(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101160561 B
(45)授权公告日 2013.10.16

(21)申请号 200680012619.8
(22)申请日 2006.02.24
(30)优先权数据
11/066,508 2005.02.24 US
(85)PCT申请进入国家阶段日
2007.10.16
(86)PCT申请的申请数据
PCT/US2006/006531 2006.02.24
(87)PCT申请的公布数据
WO2006/091778 EN 2006.08.31
(73)专利权人 高通股份有限公司
地址 美国加利福尼亚州
(72)发明人 博胡斯拉夫·雷赫利克
(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 刘国伟
(51)Int.Cl.
G06F 9/38(2006.01)
G06F 9/45(2006.01)
(56)对比文件
张幸儿．循环不变表达式外提．《计算机编译原理(第二版)》．北京科学出版社，2003,391.
审查员 孟宪超

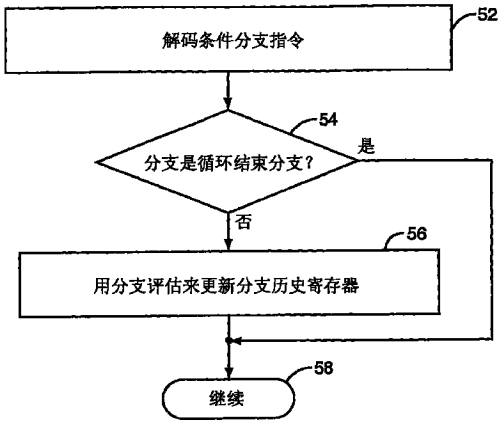
权利要求书1页 说明书6页 附图4页

(54)发明名称

通过循环结束分支来抑制分支历史寄存器的更新

(57)摘要

检测终止代码循环的条件分支指令，且防止分支历史寄存器(BHR)更新以存储循环结束分支评估。这防止实施循环迭代的分支从所述BHR中取代其它分支评估历史。可通过编译器使用特定类型分支指令或在循环结束分支指令的操作码中插入指示位来静态地检测所述循环结束分支。循环结束分支指令可被动态地检测为任何后向分支，或者通过在更新所述BHR时存储最后一个或若干个分支指令的PC并对照最后分支PC(LBPC)寄存器检验分支指令的所述PC而动态地检测。如果所述分支PC匹配，那么抑制对所述BHR的更新。将循环迭代分支保持在所述BHR之外会改进分支预测训练时间和准确性。



CN 101160561 B

1. 一种分支预测方法，其包括：

响应于确定分支指令是循环结束分支指令，在执行所述分支指令时跳过对分支历史寄存器的更新，

其中，所述分支历史寄存器配置为将对分支指令的分支行为存储为采用了或没有采用，其中，所述跳过对分支历史寄存器的更新：在将分支指令程序计数器 PC 与所述分支指令的分支目标地址 BTA 进行比较之后进行，或者，在所述分支指令 PC 与存储最后分支指令的 PC 的最后分支 PC 寄存器的内容匹配的情况下进行，或者，在所述分支指令 PC 与存储最后多个分支指令的 PC 的多个最后分支 PC 寄存器中的任一者的内容匹配的情况下进行。

2. 根据权利要求 1 所述的方法，其中确定步骤包含假设后向分支是循环结束分支。

3. 根据权利要求 1 所述的方法，其中确定步骤包含确定所述分支指令是由编译器产生以用于结束分支的唯一分支指令。

4. 根据权利要求 1 所述的方法，其中确定步骤包含确定所述分支指令包含指示其为循环结束分支指令的一个或一个以上位。

5. 一种使用分支历史寄存器的分支预测方法，所述分支历史寄存器将先前的条件分支指令的分支行为存储为采用了或没有采用，所述方法包括：

检测循环结束分支；以及

跳过对分支历史寄存器的更新，该更新存储相关联分支指令的分支行为，其中，所述跳过对分支历史寄存器的更新：在将分支指令程序计数器 PC 与所述分支指令的分支目标地址 BTA 进行比较之后进行，或者，在所述分支指令的 PC 与存储最后分支指令的 PC 的最后分支 PC 寄存器的内容匹配的情况下进行，或者，在所述分支指令的 PC 与存储最后多个分支指令的 PC 的多个最后分支 PC 寄存器中的任一者的内容匹配的情况下进行。

