



(12)发明专利申请

(10)申请公布号 CN 106528049 A

(43)申请公布日 2017.03.22

(21)申请号 201611013466.0

(22)申请日 2016.11.17

(71)申请人 上海兆芯集成电路有限公司

地址 上海市张江高科技园区金科路2537号
301室

(72)发明人 王小玲 陈国华 杨梦晨

(74)专利代理机构 北京市柳沈律师事务所
11105

代理人 王新宇

(51) Int. Cl.

G06F 7/58(2006.01)

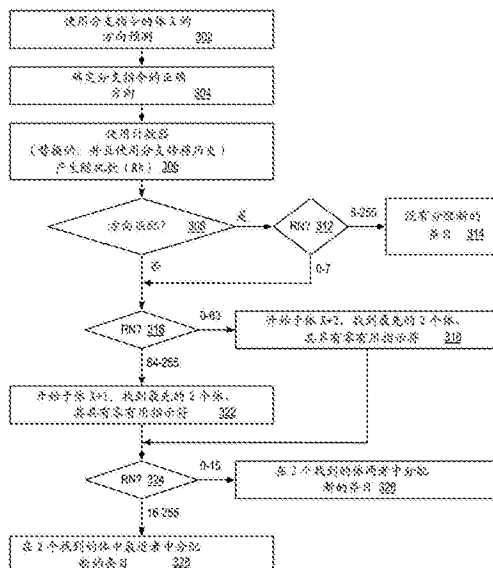
权利要求书3页 说明书8页 附图5页

(54)发明名称

在多存储体条件分支预测器中用于更新事件的随机数产生

(57)摘要

一种分支预测器,拥有具有保留预测信息的条目的多个存储体,包含该分支预测器的处理器利用这些分支历史信息来给取指并执行的分支指令进行预测方向。发生在处理器的事件的计数被提供给硬件逻辑,其对该计数的预定比特执行与/或逻辑运算(如XOR)以产生随机值。响应于该处理器确定由分支预测器预测的分支指令的正确方向,该分支预测器使用由硬件逻辑产生的随机值做出关于更新各存储体的决定。分支转移历史的比特连同该计数也可以被用于产生该随机值。计数的事件可以是提交指令的数目或者是处理器或总线时钟周期数。



1. 一种分支预测器,包括:

具有保留预测信息的条目的多个存储体,该预测信息用于预测由包括该分支预测器的处理器取指和执行的指令块所包含的分支指令的方向;

发生在处理器的事件的计数;

硬件逻辑,其对该计数的预定比特执行与/或逻辑运算以产生随机值;以及

其中响应于该处理器确定由分支预测器预测的分支指令的正确方向,该分支预测器使用由硬件逻辑产生的随机值做出关于更新各存储体的决定。

2. 如权利要求1所述的分支预测器,进一步包括:

分支转移历史,其指定该处理器遇到的分支指令的方向的历史;以及

其中该硬件逻辑除了对该计数的预定比特执行与/或逻辑运算外,还对分支转移历史

3. 如权利要求1所述的分支预测器,进一步包括:

其中与/或逻辑运算包括该计数的第一部分比特和该计数器的第二部分比特的布尔异或(XOR)运算。

4. 如权利要求1所述的分支预测器,进一步包括:

其中计数的事件包括通过处理器提交的指令条数。

5. 如权利要求1所述的分支预测器,进一步包括:

其中计数的事件包括两倍于处理器时钟周期的外部总线时钟周期的数目。

6. 如权利要求1所述的分支预测器,进一步包括:

其中计数的事件包括处理器的时钟的周期。

7. 如权利要求1所述的分支预测器,进一步包括:

其中该随机值来自可能的值的集合;

其中当由分支预测器预测的方向与正确方向匹配时,如果该随机值是可能的值的集合的预定子集之一则分支预测器在多个存储体的一个或多个中分配新的条目,否则不在多个存储体的一个或多个中分配新的条目。

8. 如权利要求1所述的分支预测器,进一步包括:

其中该随机值来自可能的值的集合;

其中当由分支预测器预测的方向不与正确方向匹配时,如果该随机值是可能的值的集合的预定子集之一则分支预测器在多个存储体的多于一个中分配新的条目,否则在多个存储体的一个中分配新的条目。

9. 如权利要求1所述的分支预测器,进一步包括:

其中该随机值来自可能的值的集合;

其中多个存储体的每个体接收使用不同长度的分支转移历史计算的索引,并且多个存储体的每个体具有一编号,并且体编号从最短长度向最长长度顺序增加;以及

其中多个存储体中的每个条目包括一指示符,其指示该条目在预测分支指令的方向中趋于有用;

其中当响应于处理器确定其方向由具有体编号X的多个存储体的第一体预测的分支指令的正确方向在多个体中分配条目时,如果该随机值是可能的值的集合的预定子集之一则分支预测器开始搜索要在编号X+1的体分配的非有用条目,否则开始搜索要在编号X+2的体

分配的非有用条目。

10. 如权利要求1所述的分支预测器,进一步包括:

其中多个存储体中的每个条目包括一指示符,其指示该条目在预测分支指令的方向中趋于有用;

计数器,按照指示符指示其条目趋于有用的体的数量递增并且按照指示符指示其条目不趋于有用的体的数量递减;以及

其中当计数器具有大于或等于随机值的值时,该分支预测器概率递减该指示符。

11. 一种用于操作分支预测器的方法,该分支预测器具有保留预测信息的条目的多个存储体,该预测信息用于预测由包括该分支预测器的处理器取指和执行的分支指令的方向,所述方法包括:

维持发生在处理器的事件的计数;

对该计数的预定比特执行与/或逻辑运算以产生随机值;以及

响应于该处理器确定由分支预测器预测的分支指令的正确方向,使用产生的随机值做出关于更新各存储体的决定。

12. 如权利要求11所述的方法,进一步包括:

维持分支转移历史,其指定该处理器遇到的分支指令的方向的历史;以及

除了对该计数的预定比特执行与/或逻辑运算外,还对分支转移历史的预定比特执行与/或逻辑运算以产生该随机值。

13. 如权利要求11所述的方法,进一步包括:

其中与/或逻辑运算包括该计数的第一部分比特和该计数的第二部分比特的布尔异或(XOR)运算。

14. 如权利要求11所述的方法,进一步包括:

其中计数的事件包括通过处理器的已提交指令。

15. 如权利要求11所述的方法,进一步包括:

其中计数的事件包括两倍于处理器时钟周期的外部总线时钟周期的数目。

16. 如权利要求11所述的方法,进一步包括:

其中计数的事件包括处理器的时钟的周期。

17. 如权利要求11所述的方法,进一步包括:

其中该随机值来自可能的值的集合;

