



(12)发明专利

(10)授权公告号 CN 104704474 B

(45)授权公告日 2017.08.15

(21)申请号 201380014658.1

(22)申请日 2013.02.28

(65)同一申请的已公布的文献号  
申请公布号 CN 104704474 A

(43)申请公布日 2015.06.10

(30)优先权数据  
13/422,538 2012.03.16 US

(85)PCT国际申请进入国家阶段日  
2014.09.16

(86)PCT国际申请的申请数据  
PCT/JP2013/001240 2013.02.28

(87)PCT国际申请的公布数据  
W02013/136701 EN 2013.09.19

(73)专利权人 国际商业机器公司  
地址 美国纽约阿芒克

(72)发明人 M.M.米特兰 岑中龙  
K.A.斯图德利

(74)专利代理机构 北京市柳沈律师事务所  
11105  
代理人 邸万奎

(51)Int.Cl.  
G06F 11/34(2006.01)

(56)对比文件  
US 2009210752 A1,2009.08.20,  
CN 101625657 A,2010.01.13,  
US 2007261032 A1,2007.11.08,  
CN 1867893 A,2006.11.22,

审查员 黄琳

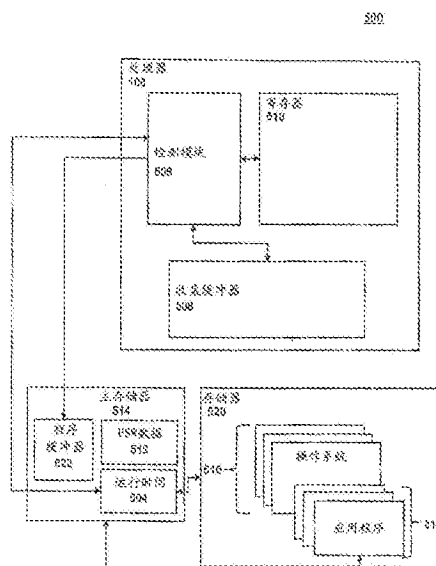
权利要求书2页 说明书25页 附图15页

(54)发明名称

用于管理的运行时间的基于硬件的运行时间检测设施

(57)摘要

本发明的实施例涉及执行运行时间检测。处理器基于在所述处理器上执行的应用程序的指令的指令流来捕获运行时间检测。所述捕获包括：在所述处理器的收集缓冲器中存储是运行时间检测数据。所述处理器检测运行时间检测样本点触发器。基于检测到所述运行时间检测样本点触发器来将所述收集缓冲器的内容作为报告群组复制到程序缓冲器内。所述程序缓冲器位于由所述应用程序可访问的地址空间中的主存储器中。



1. 一种计算机可读介质,其上存储有用于执行运行时间检测的方法的计算机程序,所述方法包括:

处理器基于在所述处理器上执行的应用程序的指令的指令流来捕获运行时间检测,所述捕获包括:在所述处理器的收集缓冲器中存储运行时间检测数据,其中,当所述应用程序执行在特权状态时,所述应用程序可访问所述收集缓冲器,当所述应用程序执行在较低特权状态时,所述应用程序不可访问所述收集缓冲器;

所述处理器检测运行时间检测样本点触发器;并且

基于检测到所述运行时间检测样本点触发器来将所述收集缓冲器的内容作为报告群组复制到程序缓冲器内,所述程序缓冲器位于由所述应用程序可访问的地址空间中的主存储器中。

2. 根据权利要求1所述的计算机可读介质,其中,通过位于所述处理器上的硬件来实现所述收集缓冲器。

3. 根据权利要求1所述的计算机可读介质,其中,所述方法进一步包括:在所述收集缓冲器中捕获与在所述指令流的所述执行期间检测到的事件对应的指令地址和元数据。

4. 根据权利要求1所述的计算机可读介质,其中,所述报告群组包括预定数量的一个或多个包括所述收集缓冲器的内容的检测记录和系统信息记录。

5. 根据权利要求1所述的计算机可读介质,其中,所述复制包括以所述程序缓冲器的当前地址开始将所述报告群组复制到所述程序缓冲器内,所述程序缓冲器被存储在由指令可访问的控制块指定的程序缓冲器原始地址处,所述指令可访问的控制块也指定在所述程序缓冲器中的最后字节的地址和在所述程序缓冲器中的所述当前地址。

