



(12) 发明专利

(10) 授权公告号 CN 102662640 B

(45) 授权公告日 2015. 04. 01

(21) 申请号 201210105659. 4

US 2007/0294518 A1, 2007. 12. 20, 全文.

(22) 申请日 2012. 04. 12

CN 1725175 A, 2006. 01. 25, 全文.

CN 102306093 A, 2012. 01. 04, 全文.

(73) 专利权人 苏州睿云智芯微电子有限公司
地址 215300 江苏省苏州市昆山市巴城镇学院路 88 号

审查员 张会

(72) 发明人 孟林

(74) 专利代理机构 苏州广正知识产权代理有限公司 32234

代理人 张利强

(51) Int. Cl.

G06F 9/38(2006. 01)

(56) 对比文件

US 5835754 A, 1998. 11. 10, 说明书第 8 栏第 15 行—第 16 栏第 40 行, 图 1-5.

US 5835754 A, 1998. 11. 10, 说明书第 8 栏第 15 行—第 16 栏第 40 行, 图 1-5.

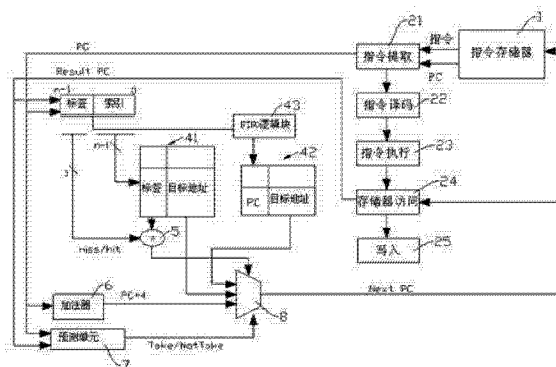
权利要求书2页 说明书6页 附图3页

(54) 发明名称

双重分支目标缓冲器和分支目标处理系统及处理方法

(57) 摘要

本发明公开了一种双重分支目标缓冲器, 包括: 第一分支目标缓冲器和第二分支目标缓冲器, 所述第一分支目标缓冲器包括多路组相连接结构的第一指令缓存单元, 所述第一指令缓存单元用于临时存储预取的指令, 每路组包括第一存储器和第二存储器; 所述第二分支目标缓冲器包括多路组相连接结构的第二指令缓存单元, 所述第二指令缓存单元用于临时存储预取指令, 每个路组包括第三存储器和第四存储器。本发明双重分支目标缓冲器, 有两个分支目标缓冲器, 减少不同的命令在相同的 index 里登陆的可能, 提高分支目标缓冲器正确的预测精度, 有效加快中央处理器的速度。



1. 一种分支目标处理系统,其特征在于,包括:CPU 整数单元、双重分支目标缓冲器、分支目标预测单元、加法器、比较器和选择器;

所述 CPU 整数单元,采用流水线结构,包括指令提取单元和位于所述指令提取单元后续的指令执行单元,所述指令提取单元将提取的指令地址分别传输给双重分支目标缓冲器、加法器和分支目标预测单元,所述指令执行单元的后续单元将执行结果地址分别传输给双重分支目标缓冲器和分支目标预测单元;

所述分支目标预测单元,对分支条件成立与否进行预测,并将结果传输给选择器;

所述加法器,将执行指令地址进行处理,得到顺序的下一条指令地址,并将结果传输给选择器;

所述双重分支目标缓冲器,包括:第一分支目标缓冲器和第二分支目标缓冲器,所述第一分支目标缓冲器包括多路组相连接结构的第一指令缓存单元,所述第一指令缓存单元用于临时存储预取的指令,每路组包括第一存储器和第二存储器,所述第一存储器用于存储地址标签位,所述第二存储器用于存储所述地址标签位所对应的有条件分支命令目标地址;所述第二分支目标缓冲器包括多路组相连接结构的第二指令缓存单元,所述第二指令缓存单元用于临时存储预取指令,每个路组包括第三存储器和第四存储器,所述第三存储器存储指令地址,所述第四存储器存储指令地址所对应的无条件分支命令目标地址;

指令地址包括索引位和标签位,所述指令地址的索引位取出对应第一分支目标缓冲器中的标签位传输给比较器,与指令地址的标签位相比较,将结果传输给选择器,同时将所述第一分支目标缓冲器中的标签位对应的目标地址传输给选择器;同时检索所述第三存储器,如果存在相同的指令地址,则取出与指令地址相对应的目标地址传输给选择器;

