

“龙腾 R”微处理器分支处理单元的研究与设计

鄢国丰, 张盛兵, 黄小平

(西北工业大学 航空微电子中心, 陕西 西安 710072)

摘 要:“龙腾 R”是西北工业大学自主研制的 32 位高性能微处理器. 该处理器的分支处理单元(BPU)能有效降低控制相关带来的延迟. 通过分析已有的分支方向预测算法和分支目标地址预测策略, 在分支处理单元总体约束下, 合理分配分支方向预测和目标地址预测的实现代价, 提出了一种基于混合分支预测器和经过改进的目标地址缓冲(BTB)结构的分支处理单元结构. 该结构不仅比传统的由 gshare 分支方向预测器构成的分支处理单元预测准确率平均高出 1%~2%, 并具有面积小、功耗低的特点.

关键词: 分支预测; 分支处理; 混合分支预测器; 目标地址缓冲

中图分类号: TN4

文献标识码: A

文章编号: 1000-7180(2010)06-0122-06

Research and Design of Branch Process Unit in “Longtium R” Microprocessor

YAN Guo-feng, ZHANG Sheng-bing, HUANG Xiao-ping

(Aviation Microelectronics Center, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: “Longtium R” is a 32-bit high performance microprocessor developed by Northwestern Polytechnical University. Branch process unit (BPU) of the processor can remarkably reduce the latency caused by control hazard. Through analyzing some algorithms of branch direction predict and some strategies of branch target address predict, under the overall constraints of BPU, appropriately allocating the hardware cost, this paper proposes a new structure of branch process unit, based on a combined branch predictor and improved branch target buffer (BTB). Comparing to the traditional BPU which is comprised of gshare branch predictor, this one increases prediction accuracy by 1%~2%, with characteristics of small area and low power consumption.

Key words: branch prediction; branch process; combined branch predictor; branch target buffer

1 引言

为了提高性能, 现代处理器大多采用了流水线技术和超标量技术. 随着指令发射宽度和流水线深度的增加, 机器 CPI 降低. 根据 Amdahl 定律可知, 随着机器 CPI 的降低, 控制相关导致的相对影响越来越大. 目前, 大多数处理器采用分支预测技术来解决控制相关导致的问题.

分支预测包括分支方向预测和分支目标地址预测两个方面. 为了提高分支预测的准确率, 目前大部分研究集中在如何提高分支方向预测的准确率上.

分支目标地址预测主要采用分支目标缓冲(BTB)结构, 通过调整 BTB 结构可以优化分支处理单元的性能.

文中通过多个实验说明了不同分支方向预测器的预测准确率的高低和不同 BTB 结构对分支目标地址预测命中率的影响, 并对分支方向预测器和 BTB 结构做出了相应的优化以及性能分析. 最后, 提出一种由基于混合分支预测算法的分支方向预测器和经过改进的 BTB 结构组成的分支处理单元, 将该单元集成到“龙腾 R”微处理器中用 VCS 进行系统级仿真测量分支预测的准确率, 并通过 Design

收稿日期: 2009-05-14; 修回日期: 2009-08-20

基金项目: 国家自然科学基金项目(60773223, 60736012)

Compiler 和 Power Compiler 等工具得到面积,功耗等参数,说明该单元具有预测命中率高、面积小和功耗低等特点.

2 分支方向预测分析

分支方向预测分为静态分支预测和动态分支预测.静态分支预测利用静态分支的信息或编译器提供的信息对分支方向进行预测;动态分支在程序执行中动态收集与分支跳转相关信息并利用这些历史信息对分支方向进行预测.文献已经表明动态分支预测在预测准确率和实用性等方面胜过静态分支预测^[1].因此,文中主要讨论的是动态分支预测.

2.1 相关研究

分支指令的行为并不是完全随机的,大多数分支指令要么经常跳转,要么经常不跳转.两位饱和计数器预测器就是利用了分支指令的这个特性,在1981年由 J. E. Smith 引入^[1],它采用的是一个两位饱和计数器的状态机,当有分支需要预测时,根据当前计数器状态来预测分支“taken”或“not-taken”.而当执行结果出来后,计数器状态按照状态迁移函数进行变化状态迁移函数有许多形式,但最有效形式如图1所示^[2].