其中当由分支预测器预测的方向与正确方向匹配时,如果该随机值是可能的值的集合的预定子集之一则在多个存储体的一个或多个中分配新的条目,否则不在多个存储体的一个或多个中分配新的条目。

18. 如权利要求11所述的方法,进一步包括:

其中该随机值来自可能的值的集合;

其中当由分支预测器预测的方向不与正确方向匹配时,如果该随机值是可能的值的集合的预定子集之一则在多个存储体的多于一个中分配新的条目,否则在多个存储体的一个中分配新的条目。

19. 如权利要求11所述的方法,进一步包括:

其中该随机值来自可能的值的集合;

其中多个存储体的每个体接收使用不同长度的分支转移历史计算的索引,并且多个存储体的每个体具有一编号,并且体编号从最短长度向最长长度顺序增加;以及

其中多个存储体中的每个条目包括一指示符,其指示该条目在预测分支指令的方向中趋于有用;

其中当响应于处理器确定其方向由具有体编号X的多个存储体的第一体预测的分支指令的正确方向在多个体中分配条目时,如果该随机值是可能的值的集合的预定子集之一则开始搜索要在编号X+1的体分配的非有用条目,否则开始搜索要在编号X+2的体分配的非有用条目。

20. 如权利要求11所述的方法,进一步包括:

其中多个存储体中的每个条目包括一指示符,其指示该条目在预测分支指令的方向中趋于有用;

按照指示符指示其条目趋于有用的体的数量递增计数器,并且按照指示符指示其条目不趋于有用的体的数量递减计数器;以及

其中当计数器具有大于或等于随机值的值时,概率递减该指示符。

21. 一种计算机程序产品,所述计算机程序产品编码在与计算设备一起使用的至少一个非瞬时计算机可用介质中,所述计算机程序产品包括:

在所述介质中具体表现的计算机可用程序代码,用于指定分支预测器,所述计算机可用程序代码包括:

第一程序代码,用于指定具有保留预测信息的条目的多个存储体,该预测信息用于预测由包括该分支预测器的处理器取指和执行的分支指令的方向;

第二程序代码,用于指定发生在处理器的事件的计数;

第三程序代码,用于指定对该计数的预定比特执行与/或逻辑运算的硬件逻辑以产生随机值;以及

其中响应于该处理器确定由分支预测器预测的分支指令的正确方向,该分支预测器使用由该硬件逻辑产生的随机值做出关于更新各存储体的决定。

在多存储体条件分支预测器中用于更新事件的随机数产生

技术领域

[0001] 本发明涉及处理器设计的领域,特别是涉及在多存储体条件分支预测器中用于更新事件的高效随机数产生的分支预测器。

背景技术

[0002] 在处理器设计的领域中对分支指令预测器的预测精度要求越来越高是公知的。随着处理器流水级、高速缓存存储器访问延迟和超标量体系结构中的指令发射宽度的增加,该需求更加强烈。分支指令预测器包括预测目标地址,和预测条件分支指令方向,即跳转(taken)或者不跳转。

[0003] 一种流行的条件分支指令方向预测器通常被称为TAGE预测器,这是对几何长度的标签预测器的缩写,这已在由Andre Seznec撰写的各种文献中描述。TAGE预测器包括用于存储分支预测信息的多个存储体(memory bank)。除开一个默认体仅通过程序计数器来索引外,预测器的每个体利用程序计数器和分支转移历史的长度的散列来索引。为了生成用于每个非默认体的索引,不同长度的分支转移历史被散列;即几何长度。此外,每个体的每个条目包括与程序计数器的标签比特进行比较的标签以确定是否在体中发生命中;因此叫做标签。

[0004] 如文献描述的,由Seznec设计的TAGE预测器已经非常成功地进入各种分支预测竞赛。各竞赛是基于分支预测器的软件仿真。TAGE文献描述了各个体按照概率方式更新的各个方法。

发明内容

[0005] 根据本发明的一方面,提供一种分支预测器,该分支预测器具有保留预测信息的条目的多个存储体,该预测信息用于预测由包括该分支预测器的处理器取指执行的指令块所包含的分支指令的方向;

[0006] 发生在处理器的事件的计数;

[0007] 硬件逻辑,其对该计数的预定比特执行与/或逻辑运算以产生随机值;以及

[0008] 其中响应于该处理器确定由分支预测器预测的分支指令的正确方向,该分支预测器使用由硬件逻辑产生的随机值做出关于更新各存储体的决定。

[0009] 根据本发明的另一方面,提供一种用于操作分支预测器的方法,该分支预测器具有保留预测信息的条目的多个存储体,该预测信息用于预测由包括该分支预测器的处理器取指和执行的分支指令的方向,所述方法包括:

[0010] 维持发生在处理器的事件的计数;

[0011] 对该计数的预定比特执行与/或逻辑运算以产生随机值;以及

[0012] 响应于该处理器确定由分支预测器预测的分支指令的正确方向,使用产生的随机值做出关于更新各存储体的决定。

附图说明

- [0013] 图1是图示多存储体条件分支指令预测器的框图。
- [0014] 图2是图1的存储体的条目的框图。
- [0015] 图3是图示图1的分支预测器的操作的流程图。
- [0016] 图4是图示根据替代实施例的多体条件分支指令预测器 (multi-bank conditional branch instruction predictor) 的框图。
- [0017] 图5是图示根据替代实施例的多体条件分支指令预测器的框图。
- [0018] 图6是图示分支预测器做出关于更新有用指示符的决定的操作的流程图。

具体实施方式

[0019] 现在参考图1, 示出了图示多体条件分支指令预测器100, 或分支预测器100的框图。分支预测器100被包含在处理器中, 用于预测处理器从高速缓存取出的指令字节块中可能的条件指令的分支跳转方向。在一个实施例中, 分支预测器100是在常规TAGE预测器上的改进。然而, 实施例不限于TAGE预测器, 并且实施例可包括其他多体预测器, 其利用随机数进行有关更新各体的决定。

[0020] 分支预测器100包括指令计数器102, 程序计数器(PC) 104, 分支转移历史(BHP) 106, 随机数产生(RNG) 逻辑108, 散列逻辑, 控制逻辑114, 比较逻辑116, 和多对多路选择器122和存储体124。图1展示了4对多路选择器122和体124, 表示为多路选择器0 122-0和体0 124-0, 多路选择器1 122-1和体1 124-1, 多路选择器2 122-2和体2 124-2, 和多路选择器N 122-N和体N 124-N。体124的数量可以在不同实施例中变化。每个多路选择器122接收两个相应的输入132和134, 并产生相应的输出136。每个体124在它索引的输入上接收其相应的多路选择器122的输出136以及来自控制逻辑114的相应条目的更新138, 并在其相应的输出139上提供所选择的条目(例如, 图2的200) 到比较和选择逻辑116。优选的是, PC 104是处理器的架构程序计数器, 或指令指针, 所述处理器指定从指令高速缓存取出该指令字节的块的地址。