6. 根据权利要求1所述的计算机可读介质,其中,所述运行时间检测样本点触发器包括下述部分的至少一个:执行样本点指令、执行阈值数量的指令和遇到自从最后样本点起的过去的时间。

7. 根据权利要求1所述的计算机可读介质,其中,所述运行时间检测样本带内触发器是硬件事件。

8. 根据权利要求1所述的计算机可读介质,其中,所述运行时间检测数据包括分支获取目标地址和发出指令寄存器值的至少一个。

9. 一种用于执行运行时间检测的计算机实现的方法,所述方法包括:

处理器基于在所述处理器上执行的应用程序的指令的指令流来捕获运行时间检测,所述捕获包括:在所述处理器的收集缓冲器中存储运行时间检测数据,其中,当所述应用程序执行在特权状态时,所述应用程序可访问所述收集缓冲器,当所述应用程序执行在较低特权状态时,所述应用程序不可访问所述收集缓冲器;

所述处理器检测运行时间检测样本点触发器;并且

基于检测到所述运行时间检测样本点触发器来将所述收集缓冲器的内容作为报告群组复制到程序缓冲器内,所述程序缓冲器位于由所述应用程序可访问的地址空间中的主存储器中。

10. 根据权利要求9所述的方法,其中,通过位于所述处理器上的硬件来实现所述收集缓冲器。

11. 根据权利要求9所述的方法,其中,进一步包括:在所述收集缓冲器中捕获与在所述

指令流的所述执行期间检测到的事件对应的指令地址和元数据。

12. 根据权利要求9所述的方法,其中,所述报告群组包括预定数量的一个或多个包括所述收集缓冲器的内容的检测记录和系统信息记录。

13. 根据权利要求9所述的方法,其中,所述复制包括以所述程序缓冲器的当前地址开始将所述报告群组复制到所述程序缓冲器内,所述程序缓冲器被存储在由指令可访问的控制块指定的程序缓冲器原始地址处,所述指令可访问的控制块也指定在所述程序缓冲器中的最后字节的地址和在所述程序缓冲器中的所述当前地址。

14. 一种用于执行运行时间检测的系统,所述系统包括:

处理器,所述系统被配置为执行一种方法,所述方法包括:

所述处理器基于在所述处理器上执行的应用程序的指令的指令流来捕获运行时间检测,所述捕获包括:在所述处理器的收集缓冲器中存储是运行时间检测数据,其中,当所述应用程序执行在特权状态时,所述应用程序可访问所述收集缓冲器,当所述应用程序执行在较低特权状态时,所述应用程序不可访问所述收集缓冲器;

所述处理器检测运行时间检测样本点触发器;并且

基于检测到所述运行时间检测样本点触发器来将所述收集缓冲器的内容作为报告群组复制到程序缓冲器内,所述程序缓冲器位于由所述应用程序可访问的地址空间中的主存储器中。

## 用于管理的运行时间的基于硬件的运行时间检测设施

### 技术领域

[0001] 本发明总体上涉及在计算环境内的处理,并且更具体地涉及用于管理的运行时间的基于硬件的运行时间检测设施。

### 背景技术

[0002] 计算机处理器使用日益复杂的分支预测及指令高速缓存逻辑来执行事务(transaction)。这些程序已被引入以增加指令吞吐量,且因此增加处理性能。用于改进性能的逻辑的引入使得难以肯定地预测特定软件应用程序将如何在计算机处理器上执行。在软件开发过程期间,常常存在功能性与性能之间的平衡。软件在基于执行软件的底层硬件的一个或多个抽象层级处执行。当将硬件虚拟化时,额外抽象层被引入。随着性能增强逻辑及各种抽象层(layer)的引入,将难以透彻地理解当程序正在执行时在硬件层级(level)实际发生的事。在不具此信息的情况下,软件开发者将更抽象的方法(诸如,执行持续时间、存储器使用率、线程的数目等)用于最佳化软件应用程序。

### 发明内容

[0003] 技术问题

[0004] 当硬件特定信息可用时,该信息通常是在事后被提供给开发者,且该信息是以聚集地、在高层级、和/或穿插着其他程序及操作系统的活动的方式提供,从而难以识别可能影响软件应用程序的效率及准确度的问题。

