

面向访问需求的数据缓存泄漏功耗管理方法

王箫音, 佟冬, 孙含欣, 程旭

(北京大学微处理器研究开发中心, 北京 100871)

摘要: 本文提出面向访问需求的数据缓存泄漏功耗管理方法, 根据访存指令对数据缓存的访问需求控制数据缓存的活动. 当流水线中未发现访存指令时, 将整个数据缓存保持在非活跃状态; 而当发现访存指令进入流水线时, 采用两种数据缓存访问控制策略以及对这两种策略的动态选择机制, 在流水线早期捕获访存地址的访问需求, 对数据缓存的活动作出精细控制. 实验结果表明, 在平均情况下, 本文方法将数据缓存的泄漏功耗降低 85.4%, 而处理器性能提升 4.41%, 比传统方法在功耗与性能方面均达到更优结果.

关键词: 嵌入式处理器; 数据缓存; 泄漏功耗

中图分类号: TP302.7 **文献标识码:** A **文章编号:** 0372-2112(2009)02-0362-05

An On-Demand Mechanism for Data Cache Leakage Power Management

WANG Xiaoyin, TONG Dong, SUN Hanxin, CHENG Xu

(Microprocessor Research and Development Center of Peking University, Beijing 100871, China)

Abstract: In this paper, we propose an on-demand mechanism for data cache leakage power management, which manages data cache activities according to the demand of memory accessing instructions. Specifically, this mechanism keeps the whole data array in leakage saving mode whenever it finds no memory accessing instructions at all; once a load instruction is detected, it employs two data cache access control policies and the dynamic selection scheme to capture the access demand of the load address early in the pipeline. Experimental results demonstrate that the data cache leakage power is reduced by an average of 85.4%. Meanwhile, the performance is increased by 4.41%. Compared to traditional methods, the mechanism proposed in this paper achieves better results in both power and performance.

Key words: embedded processor; data cache; leakage power

1 引言

在嵌入式处理器中, 高速缓存已成为最主要的能耗部件之一. 在 Strong ARM SA110 中, 高速缓存占据处理器约 43% 的能耗^[1]. 随着集成电路设计进入超深亚微米阶段, 泄漏功耗在总功耗中所占比重日益增大^[2], 如果不加以控制, 将占据高速缓存总功耗的 60% 以上^[3], 不仅制约处理器性能的进一步提升, 而且影响芯片的可靠性. 因此, 有效的高速缓存泄漏功耗管理方法对于嵌入式处理器的低功耗设计至关重要.

本文针对典型嵌入式处理器提出面向访问需求的数据缓存泄漏功耗管理方法. 该方法根据访存指令对数据缓存的访问需求控制数据缓存的活动: 当流水线中未发现访存指令时, 将整个数据缓存保持在非活跃状态, 以获得大幅度的泄漏功耗节省; 而当发现访存指令进入流水线时, 采用两种数据缓存访问控制策略以及对这两种策略的动态选择机制, 在流水线早期捕获访存地址对数据缓存的访问需求, 从而预先激活需访问的区域并仅

使该区域在访问期间处于活跃状态, 在有效降低数据缓存泄漏功耗的同时避免性能损失.

2 相关工作

Super-Drowsy^[4]采用门控预充电电路避免不必要的位线预充电过程, 降低高速缓存位线的泄漏功耗. 在降低高速缓存存储单元泄漏功耗的方法中, Cache decay^[5]将一段时间内不被访问的行的供电电压关闭. Drowsy Cache^[3]采用动态电压调节^[6]电路, 在工作模式下使存储单元的供电电压保持正常电压; 在昏睡模式下将存储单元的供电电压调低. DRU^[7]将循环程序段中的指令放在相同的子存储体中并修改替换算法以避免其被替换, 从而在大部分时间内仅使该子存储体处于正常工作模式. Early Set Resolution (ESR)^[8]方法对待访问的组进行确定性的预唤醒, 从而隐藏基址加偏移寻址的 load 指令的唤醒延迟.

