基于循环的指令高速缓存访问预测方法

梁　静，陈志坚，孟建熠
（浙江大学 超大规模集成电路研究所，杭州 310007）

摘　要：为了减少高速缓存访问功耗，提出了一种针对循环的基于历史访问路径的指令高速缓存访问预测方法。该方法以循环作为高速缓存访问路预测行为开启的先决条件，通过指令高速缓存的历史访问路径训练预测器。当循环体再次进入时选择对应的访问路径预测器，获取目标指令高速缓存的路进行访问，降低访问功耗。并进一步提出多路径路预测方法，以得到更高的预测准确率。基于 Powerstone 测试基准的实验结果表明，该预测方法能达到 99% 的预测准确率。相比传统的指令高速缓存，使用本方法的高速缓存可平均降低 65% 的访问功耗，仅增加约 0.2% 的平均指令高速缓存访问周期。

关键词：指令高速缓存；路预测；循环；路径
中图分类号：TP301.6　　**文献标志码**：A　　**文章编号**：1001-3695(2012)07-2491-03
doi:10.3969/j.issn.1001-3695.2012.07.024

Loop based trace way prediction for instruction cache

LIANG Jing, CHEN Zhi-jian, MENG Jian-yi
(*Institute of VLSI Design*, *Zhejiang University*, *Hangzhou* 310007, *China*)

Abstract: This paper proposed a new scheme of way prediction, named loop based trace way prediction, for instruction cache, to decrease the access power. The scheme made loop the premise of starting way prediction. By recording the trace of cache access when it encountered a loop, it could predict the way when getting in the same loop. Also made a research on multiple-trace way prediction scheme to get a better hit rate. Experiment results from Powerstone show this scheme can get a hit rate of 99%. On average it can decrease 65% power dissipation compared to conventional set-associative cache, and increase only 0.2% the time to access cache.

Key words: instruction cache; way prediction; loop; trace

0　引言

现代计算机体系中，高速缓存作为协调处理器核心工作速度和外部存储工作速度的缓冲器，已经成为处理器系统中不可或缺的部分。随着处理器设计技术的发展，高速缓存的功耗在处理器整体功耗中所占的比重越来越大。StrongArm SA-110 中，高速缓存所占的功耗比重为 43%[1]；Pentium Pro 中，高速缓存所占的功耗比重为 33%[2]；Alpha 21264 中，高速缓存所占的功耗比重为 16%[3]。

为了减少缺失率，提升性能，高速缓存的映射方式一般为组相连。但由于多个缓存块被同时选中，此映射方式使得高速缓存的功耗大幅增加。为了解决组相连高速缓存多个缓存块同时开启所带来的问题，当前研究采用了提前使用标志域比较得到或预测得到访问路的方法，以降低功耗。分段高速缓存[4]将高速缓存的访问分成两个阶段：a）访问标志域，得到标志域比较结果；b）访问命中的一路数据块，得到所需数据。该方法不会对其他数据路作无必要的访问，减少了功耗，但由于是分段访问，极大地降低了性能。Inoue 等人[5]首次提出了使用路预测（way-predicting）来降低功耗。该方法使用 MRU（most recently used）信息记录每组最近访问的路，下次访问时选择访问 MRU 信息中记录的路。若预测正确，则返回数据结束；若预测失败，所有路都会被访问，结束后把正确的路信息写入 MRU 中。

路预测的基本思想是以历史指令译码信息、历史指令访问信息和当前指令 PC 信息来预测当前指令的访问。预测命中时，只需访问一路即可得到所需指令；预测错误时，需要再次访问正确的路。因此，如何在降低功耗的同时，保证处理器的性能，是路预测的主要研究内容。为了提升预测准确率，Xu 等人[6]提出了根据指令流来对高速缓存路预测进行分类预测的方案，将程序取指分成了顺序取指和跳转取指，依据其各自不同的特点进行预测。但该方案在没有指令集的配合下，需要每条指令配置一个路预测信息，资源利用率低，且无法过滤无效预测。还有一些研究将路预测与其他技术结合，以寻求进一步降低预测错误带来的性能损失[7~9]；另外一些研究则将路预测用于多核，但预测方案并无更新[10]。