所述选择器,对上述指令的数据进行选择,将待执行的下一条指令传输给所述 CPU 整数单元,即如果预测结果为成立,同时比较结果为相同,选择器选择所述第一分支目标缓冲器中的目标地址传输给所述 CPU 的整数单元;如果预测结果为成立,而比较结果为不同,选择器选择所述第二分支目标缓冲器中的目标地址传输给所述 CPU 的整数单元;如果预测结果为不成立,选择器则选择加法器的计算结果传输给所述 CPU 的整数单元。

2. 一种分支目标处理方法,其特征在于,

(100) 提供一种分支目标处理系统,包括:CPU 的整数单元、双重分支目标缓冲器、分支目标预测单元、加法器、比较器和选择器;所述 CPU 的整数单元,采用流水线结构,至少包括指令提取单元和位于所述指令提取单元后续的指令执行单元;所述双重分支目标缓冲器,包括:第一分支目标缓冲器和第二分支目标缓冲器,所述第一分支目标缓冲器包括多路组相连接结构的第一指令缓存单元,所述第一指令缓存单元用于临时存储预取的指令,每路组包括第一存储器和第二存储器,所述第一存储器用于存储地址标签位,所述第二存储器用于存储所述地址标签位所对应的有条件分支命令目标地址;所述第二分支目标缓冲器包括多路组相连接结构的第二指令缓存单元,所述第二指令缓存单元用于临时存储预取指令,每个路组包括第三存储器和第四存储器,所述第三存储器存储指令地址,所述第四存储器存储指令地址所对应的无条件分支命令目标地址;

(200)、所述指令提取单元提取指令地址分别传输给双重分支目标缓冲器、加法器和分支目标预测单元,

(300)、所述分支目标预测单元,对分支条件成立与否进行预测,并将预测结果传输给

选择器；所述加法器，将执行指令地址进行处理，得到顺序的下一条指令地址，并将计算结果传输给选择器；

(400)、指令地址包括索引位和标签位，所述指令地址的索引位取出对应第一分支目标缓冲器中的标签位传输给比较器，与指令地址的标签位相比较，将比较结果传输给选择器，同时将所述第一分支目标缓冲器中的标签位对应的目标地址传输给选择器；同时检索所述第三存储器，如果存在相同的指令地址，则取出与指令地址相对应的目标地址传输给选择器；

(500)、如果预测结果为成立，同时比较结果为相同，选择器选择所述第一分支目标缓冲器中的目标地址传输给所述 CPU 的整数单元；如果预测结果为成立，而比较结果为不同，选择器选择所述第二分支目标缓冲器中的目标地址传输给所述 CPU 的整数单元；如果预测结果为不成立，选择器则选择加法器的计算结果传输给所述 CPU 的整数单元。

3. 根据权利要求 2 所述的一种分支目标处理方法，其特征在于，还包括步骤(600)、分支命令执行完毕后，进行更新；如果分支命令是无条件分支命令的话，检索第二分支目标缓冲器，如果有相同指令地址存在就不用登录，如果没有相同的存在，寻找空处，将指令地址登录到第三存储器中，下一个命令的地址登录到第四存储器中；如果命令是有条件分支命令而且预测结果为成立的情况下，将在第一分支目标缓冲器的对应的索引位内登录标签位和目标地址。

4. 根据权利要求 3 所述的一种分支目标处理方法，其特征在于，所述第二分支目标缓冲器的第四存储单元为 CAM 构造，以先进先出的原则进行覆盖。

双重分支目标缓冲器和分支目标处理系统及处理方法

技术领域

[0001] 本发明涉及具有分支预测单元的处理器,特别是涉及一种分支预测单元内的分支目标缓冲器(Branch target buffer, BTB)。

背景技术

[0002] 现有的计算器通常为流水线体系结构,“流水线”(或也称作“推测执行”)是一个术语,通常指其中使用一系列功能步骤或处理阶段来处理指令序列的操作方法,每个处理阶段通常在单个时钟周期内完成。每个处理阶段通常在单个时钟周期内完成它的一个或多个构成操作。

[0003] 与在开始下一指令之前将每条指令处理完成的非流水线处理器不同,流水线处理器在流水线的不同处理阶段同时处理几条指令。流水线阶段可以由设计者任意指定,但通常包括:指令获取(fetch)、指令译码(Decode)、指令执行(Execute)、执行解析阶段。

