

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 12/08 (2006.01)

G06F 11/34 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200480014782.9

[43] 公开日 2006年6月28日

[11] 公开号 CN 1795438A

[22] 申请日 2004.5.12

[21] 申请号 200480014782.9

[30] 优先权

[32] 2003.5.27 [33] US [31] 10/446,004

[86] 国际申请 PCT/US2004/014845 2004.5.12

[87] 国际公布 WO2004/107177 英 2004.12.9

[85] 进入国家阶段日期 2005.11.28

[71] 申请人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 G·罗尼 C·-K·卢

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 程天正 王 勇

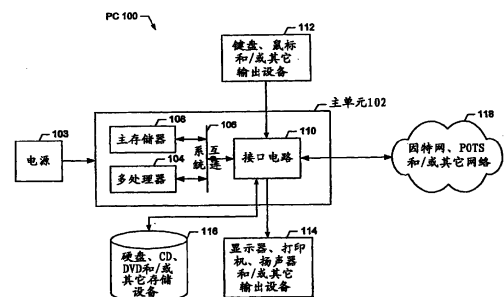
权利要求书 4 页 说明书 9 页 附图 4 页

[54] 发明名称

用于跨距剖析软件应用的方法和设备

[57] 摘要

公开了一种用于跨距剖析软件应用的方法和设备。示例性的系统使用硬件性能计数器来报告与由某个事件(诸如数据高速缓存器未命中)所触发的存储器访问指令相关联的数据地址和指令地址。当相同的指令地址被与一个以上的数据地址相关联时,记录在两个数据地址之间的差异。当对于相同的指令记录了这些数据地址差异中的两个或更多时,该系统确定与所述指令相关联的跨距是所述两个或更多差异的最大公约数。此跨距可以由编译器用来优化数据高速缓存器预取。另外,通过在审查阶段和跳跃阶段之间循环可以降低与监视数据高速缓存器未命中的地址相关联的任何开销。在审查阶段期间比在跳跃阶段期间监视更多的数据高速缓存器未命中。



1. 一种用于跨距剖析软件应用的方法，所述方法包括：
从硬件性能计数器接收用于表示第一指令地址和第一数据地址的数据；
- 5 从所述硬件性能计数器接收用于表示第二指令地址和第二数据地址的数据；
 如果所述第一指令地址等于所述第二指令地址，那么通过从所述第二数据地址减去所述第一数据地址来确定第一跨距积；并且
 根据所述第一跨距积来确定跨距。
- 10 2. 如权利要求1所述的方法，其中根据所述第一跨距积确定跨距包括：确定所述跨距等于所述第一跨距积。
3. 如权利要求1所述的方法，还包括：
 从所述硬件性能计数器接收用于表示第三指令地址和第三数据地址的数据；
- 15 如果所述第二指令地址等于所述第三指令地址，那么通过从所述第三数据地址中减去所述第二数据地址来确定第二跨距积。
4. 如权利要求3所述的方法，其中根据所述第一跨距积确定所述跨距包括：确定所述第一跨距积和所述第二跨距积的最大公约数。
5. 如权利要求3所述的方法，其中从硬件性能计数器接收表示第一指令地址和第一数据地址的数据包括：接收所述第一指令地址和所述
20 第一数据地址作为表示第一多个连续的数据高速缓存器未命中的第一数据集的一部分。
6. 如权利要求5所述的方法，还包括：
 接收用于表示第一多个连续的数据高速缓存器未命中的第一数据集；
25 并且
 接收用于表示第二多个连续的数据高速缓存器未命中的第二数据集，其中接收第一数据集和接收第二数据集按预定的周期分隔开，在所述预定的周期期间，不接收用于表示连续的数据高速缓存器未命中的数据。
- 30 7. 如权利要求1所述的方法，还包括：
 从所述硬件性能计数器接收用于表示第三指令地址和第三数据地址的数据；

从所述硬件性能计数器接收用于表示第四指令地址和第四数据地址的数据；

如果所述第一指令地址等于所述第三指令地址和所述第四指令地址，那么通过从所述第四数据地址中减去所述第三数据地址来确定第二跨距积。

8. 如权利要求 7 所述的方法，其中根据所述第一跨距积确定所述跨距包括：确定所述第一跨距积和所述第二跨距积的最大公约数。

9. 如权利要求 1 所述的方法，其中从硬件性能计数器接收用于表示第一指令地址和第一数据地址的数据包括：接收所述第一指令地址和所述第一数据地址作为用于表示多个连续的数据高速缓存器未命中的数据集的一部分。

10. 如权利要求 1 所述的方法，还包括以周期性的脉冲串系列采样硬件性能计数器。

11. 一种用于剖析软件应用的设备，所述设备包括：

15 用于执行软件应用和剖析工具的处理器；

硬件性能计数器，可操作地耦合到所述处理器，所述硬件性能计数器用于向剖析工具提供数据，所述数据用于表示第一指令地址、与所述第一指令地址相关联的第一数据地址、第二指令地址和与所述第二指令地址相关联的第二数据地址，

