

嵌入式处理器动态分支预测机制研究与设计

黄伟, 王玉艳, 章建雄

(华东计算技术研究所, 上海 200233)

摘要: 针对嵌入式处理器的特定应用环境, 通过对传统神经网络算法的改进, 结合定制的分支目标缓冲, 提出一种复合式动态分支预测机制。该机制基于全局索引方式, 对BTB结构进行定制设计, 实现对循环逻辑中最后一条分支指令的精确预测。实验结果表明, 该动态分支预测机制能降低硬件复杂度, 提高预测精度。

关键词: 复合分支预测; 神经网络; 分支目标缓冲; 嵌入式处理器; SimpleScalar 模拟

Research and Design of Dynamic Branch Prediction Mechanism for Embedded Processor

HUANG Wei, WANG Yu-yan, ZHANG Jian-xiong

(East China Institute of Computer Technology, Shanghai 200233)

【Abstract】 Aiming to the specific application environment of embedded processors, this paper gives a hybrid mechanism which combines custom-designed Branch Target Buffer(BTB) with improved neural network arithmetic for the dynamic branch prediction. In this mechanism, neural network arithmetic implements an approach of global indexing with less resource rather than the normal indexing way based on the instruction address. In use of the unique feature of embedded applications, the BTB structure makes accurate prediction for the final branch instruction in the loop logic. The result indicates that this mechanism achieves high precision with lower complexity.

【Key words】 hybrid branch prediction; neural network; Branch Target Buffer(BTB); embedded processor; SimpleScalar simulation

1 概述

现代嵌入式处理器多采用超标量流水技术, 为减少流水过程中条件转移指令带来的系统延迟, 处理器设计通常采用动态预测转移技术。随着体系结构理论的发展, 目前该技术已形成几种经典算法, 如 Bimodal 算法和二级自适应预测算法等; 同时, 随着交叉学科观点的引入, 该领域又产生一些以神经网络预测算法为典型的新算法。

在通用的测试环境中, 经典的预测算法已可达高预测率, 但算法复杂度的提高, 使可实现性降低。高预测率的算法常常占用大量处理器内部存储器资源。因此, 对于有特定应用背景的处理器的设计, 须选择合适的算法加以改进, 才能达到较高的精确度, 从而有效利用有限资源。

本文针对嵌入式处理器的应用环境特点, 通过对分支目标缓冲(BTB)结构进行定制设计, 结合神经网络算法, 在资源高效配置利用的基础上, 提出一套现实可行的分支预测方案。

2 相关算法分析

分支指令的执行有多种预测算法, 较为代表的有 Bimodal 算法、二级自适应预测算法、混合预测算法和神经网络预测算法等。

Bimodal 算法是一种最基本的预测算法。它利用一个历史跳转状态机对分支指令进行跳转预测, 如图 1 所示。该算法根据最近一次历史跳转情况, 利用 1 位或 2 位饱和计数器对下一次跳转进行预测。1 位预测器仅与最近一次跳转情况相关, 在循环分支的首尾两处可能产生错误预测, 2 位预测器

对此加以改进^[1], 能正确预测首次进入循环的跳转方向。Bimodal 算法的最大优点是控制逻辑简单和资源占用率低, 但对于整数算法占优的程序, 其预测精度不甚理想。

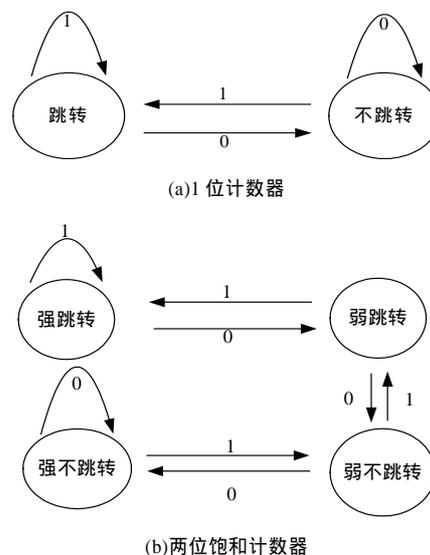


图 1 Bimodal 算法状态转换图

二级自适应预测技术^[2-3]是近年来应用最为广泛的预测

作者简介: 黄伟(1981-), 男, 硕士研究生, 主研方向: 计算机系统结构, IC 设计技术; 王玉艳, 高级工程师; 章建雄, 研究员
收稿日期: 2007-12-08 **E-mail:** vinz.huang@gmail.com

