

# 面向 MLC STT-RAM 的寄存器分配策略优化研究

倪园慧<sup>1</sup> 陈巍文<sup>1</sup> 王 磊<sup>1</sup> 邱柯妮<sup>1,2</sup>

(首都师范大学信息工程学院 北京 100048)<sup>1</sup> (北京成像技术高精尖创新中心 北京 100048)<sup>2</sup>

**摘 要** 多级自旋转移力矩磁性存储器(MLC STT-RAM)是一种新型的非易失性存储介质。不同于采用电荷方式来存储信息的 SRAM,MLC STT-RAM 利用自旋偏振电流通过磁隧道结(MTJ)改变自由层的磁层方向来存储信息,能够天然地避免电磁干扰。文章利用 MLC STT-RAM 的抗电磁辐射特性,探索在航天抗辐照环境下将其作为存储介质用于寄存器设计。在 MLC STT-RAM 中,每个存储单元有 4 种不同的阻抗状态,不同的阻抗状态之间的转换具有不同的能耗和延迟的代价。而传统的基于 SRAM 的寄存器分配技术并没有考虑不同的写状态转换的影响,其在没有考虑溢出优先级的情况下启发式地选择潜在溢出变量,因此该方法不适合用在 MLC STT-RAM 的寄存器分配中。针对该问题,提出了一种面向写状态转换的 MLC STT-RAM 寄存器分配的溢出优化策略。具体来说,首先,通过每个写状态转换频率的线性组合来构成溢出代价模型。然后,根据溢出代价模型针对性地选择溢出变量,选择代价低的变量保存在寄存器中,而代价高的变量倾向于被溢出,从而便实现了面向 MLC STT-RAM 的寄存器分配策略的优化设计。

**关键词** MLC STT-RAM,写状态转换,潜在溢出,寄存器分配

中图法分类号 TP336 文献标识码 A

## Optimization of Register Allocation Strategy for MLC STT-RAM

NI Yuan-hui<sup>1</sup> CHEN Wei-wen<sup>1</sup> WANG Lei<sup>1</sup> QIU Ke-ni<sup>1,2</sup>

(College of Information Engineering, Capital Normal University, Beijing 100048, China)<sup>1</sup>

(Beijing Advanced Innovation Center for Imaging Technology, Beijing 100048, China)<sup>2</sup>

**Abstract** Multi-level cell spin-transfer torque random access memory (MLC STT-RAM) is a promising nonvolatile memory technology. Unlike the SRAM that uses a charge mode to store information, MLC STT-RAM uses the spin polarization current to change the magnetic layer direction of the free layer through the magnetic tunneling junction (MTJ) to store information, so it can naturally avoid electromagnetic interference. This paper used the anti-electromagnetic radiation characteristics of MLC STT-RAM, and explored it as a register for its natural immunity to electromagnetic radiation in rad-hard space environment. MLC STT-RAM exhibits unbalanced write-state transitions due to the fact that the magnetization directions of hard and soft domains cannot be flipped. This feature leads to nonuniform costs of write-states in terms of latency and energy. However, current SRAM-targeting register allocations do not have a clear understanding of the impact of the different write-state transition costs. As a result, those approaches heuristically select variables to be spilled without considering the spilling priority imposed by MLC STT-RAM. Aiming to address this limitation, this paper proposed a state-transition aware spilling cost minimization (SSCM) policy to save power when MLC STT-RAM is employed in register design. Specifically, the spilling cost model is first constructed according to the linear combination of different state transition frequencies. Directed by the proposed cost model, the compiler picks up spilling candidates with the highest cost to achieve lower power and higher performance.