6. 根据权利要求 5 所述的方法，其中检测循环结束分支包括解码由编译器产生以用于结束分支的唯一分支指令。

7. 根据权利要求 5 所述的方法，其中检测循环结束分支包括检测所述相关联分支指令操作码中指示其为循环结束分支指令的一个或一个以上位。

8. 一种使用分支历史寄存器的分支预测装置，所述分支历史寄存器将先前的条件分支指令的分支行为存储为采用了或没有采用，所述装置包括：

用于检测循环结束分支的模块；以及

用于跳过对分支历史寄存器的更新的模块，该更新存储相关联分支指令的分支行为，其中，所述跳过对分支历史寄存器的更新：在将分支指令程序计数器 PC 与所述分支指令的分支目标地址 BTA 进行比较之后进行，或者，在所述分支指令的 PC 与存储最后分支指令的 PC 的最后分支 PC 寄存器的内容匹配的情况下进行，或者，在所述分支指令的 PC 与存储最后多个分支指令的 PC 的多个最后分支 PC 寄存器中的任一者的内容匹配的情况下进行。

9. 根据权利要求 8 所述的装置，其中用于检测循环结束分支的模块包括用于解码由编译器产生以用于结束分支的唯一分支指令的模块。

10. 根据权利要求 8 所述的装置，其中用于检测循环结束分支的模块包括用于检测所述相关联分支指令操作码中指示其为循环结束分支指令的一个或一个以上位的模块。

通过循环结束分支来抑制分支历史寄存器的更新

技术领域

[0001] 本发明大体上涉及处理器领域，且更确切地说涉及一种通过用循环结束分支指令抑制对分支历史寄存器的更新而改进分支预测的方法。

背景技术

[0002] 微处理器在广泛的应用中执行计算任务。几乎始终需要改进的处理器性能以允许通过软件变化来实现较快的操作和/或增加的功能性。在许多嵌入式应用（例如，便携式电子装置）中，节省功率也是处理器设计和实施的一个目标。

[0003] 许多现代处理器使用管线结构，其中连续的指令（各具有多个执行步骤）在执行时重叠。为了实现改进的性能，指令应当连续流动穿过管线。任何导致指令在管线中停滞的情形均可对性能造成不利影响。如果从管线中冲洗 (flush) 指令并随后重新取得指令，那么性能和功率消耗均会受到损害。

[0004] 大多数程序包含条件分支指令，直到在管线深处评估指令时才会知道其实际分支行为。为了避免因等待对分支指令的实际评估而产生的停滞，现代处理器可采用某种形式的分支预测，借此在管线中早期预测条件分支指令的分支行为。基于预测出的分支评估，处理器以推测方式从预测出的地址取得（预取）并执行指令，所述预测出的地址是分支目标地址（如果预测会采用分支）或分支指令之后的下一顺序地址（如果预测不会采用分支）。当确定了实际分支行为时，如果分支被错误预测，那么必须从管线中冲洗以推测方式取得的指令，并从下一正确地址取得新的指令。响应于错误的分支预测而预取指令可对处理器性能和功率消耗造成不利影响。因此，改进分支预测的准确性是一个重要的设计目标。

[0005] 已知的分支预测技术包含静态和动态两种预测。可通过编程器和/或编译器来静态地预测一些分支指令的可能行为。分支预测的一个实例是错误检验例行程序。代码通常会正确执行，且错误是罕见的。因此，实施“遇错误分支 (branch on error)”的分支指令将在非常高的百分比的时间中评估“不采用”。此种指令可在操作码中包含静态分支预测位，所述预测位是由编程器或编译器在知道分支条件的最可能结果的情况下设定的。

[0006] 动态预测一般基于正被预测的分支指令和/或同一代码中的其它分支指令的分支评估历史（且在一些情况下是分支预测准确性历史）。对实际代码的广泛分析指示，最近过去的分支评估模式可能是对未来分支指令的评估的良好指示。

[0007] 图 1 中描绘的一种已知形式的动态分支预测利用分支历史寄存器 (BHR) 100 来存储过去 n 个分支评估。在简单的实施方案中，BHR 30 包括移位寄存器。将最近的分支评估结果移入（例如，1 指示采用分支且 0 指示不采用分支），而寄存器中的最早过去的评估被取代。处理器可针对每个分支指令维持局部 BHR 100。或者（或另外），BHR 100 可含有对所有条件分支指令的最近过去的评估，其有时在此项技术中称为全局 BHR 或 GHR。如本文所使用，BHR 指代局部和全局分支历史寄存器两者。

[0008] 如图 1 中所描绘，BHR 100 可将分支预测器表 (BPT) 102 编索引，所述 BPT 102 同样可以是局部的或全局的。BHR 100 可直接将 BPT 102 编索引，或者可在 BPT 索引逻辑 104