20 其中如果所述第一指令地址等于所述第二指令地址，那么所述剖析工具根据在所述第一数据地址和所述第二数据地址之间的差异来确定跨距。

12. 如权利要求 11 所述的设备，其中：

25 由硬件性能计数器向所述剖析工具所提供的数据还包括第三指令地址和与所述第三指令地址相关联的第三数据地址，

如果所述第一指令地址等于所述第二指令地址，那么所述剖析工具通过从所述第二数据地址中减去所述第一数据地址来确定第一跨距积，

30 如果所述第二指令地址等于所述第三指令地址，那么所述剖析工具通过从所述第三数据地址中减去所述第二数据地址来确定第二跨距积，并且

所述剖析工具根据所述第一跨距积和所述第二跨距积来确定跨

距。

13. 如权利要求 12 所述的设备, 其中所述剖析工具通过确定所述第一跨距积和所述第二跨距积的最大公约数来确定所述跨距。

5 14. 如权利要求 11 所述的设备, 其中当在执行软件应用期间出现第一次数据高速缓存器未命中时, 所述硬件性能计数器向所述剖析工具提供用于表示所述第一指令地址和所述第一数据地址的数据。

15. 如权利要求 14 所述的设备, 其中当在执行软件应用期间出现第二次数据高速缓存器未命中时, 所述硬件性能计数器向所述剖析工具提供用于表示所述第二指令地址和所述第二数据地址的数据。

10 16. 如权利要求 15 所述的设备, 其中所述第一次数据高速缓存器未命中和第二次数据高速缓存器未命中是与所述第一指令地址相关联的连续的数据高速缓存器未命中。

17. 如权利要求 11 所述的设备, 其中所述剖析工具以周期性的脉冲串系列采样所述硬件性能计数器。

15 18. 一种用于剖析软件应用的设备, 所述设备包括:

硬件性能计数器;

跨距确定模块, 可操作来与所述硬件性能计数器耦合; 和

最大公约数模块, 可操作来与阶段管理器耦合。

20 19. 如权利要求 18 所述的设备, 其中所述硬件性能计数器被构造来捕获用于表示与数据高速缓存器未命中相关联的指令地址以及与数据高速缓存器未命中相关联的数据地址的数据。

20. 如权利要求 18 所述的设备, 其中所述跨距确定模块被构造来在第一指令地址等于第二指令地址的情况下从第二数据地址中减去第一数据地址。

25 21. 如权利要求 18 所述的设备, 还包括阶段管理器, 可操作来与所述跨距确定模块耦合。

22. 如权利要求 21 所述的设备, 其中所述阶段管理器被构造来产生从第一跳跃阶段到审查阶段的第一转变。

30 23. 如权利要求 22 所述的设备, 其中所述阶段管理器被构造来产生从审查阶段到第二跳跃阶段的第二转变。

24. 如权利要求 18 所述的设备, 其中所述最大公约数模块被构造来确定第一跨距积和所述第二跨距积的最大公约数。

25. 一种机器可读介质，被构造来使机器能够：

根据来自硬件性能计数器的高速缓存器未命中数据产生跨距剖析数据；并且

根据所述跨距剖析数据来编译程序。

5 26. 如权利要求 25 所述的机器可读介质，还被构造来使机器能够：

从硬件性能计数器接收第一数据，所述第一数据用于表示第一指令地址、与所述第一指令地址相关联的第一数据地址、第二指令地址和与所述第二指令地址相关联的第二数据地址；并且

10 如果所述第一指令地址等于所述第二指令地址，那么通过从所述第二数据地址中减去所述第一数据地址来确定第一跨距积。

27. 如权利要求 26 所述的机器可读介质，还被构造来使机器能够：

15 从所述硬件性能计数器接收第二数据，所述第二数据用于表示第三指令地址和与所述第三指令地址相关联的第三数据地址；并且

如果所述第二指令地址等于所述第三指令地址，那么通过从所述第三数据地址中减去所述第二数据地址来确定第二跨距积。

28. 如权利要求 27 所述的机器可读介质，还被构造来使机器能够通过确定所述第一跨距积和所述第二跨距积的最大公约数来确定跨距。

20 29. 如权利要求 25 所述的机器可读介质，还被构造来使机器能够以周期性的脉冲串系列采样所述硬件性能计数器。

用于跨距剖析软件应用的方法和设备

技术领域

5 本发明涉及软件编译器，并且尤其涉及用于跨距剖析（profiling）软件应用的方法和设备。

背景技术

在处理器和存储器速度之间的距离继续加大。结果，计算机性能逐渐地用高速缓存器分级结构的有效性来确定。然而，处理器工作量
10 典型地会招致显著的高速缓存未命中。

预取是用于改进高速缓存器分级结构有效性的公知且有效的技术。用于改进预取准确性的一种编译器技术在于统计地发现具有恒定“跨距（stride）”的存储器访问指令（例如，加载、存储等）。例如，加载每第十六个字节的加载指令易于被预取，这是因为编译器提前知道将需要什么字节。然而，由于直到运行时才被解析的间接数组引用和指针废弃，导致编译器不能统计地发现具有恒定跨距的许多存
15 储器访问指令。

为了解决此问题，可以把“工具”代码（即，测试码）添加到软件应用以便直接监视由一个或多个存储器访问指令所访问的实际数据
20 地址。然而，工具代码给软件应用增加了显著的开销，这使所述应用慢了下来（例如，慢了10倍）。