**Keywords** MLC STT-RAM, Write-state transition, Potential spilling, Register allocation

## 1 引言

随着微电子技术的飞速发展,自旋转移力矩磁性存储器(STT-RAM)、相变存储器(PCM)、磁畴壁内存(DWM)、闪存(Flash Memory)<sup>[1]</sup>等新型的存储材料越来越受到学术界和工业界的关注。特别是 STT-RAM 具有良好的抗电磁辐射特性,使其在航天嵌入式系统设计中备受青睐<sup>[2]</sup>。在航天辐

照环境下,采用以电荷方式存储数据的静态随机存取存储器(SRAM)作为寄存器时极易受到干扰,造成数据紊乱。而 STT-RAM 通过利用自旋偏振电流的磁隧道结(MTJ)改变自由层的方向来存储信息,能够避免电磁辐射引起的不良后果,提供了可靠性保障<sup>[3-4]</sup>。文献[5-6]发现,STT-RAM 的写延迟可以大大减少,使得写延迟问题不再成为替代 SRAM 作为寄存器介质的一个瓶颈,这为 STT-RAM 作为寄存器提供了

本文受国家自然科学基金(61502321),北京市教委一般项目(KM201710028016),北京市大学生“实培计划”资助基金资助。

倪园慧(1992—),女,硕士生,CCF 会员,主要研究方向为嵌入式系统及新型非易失存储器体系结构;陈巍文(1993—),男,硕士生,CCF 会员,主要研究方向为嵌入式系统及新型非易失存储器体系结构;王 磊(1995—),男,主要研究方向为嵌入式系统及新型非易失存储器体系结构;邱柯妮(1978—),女,博士,副研究员,主要研究方向为嵌入式系统及新型非易失存储器体系结构。

有力的性能保障。本文以 MLC STT-RAM 寄存器与非易失性内存为介质建立了一个完整的抗辐射存储体系结构,主要研究如何基于 MLC STT-RAM 介质进行有效的变量寄存器分配。

寄存器分配是一种十分重要的编译优化技术。传统寄存器分配通常基于图着色模型。在代码选择阶段生成的汇编指令通常不直接涉及物理寄存器,而是假定有无限数量的虚拟寄存器。寄存器分配问题是将这些虚拟寄存器映射到有限的物理寄存器,在不能完成时选择被溢出节点,执行溢出过程。由于 SRAM 的写操作不存在差异性,传统的基于 SRAM 的寄存器分配器不需要考虑不同写状态转换对程序的影响,而在没有溢出优先级的情况下启发式地选择潜在的溢出节点<sup>[7-8]</sup>。如果将传统的方法直接应用于具有不平衡的写状态转换的 MLC STT-RAM 寄存器中,将大大限制寄存器分配的有效性。

针对这一问题,本文提出了面向 MLC STT-RAM 非均匀写转换的寄存器溢出优化策略。其基本思路是把写转换代价低的变量保留在寄存器中,将写转换代价高的变量溢出到内存中。具体来说,首先,根据写操作的不同状态转换频率来构建溢出代价模型;然后根据代价模型,推导潜在溢出优先级。最后选择优先级高的变量进行潜在溢出。本文的创新点如下:

- 1) 提出了面向 MLC STT-RAM 非均匀写转换的寄存器分配的溢出优化策略。
- 2) 建立了溢出代价模型来计算每个潜在溢出节点的功耗溢出代价。
- 3) 提出了面向不同写转换的寄存器分配溢出优化算法。
- 4) 通过实验验证了上述方法的有效性。实验结果表明,本文所提方法可以减少 19.4% 的功耗和 14.4% 的延迟。

## 2 背景工作

下面主要介绍 MLC STT-RAM 的工作原理以及寄存器分配技术。

### 2.1 MLC STT-RAM 的工作原理

在航天环境中,存在大量的单粒子辐射,通常这些粒子辐射的能量大且穿透力强,极易造成电子元器件的逻辑状态发生单粒子翻转(SEU),从而导致系统功能紊乱,严重时会发生灾难性事故。在航天应用系统中,SEU 已经成为一个不能忽视的问题。传统的存储介质 SRAM 是以电荷的方式来存储数据,在空间辐射环境中会暴露出较强的辐射敏感度,因此迫切需要一种能够抵抗辐射的存储材质。STT-RAM 的出现解决了这一问题。与 SRAM 不同,STT-RAM 利用隧道型巨磁阻效应的物理机理来存储信息,具备优良的抗电磁干扰的特性,可避免太空电磁辐射导致的单粒子翻转等效应,有效保证数据的准确性,极大地降低航天电子设备可靠性设计的复杂度<sup>[3-4]</sup>。

