1 - (1-p)^n \quad (1)$$

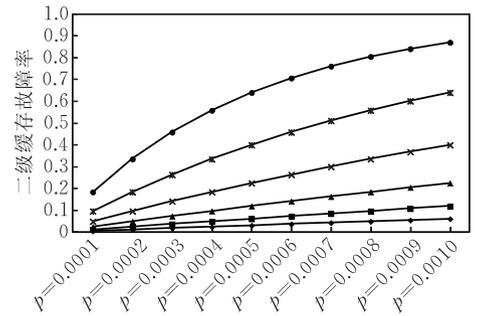
图 2(a) 给出了缓存模块数目从 64 增加到 2048,  $p$  为  $10^{-4} \sim 10^{-3}$  时,整个末级缓存出错的概率. 当  $p=10^{-3}$ ,  $n>128$  时,整个末级缓存的出错概率超过了 10%. 当  $n>1024$  时,即使  $p=10^{-4}$ ,整个末级缓存的出错概率也超过了 10%. 因此末级缓存的可靠性设计面临巨大的挑战.

当末级缓存中包含成百上千个缓存模块时,虽然至少一个缓存模块出错的概率相当高,但多个二

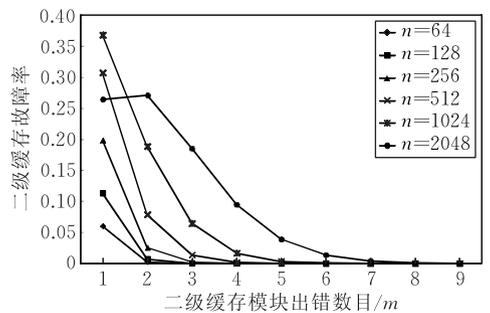
级缓存模块同时出错的概率是相对较低的.  $n$  个模块中有  $m$  个出错的概率为

$$P_{f\_m} = \binom{n}{m} p^m (1-p)^{n-m} \quad (2)$$

图 2(b) 给出了在不同缓存模块数目  $n$  下,恰好有  $m$  个缓存模块出错的概率.



(a) 至少 1 个缓存模块故障



(b) 恰好  $m$  个缓存模块故障

图 2 末级缓存的故障率

若有一种具有容错能力的设计方案,在包含  $n$  个缓存模块的末级缓存结构中,能够容忍不多于  $m$  个缓存模块出错,整个缓存系统功能正常的概率为

$$P_{LLC\_new} = 1 - \sum_{i=0}^m \binom{n}{i} p^i (1-p)^{n-i} \quad (3)$$

图 3 给出在不同  $n$  值和  $m$  值下缓存系统功能正常的概率. 例如当  $p=10^{-3}$ ,  $n=1024$  时,如果能够允许不超过 5 个缓存模块出错,整个系统功能出错的概率小于  $10^{-4}$ . 式(3)是本文提出的可扩展地址映射方法的动机. SAM 通过对可缓存地址空间进

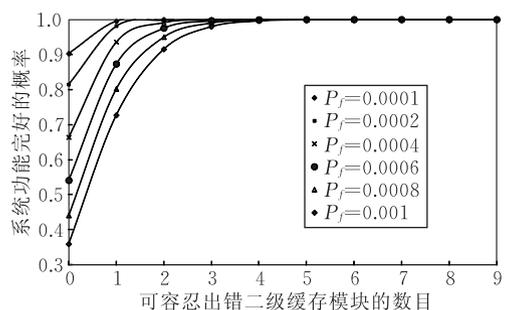


图 3 如果能容忍  $m$  个缓存模块故障,系统功能完好的概率

行重构,能在多个二级缓存模块出错的情况下保证系统正确运行.针对集群组织方式,SAM 能动态调整区域的规模.SAM 保证只要有一个末级缓存模块工作,缓存系统就能正确运行,这同时也允许系统通过关闭部分二级缓存模块来降低系统功耗.

### 3 相关工作

提高系统可靠性的研究主要可以分为三部分:

首先是电路级解决方法,此类方法目标是提高单个缓存单元的可靠性.动态电压/频率调整(DVFS)技术对较易出错的 SRAM 单元施以较高的电压或较低的频率.此类方法需要识别出一列缓存单元中最易出错的那一个,同时访问延迟也是由最易出错的那个单元决定,因而会影响系统性能.其它研究试图去改变传统 SRAM 单元的结构以提高可靠性<sup>[8,15]</sup>,代价是芯片面积增加.

第二种方法是编码解决方法,此类方法适用于故障率较低且以瞬时错误为主的情况.错误检测码(EDC)与纠错码(ECC)<sup>[10]</sup>可用来探测/纠正缓存中的错误.二维错误检测码<sup>[16]</sup>将 EDC 同时应用到数据阵列的行和列上.多位纠错码可以容忍更高的出错概率,但会带来延迟、面积与功耗上的开销.CPPC 方法<sup>[3]</sup>用两个寄存器对缓存中的写回操作提供奇偶校验保护.

最后一类是利用体系结构方法来提高缓存的可靠性.最有效的提高存储结构可靠性的方法是多模冗余,但是其面积与功耗开销太大<sup>[17]</sup>.冗余行或列可以提高存储结构的可靠性<sup>[18]</sup>.有研究将多个损坏的缓存行/块/字合并成一个功能正常的缓存行/块/字<sup>[7,19-20]</sup>,代价是增加了面积和访问延迟.