附图说明

图1是图示用于所公开系统环境的示例性计算机系统的框图。

图2是图示用于所公开系统环境的另一示例性计算机系统的框
25 图。

图3是包括加载指令的代码部分的例子。

图4是示例性的阶段图。

图5是表示示例性机器可读指令的流程图，所述机器可读指令可以由一设备执行来实现用于跨距剖析软件应用的示例性方法。

30 具体实施方式

在总体上，所公开的系统使用硬件性能计数器来报告与由某个事件（诸如数据高速缓存器未命中）所触发的存储器访问指令相关联的

数据地址和指令地址。当相同的指令地址被与一个以上的数据地址相关联时，记录在两个数据地址之间的差异。当对于相同的指令记录了这些数据地址差异中的两个或更多时，该系统确定与所述指令相关联的跨距是所述两个或更多差异的最大公约数。此跨距可以由编译器用来优化数据高速缓存器预取。另外，通过在审查阶段和跳跃阶段之间循环可以降低与监视数据高速缓存器未命中的地址相关联的任何开销。在审查阶段期间比在跳跃阶段期间监视更多的数据高速缓存器未命中。

在图 1 中图示了示例性计算机系统 100 的框图。计算机系统 100 可以是个人计算机 (PC)，也可以是任何其它计算装置。在一个例子中，计算机系统 100 包括由电源 103 供电的主处理单元 102。主处理单元 102 可以包括处理器 104，所述处理器 104 由系统互连 106 电耦合到主存储设备 108 和一个或多个接口电路 110。在一个例子中，系统互连 106 是地址/数据总线。当然，本领域内普通技术人员应当容易地理解到：可以使用不同于总线的互连来把处理器 104 连接到主存储设备 108。例如，一个或多个专用线路和/或交叉开关可以用来把处理器 104 连接到主存储设备 108。

处理器 104 可以是任何类型的公知处理器，诸如来自微处理器的 Intel Pentium®系列、微处理器的 Intel Itanium®系列、微处理器的 Intel Centrino®系列和/或微处理器的 Intel XScale®系列的处理器。另外，处理器 104 可以包括任何类型的公知高速缓冲存储器，诸如静态随机存取存储器 (static random access memory SRAM)。主存储设备 108 可以包括动态随机存取存储器 (dynamic random access memory DRAM) 和/或任何其它形式的随机存取存储器。例如，主存储设备 108 可以包括双数据率随机存取存储器 (double data rate random access memory DDRAM)。主存储设备 108 还可以包括非易失性存储器。在一个例子中，主存储设备 108 依照公知方式来存储由处理器 104 所执行的软件程序。

接口电路 (一个或多个) 110 可以使用任何类型的公知接口标准来实现，诸如以太网接口和/或通用串行总线 (Universal Serial Bus USB) 接口。一个或多个输入设备 112 可以被连接到接口电路 110 以便把数据和命令输入到主处理单元 102 中。例如，输入设备 112 可以是

键盘、鼠标、触摸屏、跟踪板、轨迹球、等电点 (isopoint) 和 / 或语音识别系统。

一个或多个显示器、打印机、扬声器和 / 或其它输出设备 114 也可以经由一个或多个接口电路 110 被连接到主处理单元 102。显示器 5 114 可以是阴极射线管 (cathode ray tube CRT)、液晶显示器 (liquid crystal displays LCD) 或任何其它类型的显示器。显示器 114 可以产生在主处理单元 102 操作期间所产生数据的可见指示。可见指示可以包括对操作员输入、所计算的值、所检测的数据等的提示。

计算机系统 100 还可以包括一个或多个存储设备 116。例如，计算机 10 系统 100 可以包括一个或多个硬盘驱动器、光盘 (compact disk CD) 驱动器、数字化多功能盘驱动器 (digital versatile disk DVD) 和 / 或其它计算机介质输入 / 输出 (I/O) 设备。

计算机系统 100 还可以经由到网络 118 的连接来与其它设备交换数据。网络连接可以是任何类型的网络连接，诸如以太网连接、数字 15 用户线路 (digital subscriber line DSL)、电话线路、同轴电缆等。网络 118 可以是任何类型的网络，诸如因特网、电话网络、有线电视网和 / 或无线网络。

在图 2 中图示了另一示例性计算机系统 200。在此例子中，计算机系统 200 包括按照所示互相连接的处理器 202、高速缓存器 204、主存 20 储器 206、硬件性能计数器 208、跨距确定模块 210、最大公约数模块 212 和阶段管理器 214。

处理器 202 也可以是任何类型的公知处理器，诸如来自微处理器的 Intel Pentium® 系列、微处理器的 Intel Itanium® 系列、微处理器的 Intel Centrion® 系列和 / 或微处理器的 Intel XScale® 系列的 25 处理器。另外，处理器 202 可以包括任何类型的公知高速缓冲存储器 204，诸如静态随机存取存储器 (SRAM)。主存储设备 206 可以包括动态随机存取存储器 (DRAM) 和 / 或任何其它形式的随机存取存储器。例如，主存储设备 206 可以包括双数据率随机存取存储器 (DRAM)。主存储设备 206 还可以包括非易失性存储器。在一个例子中，主存储 30 设备 206 依照公知方式来存储由处理器 202 所执行的软件程序。