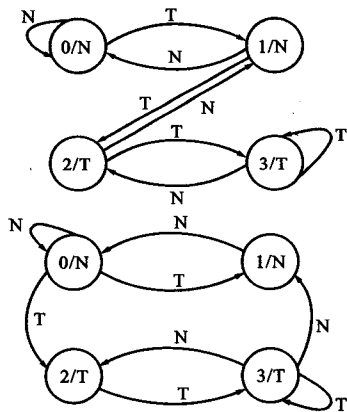


图1 两位饱和计数器预测器 FSM

为了区分不同的分支指令和利用两级饱和计数器的特点,bimodal 预测器利用分支指令地址索引两位饱和计数器.bimodal 预测器实现简单,并且命中率较高.

Bimodal 预测器只是依靠分支指令最近的行为来预测将来分支的行为.我们知道,分支指令附近已执行过的分支指令的行为也可以用来指导预测该分支的行为.两级分支(two-level)预测器就是基于这个原理提出的非常有效的预测器,由 T. Yeh 和 Y. Patt 于1992年提出^[2],它有两级预测表,第一级是

分支历史表(BHT),其中包含多个分支历史移位寄存器(BHSR);第二级是模式历史表(PHT).当有分支需要预测时,由分支地址选择一个分支历史寄存器(BHR),然后由这个 BHR 索引 PHT 选择一个两位饱和计数器对分支方向进行预测.

Gshare 预测器由 Scott M. McFaring 于1993年提出^[3],它汲取了两级分支预测的优点,减少了硬件实现的复杂度,同时在一定程度上减小不同分支在 PHT 中的冲突,提高预测准确率.它用分支地址和全局分支历史寄存器的异或(XOR)来索引分支模式表(PHT),充分利用分支执行过程中已有的信息来指导分支预测.

2.2 mgshare 预测器

对于32位指令长度和指令地址的RISC处理器,指令地址的低两位恒为2'b00.因此,有效地址为前30位.一般的gshare预测器选取有效地址的低n位作为分支PC信息,然后使用m位历史信息位与分支PC信息的高m位进行XOR,XOR的结果和分支PC信息剩下的低n-m位进行拼接后索引PHT.而笔者发现如果使用分支PC信息低地址部分与分支历史信息进行XOR,可以进一步减小不同分支在PHT中的冲突,进而提高分支方向预测的预测准确率.文中称这种分支方向预测器为mgshare预测器.各种分支方向预测器如图2所示.

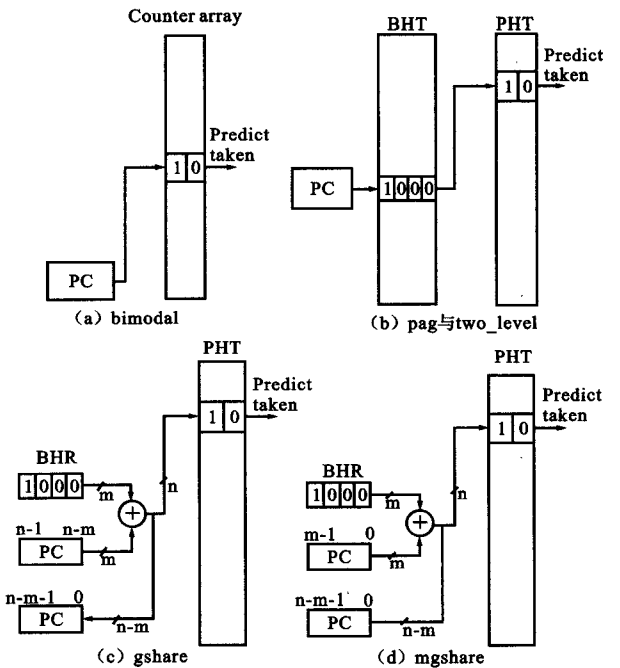


图2 各种分支方向预测器

为了说明各种预测器的性能,使用 simplescalar 的 sim-bpred 模拟器来评估预测器的预测准确率.