[0005] 对于问题的解决方案

[0006] 实施例包括用于执行运行时间检测的计算机程序、方法和系统。处理器基于在所述处理器上执行的应用程序的指令的指令流来捕获运行时间检测。所述捕获包括:在所述处理器的收集缓冲器中存储运行时间检测数据。所述处理器检测运行时间检测样本点触发器。基于检测到所述运行时间检测样本点触发器来将所述收集缓冲器的内容作为报告群组复制到程序缓冲器内。所述程序缓冲器位于由所述应用程序可访问的地址空间中的主存储器中。

[0007] 额外特征及优点是经由本发明的技术来实现。本发明的其他实施例及方面将在本文中予以详细描述且被视为所主张发明的一部分。为了更好地理解本发明的优点及特征,参看描述及附图。

### 附图说明

[0008] 在说明书结尾处的权利要求中特定指出且清楚地主张了被视为本发明的主题。将根据结合随附图进行的以下详细描述显而易见本发明的前述及其他特征及优点,其中:

[0009] 图1A为描绘实施例中的示例主机计算机系统的图;

[0010] 图1B为描绘实施例中的示例模拟主机计算机系统的图;

[0011] 图1C为描绘实施例中的示例计算机系统的图;

- [0012] 图2为描绘实施例中的示例计算机网络的图；
- [0013] 图3为描绘实施例中的计算机系统的元件的图；
- [0014] 图4A描绘实施例中的计算机系统的详细元件；
- [0015] 图4B描绘实施例中的计算机系统的详细元件；
- [0016] 图4C描绘实施例中的计算机系统的详细元件；
- [0017] 图5描绘根据实施例的用于处理器的运行时间检测的系统的示意图；
- [0018] 图6描绘实施例中的包括可由特权状态设置的控制的运行时间检测控制块(RICCB)的一部分。
- [0019] 图7描绘实施例中的当半特权位(K)经设置为1时的RICCB控制块的一部分；
- [0020] 图8描绘根据实施例的收集缓冲器；
- [0021] 图9描绘根据实施例的报告群组；
- [0022] 图10描绘根据实施例的用于实现运行时间检测设施的处理流；并且
- [0023] 图11图示了根据一个实施例的计算机程序产品。

### 具体实施方式

[0024] 本发明的实施例为用于受管理运行时间的基于硬件的运行时间检测设施。如本文中所使用，术语“受管理运行时间”指代封装状态且管理用以执行程序或应用程序(例如，Java®虚拟机或“JVM”、操作系统、中间件等)的资源的环境。该运行时间检测设施的实施例使程序能够收集关于程序执行的信息，包括中央处理单元(CPU)数据。所收集信息允许管理器运行时间环境获取关于程序的洞察(insights)，该信息是收集自该程序。该运行时间检测设施的实施例包括用于将事件(例如，采纳分支、寄存器值等)的序列收集于收集缓冲器中的硬件设施。该收集缓冲器(或含有最近记录的收集缓冲器的子集)是在发生可程序化集合的样本触发事件时在应用程序的地址空间(例如，JVM的地址空间)中复制至程序缓冲器中，这些事件诸如(但不限于)：以指令的形式的软件指示词(software directive)被插入至指令流中；所执行指令的间隔完成，从上一样本过期时起的给定经过时间，和/或诸如观察到数据或指令高速缓存未命中的给定硬件事件

[0025] 动态编译器利用运行时间信息(诸如，由本文中所描述的基于硬件的运行时间检测设施收集的运行时间信息)来执行在线反馈导向的最佳化(online feedback directed optimization)。举例而言，关于重要执行路径、设置档值及优选分支方向的信息可由动态编译器使用以执行特殊化版本码、引导内嵌(in-lining)、对执行路径重新排序及将分支拉直的最佳化。本文中所描述的实施例不限于由编译器使用，且可以诸如(但不限于)用于引导其他受管理运行时间设施的多种方式使用。举例而言，数据高速缓存事件可用以帮助废弃项目收集器程序作出关于协同定位多个物件或使多个物件彼此远离的决策以获得优选数据局部性或避免假性共用问题。