NUCA 结构<sup>[6]</sup>最早是由 Kim 等人提出,S-NUCA<sup>[21]</sup>可灵活支持不同的共享度(区域大小).Hammoud 等人<sup>[22]</sup>提出了根据缓存缺失率及平均延迟来调整共享度的方法.此外还有将缓存划分为私有与共享区域<sup>[23]</sup>及将缓存用虚拟层次<sup>[24]</sup>进行组织的方法.

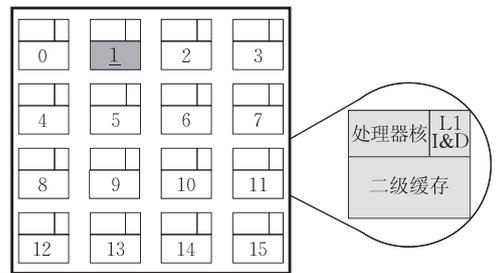
## 4 可扩展地址映射方案设计

### 4.1 常规地址映射

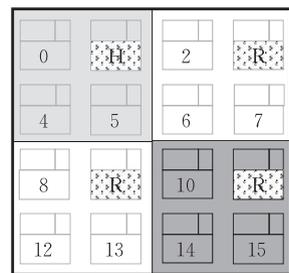
NUCA 结构通过地址映射确定每个可缓存地址所对应的缓存模块.常规地址映射可以分为三类:高位交错、低位交错和地址偏离(address skewing).

这三种映射方式实现较为简单,且具有确定性,即一个地址对应的缓存块只能存在于由地址映射所确定的缓存模块中.如果一个缓存模块出现故障,即使其它绝大部分缓存模块都功能正常,整个缓存系统也不能再正确工作.如果能够利用功能正常的缓存模块重新构建可缓存地址空间,缓存系统的可靠性将大大增加.要重建可缓存地址空间,就需要设计出新的地址映射方法,将可缓存地址空间重新分配到功能完好的缓存模块上.完成地址重新分配后,一级缓存需要知道去哪里寻找一个地址所对应的缓存块.

为说明简便且不失一般性,本文应用一个 16 核 TILE 结构的片上多处理器来说明 SAM 机制的原理与实现.如图 4(a)所示,所有节点通过二维 mesh 网络互连,每个节点中包括一个处理器核,私有的一级数据/指令缓存,和共享的二级缓存,通过路由器与片上网络连接.每一个二级缓存负责 1/16 的可缓存地址空间,并通过片上网络接收来自其它节点的请求.图 4(b)给出了 4×4 集群的组织方式.H 和 R 分别代表全局宿主与区域宿主节点.



(a) 共享组织方式



(b) 4×4 集群方式

图 4 16 核 TILE 结构处理器结构示意图

### 4.2 共享方式下的可扩展地址映射

在大规模片上多处理器中,末级缓存中一般采用目录一致性协议维护数据的一致性.在共享组织方式下,对每个保存在缓存中的数据,都有一个由地址映射确定的宿主节点,在该节点末级缓存中有对应的目录项来记录对应数据共享副本的位置.在一级缓存发生缺失时,会向对应末级缓存发送数据请求.如果宿主节点发生永久故障或者频繁的非永久

故障,属于该部分地址空间的数据就会失去控制,不能保证系统运行正确.如果缺失的地址空间可以重新映射到功能正常的一个或多个缓存模块上,就能保持系统正确运行.

本文提出的可扩展地址映射(SAM)方法可以实现上述功能.图5用一个简单的例子来说明SAM容错功能的实现机制.示例系统中包含两个处理器核,末级缓存由两个二级缓存模块组成.在没有故障发生时,两个二级缓存模块都能正常工作,每个二级缓存模块负责一半的可缓存地址空间.如果其中一个缓存模块出现故障,则整个系统都不能正常工作.而通过应用SAM机制,可以将整个可缓存地址空间都映射到功能正常的二级缓存模块上,整个系统可以继续正确运行.此时所有一级缓存的请求都会发送到左侧完好的二级缓存模块,其接收的请求数量将会是之前的2倍.这会影响到系统性能,但系统仍能正确运行.

设计SAM机制的关键是将整个可缓存地址空

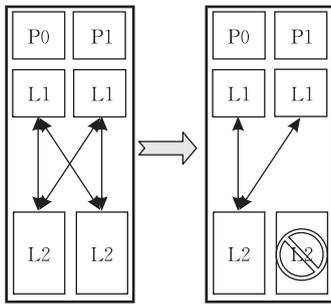


图5 SAM机制示意图

间重新分配给功能正常的缓存模块,从而提高末级缓存系统的可靠性.重新分配可缓存地址空间的关键是确定一个可缓存地址的宿主节点,这样一级缓存发生缺失后,就可以将请求发送到对应的二级缓存模块.因此一级缓存必须知道可缓存地址空间重新分配的方案.

为解决上述问题,SAM在每个一级缓存中增加了一个全局宿主节点配置表(Global Home node Configuration Table,GNCT).该配置表的项数与二级缓存模块数目一样,按照传统地址映射方式所确定的全局宿主节点编号进行索引.每个表项内容为新的全局宿主节点编号.当所有二级缓存模块功能正常时,表项中的内容与索引相同.当有二级缓存模块出现故障时,表项内容更新为经过对可缓存地址空间重新分配后的新宿主节点编号.