手段。二级自适应预测算法最初主要包括 2 个数据集：分支历史寄存器(BHR)和模式历史表(PHT)。其中BHR保存历史分支信息，PHT类似Bimodal算法中的饱和计数器，算法利用BHR对PHT进行索引获得预测结果。为提高预测精度，后续改进算法引入了地址信息，将BHR和PHT分别对应于全局、单地址和组地址进行定义，形成了表 1 中的 9 种预测器。

表 1 二级自适应预测算法

	全局 PHT	单组 PHT	单地址 PHT
全局 BHR	GAg	GAs	Gap
单组 BHR	SAg	SAs	Sap
单地址 BHR	PAg	PAs	Pap

相对于 Bimodal 算法，二级自适应预测算法性能有很大提高，但其受限于资源条件，即使在最优情况下，二级自适应预测算法仍存在 2 个问题：

- (1)对于循环预测，最后一次跳转必然产生误判。
- (2)对于一般条件预测，主要依赖饱和计数器进行预测，对于不同的条件输入，不能达到好的预测效果。

在两级自适应预测算法基础上，文献[4]提出一种混合预测算法，该算法在对程序偏向性分析的基础上，选择一种合适的预测器进行分支预测。由于要同时实现多种预测器，硬件逻辑比一般预测算法增加数倍，因此在实际中处理器很少采用该算法。

神经网络算法是近年来提出的一种分支预测算法^[5]，它将历史信息作为神经元的输入，在预测过程中进行自发学习和训练，获得相应神经元权重，最终依据输入及相应权重计算获得最终预测输出。神经网络算法存在以下 2 点较突出的不足：

- (1)未能解决循环的最后一次预测问题。
- (2)以全地址作为神经元和权重的索引方式，对每一条分支都要建立一个表项，需占用大量资源。若表项资源有限，则预测精度将大受影响。

因此从某种意义上来说，这种神经网络算法的实现并不能获得实质上的精度提升。

3 复合分支预测机制的实现

实际应用中，通用预测算法无法在任意场合都获得最佳预测效果。嵌入式处理器是针对嵌入式应用环境开发的处理器，在设计过程中，须结合嵌入式环境的特点对各个功能部件进行优化实现，才能获得整体性能的提高。复合分支预测机制就是根据嵌入式处理器应用环境的特点，提出的一种特殊优化预测算法。嵌入式环境一般具有以下 2 个特点：

- (1)单一性：针对于特定的应用，即程序的输入集合较为单一。
- (2)稳定性：程序能稳定执行，较少采用虚存管理，即指令块在内存中的位置较少改变。

基于上述特点，复合分支预测机制一方面通过定制的 BTB，对稳定的循环体实现精确预测；另一方面通过基于全局模式索引的神经网络算法，对非循环逻辑构成的较为单一的控制流实现准确的跳转判断。

3.1 定制的 BTB 结构

本文将程序执行过程中稳定多次跳转的指令定义为真循环指令和伪循环指令 2 类。真循环指令是语言逻辑上完成循环条件判断的指令，伪循环指令则是非循环却发生稳定多次跳转的指令。若同一指令连续完成 2 次以上成功跳转，则将

其定义为循环逻辑指令。

定制 BTB 结构的应用背景是在嵌入式应用的单一性条件下，对循环逻辑指令进行分支预测，包括真伪循环指令。循环的预测，由上文算法分析中可见，传统预测算法只能通过 2 位或 2 位以上饱和计数器解决进入循环时的条件预测问题，却没有提出跳出循环时的条件预测解决手段，由于仅依据少量历史信息进行分析，传统预测算法本质上不具备对任何下一位的确定预测。但结合嵌入式环境特点，在嵌入式应用的单一性条件下，对历史信息的有效记录可实现实际分支跳转再现，本文算法将借助 BTB 完成此项工作。