中与例如分支指令的程序计数器 (PC) 的其它信息组合。另外可利用对 BPT 索引逻辑 104 的其它输入。BPT 索引逻辑 104 可将输入链接在一起（此项技术中通常称为 gselect），对输入进行异或运算 (gshare)，执行散列函数，或以多种方式组合或转换输入。

[0009] 在一个实例中，BPT 102 可包括多个饱和计数器，其 MSB 充当双模态分支预测器。举例来说，每个表条目可包括 2 位计数器，所述计数器采用四种状态中的一种，所述四种状态中的每一者被指派有加权预测值，例如：

[0010] 11- 强力预测采用

[0011] 10- 弱预测采用

[0012] 01- 弱预测不采用

[0013] 00- 强力预测不采用

[0014] 每当相应的分支指令评估“采用”时，所述计数器递增，且每当指令评估“不采用”时，所述计数器递减。计数器的 MSB 是双模态分支预测器，其将预测一个分支是采用还是不采用，而不管基础预测的强度或权重如何。饱和计数器减少不频繁的分支评估的预测错误。始终单向评估的分支将使计数器饱和。相反地不频繁评估将改变计数器值（以及预测的强度），但不是双模态预测值。因此，不频繁的评估将只错误预测一次而不是两次。饱和计数器的表只是说明性实例，一般来说，BHT 可将含有多种分支预测机制的表编索引。

[0015] 不论在 BPT 102 中采用哪种分支预测机制，BHR 100（单独的或与例如分支指令 PC 的其它信息组合）将 BPT 102 编索引以获得分支预测。通过在 BHR 100 中存储先前分支评估并在分支预测中使用所述评估，使正被预测的分支指令与过去的分支行为相关 -- 在局部 BHR 100 的情况下为其自身的过去行为而在全局 BHR 100 的情况下为其它分支指令的行为。这种相关至少在高度重复的代码的情况下可能是准确的分支预测的关键。

[0016] 请注意，图 1 描绘存储在 BHR 100 中的分支评估，即对条件分支指令的实际评估，其可能只有在管线深处（例如在执行管级中）才被了解。虽然这是最终结果，但在实践中，许多高性能处理器将来自 BPT 102 的预测出的分支评估存储在 BHR 100 中，并之后在证实预测错误时作为错误预测恢复操作的一部分而对 BHR 100 进行校正。为了清晰起见，附图未反映这种实施特征。

[0017] 可能会降低采用 BHR 100 的分支预测器的功效的一种常用代码结构是循环。循环以测试循环结束条件的条件分支指令结束，所述循环结束条件例如为每次通过循环时递增的索引变量是否已达到循环结束值。如果没有，那么执行形成分支回到循环的开始处以进行另一次迭代和另一次循环结束条件分支评估。相对于 n 位 BHR 100，存在关于循环的三种相关情况：循环不执行；循环通过 m 次迭代执行（其中 m ＜ n）；和循环执行 m 次（其中 m ＞＝ n）。

[0018] 如果循环不执行，那么循环开始处的前向分支在循环主体上形成分支，从而产生一个采用的分支评估。这对 BHR 100 具有最小影响，因为 BHR 100 中的过去分支评估历史只由一个分支评估取代（实际上，预测准确性可通过与这个分支评估的相关来改进）。

[0019] 如果循环通过 m 次迭代来执行（其中 m ＞＝ n），那么循环结束分支指令的“采用”后向分支使 BHR 100 饱和。也就是说，在循环结束时，n 位 BHR 将始终含有恰好 n-1 个一且后面跟着单个零，这对应于由循环迭代产生的较长系列的采用的评估，且在循环终止时以单个未采用评估结束。这实际上损害 BHR 100 的功效，因为与先前分支评估（对于局部或全

局 BHR 100) 的所有相关全部丢失。在此情况下，BHR 100 将很可能映射到相同的 BPT 102 条目以获取给定的分支指令（视对 BPT 索引逻辑 104 的其它输入而定），而不是映射到含有反映分支指令与先前分支评估的相关的分支预测的条目。

[0020] 此外，饱和的 BHR 100 可增加 BPT 102 中的混淆。也就是说，如果 BHR 100 直接将 BPT 102 编索引，那么具有许多迭代的循环之后的所有分支指令将映射到相同的 BPT 102 条目。即使在 BHR 100 与其它信息组合的情况下，混淆的可能性也会增加。这不但对于循环之后的分支指令而且对于混淆到其在 BPT 102 中的条目的所有分支指令，均会不利地影响预测准确性。