[0026] 用于将应用进行程序概要分析的典型的现代手段是编译的检测手段，其编译重要编译单元的检测版本，并且然后执行长得足以捕获代表性数据的这些编译单元。该数据然后用于驱动执行单元的优化重新编译。该软件方法的一些缺点包括：(1) 必须编译编译单元的检测版本的开销，(2) 必须在可能对于应用的性能重要的编译单元中执行昂贵的检测路径的开销，(3) 缺少对于诸如数据高速缓存未中和分支误预测的硬件事件的洞察，(4) 捕获

事件(或轨迹)的序列以努力提取抽象和相关上下文的显著复杂度,以及,(5)不能正确地处理其中检测应用频繁地采用意料之外的分支并且执行非检测代码的情况。

[0027] 值得注意的是,程序概要相关的反馈可以被应用到的编译单元的范围被固有地绑定到由项目(1)和(2)引入的成本。更具体地,当开销增加时,候选编译单元的范围减小,因为仅更占优势的编译单元可以表示用于有可能进行开销的最终摊销的足够的执行时间机会。

[0028] 现代的动态编译环境经常包括在动态编译之前的解释阶段。包括解释阶段的、对于将应用简档化的另一种典型的现代手段是解释检测手段,该解释检测手段允许解释器收集运行时间统计。除了上面列出的点(2)、(3)和(4)之外,该软件手段也具有增加的可能缺点:要求容易在应用使用期限中早期捕获该应用的行为,导致在应用启动时的显著的开销;要求扩展该应用的解释阶段以保证收集足够代表性的数据;可能未能捕获稳态行为;以及,不能检测到和反应于在解释阶段后出现的在该应用中的阶段改变。

[0029] 也使用利用解释和编译检测两者以及随机跟踪的现代混和软件解决方案,但是该解决方案仍然受到上面列出的缺陷的影响,并且可能向运行时间引入相当大的开销和复杂度。

[0030] 除了动态编译之外,管理的运行时间的其他分量当前不受到它们对于高速缓存分级或内核作出的确定的影响。例如,可以动态地重新配置在堆上的数据的布局的垃圾收集器例程可能丢失使得下述对象共置的机会,该对象暂时高度耦合或替代地可能通过共置在独立的线程上被频繁的更新的对象而引发假共享高速缓存效果。

[0031] 在此描述的用于管理的运行时间的基于硬件的运行时间检测设施的实施例相对于现代的基于软件的程序概要分析机制提供了几个优点。这些优点可以包括但是不限于:避免了用于获得编译单元的检测版本的重新编译的成本;数据收集的开销被嵌入在硬件中,并且因此消除了对于昂贵的基于软件的编译代码的检测的需要;提供了对于诸如分支误预测、数据和指令高速缓存未命中的重硬件事件的直接洞察;用于捕获顺序事件的自然跟踪机制导致数据的自然上下文抽象和相关(例如,执行路径和在其上在被操作的相应数据类型导致给定的数据高速缓存事件);并且,可以贯穿应用的执行的使用期限而重复地被透明地使能和禁止的设施,允许改善数据收集和重新编译的成本的限制和摊销的灵活性。

[0032] 由在此所述的实施例提供的编译和检测的降低的成本固有地扩展可以被作为目标用于在线程序概要相关的编译的编译单元的范围。另外,源自去除由在此所述的实施例提供的基于软件的检测的、降低的用于收集数据的开销允许收集更多的程序概要分析数据,因此向直接重新编译和其他形式的优化(例如,数据重新组织)提供了更精确的程序概要分析数据。

[0033] 除了用于所建议的用于动态编译的使用之外,由在此所述的实施例提供的硬件事件的可用性也可以用于引导其他被管理的运行时间设施。例如,可以使用数据高速缓存事件来帮助垃圾收集器进行关于将对象共置或置于远距离的确定,以获得更好的数据本地化或避免假共享问题。

[0034] 图1A描绘实施例中的主机计算机系统50的代表性组件。组件的其他配置也可在计算机系统使用。代表性主机计算机系统50包含与主存储器(计算机存储器)2通信的一个或多个处理器1以及至存储器件11及网络10的用于与其他计算机或SAN等通信的I/O接口。