本文提出了一种基于历史取指路径的指令高速缓存路预测结构，命名为路径路预测。其利用程序循环多次执行的特点，记录历史指令高速缓存的访问路径，用于预测再次进入相同循环时将访问的路。本文研究主要内容有：a）研究单路径路预测，每个循环只有一个记录路径；b）研究多路径路预测，

收稿日期：2011-12-26；**修回日期**：2012-02-10

作者简介：梁静（1987-），男，浙江台州人，硕士，主要研究方向为计算机体系结构、超大规模集成电路设计（liangjing@zju.edu.cn）；陈志坚（1984-），男，博士，主要研究方向为高性能低功耗嵌入式 CPU 设计、超大规模集成电路设计；孟建熠（1982-），男，博士，主要研究方向为高性能低功耗嵌入式 CPU 设计、超大规模集成电路设计.

每个循环根据分支信息细分为多个记录路径。

1 路径路预测原理及架构

指令高速缓存的访问可以分为两类,即同一缓存块的访问和跨缓存块的访问。前一类只需记忆前次访问的路,即可预测后续的访问;对于后一类的访问,本文提出了基于循环的路径路预测。

基于循环的路径路预测以循环起始作为检测路径是否存在的先决条件,而路径存在则是路预测开启的条件。因此只有在循环中且路径存在的情况下,路预测才会被开启,如此就能减少不必要的预测错误,减少预测错误带来的性能损失。

路径路预测系统有三个状态:空闲状态(idle)、训练状态(training)、预测状态(predict)。空闲状态时预测系统不工作,缓存访问为传统的并行访问;训练状态时预测系统记录缓存访问的路径,缓存访问仍为传统的并行访问;预测状态时预测系统使用历史路径进行缓存访问路预测,缓存访问为路预测访问。路预测系统状态之间转换的条件因具体的实现而不同,本文分析了单路径路预测和多路径路预测两种方法。

1.1 单路径路预测

单路径路预测在每个循环中只使用单个路径记录循环中的缓存访问历史。遇到循环时,系统检查该循环是否已经有历史路径记录。若无,进入训练状态,开始记录该历史路径,直到再次遇见循环或因资源耗尽后进入空闲状态;若有,进入预测状态,开始按照已有历史路径进行路预测,直到再次遇见循环或到达路径终端或因预测错误进入空闲状态。单路径路预测流程如图 1 所示。

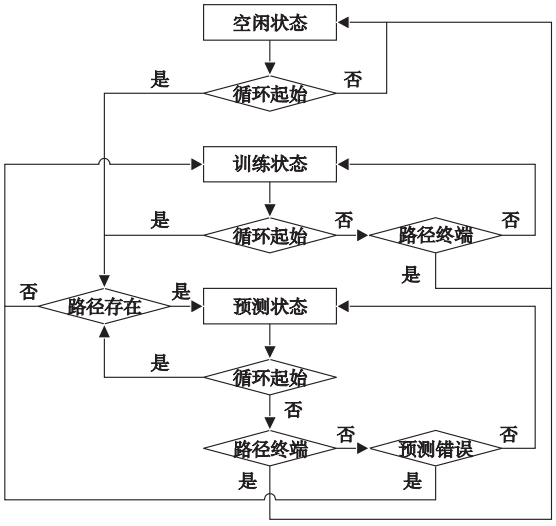

图1 单路径路预测流程

本文以反向跳转作为循环起始的标志,并记录反向跳转的 PC 值作为路径存在的标志。单路径路预测系统需要的资源有:

a)循环 PC 寄存器(rot_pc)。保存反向跳转指令的 PC,作为一个循环的起始标志。

b)路径选择寄存器(loop_sel)。在遇到反向跳转时,保存被替换的路径序号或匹配到的路径序号,用来索引之后将要替换或使用的路径。

c)路预测指针(step_ptr)。保存预测信息在当前使用路径中的位置。遇到跨缓存块的访问时加 1,路径改变时清 0。

d)状态寄存器(state)。保存当前路预测系统的状态。

e)路预测缓存(way prediction table,WPT)。在训练状态保存各个循环的路径,在预测状态提供路预测信息。

单路径路预测整体框架及其关键寄存器——路径选择寄存器、路预测指针和状态寄存器的更新与维护如图 2 所示。

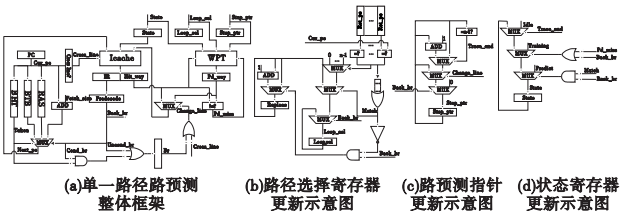

图2 单路径路预测整体框架及其关键寄存器的更新与维护

1.2 多路径路预测

由于指令流的改变,每个循环可能存在多个路径,因此本文在单路径路预测的基础上提出了多路径路预测。

多路径路预测在每个循环中使用多个路径记录循环中不同指令流的缓存访问历史。该系统与单路径预测系统在状态转换条件方面的不同之处在于,其在循环中需要检测路径是否因指令流的改变而发生了改变,并根据改变后的路径是否存在来决定之后的状态。若改变后的路径存在则进入预测状态;否则,进入训练状态。多路径路预测的流程如图 3 所示。

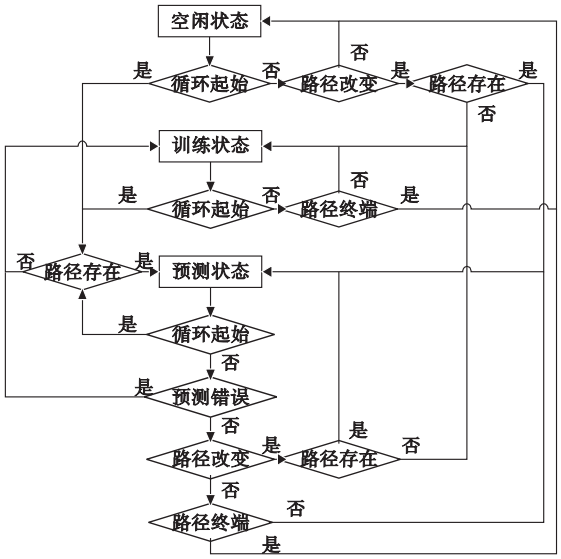

图3 多路径路预测流程

本文使用分支记录寄存器作为循环内多个路径的索引,该寄存器由循环内分支历史信息与分支指令的低位地址组成,可以体现指令流的变化。多路径路预测需要在每条路径前增加有效标志,以指示该分支路径是否有效。该标志在系统进入对应路径的训练状态时置为有效,在对应路径所属循环被替换后置为无效。多路径路预测结构与单路径预测结构类似。

2 实验与分析

本文选取嵌入式基准测试程序 Powerstone 中的部分应用作为目标程序,在 C-SKY 公司国产自主研发的 32 位高性能嵌入式处理器 CK610[11] 的仿真平台上进行实验。分析了各目标程序的高速缓存命中率和反向跳转概率,为后续的实验结果分析提供了必要的程序信息。Powerstome 程序中反向跳转和分支指令比例如表 1 所示。本文实验采用的指令高速缓存为 2 KB 大

小,4 路组相连,每个缓存块的大小为 32 B,存储 16 条指令。

表 1　Powerstone 程序中反向跳转和分支指令比例

程序名	总指令数	反向跳转比例/%	分支指令比例/%
adpcm	124 237	1.77	3.06
compress	282 569	1.27	5.24
crc	69 789	3.42	6.85
des	281 233	0.55	1.04
engine	749 784	2.63	4.49
pocsag	602 670	3.14	6.35
qurt	199 355	1.48	5.32
v42	213 513	1.99	2.68

2.1　路预测效率分析

本文以整体预测比例(pd_rat)、整体预测命中率(pd_hit)、跨块预测命中率(wpt_hit)作为预测效率的指标。整体预测比例为使用路预测进行高速缓存访问的指令数占总指令数的百分比,包括跨缓存块预测和缓存块内预测;整体预测命中率为预测路命中数占所有预测的百分比;跨块预测命中率为跨块预测中预测路命中数占所有跨块预测的百分比。