[0021] 如果循环通过 m 次迭代执行（其中 m ＜ n)，那么 BHR 100 不饱和且保持某一先前分支评估历史。然而，代表先前分支评估历史的位被 m 个位位置取代。特别在 m 变化的情况下，这对分支预测具有两个有害影响。首先，分支指令将映射到 BPT 102 中的更大数目的条目，以俘获与先前分支评估的相同相关，从而与没有循环结束分支影响 BHR 30 的情况下将需要的 BPT 102 相比，对于相同数目的分支指令需要更大的 BPT 102 来支持相同的准确性。第二，BPT 102 中的分支预测器将花费较长时间来“训练”，从而增加了在 BPT 102 开始提供准确的分支预测之前必须执行的代码量。

[0022] 举例来说，考虑 8 位 BHR 100 和一代码段，其具有分支指令 A-H，接着是循环，且接下来是分支指令 X。分支 X 与分支 G 和 H 的评估历史强力相关。插入的循环的各种迭代将在预测 X 时产生下表 1 中呈现的 BHR 结果。

[0023]

BHR								注释
A	B	C	D	E	F	G	H	循环执行一次（未采用初始前向或循环结束后向分支）
B	C	D	E	F	G	H	1	跳过循环（采用一个初始前向分支）
C	D	E	F	G	H	1	0	2 次迭代（循环结束后向分支采用一次，然后不采用）
D	E	F	G	H	1	1	0	3 次迭代
E	F	G	H	1	1	1	0	4 次迭代
F	G	H	1	1	1	1	0	5 次迭代
G	H	1	1	1	1	1	0	6 次迭代

[0024] 表 1 :各种数目的循环迭代之后的 BHR 100 内容

[0025] 在此实例中，每种情况下 BHR 100 中均存在正被预测的分支指令 X 与对分支 G 和 H 的先前评估之间的所需相关。然而，其位于 BHR 100 中的不同位置，且因此每种情况将映射到不同的 BPT 102 条目。这会浪费 BPT 102 空间，增加分支预测训练时间，并增加 BPT 102 中的混淆的可能性，所有这些都会降低预测准确性。

发明内容

[0026] 在一个或一个以上实施例中，通过识别循环结束分支指令并响应于循环结束指令而抑制对 BHR 的更新来改善在 BHR 中存储循环结束的分支指令评估的不良影响。以多种方式识别循环结束指令。

[0027] 在一个实施例中，分支预测方法包含响应于分支指令的性质而在分支指令执行时视情况抑制对 BHR 的更新。

[0028] 在另一实施例中，处理器包含 :分支预测器，其可操作以预测对条件分支指令的评

估；以及指令执行管线，其可操作以基于来自分支预测器的预测以推测方式取得和执行指令。所述处理器还包含：BHR，其可操作以存储对条件分支指令的评估；以及控制电路，其可操作以响应于分支指令的性质而抑制存储对条件分支指令的评估。

[0029] 在又一实施例中，可操作以响应于程序代码而产生指令的编译器或汇编器包含可操作以指示终止代码循环的条件分支指令的循环结束分支指令标记功能。

附图说明

[0030] 图 1 是现有技术分支预测器电路的功能方框图。

[0031] 图 2 是处理器的功能方框图。

[0032] 图 3 是执行分支指令的方法的流程图。

[0033] 图 4 是包含一个或一个以上最后分支 PC 寄存器的分支预测器电路的功能方框图。

具体实施方式

[0034] 图 1 描绘处理器 10 的功能方框图。处理器 10 根据控制逻辑 14 在指令执行管线 12 中执行指令。在一些实施例中，管线 12 可为超标量设计，其具有多个并行管线。管线 12 包含组织成管级的各种寄存器或锁存器 16，以及一个或一个以上算术逻辑单元 (ALU)18。通用寄存器 (GPR) 文件 20 提供包括存储器层级的顶部的寄存器。

[0035] 管线 12 从指令高速缓冲存储器 (I-cache)22 取得指令，其中由指令侧转译后备缓冲器 (ITLB)24 来管理存储器地址转译和许可。当在管线 12 中早期解码条件分支指令时，分支预测器 26 预测分支行为，并向指令预取单元 28 提供预测。指令预取单元 28 对于“采用的”分支预测在管线 12 中计算出的分支目标地址处或对于预测为“不采用”的分支在下一顺序地址处以推测方式从指令高速缓冲存储器 22 取得指令。在任一情况下，均将预取的指令加载到管线 12 中以用于推测性执行。

[0036] 分支预测器 26 包含分支历史寄存器 (BHR)30、分支预测器表 (BPT)32、BPT 索引逻辑 34 以及 BHR 更新逻辑 36。分支预测器 26 可额外包含一个或一个以上最后分支 PC 寄存器 38，本文下文中对其进行更充分描述。