处理器1符合具有架构化指令集及架构化功能性的架构。处理器1可具有用于将程序地址(虚拟地址)变换为存储器中的真实地址的动态地址转译(DAT)3。DAT 3通常包括用于高速缓存转译的转译后备缓冲器(TLB)7,使得对计算机存储器2的块的稍后存取不需要延迟地址转译。通常,高速缓冲存储器9是用于计算机存储器2与处理器1之间。高速缓冲存储器9可为阶层式的,其具有可供一个以上CPU使用的大的高速缓冲存储器及介于大的高速缓冲存储器与每一CPU之间的较小、较快速(较低层级)的高速缓冲存储器。在一些实施例中,较低层级高速缓冲存储器经划分以为指令提取及数据存取提供单独的低层级高速缓冲存储器。在实施例中,由指令提取单元4经由高速缓冲存储器9从计算机存储器2提取指令。该指令在指令解码单元6中被解码且(在一些实施例中与其他指令一起)被分派至指令执行单元8。通常,使用若干指令执行单元8,例如算术执行单元、浮点执行单元及分支指令执行单元。该指令是由指令执行单元8执行,从而按需要从指令指定的寄存器或计算机存储器2存取操作数。如果将从计算机存储器2存取(载入或存储)操作数,则载入存储单元5通常在被执行的指令的控制下处置该存取。指令可在硬件电路中或以内部微代码(固件)形式执行或通过两者的组合执行。

[0035] 在图1B中,提供模拟主机计算机系统21,其模拟具有主机架构的主机计算机系统,诸如图1的主机计算机系统50。在模拟主机计算机系统21中,主机处理器(CPU)1为模拟主机处理器(或虚拟主机处理器)29,且包含原生处理器27,该原生处理器具有不同于主机计算机系统50的处理器1的原生指令集架构的原生指令集架构。模拟主机计算机系统21具有原生处理器27可存取的存储器22。在实施例中,存储器22经分割成计算机存储器2部分及模拟例程存储器23部分。计算机存储器2根据主机计算机架构而可供模拟主机计算机系统21的程序使用。原生处理器27执行不同于模拟处理器29的架构的架构的架构指令集的原生指令(原生指令获得自模拟例程存储器23),且可通过使用在可解码所存取的该主机指令的序列及存取/解码例程中获得的一个或多个指令而从计算机存储器2中的程序存取用于执行的主机指令,以确定用于模拟所存取的该主机指令的功能的原生指令执行例程。针对主机计算机系统50架构定义的其他设施可由架构化设施例程模拟,其他设施包括诸如(例如)通用寄存器、控制寄存器、动态地址转译及输入/输出(I/O)子系统支持及处理器高速缓冲存储器的设施。模拟例程也可利用原生处理器27中可用的功能(诸如通用寄存器及虚拟地址的动态转译)来改进模拟例程的性能。也可提供特殊硬件及卸载引擎以帮助原生处理器27模拟主机计算机系统50的功能。

[0036] 在大型计算机中,程序设计师(通常,现今的“C”程序设计师)常常通过编译器应用程序来使用架构化机器指令。存储在存储介质中的这些指令可原生地在z/Architecture IBM服务器中执行,或替代地在执行其他架构的机器中执行。可在现有及未来的IBM大型计算机服务器中及在IBM的其他机器(例如,pSeries®服务器及xSeries®服务器)上模拟指令。可于在广泛多种机器(使用由IBM®、Intel®、AMD™、Sun Microsystems及其他公司制造的硬件)上执行Linux的机器中执行指令。除了在Z/Architecture®下在该硬件上执行之外,也可使用Linux以及由Hercules、UMX、Fundamental Software, Inc. (FSI)或Platform Solutions, Inc. (PSI)提供的使用模拟的机器,其中执行通常在模拟模式下。在模拟模式下,模拟软件是由原生处理器执行以模拟模拟处理器的架构。

[0037] 模拟主机计算机系统21的组件中的一个或多个将在“IBM®z/Architecture Principles of Operation” (公告第SA22-7832-08号,第9版,2010年8月)中予以进一步描述,该公告特此以全文引用方式并入本文中。IBM为国际商业机器公司(Armonk, New York, USA) 额注册商标。本文中所使用的其他名称可为国际商业机器公司或其他公司的注册商标、商标或产品名称。