MLC STT-RAM 是指一个存储单元可以保存多比特数据的 STT-RAM。如图 1 所示,MLC STT-RAM 的自由层包含两个区域:硬区域和软区域,本文使用 R 后面的第一位表示硬区域,第二位表示软区域。由于这两个区域有自己独立的磁性方向,因此每个存储单元可以保存 2 比特的数据。如果在两端加一个比较小的电流,则只改变软区域的磁性方向;而若两端加一个比较大的电流,则可以同时改变硬区域和

软区域的磁性方向<sup>[9]</sup>。MLC STT-RAM 的写转换主要包括以下 4 种情况:

- 1) 零转化(ZT):硬区域和软区域均不发生变化,例如  $R00 \leftrightarrow R00, R01 \leftrightarrow R01, R10 \leftrightarrow R10, R11 \leftrightarrow R11$ 。
- 2) 软转换(ST):只有软区域的磁性方向变化,例如  $R00 \leftrightarrow R01$  和  $R10 \leftrightarrow R11$ 。
- 3) 硬转换(HT):软区域和硬区域的磁性方向都变化,例如  $R00 \rightarrow R11, R01 \rightarrow R11, R10 \rightarrow R00$  以及  $R11 \rightarrow R00$ 。
- 4) 两步转换(TT):转换需要两步完成,包括一步硬转换和一步软转换,例如  $R00 \rightarrow R10, R01 \rightarrow R10, R10 \rightarrow R01$  和  $R11 \rightarrow R01$ 。

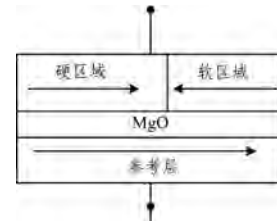


图 1 MLC STT-RAM 的 MTJ 结构

表 1 统计了在 45 nm 工艺下改变 MLC STT-RAM 的 4 种状态所需要的电流大小<sup>[10]</sup>。表中用 R00, R01, R10 和 R11 表示 MLC STT-RAM 的 4 种状态,0 表示低阻抗状态,1 表示高阻抗状态;R 后面的第一位数表示硬区域状态,第二位数表示软区域状态;表中数据表示需要的额定电流大小,因此只要大于表中的电流就能改变 STT-RAM 的状态;负数表示需要反方向的电流,“-”表示两种状态不能直接转变。

表 1 MLC STT-RAM 的转换电流

		(单位: $\mu\text{A}$ )			
转换状态		R00	R01	R10	R11
原始状态	R00	0	-38.3	-	-56.7
	R01	26.3	0	-	-56.7
	R10	66.4	-	0	-9.1
	R11	66.4	-	39.3	0

由表 1 可知,当改变状态或写入不同的变量时,MLC STT-RAM 的电流(能耗)也会相应地发生改变。如果编译器溢出不同能量代价的变量,则这对程序性能的影响具有很大差异。考虑到由这种代价变化所引起的不同影响,本文建立了溢出代价模型,并根据这个模型来选择最佳的潜在溢出节点。

### 2.2 基于图着色的寄存器分配技术

寄存器分配决定了哪些程序变量的值可以存放在哪个或者哪些寄存器中,是编译后端优化中必不可少的任务。如何最大限度地利用寄存器是编译优化的重要问题。现代编译器进行优化时都是假设有无数的虚拟寄存器,但是实际编译过程中物理寄存器的数量是有限的。当计算过程中需要一个寄存器但是所有的寄存器都被占用时,此时正在被使用的寄存器中有一个必须被溢出到内存中。寄存器分配算法有很多种,但是最典型且常用的方法是图着色<sup>[11]</sup>(Graph Coloring)。