图6给出了对16核片上多处理器可缓存地址空间进行重新分配的例子.最初所有二级缓存模块都正常工作,图6(a)中宿主节点配置表中的节点编号也与常规地址映射一样.当图6(b)中有一个二级缓存出现故障后,配置表中的内容被重置,其中第6个表项的宿主节点编号由5变为4.此时编号为4的二级缓存会承受之前约2倍的缓存请求.图6(c)和图6(d)给出二级缓存故障增加时配置表内容的变化情况.考虑到负载平衡,SAM将缺失的可缓存地址空间分布到多个二级缓存中,由于对全局宿主节点配置表的查找并不在缓存访问的关键路径上,对系统的性能不会产生影响.

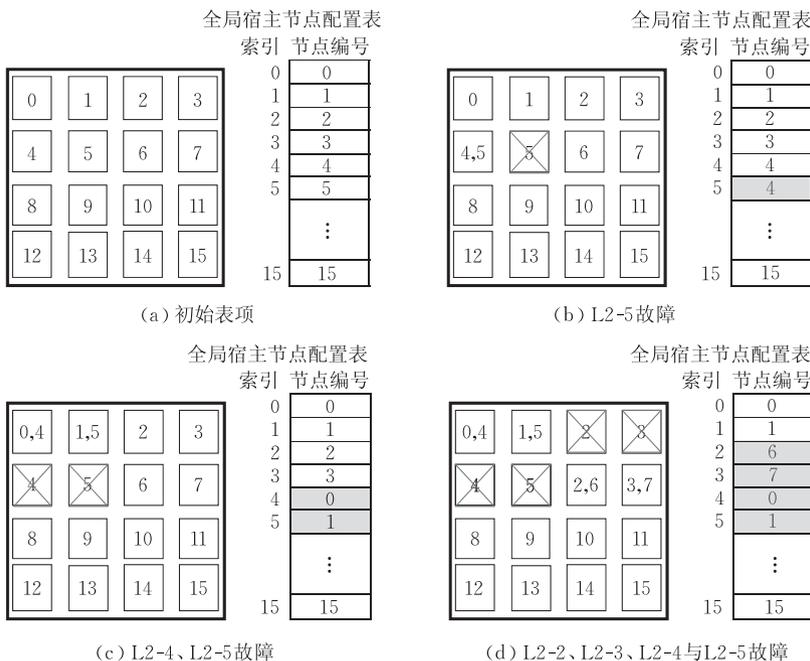


图6 共享方式下的可扩展地址映射

### 4.3 集群方式下的可扩展地址映射

随着片上处理器核数目的增加,共享组织方式下的目录存储开销也呈线性增加.例如当缓存行大小为 64 字节时,一个 512 核 TILE 结构的片上多处理器的目录存储开销与数据存储开销相等.采用集群组织方式可以节省目录存储开销,同时降低数据的访问延迟.集群组织方式具有较好的可扩展性,是设计未来众核存储系统的一种选择.在集群方式下,末级缓存被划分为多个区域,每个区域内的二级缓存可组成一个完整的可缓存地址空间.在一级缓存发生缺失时,首先向区域宿主节点发送请求,如果区域宿主节点缺失再向全局宿主节点发送请求.集群方

式下区域宿主节点的确定通常采用常规地址映射方法.对任意一个可缓存数据,其区域宿主节点和全局宿主节点(可能会映射到同一缓存模块)都是确定的.

为在集群方式下实现容错机制,SAM 在一级缓存中增加了一个区域宿主节点配置表(Regional home Node Configuration Table, RNCT),其结构与全局宿主节点配置表类似,只是每个表项中的编号是区域宿主节点.在集群方式下,全局宿主节点不再由一级缓存直接访问,而是通过二级缓存中的区域宿主节点访问,因此将全局宿主节点配置表移到二级缓存中.集群方式下 SAM 的具体设计如图 7 所示.

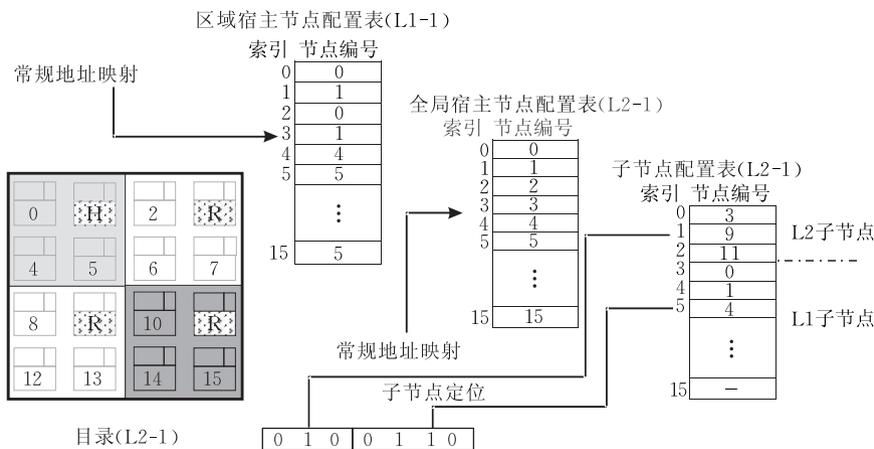


图 7 集群方式下的可扩展地址映射实现

如果一个区域中的故障二级缓存过多,该区域的缓存性能会受到较大影响. SAM 提供重新调整区域大小的机制来减少这种影响,该机制的伪代码在图 8 中给出.该算法可以调整一个区域内的二级缓存模块数目.图 9 给出了两个例子来说明对区域大小的重新调整.在图 9(a)中,只有 L2-1 出现故障,此时 SAM 将其地址空间重新分配到 L2-0 上,整个系统仍然由 4 个区域组成.而在图 10(b)中,共有 4 个二级缓存出现故障,此时 SAM 将区域数目由 4 个减少到 2 个.通过动态调整区域规模, SAM 可以避免缓存请求过多地集中到某个二级缓存,同时也能保证系统性能的平滑下降.