为降低 load 指令的访存延迟, 快速地址计算方法^[9, 10]将基址和偏移的索引位相或作为索引预测值. 基

于步长的地址预测^[11,12]、基于上下文的地址预测^[13]和值预测^[14]技术对所有最近执行的 load 指令的访存地址或数值进行预测。

在国内研究工作中,文献[15]提出了一种改进的 LRU 替换算法以降低高速缓存的泄漏功耗;文献[16]利用指令顺序执行和相对跳转的特性,减少指令缓存访问次数以节省其动态功耗。

3 两种数据缓存访问控制策略

一种典型的嵌入式处理器流水线包含以下几个阶段:IF1、IF2 阶段进行取指和分支预测,DEC 阶段进行指令译码,ISS 阶段读寄存器堆,EXE 阶段进行 ALU 运算和访存地址计算,MEM1、MEM2 阶段进行虚实地址转换和数据缓存访问,WB 阶段回写寄存器堆。

本文针对上述典型嵌入式处理器提出两种数据缓存访问控制策略:基于地址预测的控制策略与基于偏移地址预检测的控制策略。

3.1 基于地址预测的控制策略

基于地址预测的控制策略首先使用 Load 指令访存地址预测器(Load Address Predictor, LAP)在流水线早期对 load 指令的访存地址作出预测,然后根据预测结果预先唤醒并访问相应的组,最后在完成访存地址计算时检查预测结果的正确性,并根据检查结果更新 LAP 表项信息。

3.1.1 地址预测与数据缓存访问 LAP 采用指令地址索引的直接映射或组相联结构,每个表项所记录的 load 指令相关信息如下。Valid 域表示该表项是否有效,PC 域记录 load 指令地址, PredAddr 域用于预测该 load 指令下次执行的访存地址,Stride 域记录该 load 指令最近两次执行的访存地址步长变化,Confidence 域是一个两位的饱和预测器,表示对 LAP 预测正确性的信心。

使用 LAP 进行访存地址预测的过程如下。在 IF2 阶段末尾进行指令预译码以识别 load 指令,在 DEC 阶段使用 load 指令地址访问 LAP,并根据地址预测值 PredAddr 产生索引预测值和部分标签预测值。

接下来将根据地址预测结果预先唤醒并访问数据缓存。在 ISS 阶段使用索引预测值预先唤醒 Data Array 的相应组并访问 Sentry-Tag Array^[17],将各个路读出的 Sentry-Tag 与部分标签预测值相比较。如果有且只有一路比较结果相等,在 EXE 阶段只从这一路读出数据而不访问其它路,以降低数据缓存的动态功耗;同时,采用门控预充电技术^[4],只对这一路位线进行预充电,以降低其它路位线的泄漏功耗。当预先唤醒和访问结果正确时,将比原有访存过程节省 2 个周期;而当索引预测错误或者预先访问错误时,将比原有访存过程损失 1 或 2 个周期。

3.1.2 正确性检查与 LAP 更新 地址预测的正确性检查分为如下两个阶段进行。如果 MEM1 阶段的检查发现索引预测正确并且该 load 指令预先访问了某一路的 Data Array,则该阶段检查通过。如果 MEM2 阶段的检查发现该 load 指令访问数据缓存命中且命中的路与预先访问的路相同,则该阶段检查通过。

对 LAP 的更新过程如下。如果两个阶段的检查均通过,将 Confidence 域加 1(到饱和为止),并将该 load 指令本次执行的访存地址与 Stride 域相加保存在 PredAddr 域中。如果任意一个阶段的检查未通过,将 Confidence 域清零,将该 load 指令本次执行的访存地址与 PredAddr 域相减保存在 Stride 域中,然后将本次执行的访存地址与 Stride 域相加保存在 PredAddr 域中。

3.2 基于偏移地址预检测的控制策略

基于偏移地址预检测的控制策略预先检测基址加偏移寻址的 load 指令的偏移地址,以识别访存地址需访问的区域并预先将其唤醒。

首先,在 DEC 阶段判断偏移地址对应于组索引的高 m 位是否为零。如果为零,根据基址和偏移地址的低位相加是否向高 m 位产生进位,组索引高 m 位的值只有如下两种可能:(1)基址对应于组索引的高 m 位;(2)基址对应于组索引的高 m 位加 1。接下来,在 ISS 阶段读取基址寄存器,并根据这两种可能的取值在 EXE 阶段早期分别将其所对应的子存储体唤醒。

由于偏移地址往往是较小的立即数^[8],基于偏移地址预检测的控制策略能够预先唤醒大多数基址加偏移寻址的 load 指令需访问的子存储体,在获取泄漏功耗节省的同时避免出现性能损失的情况。