[0038] 原生处理器27通常执行存储在包含固件或原生操作系统的模拟例程存储器23中的模拟软件以执行对模拟处理器的模拟。该模拟软件负责提取并执行模拟处理器架构的指令。该模拟软件维持模拟程序计数器以追踪指令边界。该模拟软件一次可提取一个或多个模拟机器指令且将该一个或多个模拟机器指令转换成供原生处理器27执行的原生机器指令的相应群组。这些经转换指令可被高速缓存,以使得可实现较快速转换。该模拟软件维持模拟处理器架构的架构规则以便保证针对模拟处理器撰写的操作系统及应用程序正确地操作。此外,该模拟软件提供由模拟处理器架构识别的资源,包括(但不限于)控制寄存器、通用寄存器、浮点寄存器、包括(例如)段表及页表的动态地址转译功能、中断机制、上下文切换机制、当日时间(TOD)时钟及至I/O子系统的架构化接口,以使得经设计以在模拟处理器29上执行的操作系统或应用程序可在具有该模拟软件的原生处理器27上执行。

[0039] 被模拟的特定指令经解码,且子例程被调用以执行个别指令的功能。模拟模拟处理器29的功能的模拟软件功能被以(例如)“C”子例程或驱动程序来实施,或在理解优选实施例的描述的后将在本领域普通技术人员的能力范围内的提供特定硬件的驱动程序的某一其他方法来实施。

[0040] 在实施例中,本发明可由软件(有时被称为经授权内部码、固件、微代码、毫码(milli-code)、微微代码(pico-code)等,其中的任一个都符合本发明)实践。参看图1A,体现本发明的软件程序代码是由也被称为主机计算机系统50的CPU(中央处理单元)1的处理器从存储器件11(诸如长期存储介质、CD-ROM驱动器、带驱动器或硬驱动器)存取。软件程序代码可体现于供数据处理系统使用的多种已知介质(诸如磁盘、硬驱动器或CD-ROM)中的任一个上。程序代码可分布在这些介质上,或可从一个计算机系统的计算机存储器2或存储器经由网络10至其他计算机系统地分发给用户以供这些其他系统的用户使用。

[0041] 替代地,程序代码可体现于计算机存储器2中,且由处理器1使用处理器总线(未图示)存取。此程序代码包括操作系统,该操作系统控制各种计算机组件和一个或多个应用程序的功能及交互。程序代码通常是从诸如存储器件11的致密介质分页至计算机存储器2,程序代码可在计算机存储器中供处理器1处理。用于将软件程序代码体现于存储器中、实体介质上和/或经由网络分发软件码的技术及方法是公知的且本文中不予以进一步论述。程序代码在被产生且存储在有形介质(包括(但不限于)电子存储器模块(RAM)、闪速存储器、致密盘(CD)、DVD、磁带等)上时常常被称为“计算机程序产品”。计算机程序产品介质通常可由优选在计算机系统处理电路读取以供处理电路执行。

[0042] 图1C说明本发明可于其中实践的代表性工作站或服务器硬件系统。图1C的系统100包含诸如个人计算机、工作站或服务器的代表性基本计算机系统101,其包括可选的外围设备。基本计算机系统101包括一个或多个处理器106及总线(未图示),该总线用以连接一个或多个处理器106与基本计算机系统101的其他组件及根据现有技术实现一个或多个处理器106与基本计算机系统101之间的通信。该总线将处理器106连接至存储器105及长期



存储器107,长期存储器可包括(例如)硬驱动器(包括(例如)磁性介质、CD、DVD及闪速存储器中的任一个)或带驱动器。基本计算机系统101也可包括用户接口适配器,其经由该总线将一个或多个处理器106连接至一个或多个接口设备,诸如键盘104、鼠标103、打印机/扫描仪110和/或其他接口设备,其他接口设备可为任何用户接口设备(诸如触敏屏幕、数字化输入板等)。该总线也经由显示适配器将一个或多个处理器连接至显示器件102,诸如LCD屏幕或监视器。

[0043] 基本计算机系统101可通过能够与网络109通信108的网络适配器而与其他计算机或计算机网络通信。示例网络适配器为通信频道、令牌环(token ring)、以太网或调制解调器。替代地,基本计算机系统101可使用无线接口(诸如蜂窝数字分组数据(CDPD)卡)进行通信。基本计算机系统101可与局域网(LAN)或广域网(WAN)中的这些其他计算机相关联,或基本计算机系统101可为具有另一计算机的客户端/服务器配置中的客户端等。