其中,采用的基准程序是 SPEC95 中 gcc、go 和 li 程序。

对于 bimodal 预测器和 gshare 预测器,都选取了 1000x2bits 的 PHT 和有效地址的低 10 位作为分支 PC 信息;不同的是,bimodal 预测器采用分支 PC 信息直接索引 PHT;而 gshare 预测器先选取有效地址的低 10 位作为分支 PC 信息,然后使用 6 位历史信息位与分支 PC 信息的高 6 位进行 XOR,XOR 的结果和分支 PC 信息剩下的低 4 位进行拼接后索引 PHT. 而 mgshare 采用有效地址低 6 位作为分支 PC 信息,用 6 位历史信息位与 6 位分支 PC 信息进行 XOR,XOR 的结果和分支 PC 信息低 4 位进行拼接后索引 PHT. 对 Bimodal 预测器、gshare 预测器和 mgshare 预测器的性能分析,可以得到如下结论:

(1) 对于大多数程序,因为采用了附近其他发生过的分支指令的跳转信息,gshare 预测器的预测准确率都大于 bimodal 预测器预测准确率。

(2) 对于某些程序或者程序执行过程中的某些

指令区间,bimodal 预测器预测准确率大于 gshare 预测器预测准确率。

(3) Mgshare 预测器的预测准确率大于一般的 gshare 预测器。

2.3 基于 bimodal 和 mgshare 的混合分支预测器

根据以上结论,综合各种单个分支预测器的优点,可以实现一种混合分支方向(comb)预测器^[3],得到比单个预测器高的预测准确率. 因为 mgshare 预测器和 bimodal 预测器实现起来硬件开销小,性价比高. 文中采用这两个分支预测器的组合,该预测器的准确率如图 3 所示。

另外还需要一个选择预测器用于对这两个预测器的预测结果根据其信任度进行选择. 这个选择预测器进行选择的信息也是存在多个两位饱和计数器做成一个 PHT 中,索引是有效地址的低 10 位. 状态变化函数是当两个预测器预测判断一样时,状态不变;当预测判断不一样时,预测正确的预测器信任度加 1. comb 预测器的实现如图 4 所示。