BTB 采用的是解决由于跳转指令造成流水指令空槽的有效手段。传统的 BTB 设计主要用于记录索引信息和目标地址，对 BTB 进行索引命中后，对条件转移指令通常通过预测算法进行跳转预测。在定制设计中，笔者通过增加 BTB 表项内容的方式对循环信息进行主动记录，新增的表项域包括跳转计数器、历史计数器 1、有效位 1、历史计数器 2、有效位 2。其中，跳转计数器表示当前循环成功跳转的次数；历史计数器 1 和 2 表示该指令最近 2 次最大连续成功跳转次数；有效位 1 和 2 表示最近一次连续跳转次数是否与相应历史计数器记录相符，在同一时刻这两位相斥。

即使是嵌入式环境也可能发生程序输入变换或程序结束等事件。当事件发生时，历史计数器 1 失效，此时使历史计数器 2 被置有效，并进行预测。由于嵌入式系统事件单一特性，使用多个历史计数器可尽量保留事件的可重现性，提高预测准确率。

本文将有限的 BTB 资源交给循环逻辑使用，同时支持大概率发生事件的可重现性，在嵌入式环境中解决了传统算法中最后一次循环跳转指令预测失败的问题。

3.2 神经网络算法的实现

本文将未被 BTB 命中的非直接跳转指令判断为非循环逻辑指令，针对此类指令的预测，在不考虑资源使用和硬件复杂度的情况下，已提出的神经网络算法可达高预测精度，其基本原理是将分支指令预测为大概率发生的历史事件。在实际设计中，由于处理器内部的存储器资源有限，故应考虑在算法和资源中寻求一个平衡点。在嵌入式应用环境当中，恰能提供此类支点。

嵌入式应用具有稳定性和单一性的特点，即嵌入式应用的分支流具有较强的全局相关性，因此可采用全局模式索引方式替代原算法中基于地址的索引方式。图 2 是基本神经网络模型。

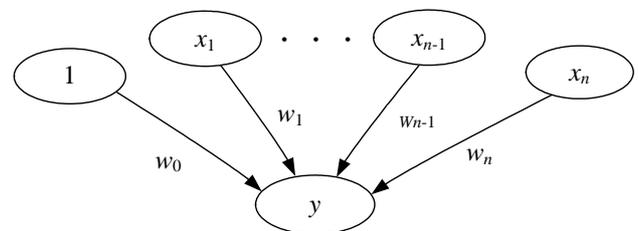


图 2 基本神经网络模型

在图 2 中， $\{1, x_1, \dots, x_{n-1}, x_n\}$ 是神经元集合，代表神经网络的输入； $\{w_0, w_1, \dots, w_{n-1}, w_n\}$ 是相应神经元的权重， w_0 是基准权重；最终输出可表示为

$$y = w_0 + \sum(x_i \times w_i)$$

其中，神经元集合是一定数量全局历史记录的集合，以 1 表

示跳转, -1 表示不跳转;神经元的权重即相应历史记录权重。神经网络算法实现如图 3 所示。

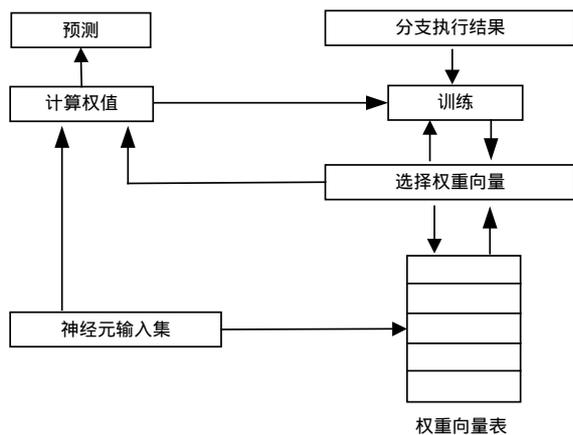


图 3 基于全局模式索引的神经网络算法

神经网络算法主要由 2 部分构成：(1)通过神经元输入集和权重计算权值进行预测。(2)利用实际分支执行结果对相应权重向量进行训练：得到分支结果后，历史记录中与分支结果相同的对应权重加 step(训练步进)，其余减 step。