[0004] 指令获取阶段从指令当前被存储的任何所在(例如,主系统存储器或者指令队伍)检索指令。一旦获取,指令被传递到译码器阶段,其通常确定指令地址和/或指令操作数。指令从译码器阶段传递到执行由指令指示的一个或多个操作的执行阶段。执行解析阶段通常涉及将通过执行指令生成的结果(例如,结果数据)回写(write-back)到一个或多个寄存器或存储器以备以后使用。

[0005] 流水线提供了毫无疑问的性能收益,只要待处理的指令序列保持高度线性或者可预测。不幸的是,大多数指令序列包含众多的能够引入非顺序执行路径的指令。所谓的“分支指令”(包括,例如,跳转,返回和条件分支指令)在流水线处理器中产生了显著的性能损害,除非实现分支预测的有效形式。在未预测到(或错误预测)的分支指令引起偏离处理器内当前流水线中的指令序列时出现性能损害。如果发生这种情况,则当前流水指令序列就必须被扔掉或“清除”,而新的指令序列必须加载到流水线中。流水线清除浪费了众多时钟周期并且通常使处理器的执行变慢。

[0006] 分支目标缓冲器预测的成败影响着中央处理器(CPU)的速度,越来越被重视。分支目标缓冲器用来预测分支命令的下一个命令的地址的。它在CPU中的回路构成如图1所示。

[0007] 现有分支目标缓冲器构成以及预测方法如下:PC为Program Counter,它为命令所在的地址,它被分为Tag和Index;BTB为现有分支目标缓冲器,它包含了分支命令的地址的Tag和分支命令所目标地址(Target Address);它的动作包括下一个地址的预测,和得到地址之后的BTB的更新。

[0008] 详细动作是:当分支命令到来时,开始预测,在BTB中,用PC的index从对应的Index中取出BTB的Tag,并与PC的tag进行比较,生成miss/hit信号,如果2个tag相同,信号为hit,如果2个tag不相同,信号为miss。这将作为BTB下方的选择器的控制信号。另外从对应的Index中取出Target Address,投入BTB下方的选择器,待被选择。

[0009] 上述BTB存在的问题为:不同的命令有在相同的index里登录的可能,这样将导致

分支目标缓冲器无法正确的预测。

[0010] 有文献(J. K. F. Lee and A. J. Smith, "Branch prediction strategies and branch target buffer design," IEEE Comput. Mag. pp. 6-22, Jan. 1984)公开了如图 2 所示的 BTB 结构,这种方法虽然能缓解 BTB 的错误,但是仍存在大量的错误。不同的命令有在相同的 index 里登录的可能,这样将导致分支目标缓冲器无法正确的预测。

[0011] 另一文献(Y. Ishii, T. Sawada, K. Kuroyanagi, M. Inaba, K. Hiraki, "Bimode Cascading: Adaptive Rehashing for ITTAGE Indirect Branch Predictor", 2nd JILP Workshop on Computer Architecture Competitions (JWAC-2), 2011) 公开了如图 3 所示的 BTB 结构,这个方法是利用不同长的 Globe History Register (GHR) 和 PC (Program Counter) 来生成 index,从而检索对因的地址来作为下一个命令的地址。

发明内容

[0012] 本发明主要解决的技术问题是提供一种双重分支目标缓冲器,能够减少预测错误,有效加快中央处理器(CPU)的速度。

[0013] 为解决上述技术问题,本发明采用的一个技术方案是:本发明提供一种分支目标处理系统,包括:CPU 整数单元、双重分支目标缓冲器、分支目标预测单元、加法器、比较器和选择器;

[0014] 所述 CPU 整数单元,采用流水线结构,包括指令提取单元和位于所述指令提取单元后续的指令执行单元,所述指令提取单元将提取的指令地址分别传输给双重分支目标缓冲器、加法器和分支目标预测单元,所述指令执行单元的后续单元将执行结果地址分别传输给双重分支目标缓冲器和分支目标预测单元;

[0015] 所述分支目标预测单元,对分支条件成立与否进行预测,并将结果传输给选择器;

[0016] 所述加法器,将所述执行指令地址进行处理,得到顺序的下一条指令地址,并将结果传输给选择器;