图着色问题是一个用于寄存器分配与管理寄存器溢出的系统化的方法,是非常著名的 NP 完全问题之一。其基本结构为冲突图  $G=(V, E)$ ,其中  $V$  是节点集合, $E$  为边集合。其中每一个节点代表一个虚拟寄存器的活跃区间。对于任意两个节点,如果一个节点在另一个被定值的地方是活跃的,即若



$$F(XT) = \sum_j S_{ji}(XT), i, j \in [0, 3] \text{ 且 } i \neq j, XT \in [ZT, ST, HT, TT] \quad (2)$$

由此,每个变量的代价模型就可以被构造为  $F(ZT)$ ,  $F(ST)$ ,  $F(HT)$  和  $F(TT)$  的线性组合,可表示如下:

$$Cost = \alpha F(ZT) + \beta F(ST) + \gamma F(HT) + \delta F(TT) \quad (3)$$

其中,  $Cost$  表示状态代价,  $\alpha, \beta, \gamma$  和  $\delta$  被定义为每个状态转换频率的权重。由于本文主要关注动态能耗,因此权重被定义为不同状态转换的执行能耗的归一化值。状态转换的动态能耗  $E_{XT}$  与每个转换的平均转换电流的平方和脉冲持续时间的乘积成正比,可表示为:

$$E_{XT} \propto I_{write(XT)}^2 \cdot t_{pulse}, XT \in [ZT, ST, HT, TT] \quad (4)$$

其中,  $I_{write(XT)}$  表示每个状态转换  $XT$  所需要的平均转换电流,可以通过表 1 得到。  $t_{pulse}$  表示脉冲持续时间。权重  $\alpha, \beta, \gamma$  和  $\delta$  是通过将  $E_{XT}$  归一化为 0-1 获得。具体来说,将最大的值  $\delta$  设置为 1,  $\alpha, \beta$  和  $\gamma$  表示相对于  $\delta$  的归一化值。

设置完参数以后,可以根据式(3)得到每个节点的状态转换代价。然后根据状态转换代价对度大于或等于  $k$  的节点进行降序排列。最终,选择具有最高代价的几个节点作为溢出候选节点。通过这种方式可以构建溢出代价最小化模型,它能够大大减小寄存器的能耗压力。在这种情况下,由于精确地利用了 MLC STT-RAM 寄存器状态转换的代价信息,在溢出代价模型指导下的分配决策可以实现更优的寄存器分配。

代价模型的建立过程如图 3 所示,其具体的执行过程如算法 1 所示。

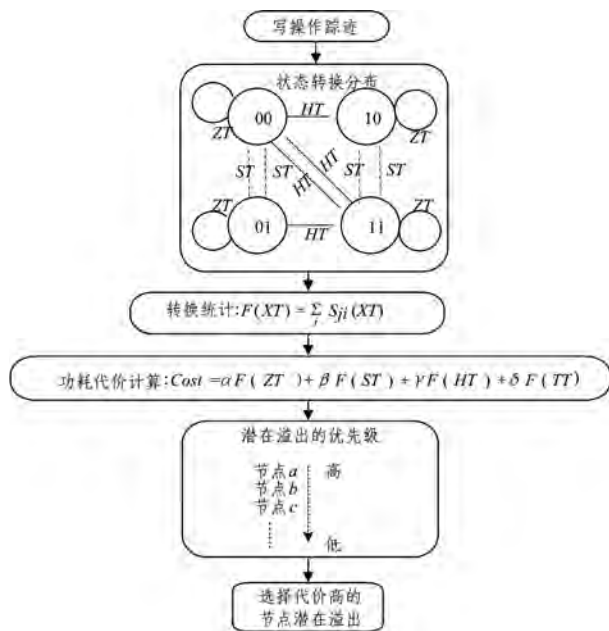


图 3 建立代价模型的步骤

溢出代价模型为选择潜在溢出节点提供了一个基本依据,其把转换能耗代价大的变量保存在内存中而不是寄存器中,避免了昂贵的溢出,进而降低了程序的能耗。

#### 算法 1 溢出代价模型建立算法

1. 收集每个状态的写数量;
2. 根据式(2)得到写状态转换频率;
3. 分析频率得到每个节点(变量)的溢出代价;
4. 按照溢出代价由大到小的顺序对节点排序;
5. 选择具有最高代价的节点进行溢出。