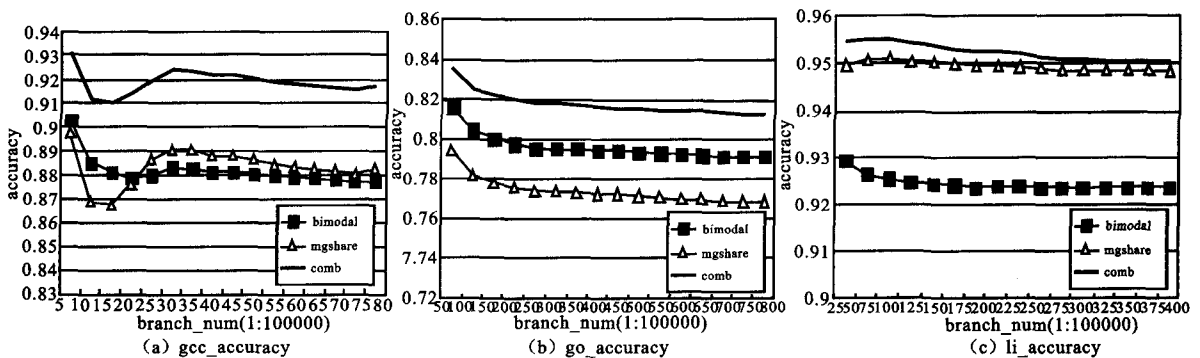


图 3 comb 分支预测器预测准确率

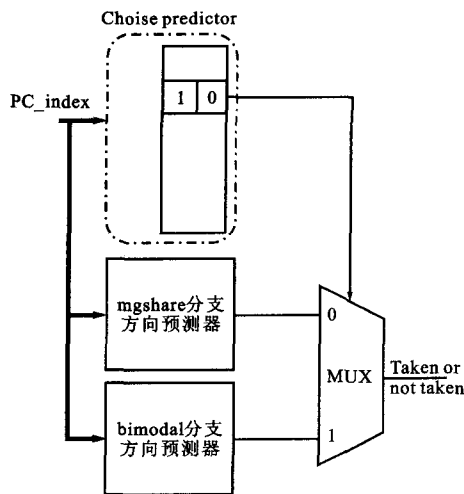


图 4 comb 预测器的组成

3 目标地址缓冲分析

BTB用于分支指令预测为跳转时得到该分支指令的下一条取指令地址. 完整的 BTB^[4]包括 4 部分:标识指令的 tag,一般用分支指令地址表示;分支方向的预测信息;分支目标地址;存在分支目标地址的指令. 这些信息不是全部都需要的,这其中存在设计时序,设计面积,执行周期的折中. 文中设计的 BTB 采用分支指令地址和分支目标地址两部分组成。

3.1 BTB 的结构

当分支指令预测为跳转时,将分支指令地址和 BTB 结构中相联 tag 作比较;一旦匹配上,则将相应的分支目标地址作为分支指令的下一条指令地址. 如图 5 所示。

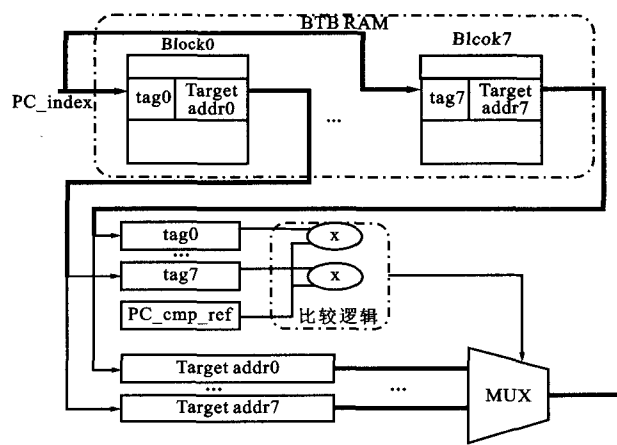


图 5 BTB 的组成

3.2 BTB 的性能分析

BTB 是一个组相联存储器,可以通过调整 BTB 实体的项数和相联程度来改变 BTB 的命中率.采用上文所用的模拟器环境和基准程序,分支方向预测器是 mgshare 预测器.分析 BTB 的命中率可见,BTB 的相联程度越高,命中率越高,并随着相联程度的增加,命中率的增幅降低;BTB 实体的项数越多,命中率越高,并随着 BTB 实体的项数的增多,命中率的增幅降低.因此,需要在 BTB 实现的代价和相联程度、BTB 实体的项数多少进行折中.

3.3 优化的 BTB 结构

对于 64 项 8 路组相连的 BTB 和 30 位的有效地址,则 BTB 的 tag 域应该是 24 位,由图 5 可知,需要 8 个 24 位的比较器.这样的组合逻辑的延时和面积都比较大,可以将原来 24 位 BTB 的 tag 域分成两部分,高 16 位称为 PC 域舍弃不用,而低 8 位作为 BTB 的新 tag,这样只有属于不同 pc 域的相同 tag 交替出现时,才会出现预测分支目标地址错误的情况,而这种情况的概率是很低的.如图 6 所示.采用改进的 BTB 结构后,BTB 的命中率几乎没有降低,但是却大大的减少了比较器的逻辑和 BTB 存储器的大小.