[0017] 所述双重分支目标缓冲器,包括:第一分支目标缓冲器和第二分支目标缓冲器,所述第一分支目标缓冲器包括多路组相连接结构的第一指令缓存单元,所述第一指令缓存单元用于临时存储预取的指令,每路组包括第一存储器和第二存储器,所述第一存储器用于存储地址标签位,所述第二存储器用于存储所述地址标签位所对应的有条件分支命令目标地址;所述第二分支目标缓冲器包括多路组相连接结构的第二指令缓存单元,所述第二指令缓存单元用于临时存储预取指令,每个路组包括第三存储器和第四存储器,所述第三存储器存储指令地址,所述第四存储器存储指令地址所对应的无条件分支命令目标地址;

[0018] 指令地址包括索引位和标签位,所述指令地址的索引位取出对应第一分支目标缓冲器中的标签位传输给比较器,与指令地址的标签位相比较,将结果传输给选择器,同时将所述第一分支目标缓冲器中的标签位对应的目标地址传输给选择器;同时检索所述第三存储器,如果存在相同的指令地址,则取出与指令地址相对应的目标地址传输给选择器;

[0019] 所述选择器,对上述指令的数据进行选择,将待执行的下一条指令传输给所述 CPU 整数单元。

[0020] 本发明还提供一种分支目标处理方法,具体步骤包括:

[0021] (100) 提供一种分支目标处理系统,包括:CPU 的整数单元、双重分支目标缓冲器、分支目标预测单元、加法器、比较器和选择器;所述 CPU 的整数单元,采用流水线结构,至少包括指令提取单元和位于所述指令提取单元后续的指令执行单元;所述双重分支目标缓冲器,包括:第一分支目标缓冲器和第二分支目标缓冲器,所述第一分支目标缓冲器包括多路组相连接结构的第一指令缓存单元,所述第一指令缓存单元用于临时存储预取的指令,每路组包括第一存储器和第二存储器,所述第一存储器用于存储地址标签位,所述第二存储器用于存储所述地址标签位所对应的有条件分支命令目标地址;所述第二分支目标缓冲器包括多路组相连接结构的第二指令缓存单元,所述第二指令缓存单元用于临时存储预取指令,每个路组包括第三存储器和第四存储器,所述第三存储器存储指令地址,所述第四存储器存储指令地址所对应的无条件分支命令目标地址;

[0022] (200)、所述指令提取单元提取指令地址分别传输给双重分支目标缓冲器、加法器和分支目标预测单元,

[0023] (300)、所述分支目标预测单元,对分支条件成立与否进行预测,并将预测结果传输给选择器;所述加法器,将所述执行指令地址进行处理,得到顺序的下一条指令地址,并将计算结果传输给选择器;

[0024] (400)、指令地址包括索引位和标签位,所述指令地址的索引位取出对应第一分支目标缓冲器中的标签位传输给比较器,与指令地址的标签位相比较,将比较结果传输给选择器,同时将所述第一分支目标缓冲器中的标签位对应的目标地址传输给选择器;同时检索所述第三存储器,如果存在相同的指令地址,则取出与指令地址相对应的目标地址传输给选择器;

[0025] (500)、如果预测结果为成立,同时比较结果为相同,选择器选择所述第一分支目标缓冲器中的目标地址传输给所述 CPU 的整数单元;如果预测结果为成立,而比较结果为不同,选择器选择所述第二分支目标缓冲器中的目标地址传输给所述 CPU 的整数单元;如果预测结果为不成立,选择器则选择加法器的计算结果传输给所述 CPU 的整数单元。

[0026] 在本发明一个较佳实施例中,还包括步骤(600)、分支命令执行完毕后,进行更新:如果分支命令是无条件分支命令的话,检索第二分支目标缓冲器,如果有相同指令地址存在就不用登录,如果没有相同的存在,寻找空处,将指令地址登录到第三存储器中,下一个命令的地址登录到第四存储器中;如果命令是有条件分支命令而且预测结果为成立的情况下,将在第一分支目标缓冲器的对应的索引位内登录标签位和目标地址。

[0027] 在本发明一个较佳实施例中,所述第二分支目标缓冲器的第四存储单元为 CAM 构造,以先进先出的原则进行覆盖。

[0028] 本发明的有益效果是:本发明双重分支目标缓冲器,有两个分支目标缓冲器,减少不同的命令在相同的 index 里登录的可能,提高分支目标缓冲器正确的预测精度,有效加快中央处理器(CPU)的速度。

附图说明

[0029] 图 1 是现有技术中一种分支目标缓存器的结构示意图;

[0030] 图 2 是现有技术中又一种分支目标缓存器的结构示意图;