4 基于循环识别的控制策略选择机制

基于地址预测的控制策略有利于大幅度降低泄漏功耗,但功耗节省和性能影响取决于 LAP 的预测正确率;基于偏移地址预检测的控制策略对泄漏功耗的节省效果相对有限,但不会出现性能损失的情况。本文提出基于循环识别的控制策略选择机制,在运行时刻识别循环程序段,并根据程序处于不同运行阶段时的访存地址规律性在两种策略之间作出动态选择。

4.1 功耗管理器状态转换

本文的功耗管理器对 SEPAS-Filter^[18]作扩展以在运行时刻识别循环程序段。在文献[18]中,SEPAS-Filter 用于记录最近执行的分支指令的相关信息以降低分支预测器的动态功耗。本文在 SEPAS-Filter 的表项中增加宽度为 1 的 Loop 域,记录从哪条分支指令进入循环程序段。

采用扩展 SEPAS-Filter 的功耗管理器状态转换如图 1 所示。初始时功耗管理器处于循环识别状态。当发现 SEPAS-Filter 中某表项的 Taken 域饱和时,推断很可能由

该分支指令进入一个循环程序段, 于是将相应表项的 Loop 域置为 1, 功耗管理器转换到循环初始统计状态。

在循环初始统计状态中, 将在初始两次迭代的执行过程中完成对 LAP 的初始化。第一次迭代执行时, 使用 load 指令地址和访存地址分别填充相应表项的 PC 域和 PredAddr 域, 将 Cor-

confidence 初始化为零, 并将 Valid 置为有效; 第二次迭代执行时, 首先将该 load 指令本次执行的访存地址与相应表项的 PredAddr 域相减保存在 Stride 域中, 然后将本次执行的访存地址与 Stride 域相加保存在 PredAddr 域中。在完成上述初始统计后, 处理器继续执行循环, 当再次遇到 Loop 域为 1 的表项中的分支指令实际方向为跳转时, 功耗管理器从循环初始统计状态转换到循环执行状态。

在循环执行状态中, 当发现 SEPAS-Filter 中 Loop 域为 1 的表项中的分支指令实际方向为不跳转或该表项被替换时, 推测很可能处理器已完成对该循环程序段的执行, 功耗管理器返回循环识别状态, 并将该表项的 Loop 域清零。

4.2 控制策略的动态选择

在循环识别状态, 处理器执行非循环程序段, 访存地址规律性往往较难捕获。为避免 LAP 的错误地址预测带来不必要的性能损失, 功耗管理器选择基于偏移地址预检测的控制策略。

在循环初始统计状态, 虽然处理器已经开始执行循环程序段, 但 LAP 尚未完成对 load 指令相关信息的初始统计, 因此, 功耗管理器仍然采用基于偏移地址预检测的控制策略。

在循环执行状态, 为充分利用循环程序段访存地址规律性较强的特点^[12], 功耗管理器采用基于地址预测的控制策略, 仅当 LAP 发生失效时才采用基于偏移地址预检测的控制策略。

此外, 当使用上述两种控制策略均无法捕获 load 指令的访问需求时, 为避免在访问昏睡的行时由于额外的唤醒周期引起性能损失, 需要在 EXE 阶段提前将数据缓存的所有组唤醒。

5 效果评估

实验环境采用 SimpleScalar^[19] 模拟器, 集成 Watch 1.02^[20] 和 HotLeakage 1.0^[21] 功耗模型, 运行嵌入式基准程

序 Dhrystone 和 MiBench。基础处理器配置如表 1 所示。

表 1 基础处理器配置参数

| | |
|--------------|-------------------------------------|
| 流水线 | 8 级, 按序执行 |
| 译码宽度 | 1 |
| 分支方向预测器 | 1024 表项 |
| 目标地址缓冲器 | 128 表项 |
| SEPAS-Filter | 8 表项, LRU 替换算法 |
| 指令 Cache | 16KB, 4 路组相联 |
| 数据 Cache | 16KB, 4 路组相联 |
| 指令 TLB | 8 表项全相联一级 TLB 64 表项 4 路组相联二级 TLB |
| 数据 TLB | 8 表项全相联一级 TLB 64 表项 4 路组相联二级 TLB |
| 寄存器堆 | 32* 4 字节 |
| 总线宽度 | 4 字节 |
| 内存访问延迟 | 40 周期 |