典型情况下，当处理器 202 遇到需要来自主存储器 206 的数据值的指令 (例如，加载或存储指令) 时，计算机系统 200 首先检查所需

5 要的值是否存储在高速缓存器 204 中。如果所需要的值存储在了高速缓存器 204 中（即，出现高速缓存器命中），那么不必从主存储器 206 获取所需要的数据值。然而，如果所述数据值没有存储在高速缓存器 204 中（即，出现高速缓存器未命中），那么可以从主存储器 206 获取所述数据值。

10 在此例子中，每当出现数据高速缓存器未命中时，硬件性能计数器 208 就捕获导致数据高速缓存器未命中的指令（例如，加载或存储指令）的地址和所需要的数据值的地址。在图 3 中图示了包括加载指令 302 的代码部分 300 的例子。在所图示的例子中，把加载指令 302 与指令地址“1005”，304 相关联。类似地，把加载指令 302 与数据地址“2000 + X”，306 相关联。用于保持由处理器 202 作为执行加载指令 302 的结果所获取的数据值的存储单元取决于变址变量“X”（308）的值。例如，如果 $X = 0$ ，那么加载指令使位于存储单元 2000 的数据值被获取。如果 $X = 48$ ，那么加载指令使位于存储单元 2048 15 的数据值被获取。

换句话说，保持加载指令 302 的存储单元是“1005”，并且每当由处理器 202 执行此指令时，地址“1005”将与操作相关联。然而，保持作为执行此指令的结果而获取的数据值的地址可能会从加载指令 302 的一次执行到所述加载指令 302 的下一次执行发生变化，这取决于 20 X 的值。例如，如果 X 在加载指令 302 的每次执行之间增量 48，那么加载指令 302 将加载存储单元 2000、2048、2096 等。在此例子中，加载指令 302 的跨距为 48。

25 跨距确定模块 210 使用从硬件性能计数器 208 接收的数据地址和指令地址来确定某些指令的跨距（如果有的话）。例如，由于每当通过包括加载指令 302 的循环 310 时 X 增加 48，所以图 3 中的加载指令 302 的跨距为 48。如果硬件性能计数器 208 捕获每个指令地址和相关联的数据地址，那么跨距确定模块 210 就可能要监视指令地址 1005 的两次连续出现，继而从第一次出现的数据地址中减去第二次出现的数据地址以便确定跨距（例如， $2048 - 2000 = 48$ ）。

30 然而，如果当出现高速缓存器未命中时硬件性能计数器 208 只捕获指令和数据地址，那么在与相同指令地址相关联的两个连续数据地址之间的差异可能是实际跨距的某个倍数。例如，图 3 中的加载指令

302 的跨距为 48。然而，如果只有加载指令 302 的第一次出现（例如，数据地址 = 2000）和加载指令 302 的第六次出现（例如，数据地址 = 2240）遭到高速缓存器未命中，那么在这两个数据地址之间的差异（即， $2240 - 2000$ ）将是 240（即， $48 * 5$ ）而不是 48。在此例子中，值 240 是“跨距积”。

由于未报告的指令的出现（例如，由于未由硬件性能计数器 208 报告的数据高速缓存器命中）所致，跨距积是跨距值（例如，48）乘以某个跨距倍数（例如，5）。可以通过从与相同指令地址相关联的前一数据地址值中减去一个数据地址值来确定跨距积（例如， $2240 - 2000 = 240$ ）。

为了根据跨距积来确定实际的跨距值，计算机系统 200 包括最大公约数模块 212。最大公约数模块 212 依照公知方式取得两个或更多的输入并且确定所述输入的最大公约数。特别地是，最大公约数模块 212 取得两个或更多的跨距积并且确定跨距值。例如，假定硬件性能计数器 208 报告相同的加载指令 302 遭受了三次数据高速缓存器未命中。在此例子中进一步假定与加载指令 302 的第一次报告相关联的数据地址是 2000，与加载指令 302 的第二次报告相关联的数据地址是 2240，并且与加载指令 302 的第三次报告相关联的数据地址是 2576。与此例子相关联的第一跨距积是 240（即， $2240 - 2000$ ）。与此例子相关联的第二跨距积是 336（即 $2576 - 2240$ ）。240 和 336 的最大公约数是 48，并且在此例子中 48 是加载指令 302 的实际跨距。

在大部分计算系统 200 中，使用来自硬件性能计数器 208 的高速缓存器未命中数据来确定跨距值给计算机系统 200 增加了处理开销。为了降低此开销，计算机系统 200 包括阶段管理器 214。阶段管理器 214 在两个或更多阶段之间转变跨距确定模块 210。在图 4 中图示了示例性阶段图。在此例子中，阶段管理器 214 在跳跃阶段 402 和审查阶段 404 之间转变跨距确定模块 210。在跳跃阶段 402 的每个实例期间，跨距确定模块 210 要比在审查阶段 404 期间处理来自硬件性能计数器 208 的、更少的高速缓存器未命中报告。例如，在跳跃阶段 402 期间，对于每 1000 个数据高速缓存器未命中，跨距确定模块 210 可能只采样硬件性能计数器 208 一次。相比之下，在审查阶段 404 期间，跨距确定模块 210 可能每个数据高速缓存器未命中都采样硬件性能计数器