#### 4.3 算法描述

这部分将描述一个面向写状态转换的寄存器分配算法。其基本原则是根据溢出代价来选择最佳的潜在溢出节点,把代价小的节点保存在寄存器中,以缓解寄存器的压力,降低程序执行过程中的能量消耗。本文所提算法主要包括以下 4 个步骤:

1) 连续地从冲突图  $G$  中将度小于或等于  $k$  的节点  $v$  删除并压入栈中,直到所有的节点的度都大于  $k$  或者  $G$  为空图,将其标记为图  $G'$ 。

2) 如果  $G'$  是空图,那么对所有的变量按照节点被删除的相反顺序着色。

3) 如果  $G'$  不是空图,则首先通过程序特征分析得到图中所有节点的写状态转换的频率,然后根据式(3)得到每个节点的溢出代价,并把这些节点按照代价的大小进行降序排列,具有最高代价的节点被标记为潜在的溢出节点。然后分配器为所有的节点着色。

4) 如果潜在溢出节点不能被着色,那么潜在溢出就变为实际溢出。如果潜在溢出变成实际溢出,那么节点的生命期会被修改,整个图着色的过程要从构造开始。

算法 2 展示了此步骤的具体细节。当冲突图中不存在度小于颜色数的节点时,有些节点必须选择潜在溢出。本算法选择代价最大的节点作为潜在的溢出,并且尝试对这个节点着色。如果该节点不能被着色,把这个潜在溢出节点标记为实际溢出。

#### 算法 2 基于溢出代价最小化的寄存器分配算法

1. while  $G$  不是空图 do
2. if 存在度小于或等于  $k$  的变量  $v$  then
3. 删除  $v$
4. else
5. 通过静态程序分析得到  $F$  的频率
6. 按照写代价降序的顺序排列变量
7. 选择代价最高的节点
8. 把  $v$  添加到潜在溢出列表中
9. 删除  $v$
10. end if
11. if 不存在需要溢出的变量 then
12. 以被删除方向相反的顺序对变量着色
13. else
14. 溢出潜在溢出列表中的变量
15. 重建冲突图并再次执行这个程序
16. end if
17. end while

#### 4.4 输入依赖

在基于静态统计信息的优化技术中很重要的问题是输入依赖,即在同一个应用中由一个具体的输入产生的优化是否适合其他的输入<sup>[19]</sup>。实际上本文所提出的优化策略并不依赖于  $F$  的具体值,而依赖于变量代价的排序。只要排序的顺序不变,即使是输入发生变化,选择的最佳的溢出变量也是不变的。

在实验中,本文在不同的输入下测试基准测试程序,并比较了其代价的减少程度。本文的优化策略采用了两种场景:理想情况和实际情况。在理想情况下采用每次都改变输入而

产生的优化结果;在实际情况下只采用一次输入,在这一次输入的基础上建立代价模型,并将这个模型运用到其他情况中。实验表明改变输入对于本文提出的策略优化几乎没有影响,因此这种基于静态信息的代价模型是可行的。

## 5 实验

### 5.1 实验设置

本文从 MLC STT-RAM 的动态能耗、访问延迟与寿命评估 3 个方面进行评价。在 45 nm 工艺技术下,MLC STT-RAM 的体系结构的参数配置如表 3 所列<sup>[20]</sup>。

表 3 MLC STT-RAM 的参数配置

MLC STT-RAM	
读延迟/ns	S:1.25 H:1.63
写延迟/ns	S:7.18 H:14.86
读能量/nJ	S:0.018 H:0.023
写能量/nJ	S:0.087 H:0.14
每一次写对应的 磨损次数	硬区域:ZT,ST,HT 和 TT 分别为 0,0,1,1; 软区域:ZT,ST,HT 和 TT 分别为 0,1,1,2.