[0031] 图 3 是现有技术中再一种分支目标缓存器的结构示意图;

- [0032] 图 4 是本发明分支目标处理系统的结构示意图；
[0033] 图 5 是本发明分支目标预测错误比率图；
[0034] 图 6 是本发明双重目标缓存器的提速效果示意图。

具体实施方式

[0035] 下面结合附图对本发明的较佳实施例进行详细阐述,以使本发明的优点和特征能更易于被本领域技术人员理解,从而对本发明的保护范围做出更为清楚明确的界定。

[0036] 请参阅图 4,本发明实施例包括:

[0037] 一种分支目标处理系统,包括:CPU 的整数单元(未标注)、双重分支目标缓冲器(未标注)、比较器 5、加法器(adder) 6、分支目标预测单元(PHT) 7 和选择器 8;

[0038] 所述 CPU 的整数单元,采用流水线结构,依次包括指令提取(fetch)单元 21、指令译码(decode)单元 22、指令执行(execute)单元 23、存储器访问(memory access)单元 24 和写入(write back)单元 25,所述指令提取单元 21 将从指令存储器 1 中提取的指令地址(PC)分别传输给双重分支目标缓冲器、加法器 6 和分支目标预测单元 7,所述指令执行单元 23 的后续单元存储器访问单元 24 将执行结果地址分别传输给双重分支目标缓冲器和分支目标预测单元 7;

[0039] 所述分支目标预测单元 7,对分支条件成立与否进行预测,并将预算结果 taken/not taken 传输给选择器 8;

[0040] 所述加法器 6,将所述执行指令地址进行处理,得到顺序的下一条指令地址,并将结果传输给选择器,以一条指令 32bit 为例,顺序的下一条指令 PC+4;

[0041] 所述双重分支目标缓冲器,包括:第一分支目标缓冲器和第二分支目标缓冲器,所述第一分支目标缓冲器包括第一指令缓存单元 41,用于临时存储预取的指令,采用多路组相连接结构,每个路包括第一存储器和第二存储器,所述第一存储器存储地址标签位(tag),所述第二存储器存储所述标志位相对应的有条件分支命令目标地址(target address);所述第二分支目标缓冲器包括第二指令缓存单元 42,用于临时存储预取指令,采用多路组相连接结构,每个路包括第三存储器和第四存储器,所述第三存储器存储指令地址(PC),所述第四存储器存储指令地址相对应的无条件分支命令目标地址(target address);

[0042] 指令地址包括索引位(index)和标签位(tag),所述指令地址的索引位取出对应第一分支目标缓冲器中的标签位传输给比较器 5,与指令地址的标签位相比较,将比较结果传输给选择器,同时将所述第一分支目标缓冲器中的标签位对应的目标地址传输给选择器 8;同时检索所述第三存储器,存在相同的指令地址,则取出与指令地址相对应的目标地址传输给选择器 8;

[0043] 所述选择器 8,将上述指令结果进行选择,将待执行的下一条指令(next PC)传输给指令存储器 1 和存储器访问单元 24。

[0044] 当分支命令到来时,开始预测:

[0045] 1. 在 CBTB 中,用 PC 的 index 从对应的 Index 中取出 BTB 的 Tag,并与 PC 的 tag 进行比较,生成 miss/hit 信号,如果 2 个 tag 相同,信号为 hit, 如果 2 个 tag 不相同,信号为 miss。这将作为 CBTB 下方的选择器的控制信号。