[0021] RNG逻辑108接收指令计数器102和分支转移历史106并在其一个或两个的选择比特上执行一个或多个与/或逻辑运算以生成提供给控制逻辑114的一个或多个随机数148。控制逻辑114使用随机数148进行有关更新存储体124的决定, 如本文中更详细描述。RNG逻辑108包含组合逻辑, 其执行在指令计数器102和分支转移历史106的一个或两个的选择比特上的与/或逻辑运算。与/或逻辑运算的示例包括, 但不限于: 实体的预定比特的选择; 包括异或(XOR)、NAND、AND、OR、NOT、循环、移位平移的布尔逻辑运算; 和包括加、减、乘、除、取模的算术运算。

[0022] 指令计数器102是计数指令事件的计数器。优选的, 指令计数器102每个时钟周期按由处理器提交的架构指令的数量递增。可替代地, 指令计数器102每个时钟周期按由处理器提交的微指令的数量递增。此外, 下面对应图4和图5中的替代实施例进行描述, 其中与指令事件不同的事件的计数的比特被提供给RNG逻辑108用于产生随机数148。指令计数器102比特在图1中表示为IC[x:y], 其中[x:y]表示指令计数器102的比特的范围。在一个实施例中, RNG逻辑108通过对比特IC[15:8]和比特IC[7:0]执行布尔异或(XOR)运算以产生8比特

结果(如图1的RANDOM1所示)来产生随机数148。

[0023] 分支转移历史106,按照其他术语也称为如全局历史寄存器(GHR),是N比特移位寄存器。当处理器看到条件分支指令,处理器向移位寄存器移入所述条件分支指令的方向,即,跳转或不跳转,其在一个实施例中分别对应于二进制1或0。因此,分支转移历史106跟踪处理器所见的最后N个条件分支的方向历史。在一个实施例中,在指令提交时看到条件分支指令;替换地,如果处理器预测条件分支指令存在于从指令高速缓存取出的指令字节的块中并且其处于在当前PC 104值或之后(但是不在其他任何预测跳转的条件分支之后)值的块的位置,则看到条件分支指令。在一个实施例中,N大约是100比特。在图1分支转移历史106比特表示为BHP [x:y],其中[x:y]表示分支转移历史106的比特的范围。在一个实施例中,RNG逻辑108通过对比特IC [15:8]和比特IC [7:0]以及比特BHP [msb:msb-7]执行布尔异或(XOR)运算以产生8比特结果(如图1的RANDOM2所示)来产生随机数148,其中msb表示分支转移历史106的最高有效位。

[0024] 虽然实施例已描述在其中由RNG逻辑108产生的随机数148是8比特,可以设想其他实施例,其中随机数148的大小是不同的,并且使用指令计数器的102和/或分支转移历史106的不同比特。例如,在一个实施例中,随机数148是10比特,例如,RANDOM1 = IC [19:10] ^ IC [9:0]和RANDOM2 = IC [19:10] ^ IC [9:0] ^ BHP [MSB:MSB-9]。还应该理解的是,除了这里描述的实施例外,指令计数器102和/或分支转移历史106的其它比特可以使用,例如,RANDOM1 = IC [22:13] ^ IC [9:0]和RANDOM2 = IC [30:21] ^ IC [13:4] ^ BHP [40:31]。

[0025] 分支预测器100做出关于是否和如何使用由RNG逻辑108产生的随机数148更新存储体124的决定。有利的,本文描述的RNG逻辑108的实施例以非常高效的方式产生随机数148。此外,RNG逻辑108的实施例可以使用软件仿真工具比调用在软件仿真工具(如,Verilog中的函数random())中提供的随机数产生器的仿真方法来更精确地仿真,这可以使能分支预测器100的更精确的性能建模。

[0026] 散列逻辑112将程序计数器104的一部分与分支转移历史106的一部分进行散列运算以产生用于每个体124的相应索引。相应索引132在图1中表示为132-0、132-1、132-2和132-N并且分别提供作为对多路选择器0 122-0、多路选择器1 122-1、多路选择器2 122-2到多路选择器N 122-N的第一输入。在一个实施例中,如由TAGE预测器执行的,散列逻辑112简单地让程序计数器104的该部分作为索引132-0而非将之与分支转移历史106进行散列运算,并且散列逻辑112将程序计数器104与分支转移历史106的不同长度进行散列运算以产生剩余索引132-1、132-2到132-N中的每个。在一个实施例中,散列逻辑112执行程序计数器104的低比特与分支转移历史106的相应选择长度的比特的XOR以产生索引132-1、132-2到132-N。

[0027] 每个多路选择器122也在第二输入上从控制逻辑114接收相应的更新索引134。控制逻辑114控制每个多路选择器122以选择由散列逻辑112产生的索引132或由控制逻辑114产生的更新索引134,以在其相应的输出136上提供给相应体124的索引输入。当控制逻辑114要更新体124时,控制逻辑114产生关于体124的相应更新索引134的值来选择要更新的条目,并控制相应的多路选择器122选择更新索引134,并且控制体124写入由控制逻辑114产生的更新值138。当控制逻辑114希望从体124读取条目时,控制逻辑114控制相应多路选择器122选择索引132并且作为响应体124在其输出139上提供选择的条目给比较和选择逻

辑116。

[0028] 简要地参考图2,示出图1的体124的条目200的框图。优选地,在每个体124的每个条目200包括有效比特202,标签206,预测204和有用指示符208。有效比特202表示该条目是否有效。标签206是对应的条件分支指令的地址(即,程序计数器值)的高位比特。预测204表示条件分支指令是否将跳转或不跳转。优选地,条目200包括计数器(例如,3比特饱和计数器),并且预测204是计数器的最高有效位。在一个实施例中,当条件分支指令跳转时计数器递增以及当未跳转时计数器递减。在另一个实施例中,计数器根据基于由条目所提供的预测204是否是正确的预测或错误的预测的状态机来更新。有用指示符208是条目200是否在预测条件分支指令中有用的指示。在一个实施例中,有用指示符208由分支预测器100用来做出关于是否要分配条目200的决定,如下更详细描述。在一个实施例中,有用指示符208包括单个比特;可替代地,有用指示符208包括多比特计数器,其计数指示条目200的有用性的程度。

[0029] 再参考图1,当分支预测器100进行预测时,每个体124提供其相应选定的条目200到比较和选择逻辑116。优选的,比较和选择逻辑116选择由来自最高体124的、具有匹配程序计数器104的标签部分的有效标签206的条目200所提供的预测作为最终的预测142。最高体124是其索引132具有由散列逻辑112所使用的最长的分支转移历史106长度的体124。在一个实施例中,存储在条目的标签206是与上述分支转移历史106进行散列的条件分支指令的地址的高位比特,并且程序计数器104的标签部分与分支转移历史106进行散列,并且两者由比较和选择逻辑116进行比较。比较和选择逻辑116向控制逻辑114提供哪个体124被选作最终的预测142的指示。