[0037] 从数据高速缓冲存储器 (D-cache)40 存取数据，其中由主转译后备缓冲器 (TLB)42 管理存储器地址转译和许可。在各种实施例中，ITLB 24 可包括 TLB 42 的一部分的副本。或者，ITLB 24 与 TLB 42 可集成。类似地，在处理器 10 的各种实施例中，I-cache 22 与 D-cache 40 可集成或合并。I-cache 22 和 / 或 D-cache 40 中的未中导致在存储器接口 46 的控制下存取主（芯片外）存储器 44。

[0038] 处理器 10 可包含输入 / 输出 (I/O) 接口 46，其控制对各种外围装置 50 的存取。所属领域的技术人员将认识到，处理器 10 的许多变化形式是可能的。举例来说，处理器 10 可包含用于 I-cache 22 和 D-cache 40 中的任一者或两者的第二层 (L2) 高速缓冲存储器。此外，特定实施例中可省略处理器 10 中描绘的功能块中的一者或一者以上。

[0039] 根据一个或一个以上实施例，通过防止循环结束分支破坏分支预测器 26 中的一个或一个以上 BHR 30 来改进分支预测准确性。图 3 中将这个过程描绘成流程图。将条件分支指令解码（方框 52）。确定分支是否为循环结束分支（方框 54）。如果不是，那么更新 BHR 30 以记录分支评估（方框 56），即，分支指令评估为“采用”还是“不采用”。执行接着

分别在分支目标地址或下一顺序地址处继续（方框 58）。如果分支不是循环结束分支，那么抑制更新 BHR 30 以记录对循环结束分支指令的分支评估（如从方框 54 到方框 58 的路径所指示）。以此方式，循环迭代分支不会通过取代相关分支评估历史而破坏 BHR 30 的内容。查询（方框 54）将分支指令识别为循环结束分支指令，所述步骤可通过多种方式来实现。

[0040] 循环通过从循环结束处向后向循环开始处形成分支来迭代。根据一个实施例，假设每个分支目标地址小于分支指令地址或 PC 的条件分支指令（即后向分支）是循环结束分支指令，且防止其更新 BHR 30。这个实施例提供简化的优点。在 BHR 30 更新时间，当分支指令实际上在管线中评估时将分支指令 PC 与分支目标地址 (BTA) 进行比较。如果 BTA ＜ PC，那么不更新 BHR 30。这个实施例具有以下缺点：当确定分支目标地址时需要进行地址比较，且一些不是循环结束分支的后向分支将不使其评估记录在 BHR 30 中。

[0041] 另一种检测循环结束分支的方式是辨别对同一分支指令的重复执行。在图 4 中描绘的一个实施例中，最后分支 PC(LBPC) 寄存器 38 存储其评估被存储在 BHR 30 中的最后分支指令的 PC。在简单循环的情况下，如果分支指令的 PC 与 LBPC 38 匹配（也就是说，分支指令是所评估的最后分支指令），那么假设分支指令是循环结束分支指令，且抑制对 BHR 30 的进一步更新。如上文参看图 1 所论述，虽然图 4 描绘在任何给定实施方案中将 LBPC 38 的内容与 BHR 更新逻辑 36 中的实际分支评估进行比较，但可将 LBPC 38 与预测的分支评估进行比较，其中在错误预测的情况下对 BHR 30 进行校正。这个实施例只存储循环的第一次迭代，只从 BHR 30 中取代一个先前分支评估。这个实施例不需要编译器支持，且不需要在 BHR 30 更新时间确定分支的方向。

[0042] 循环可含有一个或一个以上嵌套循环，或者可在循环内包含其它分支。在此情况下，可通过 LBPC 方法抑制内部循环使 BHR 30 饱和；然而，外部循环结束分支仍然将被存储在 BHR 30 中。在一个实施例中，可提供两个或两个以上 LBPC 寄存器 38，其中将连续评估的分支指令的 PC 存储在相应的 LBPC 寄存器 ($LBPC_0$，$LBPC_1$…$LBPC_M$) 38 中。如果分支指令的 PC 与 $LBPC_N$ 寄存器 38 中的任一者匹配，那么可抑制对 BHR 30 的更新。

[0043] 也可由编译器或汇编器静态地标记循环结束分支指令。在一个实施例中，编译器产生只用于循环结束分支的特定类型的分支指令，例如“BRLP”。辨别 BRLP 指令，且当 BRPE 指令在执行管级中评估时决不更新 BHR 30。在另一实施例中，编译器或汇编器可例如通过在操作码中设定一个或一个以上预界定的位而在分支指令中嵌入循环结束分支指示。检测循环结束分支位，且当所述分支指令在执行管级中评估时抑制对 BHR 30 的更新。对循环结束分支的静态识别通过将循环结束识别功能移动到编译器或汇编器中而减小硬件和计算复杂性。