由于每个区域内包含的二级缓存和一级缓存节点会在调整后发生改变,同时一个地址对应的区域宿主节点也会改变.此时在全局宿主节点需要记录每个区域宿主节点的位置,同时区域宿主节点也需要记录每个一级缓存子节点的位置.为记录这些信息, SAM 在每个二级缓存中增加子节点配置表(Sub Node Configuration Table, SNCT). SNCT 由两部分组成,分别用来记录区域宿主节点的位置和

一级缓存子节点的位置. SNCT 依照节点在目录中的位置来索引,表项中的值则是子节点的具体编号,同时每个表项中增加 1 位以标记对应编号是区域宿主节点还是一级缓存子节点.

### 4.4 可扩展地址映射的硬件开销

共享组织方式下, SAM 设计的硬件开销仅为全

```

Begin
{
    实时检测每个二级模块并返回结果( $N_{failure}$ );
    if ( $N_{failure} = 0$ ) {
        return;
    } else if ( $N_{failure} < \text{预定义的阈值}(N_{threshold})$ ) {
        for each 故障二级缓存
            将相应地址空间映射到相同区域内的完好二级缓存;
    } else {
        选择新的区域大小;
        为每个二级缓存分配区域;
        for each 故障二级缓存
            将相应地址空间映射到相同区域内的完好二级缓存;
    }
}
End

```

图 8 重组算法的伪代码

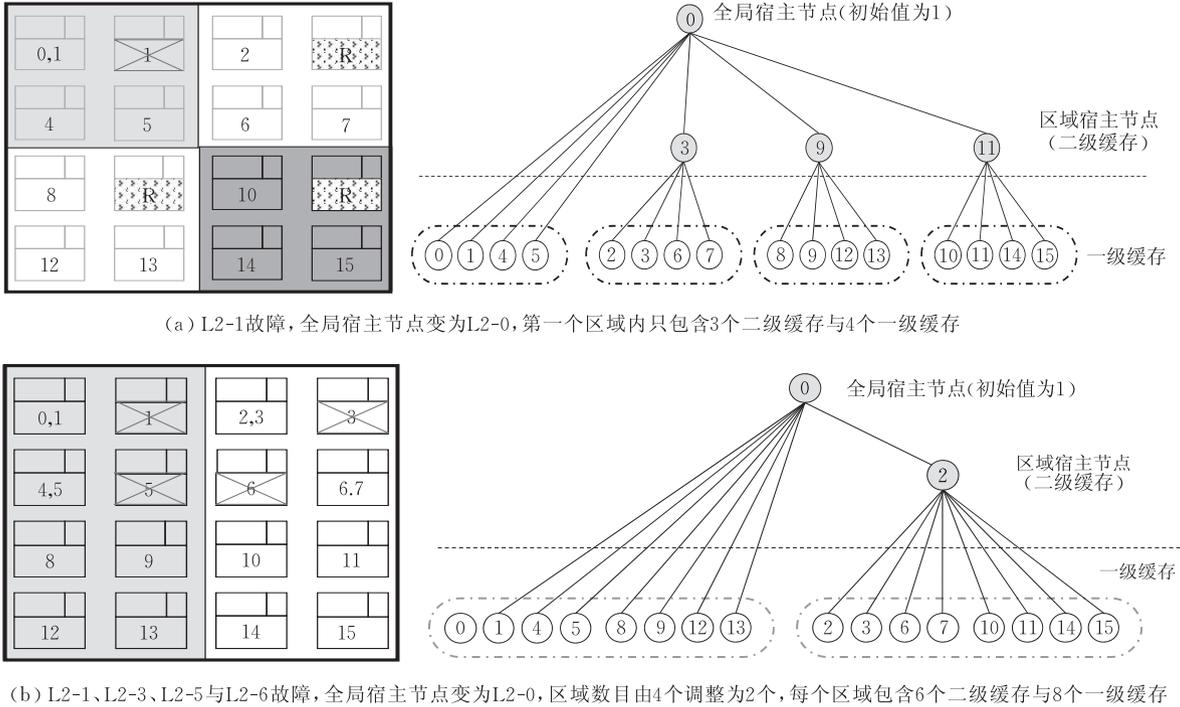


图 9 集群方式下 SAM 自动调整示例

局宿主节点配置表. 配置表项数与二级缓存数目相同, 每一个表项需要  $\log_2 n$  位来存储新的全局宿主节点. 共享方式 SAM 的硬件开销是  $n \log_2 n$  位.

集群组织方式下, 需要增加 3 个配置表, 即全局宿主节点配置表、区域宿主节点配置表与子节点配置表. 前两个配置表的硬件开销在最坏情况下与共享组织方式相同, 而子节点配置表每个表项还需要额外 1 位来标记表项是二级缓存还是一级缓存. 表 1 给出了在每个二级缓存模块大小为 1 MB 时, 不同规模的片上多处理器实现 SAM 机制所需的硬件开销. 在 1024 核时, 集群方式下 SAM(SAM-C) 的硬件开销要小于 1%.