本文的测试程序集来自于数字信号处理程序与 Livermore 测试程序。实验的主要步骤为:首先,通过编译优化工具 LLVM(Low Level Virtual Machine)<sup>[21]</sup>得到相应的汇编代码和寄存器的写状态转换的静态统计信息;然后,建立代价模型,从而得到最佳的潜在溢出节点。

### 5.2 实验结果与分析

#### 5.2.1 动态能耗

本文设定 MLC STT-RAM 中每两位转换能量的积累和为总能耗。假设寄存器的大小为 64 位,同一寄存器中的所有位可以同时写入,那么每两位状态转换的数量乘以转换的能量就是寄存器的动态能耗。图 4 给出了在考虑溢出优先级的情况下溢出代价最小化方案(SSCM-MLC)与传统的基于 MLC STT-RAM(C-MLC)的寄存器分配的能耗对比结果。如图 4 所示,对于所有测试程序,wdf 能达到最佳性能,这主要是因为 wdf 存在的溢出情况多,并且大量的溢出选择了能耗大的节点。总体来看,SSCM-MLC 方案比 C-MLC 方案平均节省了 19.4% 的能耗。其主要原因是本文提出的面向 MLC STT-RAM 寄存器分配策略的溢出优化可以在寄存器中保留高能效变量,从而节省了更多的写能耗。

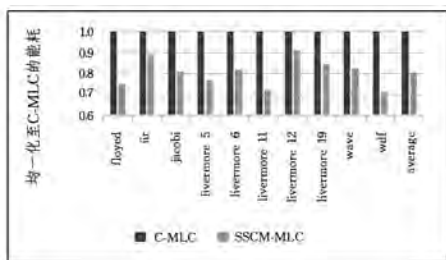


图 4 C-MLC 与 SSCM-MLC 寄存器的能耗评估

#### 5.2.2 访问延迟

延迟的优化计算方式与能耗的优化略有不同。对于每个 64 位的寄存器,寄存器的延迟时间由最慢的状态转换位决定。如图 5 所示,与传统的采用溢出优化技术的 MLC STT-RAM 寄存器方式相比,本文所提方法降低了约 14.4% 的访问延迟。这种延迟性能的增益主要来自于最慢状态变量的溢出。而最慢状态变量又决定了寄存器的访问时间,因此

整体的延迟会大大降低。

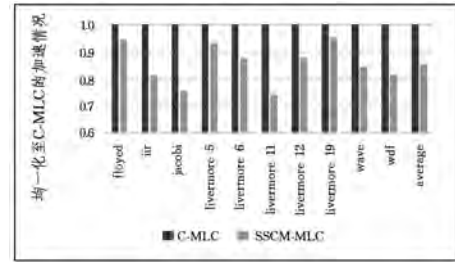


图 5 C-MLC 与 SSCM-MLC 寄存器的访问延迟

#### 5.2.3 寿命评估

对于 MLC STT-RAM 来说,一方面,由于大的写入电流绝缘击穿而导致寄存器的寿命呈指数级下降;另一方面,频繁访问导致寄存器的寿命缩短,因此评估 MLC STT-RAM 的寿命是非常必要的。针对寿命评估,由于寄存器的物理属性相同,因此它们的寿命取决于写入次数(写操作的转换次数)。由于 MLC 单元在同一转换中软硬区域的写次数是不同的,本文在软区域与硬区域中分别进行统计,分别记为 0,0,1 和 1。对于硬区域,ZT,ST,HT 和 TT 操作的转换次数分别为 0,1,1 和 2<sup>[22]</sup>。如图 6 所示,总的转换次数表示软区域和硬区域之和。结果表明,本文提出的 SSCM-MLC 设计比 C-MLC 的转换次数大大降低。具体来说,软区域与硬区域中的转换总数平均减少了 9.35%。且与 C-MLC 设计相比,SSCM-MLC 的寿命提高了 23.2%。这主要是因为本文提出的面向 MLC STT-RAM 寄存器分配策略的溢出优化方案删除了更多的两步状态转换的变量,导致写入总数减少,从而提高了寄存器的寿命。