[0044] 图2说明本发明可于其中实践的数据处理网络200。数据处理网络200可包括诸如无线网络及有线网络的多个个别网络,该多个个别网络中的每一个可包括多个个别工作站201、202、203、204和/或图1C的基本计算机系统101。另外,如本领域普通技术人员将了解,可包括一个或多个LAN,其中LAN可包含耦接至主机处理器的多个智能工作站。

[0045] 程序设计码111可体现于存储器105中,且由处理器106使用处理器总线存取。此程序设计码包括操作系统,其控制各种计算机组件和一个或多个应用程序112的功能及交互。程序代码通常是从长期存储器107分页至高速存储器105,程序代码可在高速存储器中供处理器106进行处理。用于将软件程序设计码体现于存储器中、实体介质上和/或经由网络分发软件码的技术及方法是公知的且本文中不予以进一步论述。程序代码在被产生且存储在有形介质(包括(但不限于)电子存储器模块(RAM)、闪速存储器、致密盘(CD)、DVD、磁带等)上时常常被称为“计算机程序产品”。计算机程序产品介质通常可由优选在计算机系统处理电路读取以供处理电路执行。

[0046] 处理器最容易利用的高速缓冲存储器(通常比处理器的其他高速缓冲存储器快速且小)为最低(L1或层级一)高速缓冲存储器,且主存储器(主存储器)为最高层级高速缓冲存储器(L3,如果存在3个层级)。最低层级高速缓冲存储器常常被划分为保存待执行的机器指令的指令高速缓冲存储器(I高速缓冲存储器)及保存数据操作数的数据高速缓冲存储器(D高速缓冲存储器)。

[0047] 仍参看图2,网络也可包括大型计算机或服务器,诸如可存取数据存储库也可从工作站205直接存取的网关计算机(客户端服务器)206或应用程序服务器(远程服务器)208。网关计算机206充当至每一网络207的进入点。网关在将一个网络连接协议连接至另一者时是必需的。网关计算机206可优选地借助于通信链路耦接至另一网络(例如,因特网207)。网关计算机206也可使用通信链路直接耦接至一个或多个工作站101、201、202、203及204。网关计算机可利用可从国际商业机器公司购得的IBM eServer™ zSeries® z9®服务器来实施。

[0048] 在实施例中,体现本发明的软件程序设计码是由基本计算机系统101的处理器106从长期存储介质(诸如图1C的长期存储器107)存取。软件程序设计码可体现于供数据处理系统使用的多种已知介质(诸如磁盘、硬驱动器或CD-ROM)中的任一个上。该码可分布在这些介质上,或可从一个计算机系统的存储器或存储器经由网络至其他计算机系统地分发给

用户210及211以供这些其他系统的用户使用。

[0049] 参看图3,描绘处理器106的示例性处理器实施例。将一个或多个层级的高速缓冲存储器303用以缓冲存储器块以便改进处理器106的性能。高速缓冲存储器303为保存可能被使用的存储器数据的高速缓存行的高速缓冲器。典型高速缓存行为64、128或256个字节的存储器数据。在实施例中,将单独的高速缓存用于高速缓存指令而非用于高速缓存数据。常常由本领域公知的各种“窥探”演算法来提供高速缓存一致性(存储器及高速缓冲存储器中的行的复本的同步)。诸如处理器系统的存储器105的主存储器常常被称为高速缓冲存储器。在具有4个层级的高速缓冲存储器303的处理器系统中,存储器105有时被称为层级5(L5)高速缓冲存储器,因为存储器105通常较快且仅保存可供计算机系统使用的非易失性存储器(DASD、磁带等)的一部分。存储器105“高速缓存”由操作系统分页到存储器105内及外的数据页。