表 1 每个缓存模块为 1 MB 时 SAM-C 的存储开销

节点数目	HNCT+RNCT	SNCT	总开销/位	占二级缓存百分比/%
64	$2 \times 64 \times 6$	$64 \times 7$	1216	0.0145
128	$2 \times 128 \times 7$	$128 \times 8$	2816	0.0336
256	$2 \times 256 \times 8$	$256 \times 9$	6400	0.0763
512	$2 \times 512 \times 9$	$512 \times 10$	14336	0.1709
1024	$2 \times 1024 \times 10$	$1024 \times 11$	31744	0.3874
2048	$2 \times 2048 \times 11$	$2048 \times 12$	69632	0.8301

## 5 实验评测

### 5.1 实验设置

为验证可扩展地址映射方法的效果, 我们利用

Simics<sup>①</sup> 和 GEMS<sup>[25]</sup> 建立实验环境. Simics 是一个事件驱动的全系统模拟平台. Wisconsin 大学的 GEMS 在 Simics 的基础之上实现了完整的片上存储层次, 为片上存储层次研究提供了平台.

表 2 中给出了片上多处理器系统的配置, 二级缓存与一级缓存的数据是全包含关系, 片上网络中传输的是所有维护一致性所需的消息. 为评估 SAM 的效果, 本文分别将共享方式与集群方式作为对比的基准系统, 同时分别用 1 个或 8 个二级缓存模块出错时的系统性能与缓存性能作为对比. 表 3 给出了不同方案的简写及其描述.

表 2 系统配置参数

部件名称	配置参数
CMP 规模	64 核
缓存行大小	64 字节
L1 I-Cache 大小/相连度	16 KB/2 路组相连
L1 D-Cache 大小/相连度	16 KB/2 路组相连
L1 延迟	2 周期
L1 替换策略	伪 LRU
L2 Cache 大小/相连度	256 KB/(16 路/TILE)
L2 延迟	15 周期
L2 替换策略	伪 LRU
网络拓扑	$8 \times 8$ Mesh
每跳延迟	3 周期
片外存储延迟	300 周期

① <https://www.simics.net/>

表 3 实验方案名称与描述

方案名称	方案描述
BASE	共享组织方式
SAM-1	SAM 方式,有 1 个二级缓存出错
SAM-8	SAM 方式,有 8 个二级缓存出错
BASE-C	集群组织方式
SAM-C1	SAM-C 方式,有 1 个二级缓存出错
SAM-C8	SAM-C 方式,有 8 个二级缓存出错

实验选择的基准测试程序来自 SPLASH-2<sup>[26]</sup>和 PARSEC<sup>[27]</sup>测试程序集,这些程序覆盖科学计算及通用数据处理.表 4 给出了本文选择的测试程序及问题规模.本文采用执行时间作为系统性能的标准,并用平均片上延迟作为评价缓存性能的标准.

表 4 测试程序描述

负载名称	问题规模
fft	256 K points
lu	1024×1024 matrix, 16×16 blocks
radix	1048576 numbers, 1024 radix
ocean	1024×1024 ocean
fmm	16 k particles
volrend(vol.)	head
barnes(barn.)	16 k particles
cholesky(chol.)	Tkl5.o
bodytrack(bod.)	Simsmall
fluidanimate(flui.)	Simsmall
x264	Simsmall
vips	Simsmall
swaption.swap.)	Simsmall