208。

在图 5 中图示了用于跨距剖析软件应用的示例性过程 500。优选地是，过程 500 被实现为一个或多个软件程序，所述软件程序依照公知方式存储在一个或多个存储器中并且由一个或多个处理器（例如，处理器 104 和 / 或处理器 202）执行。然而，过程 500 的某些或全部块可以人工地执行和 / 或由某个其它设备执行。尽管参考在图 5 中所图示的流程图描述了过程 500，然而本领域内普通技术人员应当容易地理解可以使用用于执行过程 500 的许多其它方法。例如，可以修改许多块的次序，可以改变一个或多个块的操作，可以组合块，和 / 或可以删去块。

通常，示例性的过程 500 使用硬件性能计数器来报告与由某个事件（诸如数据高速缓存器未命中）所触发的存储器访问指令相关联的数据地址和指令地址。当相同的指令地址被与一个以上的数据地址相关联时，记录在两个数据地址之间的差异。当对于相同的指令记录了这些数据地址差异中的两个或更多时，系统确定与所述指令相关联的跨距是所述两个或更多差异的最大公约数。此跨距可以由编译器用来优化数据高速缓存器预取。另外，通过在审查阶段和跳跃阶段之间循环可以降低与监视数据高速缓存器未命中的地址相关联的任何开销。在审查阶段期间比在跳跃阶段期间监视更多的数据高速缓存器未命中。

示例性过程 500 通过初始化几个变量开始（块 502）。特别地是，示例性过程 500 初始化跳跃计数器、跳跃阈值、审查计数器、审查阈值和捕获数组。可以把这些变量中的每个都存储在任何存储器（诸如主存储器 108、206）中。跳跃计数器用来计数在跳跃阶段期间所出现的数据高速缓存器（D 高速缓存器）未命中的数目，而跳跃阈值用来触发从跳跃阶段 402 到审查阶段 404 的改变。跳跃阶段 402 用来降低采样硬件性能计数器 208 的速率，以便降低与正在执行的跨距剖析相关联的开销。例如，对于每个跳跃阶段 402 可以跳过一千个 D 高速缓存器未命中。

类似地，审查计数器用来计数在审查阶段 404 期间所出现的 D 高速缓存器未命中的数目，而审查阈值用来触发从一个审查阶段 404 到下一跳跃阶段 402 的改变。审查阶段 404 用来提高采样硬件性能计数器 208 的速率，以便收集用于执行跨距剖析的数据。例如，在每个审

查阶段 404 期间可以收集与一千个连续的 D 高速缓存器未命中相关联的指令和数据地址。捕获数组可以用来记录在每个审查阶段 404 期间所收集的地址。

一旦初始化这些变量，那么过程 500 通过等待 D 高速缓存器未命中来开始跳跃阶段 402 (块 504)。在一个例子中，硬件性能计数器 208 被配置为每当出现 D 高速缓存器未命中时触发。因为这是跳跃阶段 402，所以过程 500 只在每当出现 D 高速缓存器未命中时才增量跳跃计数器 (块 506)。当 D 高速缓存器未命中的数目超过跳跃阈值时，过程 500 继续至审查阶段 404 (块 508)。

在审查阶段 404 中，对于每个 D 高速缓存器未命中，都从硬件性能计数器 208 接收指令地址和相关联的数据地址 (块 510)。所述指令地址是访问特定存储器段的指令的地址。数据地址是由所述指令所访问的特定存储器段的地址。当在审查阶段 404 期间出现 D 高速缓存器未命中时，过程 500 确定在捕获数组中是否已经记录了相同的指令地址 (块 512)。换句话说，过程 500 检查在此审查阶段 404 的早些时候该相同的指令是否遭到 D 高速缓存器未命中。

如果相同的指令地址尚未被记录在捕获数组中 (即，这是此特定的指令在此审查阶段 404 期间第一次遭到 D 高速缓存器未命中)，那么过程 500 就增量审查计数器 (块 514) 并且检查审查计数器是否超过审查阈值 (块 516)。如果没有超过审查阈值 (即，仍然出现审查阶段 404)，那么过程 500 等待另一 D 高速缓存器未命中 (块 518)。当出现下一 D 高速缓存器未命中时，过程再次从硬件性能计数器 208 接收指令地址和相关联的数据地址 (块 510)。如果此 D 高速缓存器未命中与不同的指令地址相关联 (块 512)，那么过程 500 再次增量审查计数器 (块 514)，并且再次检查审查计数器是否超过审查阈值 (块 516)。

然而，如果此 D 高速缓存器未命中与先前已经记录在捕获数组中的指令地址相关联 (块 512)，那么过程 500 为此指令地址计算“跨距积” (块 520)。跨距积是跨距值乘以跨距倍数。例如，如果跨距值是 48 字节而跨距倍数是 5，那么跨距积是 240。跨距倍数是 D 高速缓存器命中的结果。例如，如果特定的存储器访问指令遭到 D 高速缓存器未命中，后面是四个 D 高速缓存器命中，后面是另一 D 高速缓存器未命中，那么所述跨距倍数是 5。通过从当前数据地址中减去先前与此指令