[0044] 条件分支指令可具有许多性质，其中包含（例如）分支指令地址或 PC、指令类型和操作码中是否存在指示位。如本文所使用，将分支操作的性质和 / 或与分支有关的程序的性质视为分支指令的性质。举例来说，分支指令 PC 是否与一个或一个以上 LBPC 寄存器 38 的内容匹配以及分支目标地址相对于分支指令 PC 是正向还是后向的，这些是分支指令的性质。

[0045] 虽然本文中已就本发明的特定特征、方面和实施例描述了本发明，但将了解，在本发明的广泛范围内可能有许多变化、修改和其它实施例，且因此，应认为所有变化、修改和

实施例均在本发明的范围内。因此，应在所有方面均将当前实施例理解为说明性的而非限制性的，且希望属于所附权利要求书的意义和等效范围内的所有变化均包含在其中。

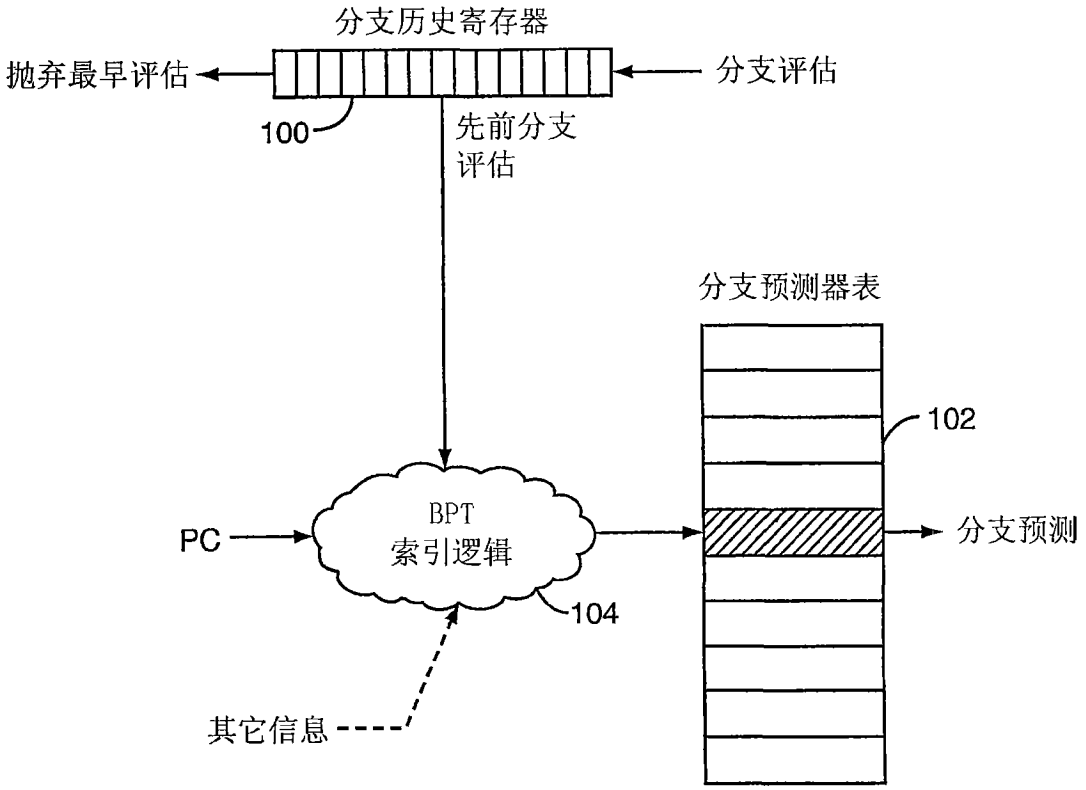

分支历史寄存器
抛弃最早评估
分支评估
100
先前分支
评估
分支预测器表
102
PC
BPT
索引逻辑
104
分支预测
其它信息

图1

(现有技术)