## 5.2 共享方式下 SAM 的性能

图 10 给出了 SAM-1 和 SAM-8 方案下整个系

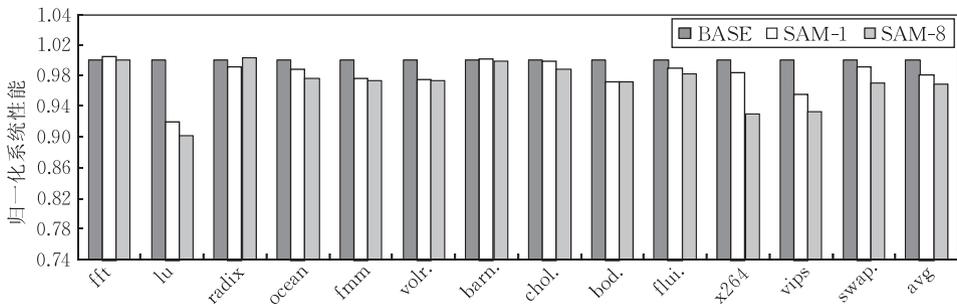


图 10 共享组织方式下的系统性能

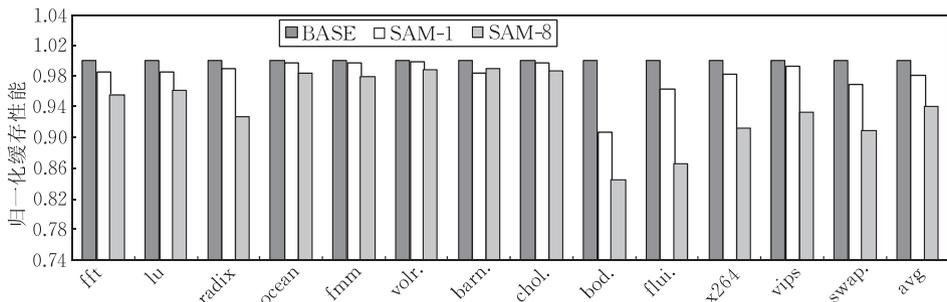


图 11 共享组织方式下的缓存性能

统与基准系统 BASE 相对比的性能.在 SAM-8 方案中,系统性能下降最大的程序为 lu,下降了约 10%,其它程序的性能下降幅度都低于 7%.SAM-1 和 SAM-8 的系统性能平均分别下降了 1.9% 和 3.1%.图 11 给出了 SAM-1 和 SAM-8 的缓存性能,其中 SAM-8 在 bodytrack 程序上缓存性能下降最大,性能下降达 15%.SAM-1 和 SAM-8 的系统性能平均分别下降了 2.0% 和 5.9%.

## 5.3 集群方式下 SAM 的性能

图 12 给出了 SAM-C1 和 SAM-C8 方案下整个系统与基准系统 BASE 相对比的性能.在 SAM-8 方案中,系统性能下降最大的程序为 volrend,下降了约 9%.SAM-1 和 SAM-8 的系统性能平均分别下降了 1.0% 和 2.7%.图 13 给出了 SAM-1 和 SAM-8 的缓存性能.其中 SAM-8 在 fluidanimate 程序上缓存性能下降最大,达 12%.SAM-1 和 SAM-8 的系统性能平均分别下降了 1.9% 和 4.3%.

## 5.4 敏感性分析

为研究重建可缓存地址空间对缓存系统性能的影响,选择 fft 和 radix 程序测试从 1 个二级缓存故障直到 32 个二级缓存故障时的缓存系统性能.图 14 和图 15 分别给出了共享方式与集群方式下缓存系统的相对性能.在共享方式下,fft 与 radix 的缓存性能分别平滑下降到了基准系统的 69% 和 80%.在集群方式下,最初系统是由 16 个区域组成,每个区域内包含 4 个节点.当出故障的二级缓存数目超