在该算法中，历史记录能以某种加权方式实现分支预测相对真实结果的大概率逼近，最终通过训练获得稳定概率

4 性能模拟

针对提出的复合分支预测算法，本文利用 SimpleScalar 进行相关性能模拟试验。

4.1 试验结果

选用 SPEC2000 部分程序作为测试程序，由表 2 可见利用 GCC 优化编译后的结果。

表 2 SPEC2000 部分程序信息

SPEC2000	静态指令数	动态执行指令数	转移指令数
1641gzip	110 032	2 601 833 368	401 247 847
1761gcc	416 483	1 441 579 905	339 802 089
1811mcf	100 240	164 413 518	38 323 047
1971parser	120 448	3 610 491 690	822 986 833
2551vortex	275 163	9 047 485 398	1 585 575 907
300 twolf	171 455	1 239 924 029	227 348 149

对测试程序进行简单分析后，结合 Bimodal, Gshare 及混合算法的试验数据可得失效率对比见图 4。

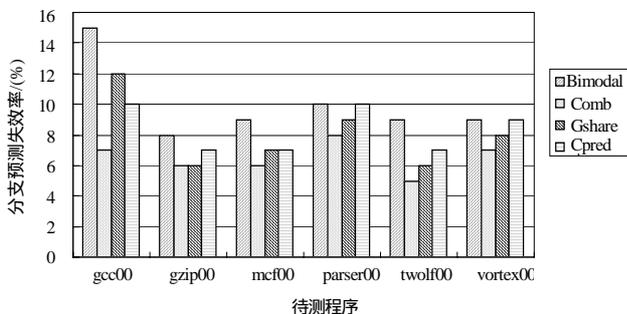


图 4 失效率对比结果

对比结果表明，复合分支预测算法与其他几种算法失效

率相当，平均成功预测率在 90%~93%左右。上述试验仅在优化编译后单次执行进行预测，实际上复合分支算法是一种针对嵌入式应用环境的预测算法，在满足单一性与稳定性的条件下将可达到更优的预测效果。

4.2 相关参数对预测结果的影响

复合分支预测算法主要包含以下 2 个部分：

(1) 针对循环预测的 BTB 结构

循环预测 BTB 结构产生的预测精度由 BTB 表项及历史计数器的位数决定。增加 BTB 表项可增加历史信息的记录量，提高命中率；增加计数器位数可降低误测率的下限值。本文采用 10 位计数器，使误测率下限降低到 0.1%左右。

(2) 针对非循环控制流的神经网络算法

神经网络算法同样受资源的影响。本文采用 6 位神经元输入，使用 6 位最近历史信息输入，因此权重向量表包含 64 个表项，权重向量表项中的每一位权重用 5 位表示，最大权重为 32，训练步进设置为 2。理论上神经元的数量、权重的位数与预测精度成正比，选择合适的训练步进能调整训练时间，使预测率较快达到稳定状态。本文通过对参数的多次调整，使算法的性能和资源占用达到平衡。

5 结束语

动态分支预测是计算机体系结构的研究重点，由目前动态分支预测算法的发展可见，动态分支理论在成熟的体系结构下几乎已难有实质性突破。但在具体应用中，仍可有较多途径来提升预测精度。

目前已有较多新的预测手段应用在专有领域。本文根据嵌入式应用的特殊环境及特点，提出一种复合式分支预测算法，可在嵌入式条件下解决一般预测算法无法避免的 2 个基本问题，同时能以更低的硬件复杂度和资源占用得到较高的预测精度。SimpleScalar 性能模拟表明，这种复合预测算法即使在通用预测领域也能达到较好的预测效果。

参考文献

- [1] Hennessy J L, Patterson D A. Computer architecture: A Quantitative Approach[M]. 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.
- [2] Yeh T Y, Patt Y N. Two-level Adaptive Training Branch Prediction[C]//Proc. of the 24th Annual Int. Symp. on Microarchitecture. Albuquerque, USA: [s. n.], 1991: 51-61.
- [3] Yeh T Y, Patt Y N. Alternative Implementations of Two-level Adaptive Branch Prediction[C]//Proc. of the 19th Int. Symp on Computer Architecture. Queensland, Australia: [s. n.], 1992: 124-134.
- [4] McFaring S M. Combining Branch Predictors[R]. Digital Western Research Laboratory, Tech.Rep.:TN-36, 1993.
- [5] Jiménez A D, Improved Latency and Accuracy for Neural branch prediction[J]. ACM Transaction on Computer System, 2005, 23(2): 197-218.