[0030] 控制逻辑114还从执行条件分支指令的处理器执行单元接收关于每个执行的条件分支指令的信息144,诸如条件分支指令的正确方向及其地址。控制逻辑114维持关于每个预测的条件分支指令的信息,直到它确定该条件分支指令被执行或从处理器流水线被刷去。控制逻辑114使用维持的关于每个预测的条件分支指令的信息和从执行单元接收的信息144,以使其能够做出关于使用随机数148(使用指令计数器102和/或分支转移历史106产生的)更新存储体124的决定,如下面更详细描述。在一个实施例中,处理器包括分支顺序表(BOT),其工作机制类似于重新排序缓冲器(ROB)并储存流水线中存在的分支指令的相关信息(包括地址)。

[0031] 优选地,包括分支预测器100的处理器包括取指单元、指令高速缓存、分支目标地址高速缓存(branch target address cache, BTAC)、指令译码器和执行流水线。在一个实施例中,执行流水线是包括一个或多个架构寄存器、寄存器重命名单元、重排序缓冲器、保留站、多个执行单元和用于安排发射微指令到执行单元的指令调度器的超标量乱序执行流水线。执行单元可以包括以下执行单元类型的一个或多个:整数单元、浮点单元、媒体单元、单指令多数据(SIMD)单元、分支执行单元、访存单元。优选地,处理器103还包括存储器子系统,其包括存储器次序缓冲器、地址变换高速缓存(translation-lookaside buffer, TLB)、tablewalk引擎、多级缓存和各种请求队列,例如一个或多个存储器访问队列、存储器存储队列、填充队列和/或存储器监听队列。优选地,取指单元基于程序计数器值产生提供给指令高速缓存和分支目标地址高速缓存的块地址。指令高速缓存响应于块地址提供架构指令字节块,其可以包括一个或多个架构分支指令。优选地,指令字节块由指令译码器接收,该

指令译码器将架构指令翻译为提供给可被X86处理器架构的流水线执行的微指令。

[0032] 现在参考图3,示出图1的分支预测器100的操作的流程图。流程始于块302。

[0033] 在块302,从处理器的指令高速缓存取指令字节块,其预测包括至少一个条件分支指令。优选地,分支目标地址高速缓存通过查找图1的程序计数器104的值预测条件分支指令的存在。另外,散列逻辑112将程序计数器104值与分支转移历史106的不同长度进行散列运算以产生索引132而施加于图1的体124。选择的条目139被提供给比较和选择逻辑116,其选择提供给处理器的执行流水线的最终的预测142。具体地,分支预测器100将来自一个体124的条目选择为最终的预测142。其条目被选择的体124在图3中被称为体X。如上所述,优选地,比较和选择逻辑116选择具有匹配程序计数器104的标签部分的有效标签的最高体124的条目,其中最高体124是其索引132具有由散列逻辑112所使用的最长的分支转移历史106长度的体124。优选地,最终的预测142也提供给处理器的执行流水线,从而执行条件分支指令的执行单元能够比较该预测和解析的条件分支指令的正确方向并且向分支预测器100提供关于预测142是否正确的信息144。分支预测器100使用信息144更新体124。流程前进到块304。

[0034] 在块304中,执行单元执行条件分支指令,以得到其正确的方向,即,跳转或不跳转,并向分支预测器100提供正确的方向144。流程前进至块306。

[0035] 在块306中,控制逻辑114确定需要更新体124的一个或多个,所以RNG逻辑108使用指令计数器102产生图1的随机数148。如上所述,RNG逻辑108也同时使用指令计数器102和分支转移历史106产生随机数148。流程前进至块308。

[0036] 在决定块308中,控制逻辑114确定由分支预测器100预测的方向是否匹配通过执行单元在块304提供的正确的预测144。如果是,则流程进行到决定块312;否则,流程前进到决定块316。

[0037] 在决定块312,控制逻辑114检查RNG逻辑108产生的随机数148。如果随机数148是在值8-255的范围内,则流程前进到块314;否则,如果该随机数148是在范围0-7,则流程前进至决定块316。以这种方式,控制逻辑114有效地决定当预测的方向不匹配正确的方向时是否根据1:31的比例来分配一新的条目。有利的,图1的RNG逻辑108以高效的方式提供随机数148并且通过包括容易仿真的组合逻辑提供了测量由RNG逻辑108所提供的随机数148的质量(而不是通过将由仿真工具的随机数产生函数提供的随机数的质量)得到的性能增益的更精确的能力,如以上所述。

[0038] 在块314,分支预测器100没有在体124中分配新的条目,流程在块314结束。

[0039] 在决定块316,控制逻辑114检查由RNG逻辑108产生的随机数148。如果随机数148是在值0-63的范围内,则流程前进到块318;不然,如果随机数148是在范围64-255内,则流程进行到块322。以这种方式,控制逻辑114有效地决定是否开始寻找一个体124,其根据3:1的比率决定在X+1还是X+2的体进行分配。在一个实施例中,在决定块316中检查的随机数148是第二随机数148,也即,不同于在决定块308中检查的随机数148。应当理解,分支预测器使用的随机数是148,而决定块312、316和324中是作为示例描述的,并且可以设想使用其他比率的其他实施例。此外,应当理解,尽管实施例被描述,其中产生和使用的随机数148是8比特,但是可以设想其他实施例,其中不同大小的随机数148被产生并使用。

[0040] 在块318中,控制逻辑114开始于体X+2以找到最先的两个体124,其有用指示符208

具有零值。例如,如果体X是体1 124-1(即,如在块302确定的做出预测的体124),则体X+2是体3 124-3。分支预测器100可能无法找到具有零有用指示符208的两个体124,甚至具有零值的有用指示符208的一个体124。此外,分支预测器100可以只需要具有零值的有用指示符208的一个体124,例如,如果流程进至块328。流程进行到决定块324。

[0041] 在块322中,控制逻辑114开始于体X+1以找到最先的两个体124,其有用指示符208具有零值。例如,如果体X是体1 124-1(即,如在块302确定的做出预测的体124),则体X+1是体2 124-2。流程进行到决定块324。

[0042] 在决定块324中,控制逻辑114检查由RNG逻辑108产生的随机数148。如果随机数148是在值0-15的范围内,则流程前进到块326;不然,如果随机数148是在范围16-255内,则流程进行到块328。以这种方式,控制逻辑114有效地决定根据15:1的比率是否在一个体或两个体中进行分配。在一个实施例中,在决定块324中检查的随机数148是第三随机数148,即,与在决定块308和316中检查的随机数148不同。