地址相关联地记录的数据地址来确定跨距积。

然后过程 500 确定所计算的跨距积是否是为该指令地址所计算的第一个跨距积 (块 522)。例如, 如果与该指令地址相关联只出现了两个高速缓存器未命中, 那么只确定了一个跨距积 (即, 这是第一个跨距积)。然而, 如果与该指令地址相关联已经出现了三个 (或更多) 高速缓存器未命中, 那么可能已经确定了一个以上的跨距积。例如, 在用于第一次高速缓存器未命中的数据地址和用于第二次高速缓存器未命中的数据地址之间的差异可能会产生第一跨距积, 而在用于第二次高速缓存器未命中的数据地址和用于第三次高速缓存器未命中的数据地址之间的差异可能会产生第二跨距积。

如果所计算的跨距积是为该指令地址所计算的第一跨距积 (块 522), 那么过程 500 增量审查计数器 (块 514) 并且检查审查计数器是否超过审查阈值 (块 516)。如果没有超过审查阈值 (即, 仍然出现审查阶段 404), 那么过程 500 等待另一 D 高速缓存器未命中 (块 518)。

然而, 如果所计算的跨距积不是为该指令地址所计算的第一跨距积 (块 522), 那么过程 500 确定此指令的跨距值 (块 524)。特别地是, 通过确定与该指令地址相关联的当前跨距积和与相同指令地址相关联的前一跨距积的最大公约数来计算跨距值。作为选择, 可以用任意数目的跨距积来确定所述跨距值。例如, 可以把三个跨距积的最大公约数用作为跨距值。

随后, 过程 500 增量审查计数器 (块 514) 并且检查所述审查计数器是否超过审查阈值 (块 516)。如果没有超过审查阈值 (即, 仍然出现审查阶段 404), 那么过程 500 等待另一 D 高速缓存器未命中 (块 518)。然而, 如果已经超过审查阈值, 那么过程 500 重新初始化一定的变量 (块 502) 并且返回到跳跃阶段 402。

尽管以上公开的示例性系统在其它组件中特别包括在硬件上执行的软件, 然而应当注意, 这种系统仅仅是说明性的而不应当被认为是限制性的。例如, 预计这里所公开的任何或所有硬件和软件组件可以专门地用专用硬件、软件、固件或用硬件、固件和 / 或软件的某种组合来实现。