图 4 显示了在路径长度足够的情况下,采用单路径预测的测试程序预测效率平均值与循环记录数的关系。由图 4 可知,该测试程序集只需四个循环记录数即可达到很好的预测效率。图 5 显示了在循环记录数足够的情况下,采用单路径预测的测试程序预测效率平均值与路径长度的关系。由图 5 可知,该测试程序集在路径长度为 20 左右即可达到很好的预测效率。

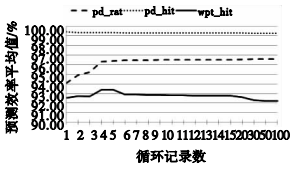

图4　预测效率平均值与循环记录数关系

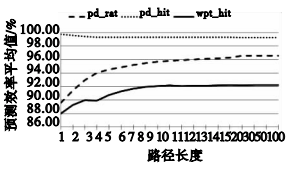

图5　预测效率平均值与路径长度关系

图 6 显示了在循环记录数和路径长度足够的情况下,采用多路径预测的预测效率平均值与分支记录寄存器位数的关系。由图 6 可知,在资源充足的情况下,采用多路径预测能够有更好的预测效率,而且分支记录寄存器的位数为 2,即每个循环有四个路径就能有很好的预测效率。

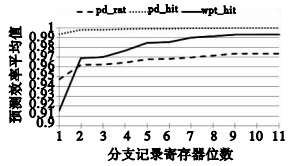

图6　预测效率平均值与分支记录寄存器位数关系

表 2 是在循环记录数为 4,单循环内总路径长度分别为 8、16、32 的条件下,分别对单路径和多路径进行实验的结果。由表 2 可知,在资源较少的情况下,路径长度对预测效率有很大的影响,因此单路径比多路径有更高的预测比例和准确率;而当资源增加时,路径长度已能满足需求,多路径预测的效果就能够得到体现。

2.2　功耗分析

高速缓存的访问功耗大致与访存位宽成正比。本文将以式(1)计算高速缓存的访问功耗:

$$E = N_{\text{bit}} \times C_{\text{acc}} \times e_{\text{bit}} \tag{1}$$

其中:N_{bit}为每次访存时的位宽;C_{acc}为程序中的总访存数;e_{bit}为每位访存的功耗。

表 2　同等资源下单路径预测与多路径预测的效率比较

总路径长度	路径类型	pd_rat/%	pd_hit/%	wpt_hit/%
8	单路径	95.24	99.38	93.21
	多路径	92.66	99.75	92.75
16	单路径	96.12	99.36	93.37
	多路径	95.19	99.78	95.47
32	单路径	96.32	99.36	93.39
	多路径	96.21	99.78	96.91

传统的高速缓存每次访存将开启所有路,采用四路组相连结构时,其访存功耗为

$$E = (4 \times N_{\text{data}} + 4 \times N_{\text{tag}}) \times C_{\text{acc}} \times e_{\text{bit}} \tag{2}$$

其中:N_{data}为高速缓存中数据域的位宽,本文采用 32 B,即 256;N_{tag}为高速缓存中标志域的位宽,本文采用 23。

采用路径路预测的高速缓存,在路预测开启并预测准确的情况下,只有标志域和一路数据域会被开启,其访存功耗为

$$E = (4 \times N_{\text{tag}} + 4 \times N_{\text{data}} - 3 \times N_{\text{data}} \times R_{\text{pd}} \times R_{\text{hit}} + N_{\text{way}} \times R_{\text{pd}} \times R_{\text{hit}}) \times C_{\text{acc}} \times e_{\text{bit}} \tag{3}$$

其中:R_{pd}为整体预测比例;R_{hit}为整体预测命中率;N_{way}为路预测需要的位数,四路组相连时为 2。

图 7 显示了循环记录数为 4,每个循环内总路径长度为 16 的单路径路预测高速缓存、多路径路预测高速缓存与传统四路组相连高速缓存的功耗对比。单路径路预测高速缓存访问功耗平均为传统高速缓存的 34.45%,多路径路预测高速缓存访问功耗平均为传统高速缓存的 34.81%。由于资源较少,多路径路预测的功耗优势未能够体现出来,但是在一些分支和反向跳转频率较高的程序上,如 crc 和 pocsag,多路径路预测已经能够有更好的效果。