[0043] 在块326中,控制逻辑114在块318/322中找到的两个体124两者中分配新的条目用于条件分支指令。流程在块326结束。

[0044] 在块328中,控制逻辑114仅在块318/322中找到的两个体124到体X中的最短者(即使用较短的分支预测器100的长度)中分配新的条目用于条件分支指令。流程在块328结束。

[0045] 现在参考图4,示出根据替代实施例的多体条件分支指令预测器100的框图。图4的分支预测器100在许多方面类似于图1的分支预测器100并且相同编号的元件是类似的。但是,图4的分支预测器100利用处理器时钟周期计数器(CCCC)402取代图1的指令计数器102。CCCC 402计数处理器的时钟周期的数量,优选地从处理器复位开始。CCCC 402被提供给RNG逻辑108,RNG逻辑108将它用来产生提供给控制逻辑114的随机数148,控制逻辑114使用随机数148做出有关更新存储体124的决定。相应地,在图4所示的RANDOM1和RANDOM2等式更新为 $\text{RANDOM1} = \text{CCCC}[19:10] \wedge \text{CCCC}[9:0]$ 和 $\text{RANDOM2} = \text{CCCC}[19:10] \wedge \text{CCCC}[9:0] \wedge \text{BHP}[\text{MSB}:\text{MSB}-9]$ 来说明使用CCCC 402而非指令计数器102。图4的分支预测器100的操作类似于相对于图3描述的操作;然而,在块306中RNG逻辑108使用CCCC 402而非指令计数器102比特来产生随机数148。

[0046] 现在参考图5,示出根据替代实施例的多体条件分支指令预测器100的框图。图5的分支预测器100在许多方面类似于图1的分支预测器100并且相同编号的元件是类似的。但是,图5的分支预测器100利用总线时钟周期计数器(BCCC)502取代图1的指令计数器102。BCCC 502计数处理器外部的总线的时钟周期的数量,优选地从处理器复位开始。例如,总线时钟可以是系统总线,其耦合处理器和系统的外围设备和/或存储器。BCCC 502被提供给RNG逻辑108,RNG逻辑108将它用来产生提供给控制逻辑114的随机数148,控制逻辑114使用随机数148做出有关更新存储体124的决定。相应地,在图5所示的RANDOM1和RANDOM2等式更新为 $\text{RANDOM1} = \text{BCCC}[19:10] \wedge \text{BCCC}[9:0]$ 和 $\text{RANDOM2} = \text{BCCC}[19:10] \wedge \text{BCCC}[9:0] \wedge \text{BHP}[\text{MSB}:\text{MSB}-9]$ 来说明使用BCCC 502而非指令计数器102。图5的分支预测器100的操作类似于相对于图3描述的操作;然而,在块306中RNG逻辑108使用BCCC 502而非指令计数器102比特来产生随机数148。

[0047] 现在参考图6,示出图示图1的分支预测器100做出关于更新有用指示符208的决定的操作的流程图。在一个实施例中,每次分支预测器100选择体124做出预测142(图3中表示

为体X),分支预测器100还记住备用的预测体124,在此被称为体Y。体Y是如果在体X中出现未命中则将用来做出预测142的体124。也即,体Y是下一个最低的体124,其中程序计数器104的标签部分也命中了,以及如果没有其中存在命中的较低的体124,则体Y是默认体124,例如,体0。优选的,每当体X正确地预测条件分支指令的方向而体Y错误地预测,则控制逻辑114将有用指示符208递增。如相对于图3描述的,有用指示符208用来在体124中分配条目。然而,如果有用指示符208仅曾递增,并且从不递减,则最终将没有零值的有用指示符208,这将对分配方案产生负面影响。所以,如在Seznec的TAGE文章所述的,有必要以老化有用指示符208以将它们复位至零。发表在2011年5月20日的、Andre Seznec的题为“A 64Kbytes ISL-TAGE branch predictor”的文章的第2.2节描述了使用8位计数器,简称TICK,用于动态监测当试图在错误预测之后分配新的条目时成功和失败的次数。TICK计数器当在分配遭遇失败多于成功时饱和,并且此时Seznec的预测器重置预测器的所有的有用比特。图6描述了Seznec方案的变体。优选地,图1的分支预测器100的控制逻辑114还包括一计数器(未示出),这里称为T。针对图6描述了结合由RNG逻辑108基于指令计数器102(或CCCC 402或BCCC 502)和/或分支转移历史106产生的随机数148的T的使用。流程开始于块602。

[0048] 在块602中,结合在块318或块322执行的操作,控制逻辑114确定两个值,N和P。N是来自在块318/322查找(即搜索)的、具有零值的有用指示符208的体124的有用指示符208的数量,而P是来自查找的体124的非零值的有用指示符208的数量。流程前进至块604。

[0049] 在块604,控制逻辑114将T的值增加P并且将T的值减少N。在一个实施例中,T是10比特计数器因此具有0-1023的范围并且当处理器复位时初始化到0。流程前进到决定块606。

[0050] 在决定块606,控制逻辑114确定T的值是否大于或等于随机数148的值。如果不是,则流程结束;否则,流程继续到决定块608。在一个实施例中,在决定块606中比较的随机数148是7比特随机数148,因此具有范围0-127。

[0051] 在决定块608中,控制逻辑114检查由RNG逻辑108产生的随机数148。如果随机数148是在值0-127的范围内,则流程结束;不然,如果随机数148是在范围128-255,则流程前进到块612。以这种方式,控制逻辑114有效地决定是否根据1:1比率递减有用指示符208。在一个实施例中,在决定块608中检查的随机数148是第二随机数148,也即,不同于在决定块606中检查的随机数148。应当理解的是,由分支预测器100基于例如在决定块608的随机数148使用的比率被描述为一个例子,其他实施例可以设想使用其它比率。此外,应当理解,尽管实施例被描述,其中在块606和608中产生和使用的随机数148分别是7比特和8比特,可以设想其他实施例,其中不同大小的随机数148被产生和使用。

[0052] 虽然在这里已经描述了本发明的各种实施例,应当理解它们已经通过示例的方式呈现,而不是限制。对于相关计算机领域中的普通技术人员很明显可以在其中做出形式和细节的各种改变而不脱离本发明的范围。例如,软件例如可以使能在这里描述的装置和方法的功能、制造、建模、模拟、描述和/或测试。这可以通过通用编程语言(例如,C,C++)、包括Verilog HDL、VHDL等的硬件描述语言(HDL)或者其他可用程序的使用实现。这种软件可以以任何已知的计算机可用介质配置,比如磁带、半导体、磁盘或者光盘(例如,CD-ROM、DVD-ROM等)、网络,有线线路或者其他通信介质。在这里描述的设备和方法的实施例可以包括在半导体知识产权核心中,比如处理器核心(例如,以HDL实现或者指定)和可以变换为集成