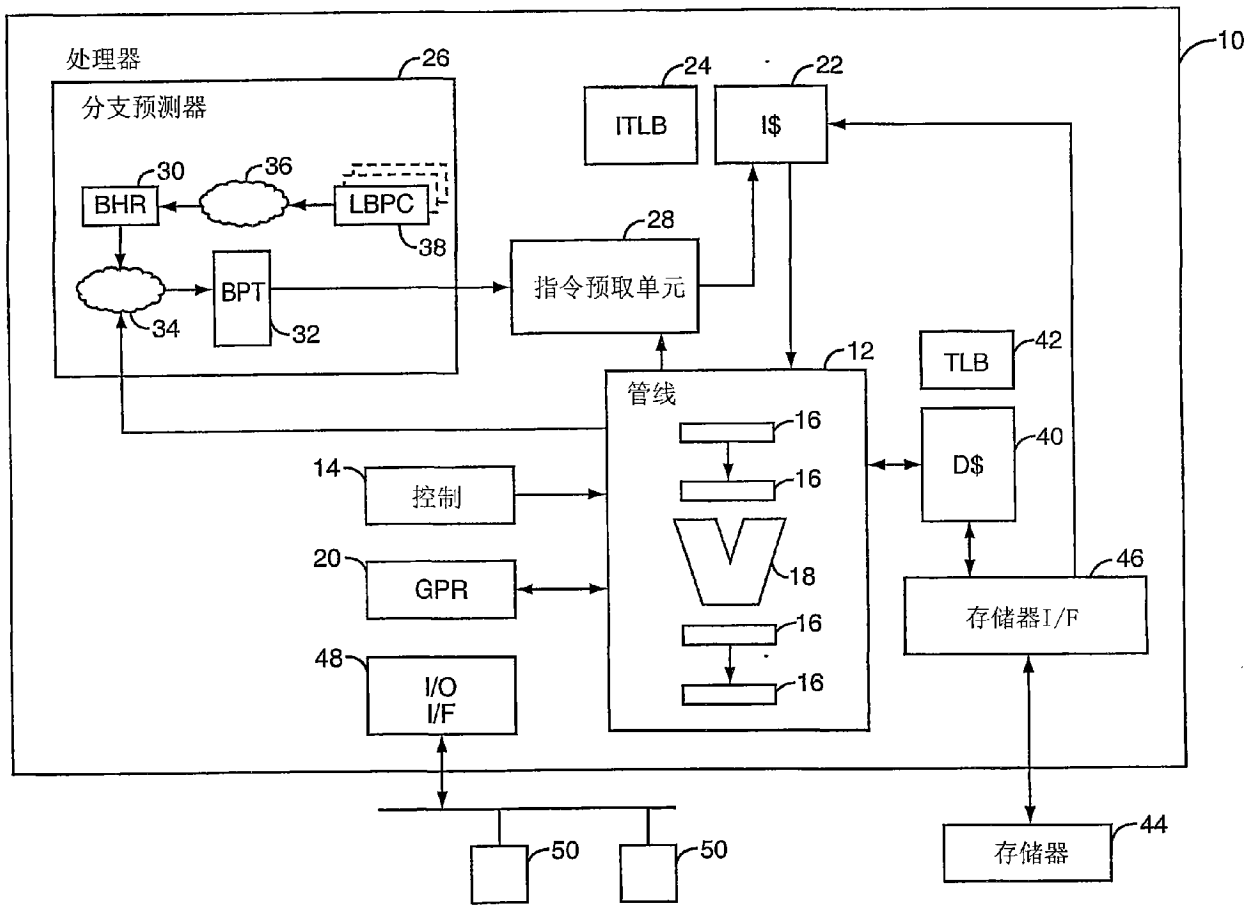

处理器
26
分支预测器
30
BHR
36
LBPC
38
BPT
34
32
28
指令预取单元
24
ITLB
22
I$
10
12
管线
16
16
18
16
16
14
控制
20
GPR
48
I/O
I/F
TLB
42
D$
40
46
存储器I/F
50
50
存储器
44