在实验中, 对于 ESR 方法^[8], 选取 update window 为 1024 个时钟周期。对于本文方法, LAP 为 32 表项直接映射结构, 当表项中 Confidence 域的数值达到 2 或 3 时表示预测结果可用; 数据缓存 Sentry-Tag 宽度为 5。基于偏移地址预检测的控制策略将数据缓存划分为 16 个子存储体, 相应地, 每次判断偏移地址对应于组索引的高 4 位是否为零。动态电压调节电路中正常供电电压为 0.9V, 昏睡模式供电电压为 0.3V。环境温度为 80 摄氏度, N 管和 P 管的阈值电压分别为 0.19V 和 0.21V。

5.1 访问需求捕获比例与地址预测正确率

本文将可以由基于地址预测的控制策略(策略一)预测出可用的访存地址或符合基于偏移地址预检测的控制策略(策略二)的检测条件的 load 指令称为访问需求被捕获的 load 指令。如图 2 所示, 在平均情况下, 如果单独在循环执行状态使用策略一, 访问需求捕获比例可达到 64.4%, 并且 LAP 的地址预测正确率达到 93.4%; 如果单独使用策略二, 访问需求捕获比例可达到 60.3%; 如果按照本文的功耗管理方法将两者结合使用, 策略一的访问需求捕获比例与地址预测正确率不变, 策略二的访问需求捕获比例为 10.3%。

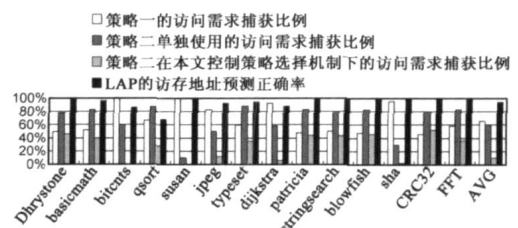


图 2 访问需求捕获比例与地址预测正确率

同时, 本文实验结果显示, ESR 方法^[8] 仅能预先唤醒 24.5% 的动态执行的 load 指令需访问的组。因此, 与 ESR 方法相比, 本文的两种控制策略能够捕获更多的 load 指令的访问需求, 并且 LAP 具有较高的地址预测正

确率, 从而为有效控制数据缓存的活动提供更大机会.

5.2 功耗

5.2.1 泄漏功耗 在平均情况下, ESR 方法^[8]和本文方法分别将存储单元泄漏功耗降低 62.8% 和 90.9%, 如图 3 所示. 与 ESR 方法相比, 本文方法仅使需访问的区域在访问期间处于工作模式, 因此具有更优的存储单元泄漏功耗节省效果.

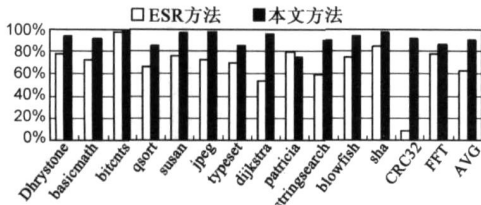


图3 数据缓存存储单元泄漏功耗节省比例

本文方法和 ESR 方法^[8]对数据缓存总泄漏功耗的节省情况如图 4 所示. 由于基于地址预测的控制策略对数据缓存中不需访问的路有效避免位线预充电过程, 本文方法在平均情况下将位线的泄漏功耗降低 64.5%. 由于存储单元和位线的泄漏功耗均被降低, 本文方法将数据缓存总泄漏功耗降低 85.4%. 而 ESR 方法仅降低存储单元的泄漏功耗, 在平均情况下将数据缓存总泄漏功耗降低 50.2%. 因此, 本文方法比 ESR 方法具有更优的数据缓存泄漏功耗节省效果.

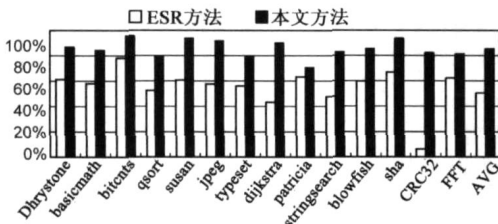


图4 数据缓存总泄漏功耗节省比例

5.2.2 动态功耗与总功耗 由于基于地址预测的控制策略能够预先识别不需访问的路并避免对其不必要的访问, 本文方法将数据缓存动态功耗(考虑 LAP 和 Sentry-Tag Array^[7]的动态功耗)节省 61.1%. 由于数据缓存的泄漏功耗和动态功耗均被降低, 数据缓存的总功耗和处理器的总功耗得以降低. 在平均情况下, 本文方法将数据缓存总功耗降低 75.02%, 从而使处理器的