电路的生产中的硬件。另外,在这里描述的设备和方法可以具体表现为硬件和软件的组合。因此,本发明不应该由在这里描述的任何示例性实施例限制,而是应该仅根据以下权利要求和它们的等同物来定义。特别地,本发明可以在通用计算机中使用的处理器装置内实现。最终,本领域技术人员应该认可,他们可以容易地使用公开的概念和特定实施例作为设计或者修改其他结构以进行本发明的相同目的的基础,而不脱离本发明如所附权利要求所定义的保护范围。

100

$$\text{RANDOM1} = \text{IC}[15:8] \wedge \text{IC}[7:0]$$
$$\text{RANDOM2} = \text{IC}[15:8] \wedge \text{IC}[7:0] \wedge \text{BHP}[\text{MSB}:\text{MSB}-7]$$

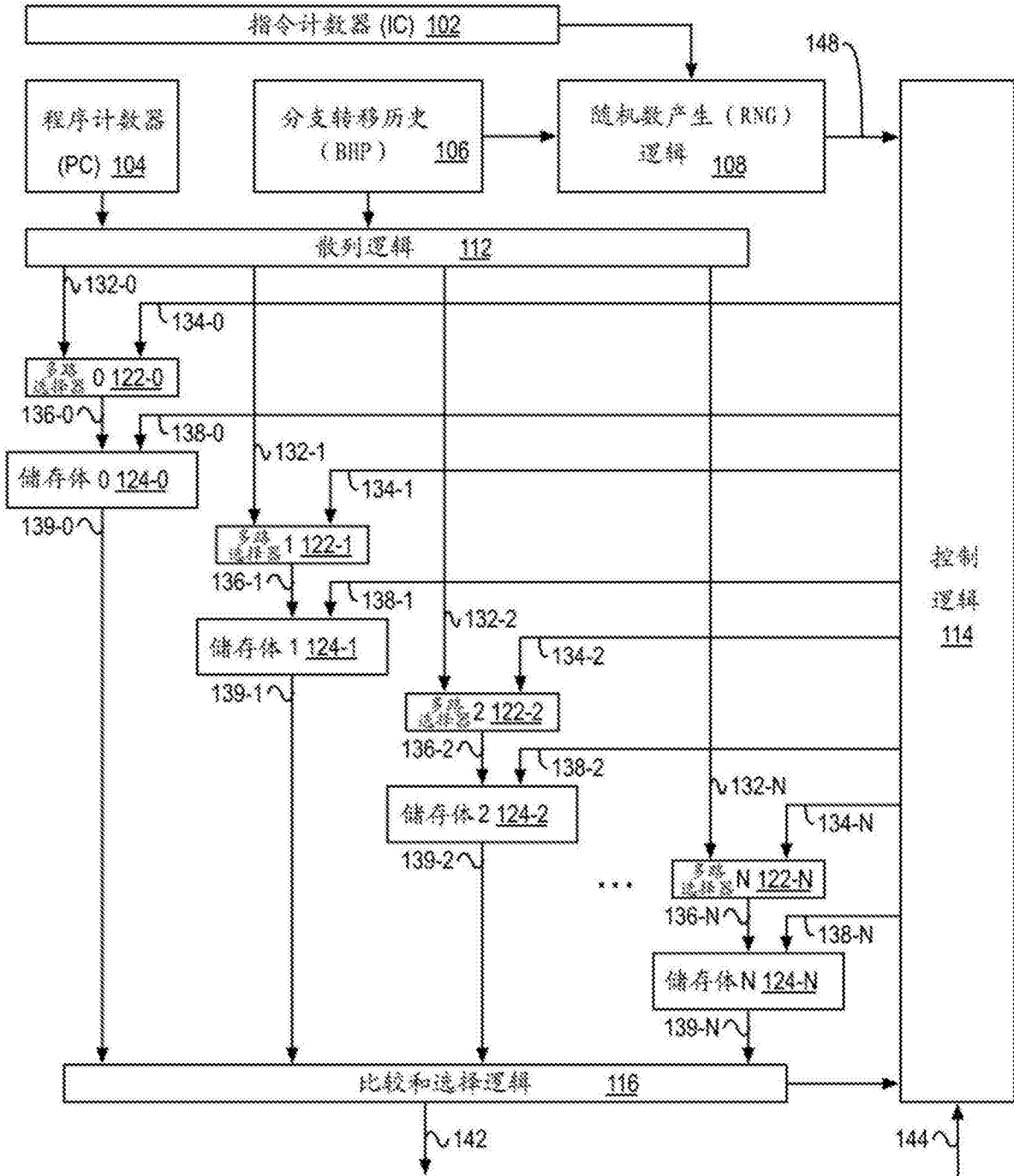


图1



图2