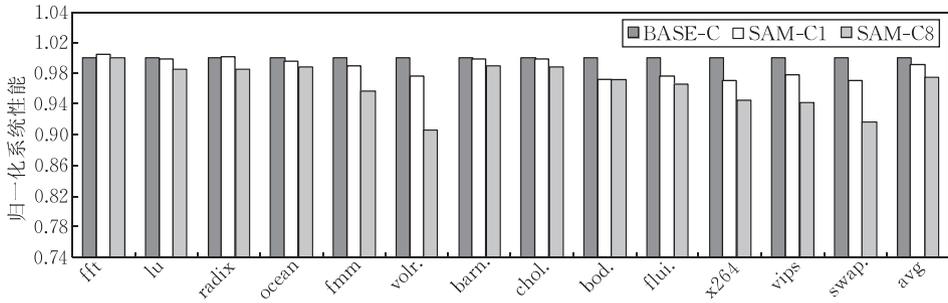


图 12 集群组织方式下的系统性能

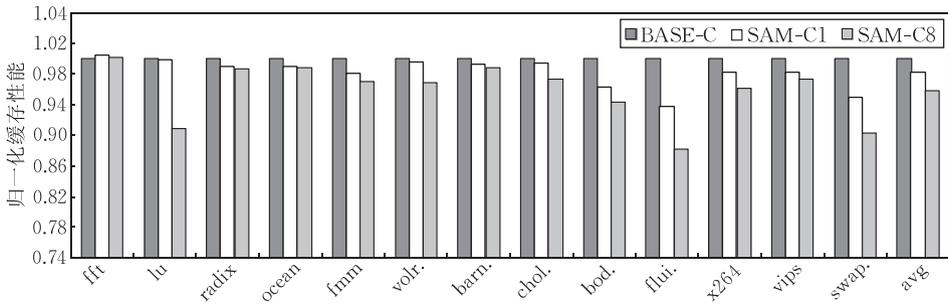


图 13 集群组织方式下的缓存性能

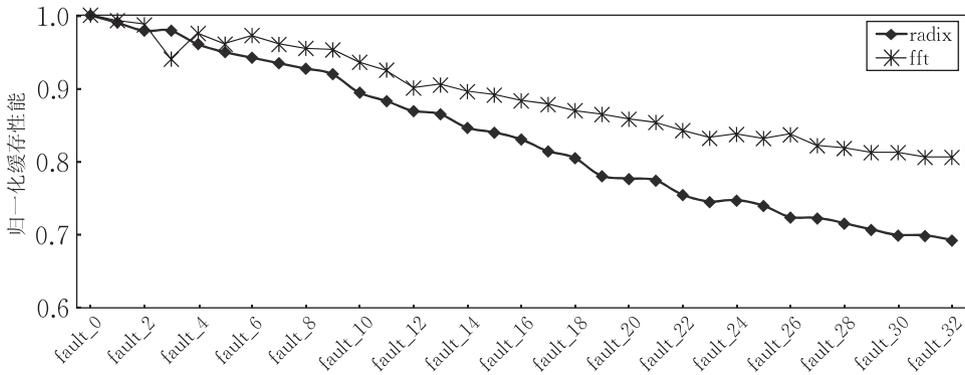


图 14 共享组织方式下缓存性能下降曲线

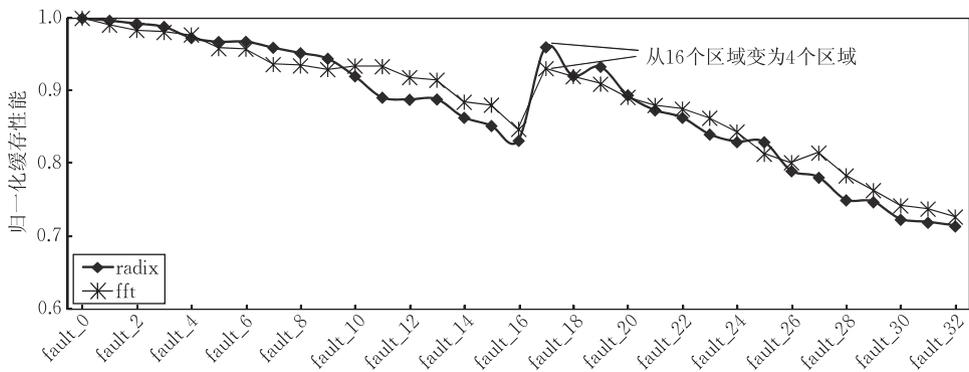


图 15 集群组织方式下缓存性能下降曲线

过 16 时, SAM-C 将系统动态调整为 4 个区域, 每个区域包含 16 个节点, 从图 15 中可以看出, 当二级缓存故障数目达到 17 个时, 缓存系统的性能出现了波动, 这说明了动态调整片上区域划分方法的效果。

## 6 结 论

片上多处理器的片上末级缓存通常采用 NUCA

结构,即使单个缓存模块的出错概率较低,当 NUCA 中包含成百上千个缓存模块时,整个缓存出错的概率也较为可观.本文提出可扩展地址映射(SAM)方法,目的是提高片上末级存储系统的可靠性.SAM 在不十分可靠的缓存模块基础上构建了高度可靠的片上存储系统.SAM 的硬件开销较小,由 1024 个 1MB 模块组成的二级缓存,实现 SAM 所需的硬件开销小于 1%.SAM 能够在相当数量的二级缓存出现故障时保证系统的正确运行.

SAM 机制可以应用在共享缓存组织方式和集群缓存组织方式下.在集群方式下,SAM 可以通过动态算法选择一个适当的区域大小.模拟实验表明,在两种组织方式下,SAM 都能够实现系统性能的平滑下降.SAM 可以与其它提高系统可靠性的方案共同工作.SAM 对提高将来大规模众核处理器存储系统可靠性是一个可行的设计方案.

## 参 考 文 献

- [1] Borkar S. Thousand core chips: A technology perspective// Proceedings of the 44th Annual Design Automation Conference. San Diego, USA, 2007: 746-749
- [2] Chishti Z, Alameldeen A R, Wilkerson C, Wu W, Lu S L. Improving cache lifetime reliability at ultra-low voltages// Proceedings of the 42nd International Symposium on Microarchitecture. New York, USA, 2009: 89-99
- [3] Manoochehri M, Annavam M, Dubois M. CPPC: Correctable parity protected cache// Proceedings of the 38th Annual International Symposium on Computer Architecture. San Jose, USA, 2011: 223-234
- [4] Ansari A, Gupta S, Feng S, Mahlke S. ZerehCache: Armoring cache architectures in high defect density technologies// Proceedings of the 42nd International Symposium on Microarchitecture. New York, USA, 2009: 100-110
- [5] Ansari A, Gupta S, Feng S, Mahlke S. Maximizing spare utilization by virtually reorganizing faulty cache lines. IEEE Transactions on Computers, 2011, 60(1): 35-49
- [6] Kim C, Burger D, Keckler S W. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches// Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems. San Jose, USA, 2002: 211-222
- [7] Wilkerson C, Gao H, Alameldeen A R, Chishti R, Khellah M, Lu S L. Trading off cache capacity for reliability to enable low voltage operation// Proceedings of the 35th Annual International Symposium on Computer Architecture. Beijing, China, 2008: 203-214
- [8] Kulkarni J P, Kim K, Roy K. A 160 mV, fully differential, robust cchmitt trigger based sub-threshold SRAM// Proceedings of the 2007 International Symposium on Low Power Electronics and Design. Portland, USA, 2007: 171-176
- [9] Sridharan V, Asadi V, Tahoori M B, Kaeli D. Reducing data cache susceptibility to soft errors. IEEE Transactions on Dependable and Secure Computing, 2006, 3(4): 353-364
- [10] Hamming R. Error-detecting and error-correcting codes. The Bell System Technical Journal, 1950, 29(1): 147-160
- [11] Agostinelli M et al. Erratic fluctuations of SRAM cache Vmin at the 90 nm process technology node// Proceedings of the 2005 IEEE International Electron Devices Meeting. Washington, USA, 2005: 655-658
- [12] Borkar S. Designing reliable systems from unreliable components: The challenges of transistor variability and degradation. IEEE Micro, 2005, 25(6): 10-16
- [13] Agarwal A, Paul B C, Mahmoodi H, Datta A, Roy K. A process-tolerant cache architecture for improved yield in nanoscale technologies. IEEE Very Large Scale Integration (VLSI) Systems, 2005, 13(1): 27-38
- [14] Lee H, Cho S, Childers B R. Perfectory: A fault-tolerant directory memory architecture. IEEE Transactions on Computers, 2010, 59(5): 638-650
- [15] Liang X, Canal R, Wei G Y, Brooks D. Replacing 6T SRAMs with 3T1D DRAMs in the L1 data cache to combat process variability. IEEE Micro, 2008, 20(1): 60-68
- [16] Kim J, Hardavellas N, Mai K, Falsafi B, Hoe J. Multi-bit error tolerant caches using two-dimensional error coding// Proceedings of the 40th Annual International Symposium on Microarchitecture. Chicago, USA, 2007: 197-209
- [17] Aichelmann F J. Fault-tolerant design techniques for semiconductor memory applications. IBM Journal of Research and Development, 1984, 28(2): 177-183
- [18] Lee J, Lee Y J, Kim Y B. SRAM word-oriented redundancy methodology using built in self-repair// Proceedings of the 2004 IEEE International SOC Conference. Santa Clara, USA, 2004: 219-222
- [19] Ozdemir S, Sinha D, Memik G, Adams J, Zhou H. Yield-aware cache architectures// Proceedings of the 39th International Symposium on Microarchitecture. Orlando, USA, 2006: 15-25
- [20] Roberts D, Kim N S, Mudge T. On-chip cache device scaling limits and effective fault repair techniques in future nanoscale technology// Proceedings of the 10th EuroMicro Conference on Digital System Design. Lübeck, Germany, 2007: 570-578
- [21] Huh J, Kim C, Shafi H, Zhang H, Burger D, Keckler W. A NUCA substrate for flexible CMP cache sharing// Proceedings of the 19th Annual International Conference on Supercomputing. Massachusetts, USA, 2005: 31-40
- [22] Hammoud M, Cho S, Melhem R. Dynamic cache clustering for chip multiprocessors// Proceedings of the 23rd Annual International Conference on Supercomputing. Yorktown Heights, USA, 2009: 56-67