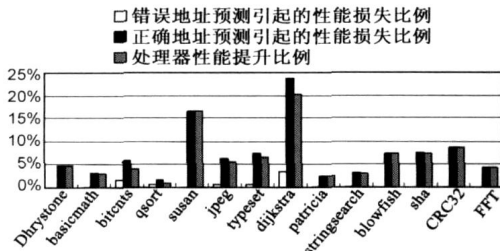


图5 本文方法的处理器性能影响比例

总功耗降低 9.75%.

5.3 性能

基于地址预测的控制策略具有较高的访问需求捕获比例, 并且 LAP 地址预测正确率较高, 因此, LAP 所作出的正确的地址预测带来的性能提升要大于错误的地址预测引起的性能损失, 从而使处理器性能得到提高. 如图 5 所示, 在平均情况下, 本文方法使处理器性能提高 4.41%, 比传统方法达到了更优的性能结果.

5.4 LAP 设计空间探索

以采用 16 表项直接映射 LAP(LAP_16_1)的功耗降低与性能提升结果为基准值, 将采用各种配置的 LAP 的结果作规格化, 得出的数据如图 6 所示.

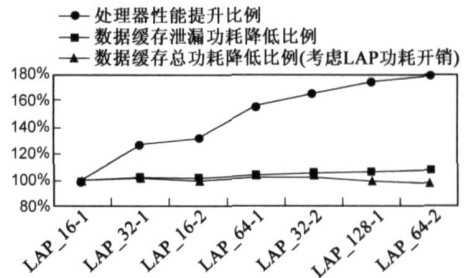


图6 采用不同配置LAP的功耗降低与性能提升结果(以采用 LAP_16-1的结果为基准作规格化)

实验结果表明, 随着 LAP 容量或相联度的增大, 处理器性能提升比例和数据缓存泄漏功耗降低比例均逐渐增大, 但数据缓存总功耗降低比例呈先升后降的趋势. 因此, 如果不经过功耗管理器的指导而直接采用容量较大的 LAP, 虽然有利于降低数据缓存的泄漏功耗, 但是会由于 LAP 自身功耗开销较大而不利于对数据缓存的总功耗进行优化. 在实际处理器设计中, 需要综合考虑数据缓存功耗降低比例和处理器性能提升比例, 选择容量与相联度适当的 LAP, 达到较优的功耗与性能结果.

6 结论

本文根据访存指令对数据缓存的访问需求控制数据缓存的活动. 当未发现访存指令时将整个数据缓存保持在非活跃状态, 而当发现访存指令进入流水线时, 本文的两种控制策略能够在流水线早期及时捕获访存地址的访问需求, 仅使需访问的区域在访问期间处于活跃状态, 从而有效降低数据缓存的泄漏功耗. 与传统方法相比, 本文方法在功耗与性能方面均达到更优结果.

参考文献:

[1] J Montanaro, et al. A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor[J]. IEEE Journal of Solid State Circuits, 1996, 31(11): 1703-1714.

[2] B Doyle, R Arghavani, et al. Transistor elements for 30 nm physical gate lengths and beyond[J]. Intel Technology Journal, 2002, 6(2): 42-54.