- [0046] 另外从对应的 Index 中取出 Target Address,投入 CBTB 下方的选择器,待被选择。
- [0047] 2. 在 NBTB 中,检索相同的 PC,如果存在取出对应的 Target Address,投入 CBTB 下方的选择器,待被选择。
- [0048] 同时利用加法器(Adder) 将 Address 加上 4,投入 CBTB 下方的选择器,待被选择。
- [0049] 最后如果 PHT 的控制信号是 Taken 时,
- [0050] 如果 miss/hit 是 Hit, 选择 CBTB,生成的 Target Address 作为下一个命令的地址。
- [0051] 如果检索到相同的 PC,选择 NBTB,生成的 Target Address 作为下一个命令的地址。
- [0052] 如果都没有,则等待命令执行完毕。
- [0053] 如果 PHT 的控制信号是 Not Taken 时,则选择加法器(Adder) 的结果
- [0054] 当分支命令执行完毕后,开始登录(更新)
- [0055] 这时候由于结果都出来了 所以利用计算结果登录即可。
- [0056] 关于 NBTB :
- [0057] 如果命令是 non-conditional 分支命令的话,检索 NBTB,如果有相同命令存在就不用登录,如果没有相同的存在,寻找空处,将地址等陆到 PC 里,下一个命令的地址等陆到 Target Address 里。
- [0058] 另外因为 NBTB 为 CAM 构造,所以存在变满的情况,所以我们给 CAM 一个 Check point,实现 CAM 的 FIFO(First In First Out) 机能。也就是说当满了的时候,新的 non-conditional 分支命令将覆盖掉最先被登录 NBTB 的 index 中。
- [0059] 关于 CBTB
- [0060] 如果命令是 conditional 分支命令而且结果是 PHT 为 Taken 的情况下,将在 CBTB 的对应的 index 内登录 Tag 和 Target Address。
- [0061] 分支命令分为无条件(non-conditional)分支命令和有条件(conditional)分支命令。
- [0062] 分支命令出错主要分成以下几种情况 :
- [0063] 1、不一样的 non-conditional 分支命令连续 Access 相同的 BTB entry 时会出错 ;
- [0064] 2、non-conditional 分支命令后为 conditional 分支命令 Taken 的话,连续 Access 相同的 BTB entry 时会出错 ;
- [0065] 3、non-conditional 分支命令后为 conditional 分支命令 NotTaken 的话,连续 Access 相同的 BTB entry 时不会出错 ;
- [0066] 4、conditional 分支命令 Taken 后为的 non-conditional 分支命令后为话,连续 Access 相同的 BTB entry 时会出错 ;
- [0067] 5、conditional 分支命令 NotTaken 后为的 non-conditional 分支命令后为话,连续 Access 相同的 BTB entry 时不会出错 ;
- [0068] 以上为所有 Parten 3/5 都会出错。
- [0069] 图 5 为各种情况的在相关程序中所有错误中的比率
- [0070] CB_CB _Miss 为 :conditional 分支命令后为 conditional 分支命令,出错的比

率；

[0071] CB_NB _Miss 为 :conditional 分支命令后为 non-conditional 分支命令,出错的比率；

[0072] NB_CB _Miss 为 :non-conditional 分支命令后为 conditional 分支命令,出错的比率；

[0073] NB_NB _Miss 为 :non-conditional 分支命令后为 non-conditional 分支命令,出错的比率；

[0074] 我们发现预测的失败与无条件命令的关系比较强,无条件跳转总是要执行的,所以总是要使用 BTB 表,有条件跳转不一定执行,当判断条件不成立(not taken),就不需要使用 BTB 表。而且,non-conditional 的种类不是十分很多。因而使用 non-conditional 分支命令用和 conditional 分支命令用的新的双重分支目标缓冲器的方案,可有效提高预测正确率且不会带来较高的硬件成本。第一分支目标缓冲器是利用了原来的 BTB,用于 conditional 分支命令的预测的机构(第一分支目标缓冲器为 SRAM 构成),第二分支目标缓冲器是追加的用于 non-conditional 分支命令的预测(CAM)。第二分支目标缓冲器里放的是 non-conditional 分支命令的地址(PC)和目标地址(Target Address),更新和预测是通过检索 PC 来实现的。