图2

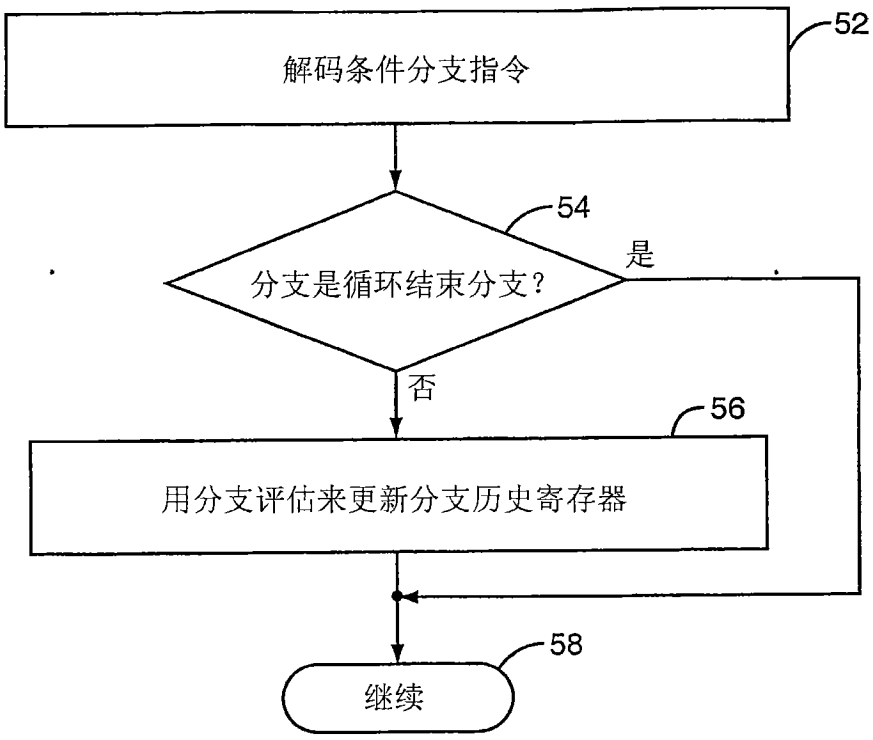


图3

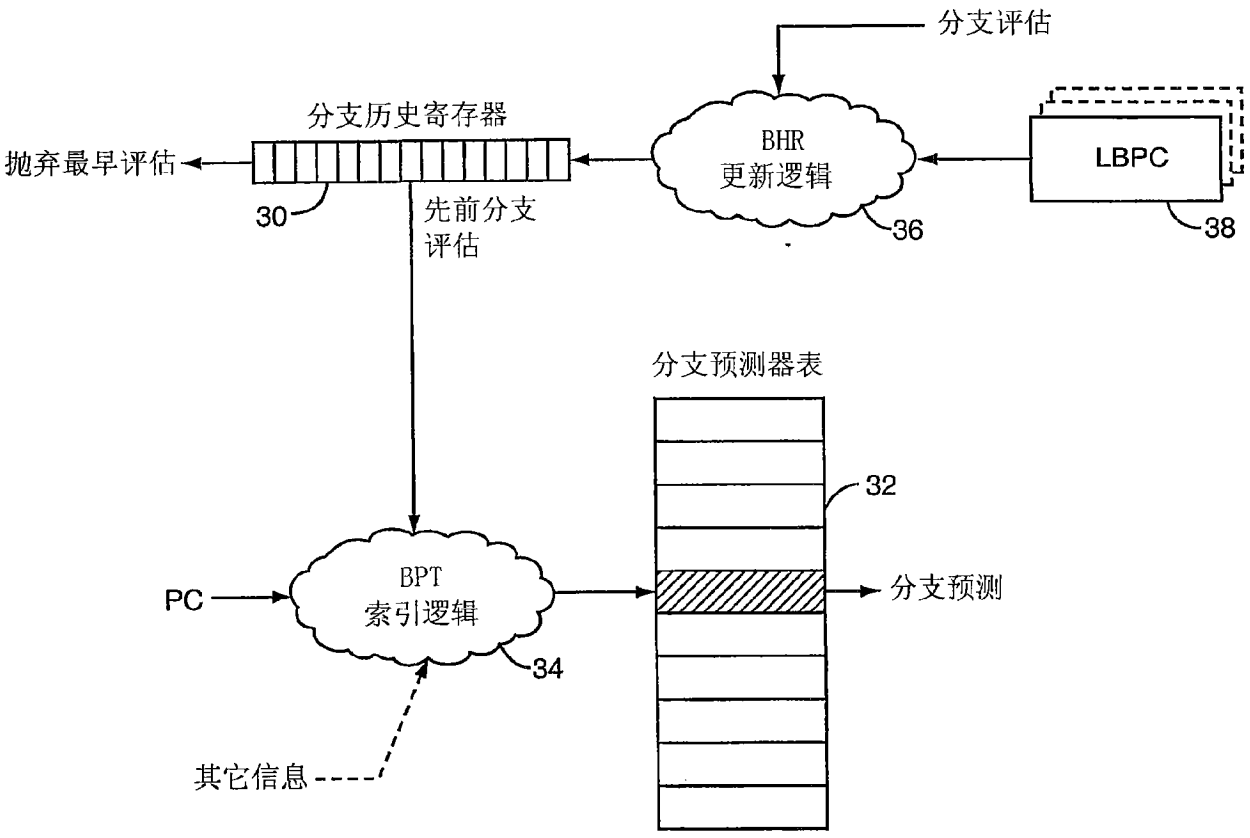

分支评估
分支历史寄存器
抛弃最早评估
30
先前分支
评估
BHR
更新逻辑
36
LBPC
38
分支预测器表
32
PC
BPT
索引逻辑
34
分支预测
其它信息

图4