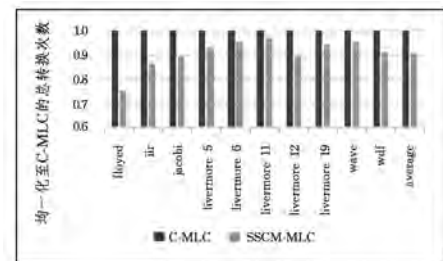


图 6 C-MLC 与 SSCM-MLC 寄存器写操作次数

**结束语** 针对 MLC STT-RAM 中不同的写转换具有能耗差异的特征,本文提出了一种基于 MLC STT-RAM 的寄存器设计的面向写状态转换的溢出成本最小化策略。首先,建立能耗代价模型来计算每个度大于或等于  $k$  的变量的潜在溢出代价;然后,提出了基于代价模型的寄存器分配算法,选择出能耗代价最大的潜在溢出变量。实验结果表明,本文提出的策略可以降低能源消耗和访问延迟,并延长使用寿命。

## 参考文献

- [1] LIANG S, CHUN X, HU J, et al. Write activity reduction on flash main memory via smart victim cache[C]// Great Lakes Symposium on VLSI Systems. 2010:91-94.
- [2] 姚君. STT-RAM 缓存优化策略的研究[D]. 杭州:浙江工业大学, 2014.
- [3] CHABI D, ZHAO W S, JACQUES-OLIVIER K, et al. Design and analysis of radiation hardened sensing circuits for spin trans-