4 分支处理单元实现与优化

4.1 分支处理单元的结构

在“龙腾 R”微处理器分支处理单元(BPU)的实现中,4 条指令同时进入指令预处理单元,当有分支指令需要进行方向预测时,则将该指令加入动态方向预测器中进行方向预测;当有分支指令需要进行目标地址预测时,则将该指令加入 BTB 中进行目标地址预测;当预测为跳转且通过 BTB 预测分支目

标地址后,将该分支目标地址作为下条指令的取指地址.“龙腾 R”微处理器分支处理单元的实现结构如图 7 所示.

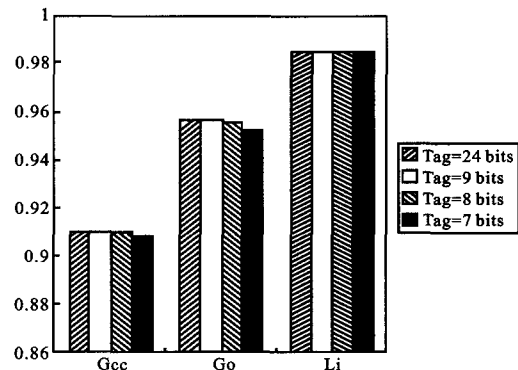


图 6 改进后的 BTB 命中率

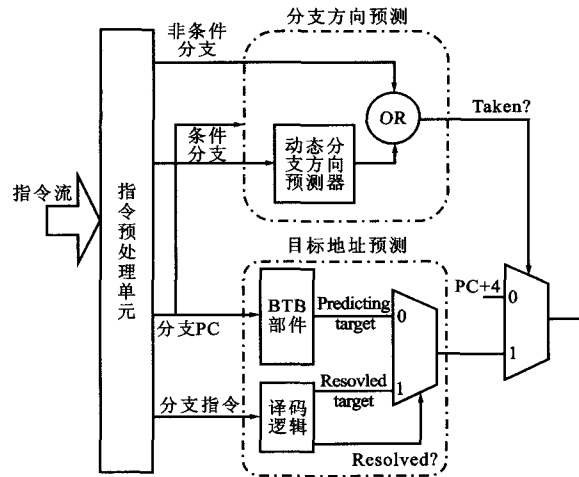


图 7 “龙腾 R”分支处理单元结构图

4.2 分支方向预测与分支目标地址预测的权衡

通过对分支方向预测及 BTB 的分析可以得出如下结论:

(1) Bimodal 预测器和 gshare 预测器实现简单,预测准确率较高,gshare 预测器因为利用了相关分支信息,在大多数测试程序中该预测器预测准确率大于 bimodal 预测器,但是 bimodal 预测器在某些测试程序中表现良好.

(2) 从预测器预测准确率上看,基于 bimodal 算法和 mgshare 算法的混合分支方向预测器,比 bimodal 预测器和 mgshare 预测器的性能都要好.

(3) BTB 的大小和相联程度影响 BTB 的命中率,但是随着 BTB 大小的增加和相联程度的增大,命中率的增幅降低.

(4) 简化的 BTB 结构虽然使得 BTB 命中率略有下降,但是却大大减少了 BTB 存储体和比较逻辑的面积.

另外,可以建立分支处理单元预测准确率与分支方向预测准确率、分支目标地址预测命中率关系的数学模型.

$$Ph = Pntnt + Ptt * Pb;$$

$$Pd = Ptt + Pntnt;$$

$$Ptt = \theta * Pd;$$

其中,各个变量的含义如下:

Ph :整个分支处理单元的预测准确率;

$Pntnt$:分支方向预测器预测某分支指令不跳转,结果该指令不跳转的概率;

Ptt :分支方向预测器预测某分支指令跳转,结果该指令跳转的概率;

Pb :分支目标地址预测单元预测目标地址的命中率;

Pd :分支方向预测器的预测准确率;

θ :分支方向预测器预测准确且预测为跳转的指令在所有方向预测准确的指令的比例;

$D(x)$:对 x 变量取微分;

为了简化计算,现假定 θ 为常量,且等于程序中分支指令实际跳转数目和分支指令数目的平均比率,大量统计表明^[5],这个平均比率约为 0.67. 则

$$Ph = (1 - \theta) * Pd + \theta * Pd * Pb$$

保持分支目标地址预测单元预测命中率不变,改变分支方向预测器预测准确率,根据之前的实验数据,假定目标地址平均预斥命中率约为 0.9.