另外, 尽管这里已经描述了一定的方法、设备和制造产品, 然而本专利的覆盖范围并不局限于此。相反地, 本专利完全覆盖了落入所

附权利要求字面或等效原则范围内的所有设备、方法和制造产品。

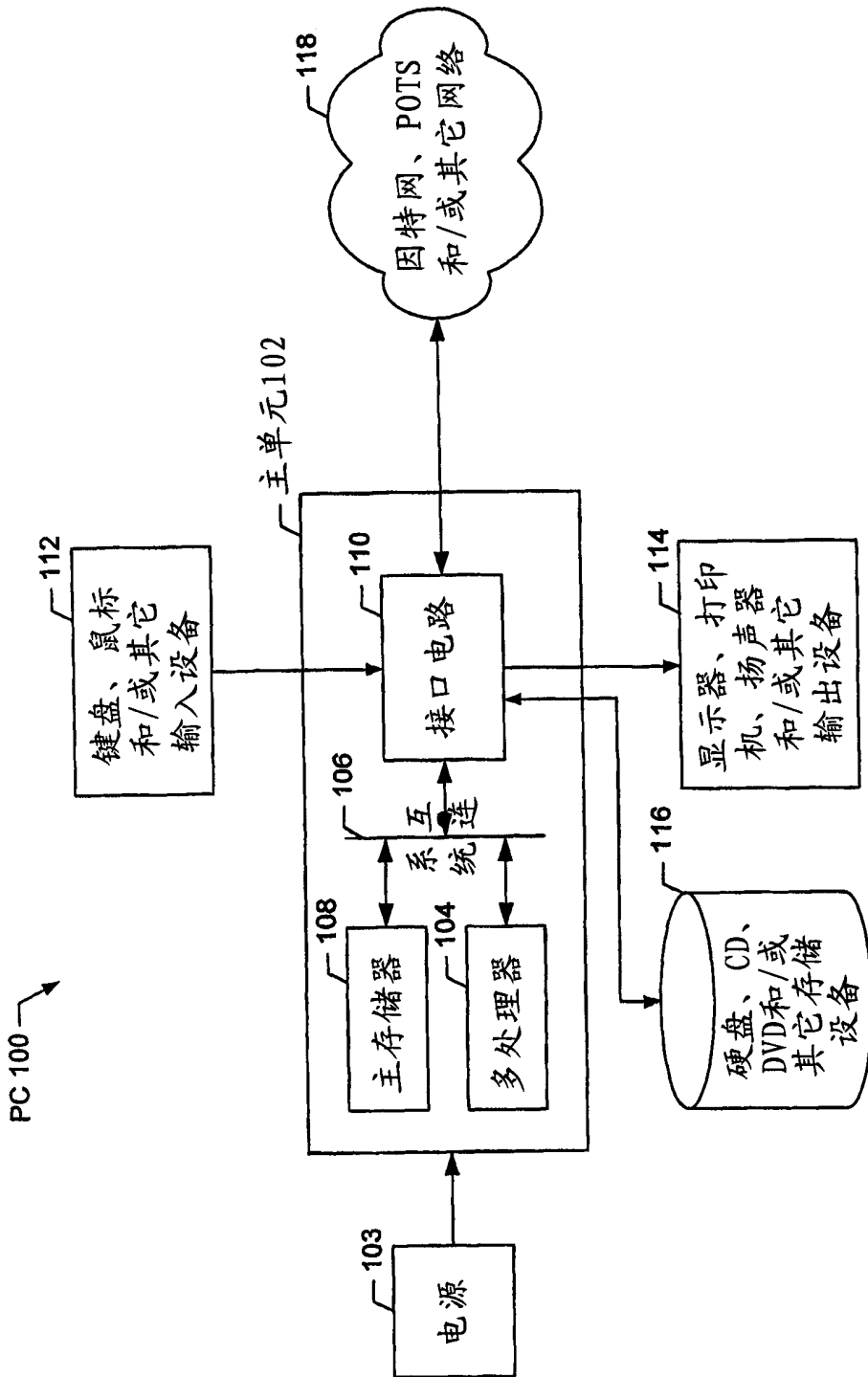


图 1

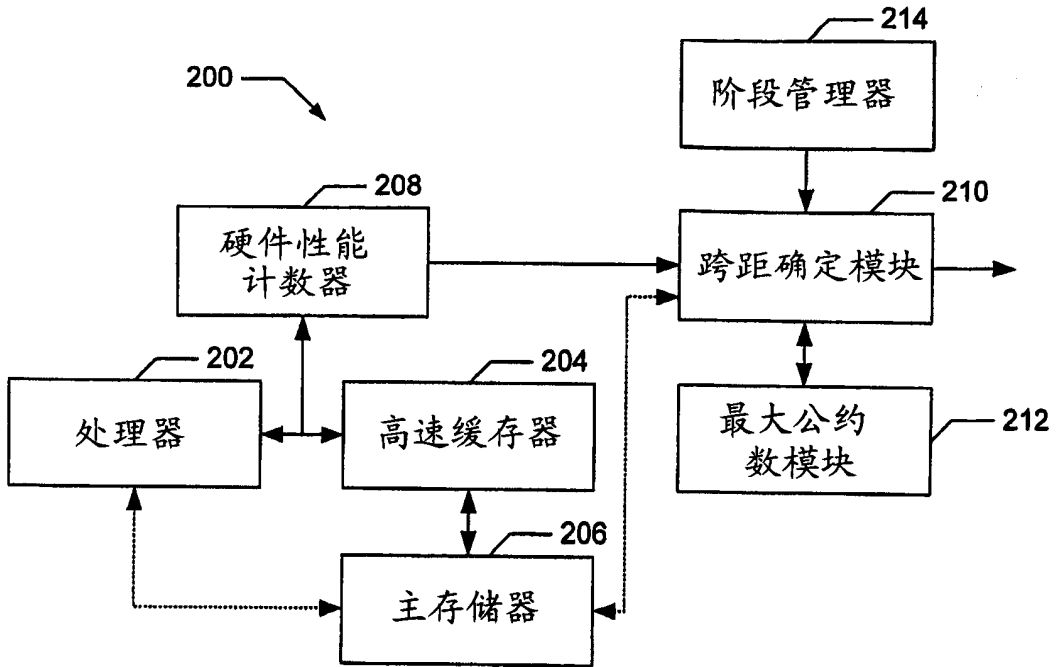