- fer torque magnetic memory and logic[J]. IEEE Transactions on Nuclear Science, 2014, 61(6): 3258-3264.
- [4] HUGHES H, BUSSMANN K, MCMARR P J, et al. Radiation studies of spin-transfer torque materials and devices[J]. IEEE Transactions on Nuclear Science, 2012, 59(6): 3027-3033.
- [5] BISHNOI R, EBRAHIMI M, OBORIL F, et al. Improving write performance for STT-MRAM[J]. IEEE Transactions on Magnetism, 2016, 52(8): 1-11.
- [6] 赵巍胜, 王昭昊, 彭守仲, 等. STT-MRAM 存储器的研究进展[J]. 中国科学: 物理学力学天文学, 2016, 46(10): 107306.
- [7] 董伟. 基于 NVM 的写操作优化策略研究与设计[D]. 济南: 山东大学, 2016.
- [8] 李清安. 面向非易失性片上存储的编译技术研究[D]. 武汉: 武汉大学, 2013.
- [9] CHEN Y R, WANG X B, ZHU W Z, et al. Access scheme of multi-level cell spin-transfer torque random access memory and its optimization[C]// Midwest Symposium on Circuits and Systems. 2010: 1109-1112.
- [10] 何炎祥, 沈凡凡, 张军, 等. 新型非易失性存储器架构的缓存优化方法综述[J]. 计算机研究与发展, 2015, 52(6): 1225-1241.
- [11] CHAITIN, AUSLANDER, CHANDRA, et al. P. W. : Register allocation via graph coloring[J]. Journal of Computer Languages, 1981, 238(16): 265-266.
- [12] COOPER K, DASGUPTA A, et al. Tailoring graph-coloring register allocation for runtime compilation[C]// Code Generation and Optimization. 2006: 39-49.
- [13] 张军超. 相连多寄存器组体系结构上的寄存器分配技术[D]. 中国科学院研究生院(计算技术研究所), 2005.
- [14] 邓宇. 基于图着色的存储层次优化技术研究[D]. 长沙: 国防科学技术大学, 2007.
- [15] 薛丽萍. 基于龙芯 I 的全局寄存器分配研究[D]. 中国科学院研究生院(计算技术研究所), 2004.
- [16] FALK H. WCET-aware register allocation based on graph coloring[C]// Design Automation Conference. 2009: 726-731.
- [17] LOU X H, GAO Z, DIMITORV, et al. Demonstration of multi-level cell spin transfer switching in mgo magnetic tunnel junctions[J]. Applied Physics Letters, 2008, 93(24): 242-502.
- [18] ZHAO M Y, XUE Y, YANG C M, et al. Minimizing MLC PCM write energy for free through profiling-based state remapping[C]// Asia and South Pacific Design Automation Conference. 2015: 502-507.
- [19] ZHAO M Y, XUE Y, HU J T, et al. State asymmetry driven state remapping in phase change memory[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 36(1): 27-40.
- [20] LIU X X, MAO M J, BI X Y, et al. An efficient stt-ram-based register file in gpu architectures[C]// Asia and South Pacific Design Automation Conference. 2015: 490-495.
- [21] LATTNER C, ADVE V. LLVM: A compilation framework for lifelong program analysis & transformation[C]// Code Generation and Optimization. 2004: 75-86.
- [22] LUO H Z, HU J T, SHI L, et al. Two-step state transition minimization for lifetime and performance improvement on MLC STT-RAM[C]// Design Automation Conference. 2016: 1-6.
- (上接第 548 页)
- [7] ANDRIKOPOULOS V, BENBERNOU S, PAPAZOGLOU M P. Managing the Evolution of Service Specifications[C]// International Conference on Advanced Information Systems Engineering. Springer-Verlag, 2008: 359-374.
- [8] RINDERLE S, REICHERT M, DADAM P. Correctness Criteria for Dynamic Changes in Workflow Systems- A Survey[J]. Data & Knowledge Engineering, 2004, 50(1): 9-34.
- [9] WEIDLICH M, WESKE M, MENDLING J. Change Propagation in Process Models Using Behavioural Profiles[C]// IEEE International Conference on Services Computing. IEEE, 2009: 33-40.
- [10] WEIDLICH M, MENDLING J, WESKE M. Propagating Changes between Aligned Process Models [J]. Journal of Systems and Software, 2012, 85(8): 1885-1898.
- [11] GUPTA C, SINGH Y, SINGH C D. A Dynamic Approach to Estimate Change Impact Using Type of Change Prpagation [J]. Journal of Information Processing Systems, 2010, 6(4): 597-608.
- [12] EKANAYAKE C C, ROSA M L, HOFSTED E A H M T, et al. Fragment-Based Version Management for Repositories of Business Process Models[M]// On the Move to Meaningful Internet Systems: OTM 2011. Springer Berlin Heidelberg, 2011: 20-37.
- [13] KURNIAWAN T A, GHOSE A K, DAM H K. Relationship Preserving Change Propagation in Process Ecosystems[C]// International Conference on Service-Oriented Computing. Springer-Verlag, 2012: 63-78.
- [14] ALAM K A, AHMAD R B, AKHUNZADA A, et al. Impact analysis and change propagation in service-oriented enterprises: A systematic review[J]. Information System, 2015(54c): 43-73.
- [15] 吴哲辉. Petri 导论[M]. 北京: 机械工业出版社, 2006, 4: 1-27.
- [16] SERGEY S, MATTHIAS W, JAN M. Business Process Model Abstraction Based on Synthesis From Well-Structured Behavioral Profiles[J]. International Journal of Cooperative Information Systems, 2012, 21(1): 55-83.
- [17] WEIDLICH M, MENDLING J, WESKE M. Efficient Consistency Measurement Based on Behavioral Profiles of Process Models [J]. IEEE Transactions on Software Engineering, 2010, 37(3): 410-429.
- [18] WANG Y, YANG J, ZHAO W. Change Impact Analysis for Service Based Business Processes[C]// IEEE International Conference on Service-oriented Computing and Applications. IEEE, 2010: 1-8.
- [19] HAO W J, FANG X W. The Approach of Analyzing the Smallest Change Domain in Process Models Based on Petri Nets [J]. Computer Science, 2012, 39(Z11): 76-78.