$$D(Ph) = (1 - \theta + \theta * Pb) D(Pd) \\ \approx 0.933 D(Pd)$$

保持分支方向预测器预测准确率不变,改变分支目标地址预测单元预测命中率,根据之前的实验数据,假定分支方向平均预测准确率约为 0.9.

$$D(Ph) = \theta * Pd * D(Pb) \approx 0.603 D(Pb)$$

由此可见,在增加同等硬件复杂度的条件下如果能提高相同的分支方向预测准确率和目标地址预测命中率,应优先考虑改进分支方向预测器.

4.3 基于混合分支预测器和优化的 BTB 结构的分支处理单元的实现

基于以上结论,可以适量地增加方向预测器的复杂度来提高方向预测的准确率,同时为了减少面积和功耗,在可以接受的范围内降低目标地址预测

单元预测准确率的前提下,简化 BTB 结构,最终使得整个分支处理单元预测准确率高、面积小、功耗低.文中建立了一种优化的分支处理单元实现方案如下.分支方向预测器采用基于 bimodal 算法和 mgshare 算法的混合分支预测器,其中选择预测器, bimodal 预测器和 mgshare 预测器均采用 $1000 \times 2\text{bits}$ 的 PHT.选择预测器和 bimodal 预测器采用有效地址的低 10 位索引 PHT,而 mgshare 预测器有效地址的低 6 位与 6 位分支历史信息进行 XOR,然后与有效地址的低 4 位拼接后索引 PHT;BTB 采用优化的 BTB 结构,即 8 个 64×38 的存储器,即 64 项 BTB 实体,tag 域为 8 位,预测目标地址 30 位,8 路组相连.

为了验证该优化方案实现的分支处理单元的性能,现使用一种典型的分支处理单元作为参考方案如下.分支方向预测器采用 gshare 预测器;gshare 预测器采用有效地址的低 10 作为分支 PC 信息,分支 PC 信息的高 6 位与 6 位分支历史信息进行 XOR,然后与有效地址的低 4 位拼接后索引 PHT;BTB 采用 8 个 64×54 的存储器,即 64 项 BTB 实体,tag 域为 24 位,预测目标地址 30 位,8 路组相连.

4.4 仿真环境

针对以上提出的两种分支处理单元分别用可综合的 verilog HDL 代码实现,使用 Design Compiler 进行综合.其中时钟约束是 100MHz,对两种 BPU 进行面积分析,同时提取相应网表后分别加入“龙腾 R”的代码中用 VCS 进行仿真并测量 BPU 的预测准确率.使用的仿真程序是 VxWorks 操作系统的启动程序的一部分,功耗分析是使用 Power Compiler 分析 VCS 仿真后产生的后标 SAIF 文件得到的数据.

4.5 实验结果分析

由表 1 可知,优化的分支处理单元的预测准确率明显高于参考方案的分支处理单元的预测准确率,分支目标地址命中率略低于参考方案的分支目标地址命中率,但是比参考方案面积小、功耗低.

参考方案分支方向预测器实现简单,在 BTB 硬件实现中存储器和比较逻辑面积较大,分支方向预测和分支目标地址预测实现分配不够合理,整体预测准确率低.

表 1 两种预测器预测启动 VxWorks 的预测准确率

分支处理单元实现	频率 /MHz	面积 / μm^2	功耗 /mW	分支总数 (万)	方向预测准确率 (%)	分支目标地址命中率 (%)	整体预测准确率 (%)
gshare + BTB	100	2085674.0	7.2108	34.7	91.3452	86.4668	83.6475
Comb + 改进 BTB	100	1749487.5	6.5716	34.7	93.0433	86.2572	85.3293

优化方案分支方向预测器实现较复杂,但是分支方向预测准确率比 mgshare 预测器和 bimodal 预测器都高;采用优化的 BTB,硬件实现存储器和比较逻辑面积大幅降低,分支方向预测和分支目标地址预测实现分配合理.和参考方案比较,整体实现面积小、功耗低,整个分支处理单元预测准确率高^[6].