图 2

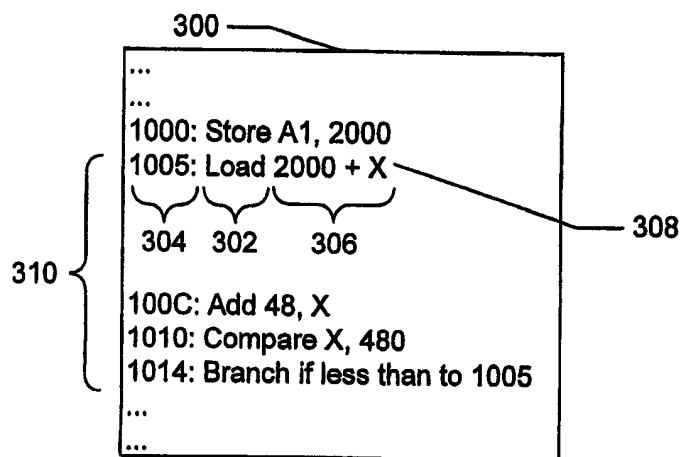


图 3

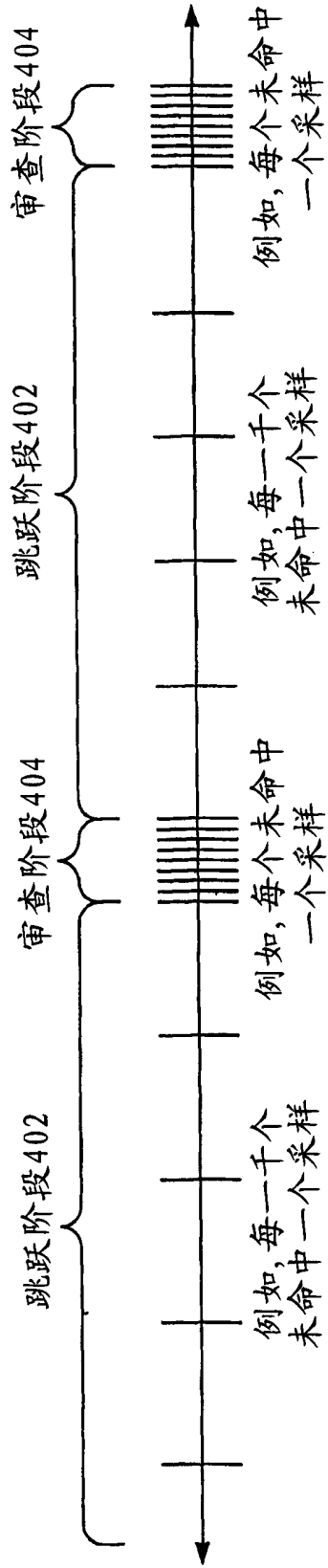


图 4

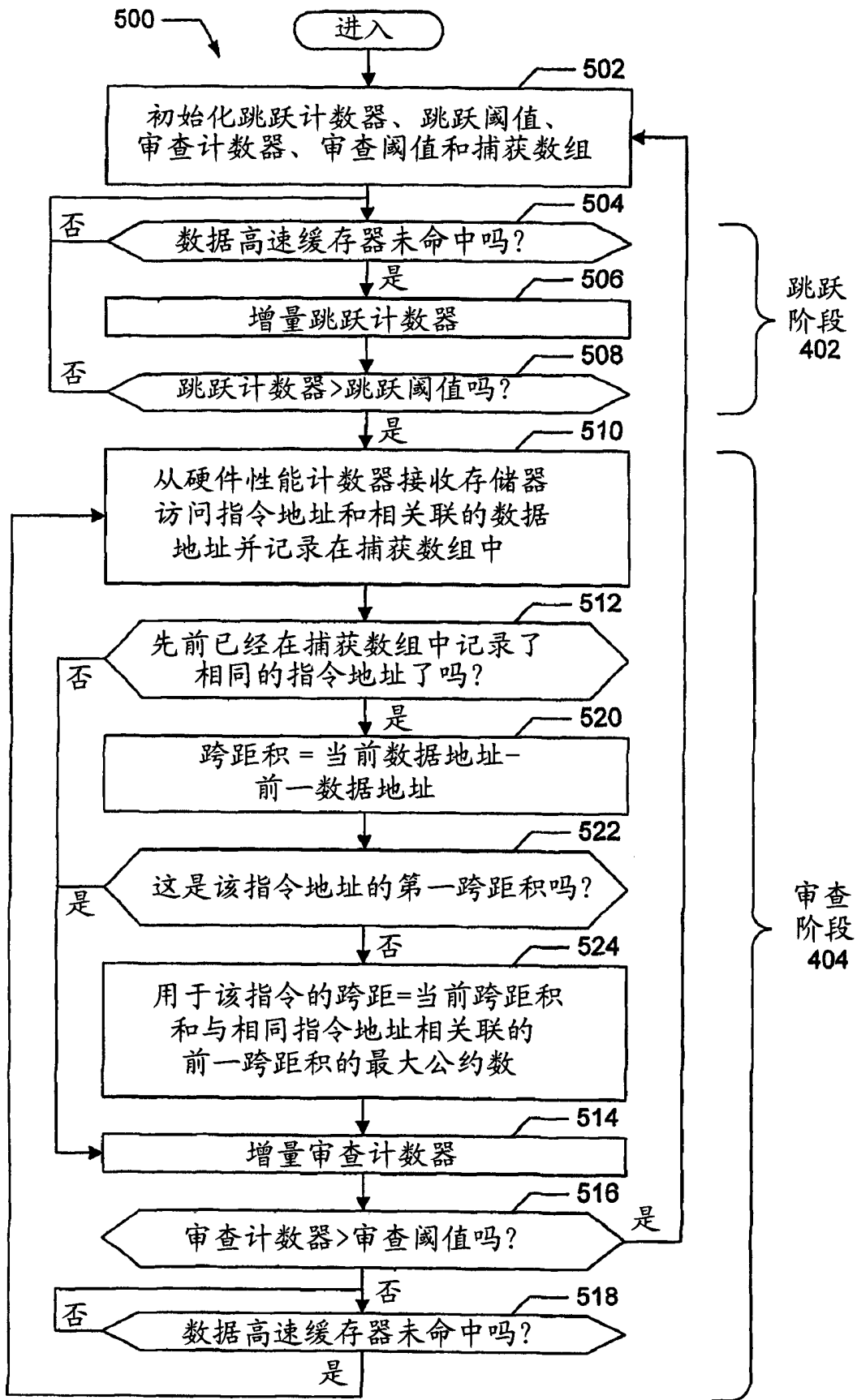


图 5