2.3　性能分析

路径路预测高速缓存在预测命中时,访存只需一个周期;而在预测错误时,访存则需两个周期。图 8 显示了循环记录数为 4,每个循环内总路径长度为 16 的单路径路预测高速缓存、多路径路预测高速缓存相比传统四路组相连高速缓存平均每个访存行为增加的周期数。由图 8 可知,单路径路预测高速缓存访问周期平均增加 0.61%,多路径路预测高速缓存访问周期平均增加 0.21%。多路径路预测因其能更准确地定位路径使得整体预测错误减少,进而降低了性能损失。

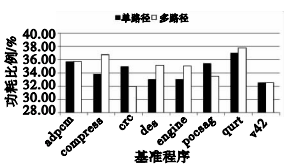

图7　路径预测高速缓存与传统高速缓存功耗对比

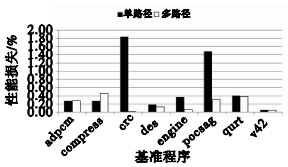

图8　路径预测高速缓存相比传统高速缓存的性能损失

3　结束语

本文提出了一种基于循环的路径路预测高速缓存,有选择性地开启路预测,在提高预测命中率的同时,减少了预测错误带来的性能损失,有效地降低了高速缓存的访问功耗。实验表明,对于 4 KB 的高速缓存,使用 16 B 的路预测缓存资源,单路径路预测高速缓存能够有 96.12% 的整体预测(下转第 2499 页)

(上接第 2493 页)比例、99.36% 的整体预测命中率和 93.37% 的跨块预测命中率,功耗减少达 65.55%,而只带来平均每个访存行为 0.61% 的性能损失;多路径路预测高速缓存能够有 95.19% 的整体预测比例、99.78% 的整体预测命中率和 95.47% 的跨块预测命中率,功耗减少达 65.19%,而只带来平均每个访存行为 0.21% 的性能损失。

参考文献:

[1] PATTERSON D A, HENNESSY J L. Computer organization and design: the hardware/software interface[M]. 3rd ed. San Francisco: Morgan Kaufmann,2005.

[2] MANNE S, KLAUSER A, GRUNWALD D. Pipeline gating: speculation control for energy reduction[C]//Proc of Annual International Symposium on Computer Architecture. Washington DC:IEEE Computer Society,1998:132-141.

[3] GOWAN M K, BIRO L L, JACKSON D B. Power considerations in the design of the alpha 21264 microprocessor[C]//Proc of Design Automation Conference. San Francisco: Association for Computing Machinery,1998:726-731.

[4] HASEGAWA A, KAWASAKI I, YAMADA K, *et al*. SH3: high code density, low power[J]. IEEE Micro,1995,15(6):11-19.

[5] INOUE K, ISHIHARA T, MURAKAMI K. Way-predicting set associative cache for high performance and low energy consumption[C]//Proc of International Symposium on Low Power Electronics and Design. New York:ACM Press,1999:273-275.

[6] XU Cui-ping, ZHANG Ge, HAO Shou-qing. Fast way-prediction instruction cache for energy efficiency and high performance[C]//Proc of IEEE International Conference on Networking, Architecture, and Storage:IEEE Computer Society,2009:235-238.

[7] 彭方. 路预测与可重构 Cache 的自适应低能耗算法研究[D]. 长沙:湖南大学,2008.

[8] CHEN Hsin-chuan, CHIANG Jen-shiun. Low-power way-predicting cache using valid-bit pre-decision for parallel architectures[C]//Proc of the 19th International Conference on Advanced Information Networking and Applications. [s.l.]: IEEE Press,2005:203-206.

[9] POWELLR M D, AGARWALR A, VIJAYKUMAR T N, *et al*. Reducing set-associative Cache energy via way-prediction and selective direct-mapping[C]//Proc of Annual International Symposium on Microarchitecture. Washington DC:IEEE Computer Society,2001:54-65.

[10] CHUNG C M, KIM J. Low-power L2 cache design for multi-core processors[J]. Electronics Letters,2010,46(9):618-620.

[11] C-SKY Microsystems. 32-bit high performance and low power embedded processor [EB/OL]. (2011-04-25) [2011-12-20]. http://www.c-sky.com.