5 结束语

分支处理单元由分支方向预测器和分支目标地址预测单元两部分组成,文中通过理论分析、实验论证说明了这两大部分各种实现方案的优缺点,并分别进行了优化设计.进一步综合数学推导的结论和实验得到的结论,提出了自己的分支预测单元:comb 分支方向预测器和优化的 BTB 结构.为了显示该设计的优越性,同时实现了一种简单的分支处理单元的参考模型: gshare 预测器^[7]和 BTB.在搭建的 VCS 仿真平台下进行仿真验证,并用 Power Compiler 进行功耗分析;用 Design Compiler 对分支处理单元进行了综合,其中时钟约束在 100MHz,并用 Primetime 进行了静态时序分析.结果表明,该设计相比参考模型具有面积小、功耗低及预测准确率高的特点.

参考文献:

- [1] Smith J E. A study of branch prediction strategies[C]// Minneapolis: Proceedings of the 8th Annual International

Symposium on Computer Architecture. Widely, 1981:135-148.

- [2] Yeh T, Patt Y. Two-level adaptive branch prediction [C]// Proceedings of the 24th Annual ACM/IEEE International Symposium and Workshop on Microarchitecture. USA:Ann Arbor, 1991:51-61.
- [3] Scott M, McFaring. Combining branch predictors[R]. California, Technical Report TN-36, Digital Western Research Laboratory, 1993.
- [4] Perleberg C, Smith A J. Branch target buffer design and optimization[J]. IEEE Transactions on Computers, 1993 (23):396-412.
- [5] Brian K Bray, Flynn M J. Strategies for branch target buffers[M]. New York:ACM,1991.
- [6] 汪永威,樊晓桢,黄小平. 32 位 RISC 微处理器中分支预测器的硬件实现[J]. 计算机应用与研究,2008(2):419-422.
- [7] 武萌,沈海斌. 一种 gshare 分支预测器的低功耗设计方法[J]. 微电子学与计算机,2007,24(3):200-206.

作者简介:

鄢国丰 男,(1984-),硕士研究生.研究方向为计算机系统结构、专用集成电路设计.

张盛兵 男,(1968-),教授.研究方向为 VLSI 系统设计、计算机系统结构、电子设计自动化.

黄小平 男,(1980-),博士研究生.研究方向为计算机系统结构、专用集成电路设计、VLSI 系统设计.

(上接第 121 页)

明距离的方法,并针对 QC LDPC 码校验矩阵的准循环结构特点提出改善校验矩阵的方法,使最小汉明距离得到增加,从而提高了 QC LDPC 码的译码性能.对于分析和优化 QC LDPC 码具有一定的意义.

参考文献:

- [1] Robert G Gallager. Low density parity check codes[J]. IRE Transon. Inform. Theory, 1962,8(1):21-28.
- [2] Zongwang Li, Lei Chen, Lingqi Zeng, et al. Efficient encoding of quasi-cyclic low-density parity-check codes [J]. IEEE Transaciton On Communications, 2006, 54 (1):71-81.

- [3] Tomas J Richardson, Rudiger L Urbanke. Efficient encoding of low-density parity-check codes [J]. IEEE Transaciton on Communications Theory, 2001,47(2):638-656.
- [4] Xiaoyu Hu, Fossorier M P C, Evangelos E. On the computation of the minimum distance of low-density parity-check codes[J]. IEEE International Conference on Communications, 2004(2):767-771.
- [5] Shu Lin, Lei Chen, Jun Xu, et al. Near shannon limit quasi-cyclic low-density parity-check codes[J]. IEEE Globecom, 2003:2030-2035.
- [6] 肖旻,王琳,罗智勇. LDPC 码的最小汉明距离估算. 信息技术,2005(1):4-6.