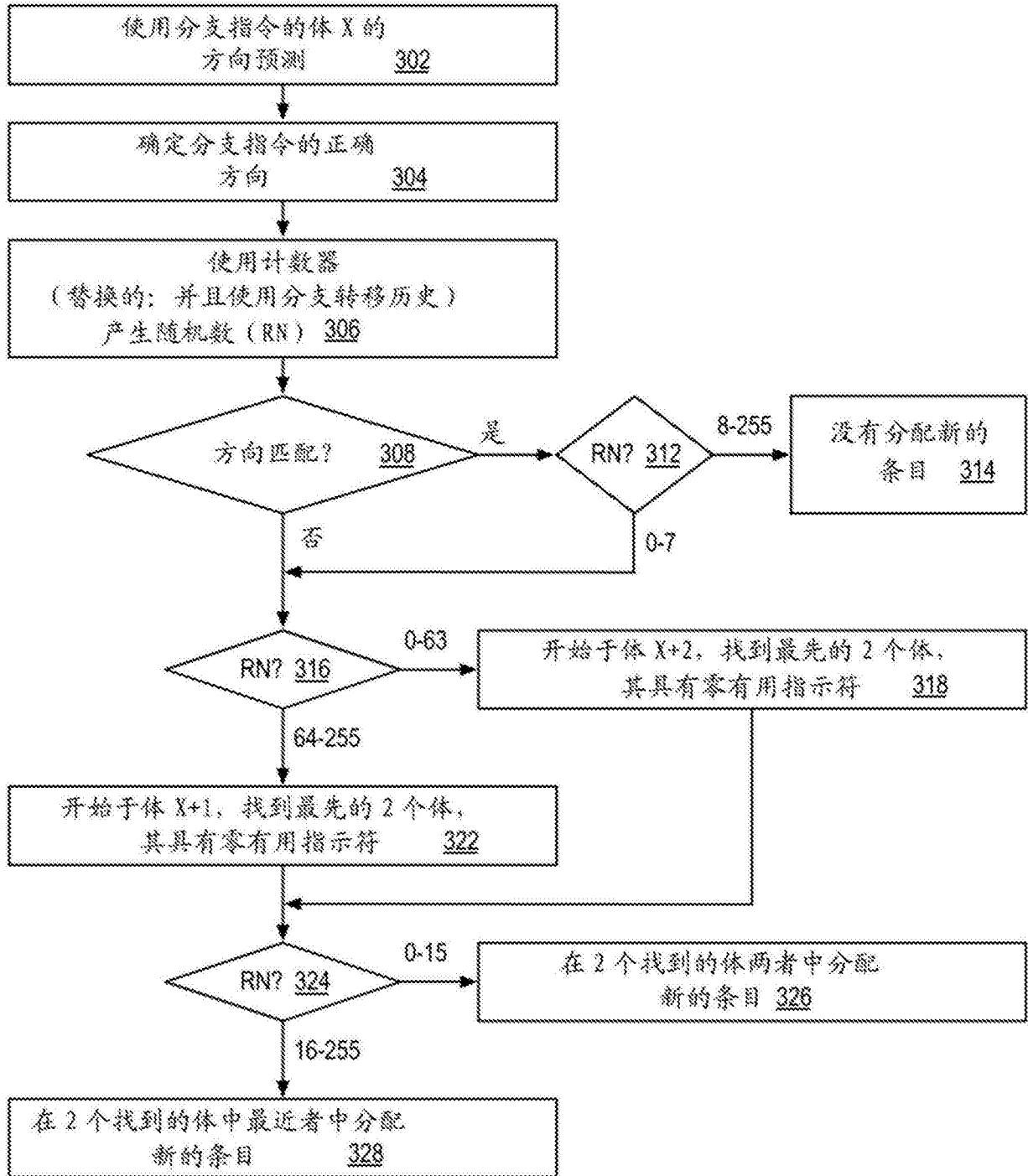


图3

100

$$\text{RANDOM1} = \text{CCCC}[15:8] \wedge \text{CCCC}[7:0]$$

$$\text{RANDOM2} = \text{CCCC}[15:8] \wedge \text{CCCC}[7:0] \wedge \text{BHP}[\text{MSB}:\text{MSB}-7]$$

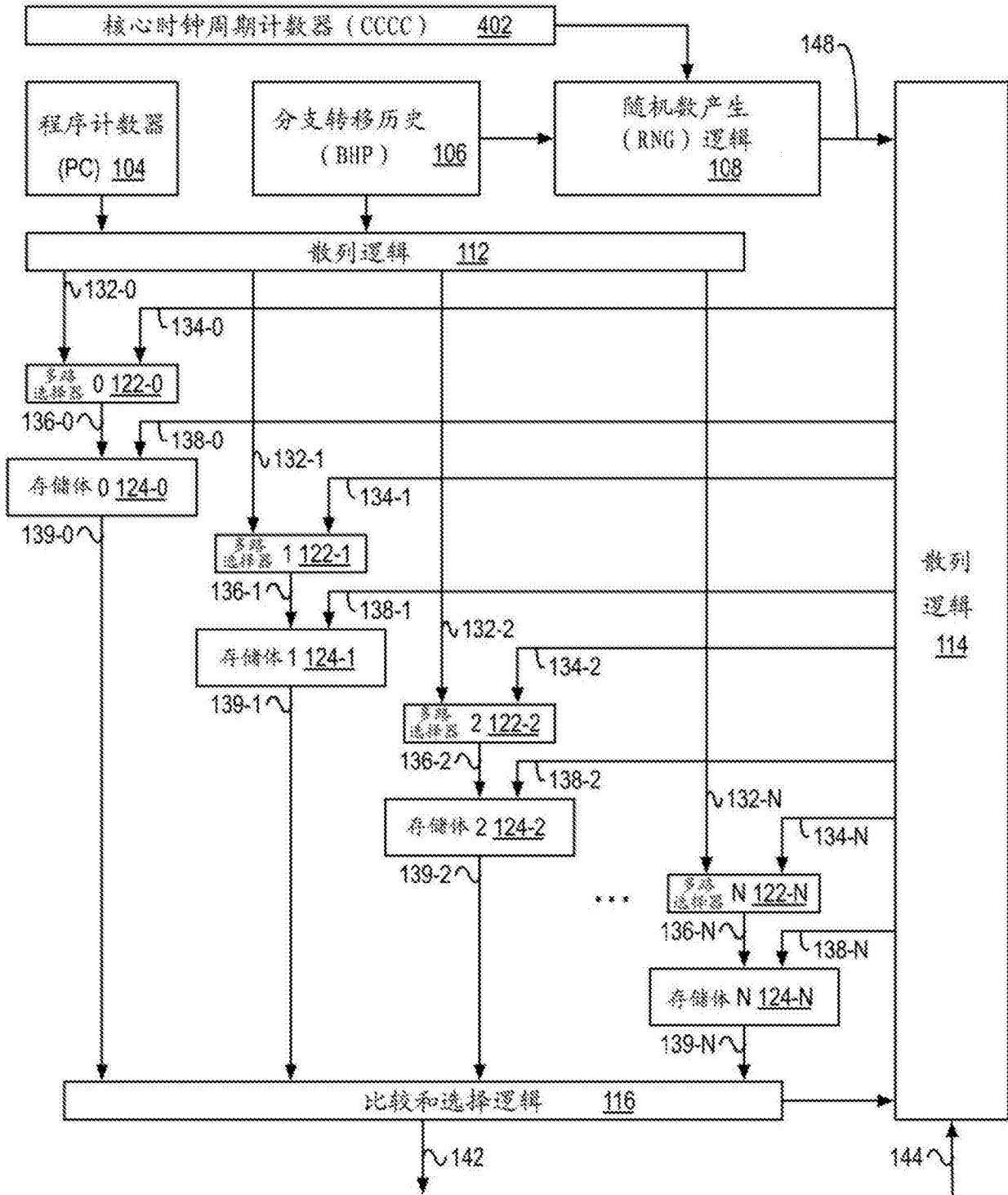


图4

$$\text{RANDOM1} = \text{BCCC}[15:8] \wedge \text{BCCC}[7:0]$$
$$\text{RANDOM2} = \text{BCCC}[15:8] \wedge \text{BCCC}[7:0] \wedge \text{BHP}[\text{MSB}:\text{MSB}-7]$$