[0075] 图 6 表示了提案的效果。128entry 的时候得到了 4% 的提速,256entry 的时候达到 5% 的提速。

[0076] 以上所述仅为本发明的实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

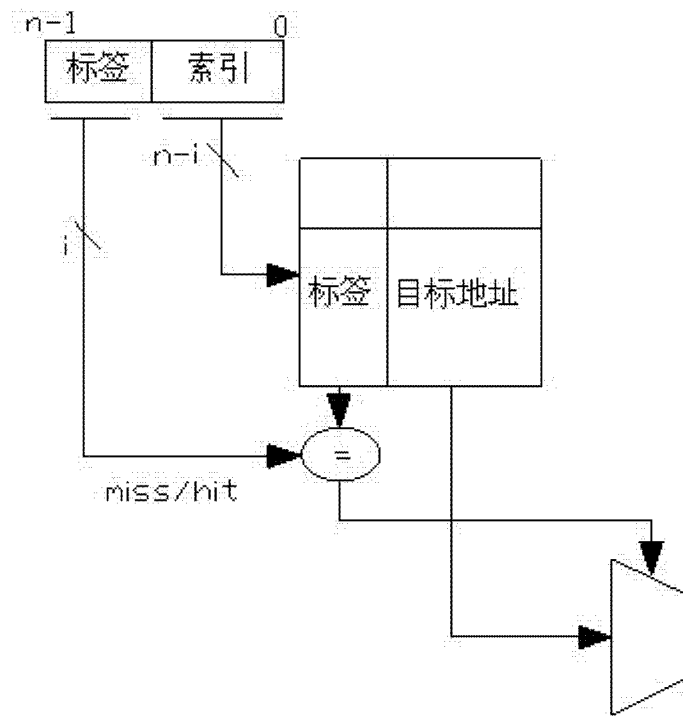


图 1

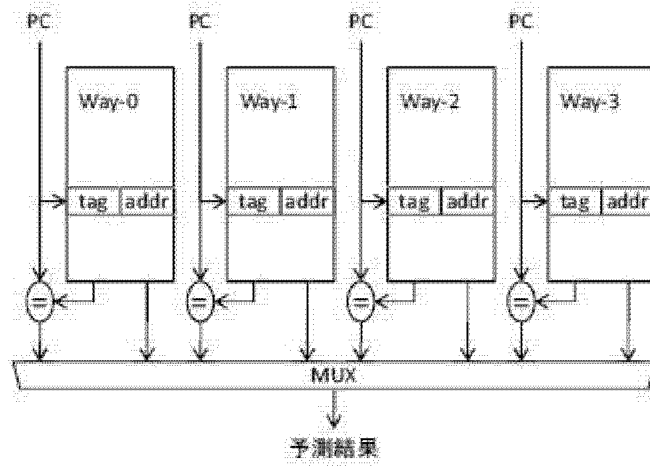