- [23] Dybdahl H, Stenstrom P. An adaptive shared/private NUCA cache partitioning scheme for chip multiprocessors// Proceedings of the 13th International Symposium on High Performance Computer Architecture. Phoenix, USA, 2007; 2-12
- [24] Marty M R, Hill M D. Virtual hierarchies to support server consolidation// Proceedings of the 34th Annual International Symposium on Computer Architecture. San Diego, USA, 2007; 46-56
- [25] Martin M M, Sorin D J, Beckmann B M, Marty M R, Xu M, Alameldeen A R, Moore K E, Hill M D, Wood D A. Multifacet's general execution-driven multiprocessor simula-

tor (GEMS) toolset. Computer Architecture News, 2005, 33(4): 92-99

- [26] Woo S C, Ohara M, Torrie E, Singh J P, Gupta A. The SPLASH-2 programs: Characterization and methodological considerations. Computer Architecture News, 1995, 23(2): 24-36
- [27] Bienia C, Kumar S, Clara S, Singh J P, Li K. The PARSEC benchmark suite: Characterization and architectural implications// Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques. Toronto, Canada, 2008; 272-281



**LI Chong-Min**, born in 1973, Ph.D., assistant professor. His research interests focus on multiprocessor architecture.

**WANG Dong-Sheng**, born in 1966, Ph. D., professor, Ph.D. supervisor. His research interests include computer

architecture, high performance computing, storage and file systems and network security.

**WANG Hai-Xia**, born in 1977, Ph. D., associate professor. Her research interests include high performance computing, multiprocessor architecture and formal check.

**XUE Yi-Bo**, born in 1967, Ph. D. professor. His research interests include computer network, information security, parallel processing and distributed system.

## Background

As more processor cores can be integrated on one die, in the near future, a Chip Multiprocessor (CMP) may contain hundreds or thousands of cores. On-chip memory will occupy more than 60% of the die area, which have a considerable impact on CMP reliability.

Many cache failures come from particle strikes (soft errors) as well as erratic bit failures. Soft errors are transient and recoverable, erratic bit failures are unrecoverable. Both kinds of errors strongly depend on process parameters. The smaller of the feature size, the higher of the exponential increasing fault probability.

Previous techniques provide fault tolerance for either soft errors or hard faults within an L1/L2 module. However, future CMP may contain hundreds or even thousands of cache units. Even if the error rate of single unit is very low, there is a remarkable failure probability that at least one cache unit can fail.

In NUCA architecture, the cacheable address space is composed of all cache modules. When an LLC cache module gets a hard failure, it destroys the integrity of the whole cacheable address space which can affect the correctness of

the system. Despite recent advances aimed at improving the efficiency of single cache module, little attention has been paid to reliability of multiple LLC modules as a whole.

In this paper, we propose Scalable Address Mapping (SAM) method which can perform a flexible address mapping of the cacheable address space at large scale. When there is a hard fault in an LLC module, SAM can reconstruct the cacheable address space by reassigning the missing address space to other functional modules. SAM can greatly improve the reliability of LLC. In extreme cases, SAM can guarantee that the system functions until all LLC modules except one fail. Both shared and cluster organizations can use SAM. A reorganization algorithm is proposed to dynamically choose an appropriate cache region size and to achieve graceful performance degradation. SAM is orthogonal to other fault-tolerant cache techniques to extend the lifetime of LLC.

The authors acknowledge the support of the National Natural Science Foundation of China (Grant No. 60833004 and No. 60970002), and the National High Technology Research and Development Program (863 Program) of China (No. 2012AA010905).