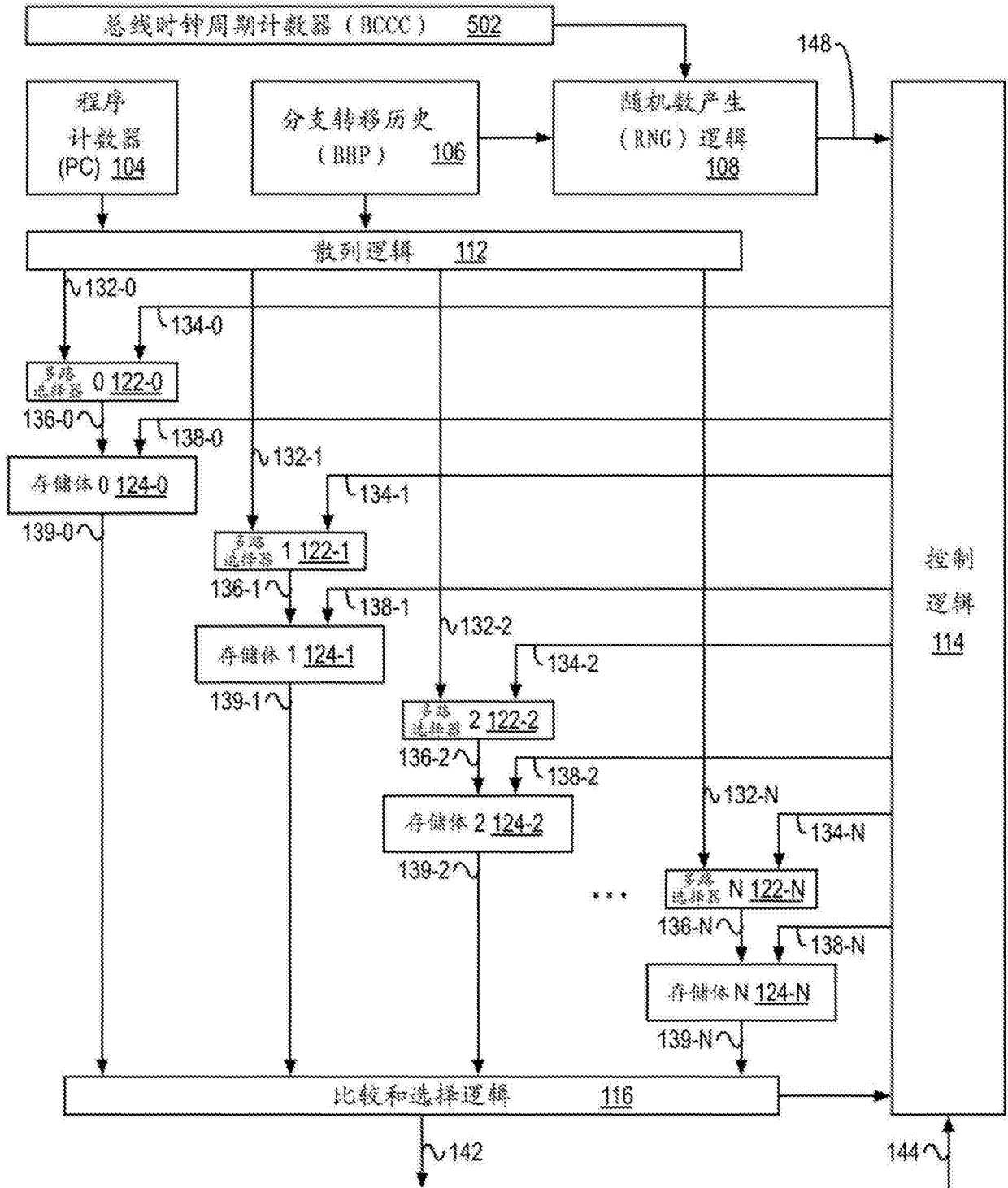


图5

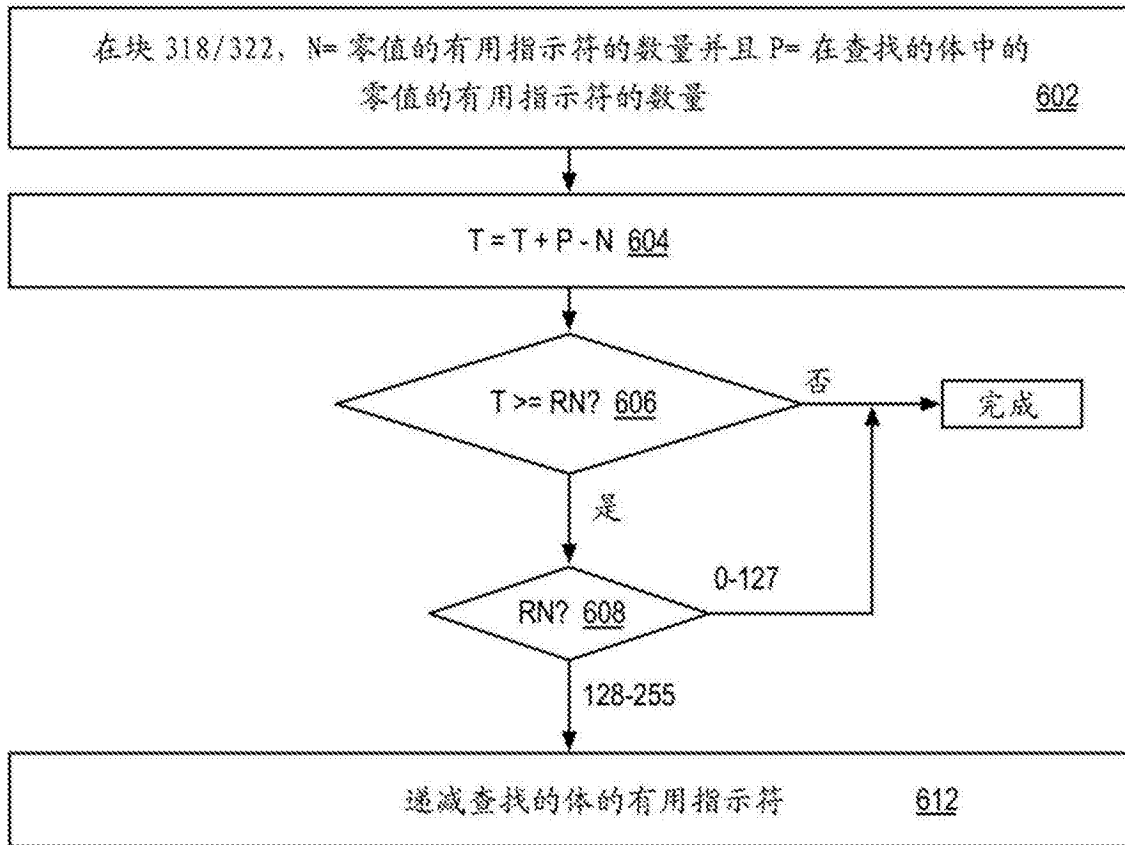


图6