图 2

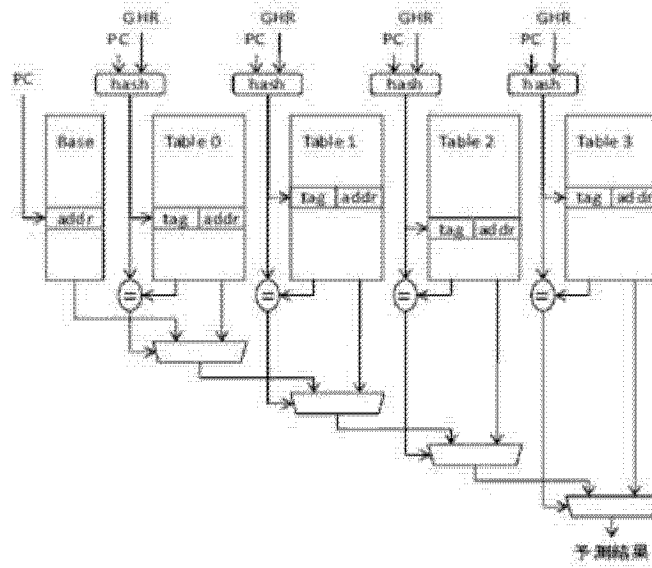


图 3

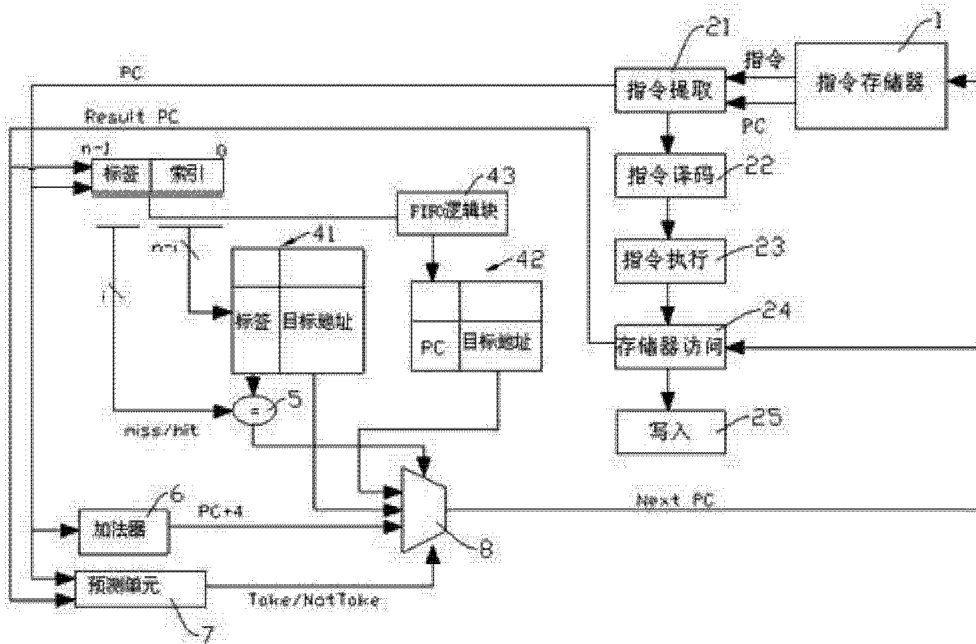


图 4

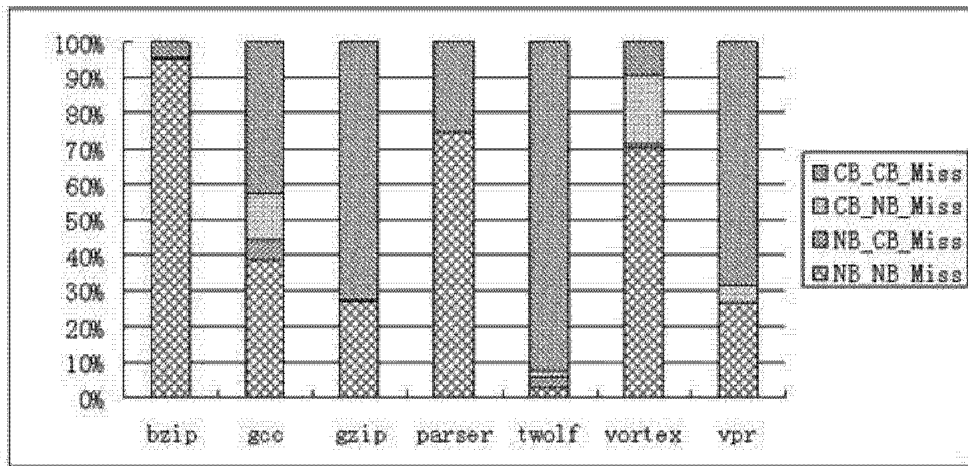


图 5

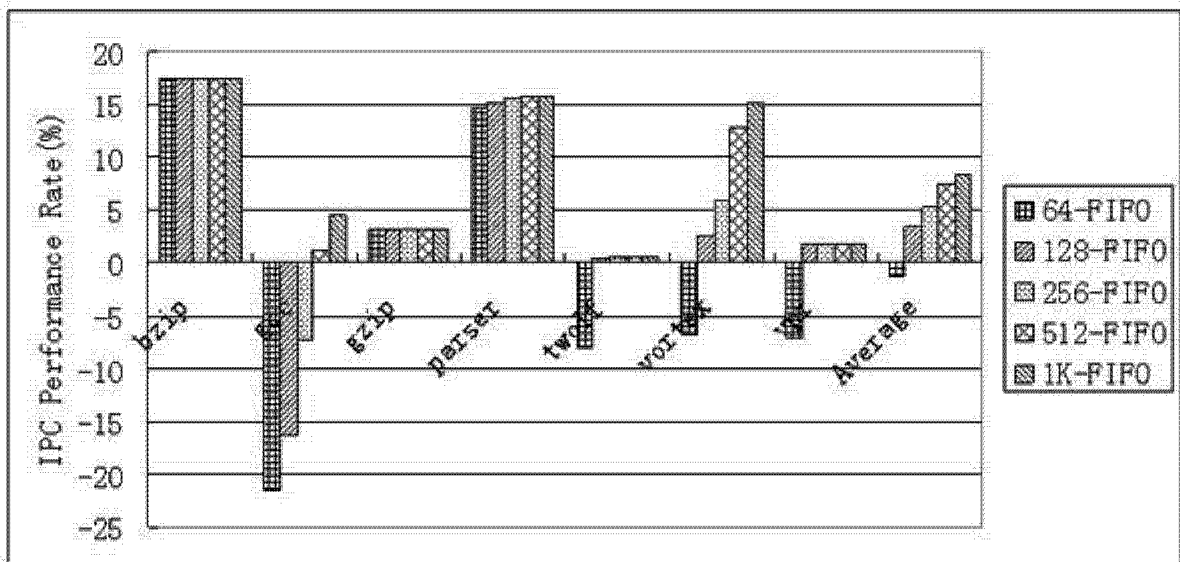


图 6