- [3] N S Kim, K Flautner, et al. Circuit and microarchitectural techniques for reducing cache leakage power[J]. IEEE Transactions on Very Large Scale Integration Systems, 2004, 12(2): 167–184.
- [4] N S Kim, K Flautner, D Blaauw, T Mudge. Single V_{DD} and single V_T super drowsy techniques for low leakage high performance instruction caches[A]. Int'l Symposium on Low Power Electronics and Design[C]. Newport Beach, California, USA: IEEE Computer Society, 2004. 54–57.
- [5] S Kaxiras, Z Hu, M Martonosi. Cache decay: exploiting generational behavior to reduce cache leakage power[A]. Int'l Symposium on Computer Architecture[C]. Goteborg, Sweden: IEEE Computer Society, 2001. 240–251.
- [6] T Pering, T Burd, R Brodersen. The simulation and evaluation of dynamic voltage scaling algorithms[A]. Int'l Symposium on Low Power Electronics and Design[C]. California, USA: IEEE Computer Society, 1998. 76–81.
- [7] P Kalla, et al. Distance based recent use(DRU): an enhancement to instruction cache replacement policies for transition energy reduction[J]. IEEE Transactions on Very Large Scale Integration Systems, 2006, 14(1): 69–80.
- [8] S Kim, N Vijaykrishnan, M J Irwin, L K John. On load latency in low power caches[A]. Int'l Symposium on Low Power Electronics and Design[C]. Seoul, Korea: IEEE Computer Society, 2003. 258–261.
- [9] T M Austin, D N Pnevmatikatos, G S Sohi. Streamlining data cache access with fast address calculation[A]. Int'l Symposium on Computer Architecture[C]. Ligure, Italy: IEEE Computer Society, 1995. 369–380.
- [10] T M Austin, G S Sohi. Zero cycle loads: microarchitecture support for reducing load latency[A]. Int'l Symposium on Microarchitecture[C]. Ann Arbor, Michigan, USA: IEEE Computer Society, 1995. 82–92.
- [11] R J Eickemeyer, S Vassiliadis. A load instruction unit for pipelined processors[J]. IBM Journal of Research and Development, 1993, 37(4): 547–564.
- [12] J Gonzalez, A Gonzalez. Speculative execution via address prediction and data prefetching[A]. ACM Int'l Conference on Supercomputing[C]. Vienna, Austria: ACM Press, 1997. 196–203.
- [13] M Bekeman, S Jourdan, R Ronen, G Kirshenboim. Correlated load address predictors[A]. Int'l Symposium on Computer Architecture[C]. Atlanta, Georgia, USA: IEEE Computer Society, 1999. 54–63.
- [14] M H Lipasti, C B Wilkerson, J P Shen. Value locality and load value prediction[A]. Int'l Conference on Architectural Support for Programming Languages and Operating Systems[C]. Cambridge, Massachusetts, USA: ACM Press, 1996. 138–
- [15] 张承义, 张民选, 邢座程, 王永文. LRU-Assist: 一种高效的 Cache 漏流功耗控制算法[J]. 电子学报, 2006, 34(9): 1626–1630.
Zhang Chengyi, Zhang Minxuan, Xing Zuocheng, Wang Yongwen. LRU-Assist: An efficient algorithm for cache leakage power controlling[J]. Acta Electronica Sinica, 2006, 34(9): 1626–1630. (in Chinese)
- [16] 张宇弘, 王界兵, 严晓浪, 汪乐宇. 标志预访问和组选择历史相结合的低功耗指令 cache[J]. 电子学报, 2004, 32(8): 1286–1289.
Zhang Yuhong, Wang Jiebing, Yan Xiaolang, Wang Leyu. Pre visiting tag and keeping way history to reduce power in instruction cache[J]. Acta Electronica Sinica, 2004, 32(8): 1286–1289. (in Chinese)
- [17] Y J Chang, S J Ruan, F Lai. Design and analysis of low power cache using two level filter scheme[J]. IEEE Transactions on Very Large Scale Integration Systems, 2003, 11(4): 568–580.
- [18] A Baniasadi, A Moshovos. SEPA: A highly accurate energy efficient branch predictor[A]. Int'l Symposium on Low Power Electronics and Design[C]. Newport Beach, California, USA: IEEE Computer Society, 2004. 38–43.
- [19] D Burger, T M Austin. The SimpleScalar tool set, version 2.0[J]. ACM Computer Architecture News, 1997, 25(3): 13–25.
- [20] D Brooks, V Tiwari, M Martonosi. Wattch: A framework for architectural level power analysis and optimizations[A]. Int'l Symposium on Computer Architecture[C]. Vancouver, British Columbia, Canada: IEEE Computer Society, 2000. 83–94.
- [21] Y Zhang, D Parikh, et al. HotLeakage: A temperature aware model of subthreshold and gate leakage for architects[R]. Virginia, USA: Department of Computer Science, University of Virginia, 2003.

作者简介:



王箫音 女, 1983 年生于河北保定, 北京大学信息科学技术学院博士研究生. 主要研究方向为计算机体系结构和低功耗集成电路设计.

E-mail: wangxiaoyin@mprc.pku.edu.cn



佟冬 男, 1971 年生于吉林长春, 北京大学信息科学技术学院副教授. 主要研究方向为计算机体系结构、超大规模集成电路设计、系统芯片及软硬件协同设计.