[0050] 程序计数器(指令计数器)311追踪待执行的当前指令的地址。z/Architecture处理器中的程序计数器是64位的且可截断至31或24位以支持先前寻址极限。程序计数器通常体现于计算机的程序状态字(PSW)中,以使得程序计数器在上下文切换期间保持不变。因此,进行中的程序(具有程序计数器值)可被(例如)操作系统中断(即,当前上下文从程序环境切换至操作系统环境)。程序的PSW在程序不在作用中时维持程序计数器值,且在操作系统在执行中使用操作系统的程序计数器(在PSW中)。在实施例中,程序计数器被递增等于当前指令的字节数目的量。精简指令集计算(RISC)指令通常具有固定长度,而复杂指令集计算(CISC)指令通常具有可变长度。IBM z/Architecture的指令为具有2、4或6个字节的长度的CISC指令。程序计数器311由(例如)上下文切换操作或分支指令的分支采纳操作来修改。在上下文切换操作中,当前程序计数器值是与关于执行中的程序的其他状态信息(诸如条件码)一起存储在PSW中,且载入指向待执行的新程序模块的指令的新程序代码计数器值。执行分支采纳操作以便通过将分支指令的结果载入至程序计数器311中而准许程序做出决策或在程序内循环。

[0051] 在实施例中,指令提取单元305是用以代表处理器106提取指令。指令提取单元305提取“下一顺序指令”、分支采纳指令的目标指令或在上下文切换后程序的第一指令。在实施例中,指令提取单元305使用预提取技术而基于经预提取的指令可能被使用的可能性来推测性地预提取指令。举例而言,指令提取单元305可提取包括下一顺序指令的16个字节的指令及额外字节的其他顺序指令。

[0052] 所提取指令接着由处理器106执行。在实施例中,该所提取指令被传递至指令提取单元305的解码/分派单元306。解码/分派单元306解码该指令且将关于该经解码指令的信息转递至适当执行单元307、308和/或310。执行单元307从指令提取单元305接收关于经解码算术指令的信息且将根据指令的操作码(运算码)对操作数执行算术运算。操作数是从存储器105、架构化寄存器309或从执行中的指令的立即字段提供至执行单元307。当存储时,执行的结果是存储在存储器105、架构化寄存器309或存储在其他机器硬件(诸如控制寄存器、PSW寄存器等)中。

[0053] 处理器106通常具有用于执行指令的功能的一个或多个执行单元307、308及310。参看图4A,执行单元307可通过接口逻辑407而与架构化寄存器309、解码/分派单元306、载入/存储单元310及其他处理器单元401通信。执行单元307可使用若干寄存器电路403、404

及405来保存信息,算术逻辑单元 (ALU) 402将对该信息进行操作。ALU 402执行诸如加减乘除的算术运算,以及诸如及 (and)、或 (or)、以及互斥或 (xor)、旋转及移位的逻辑功能。在实施例中,ALU支持与设计依赖的特殊化运算。其他电路可提供包括(例如)条件码及复原支持逻辑的其他架构化设施408。通常,ALU运算的结果被保存在输出寄存器电路406中,该输出寄存器电路可将该结果转递至多种其他处理函数。在其他实施例中,存在处理器单元的许多配置,本描述仅意欲提供对一个实施例的代表性理解。

[0054] ADD指令(例如)将在具有算术及逻辑功能性的执行单元307中执行,而浮点指令(例如)将在具有特殊化浮点能力的浮点执行单元(未图示)中执行。优选地,执行单元通过对由指令识别的操作数执行运算码定义的函数而对操作数进行运算。举例而言,可由执行单元307对在由指令的寄存器字段识别的两个架构化寄存器309中发现的操作数执行ADD指令。

[0055] 执行单元307对两个操作数执行算术加法且将结果存储在第三操作数中,其中该第三操作数可为第三寄存器或该两个源寄存器中的一个。执行单元307优选利用算术逻辑单元 (ALU) 402,其能够执行诸如移位、旋转、与、或及XOR的多种逻辑函数,以及包括加减乘除中的任一个的多种代数函数。一些ALU 402是为了纯量运算而设计且一些ALU是为了浮点运算而设计。在实施例中,数据可视架构而为大端序 (big endian) (其中最低有效字节在最高字节地址处)或小端序 (little endian) (其中最低有效字节在最低字节地址处)。IBM z/Architecture为大端序。带正负号字段可具正负号及量值,其视架构而定为二进位反码或二进位补码。二进位补码是有利的,因为ALU不必设计减法能力,因为呈二进位补码的负值或正值在ALU内仅需要加法。数字通常以速记 (shorthand) 描述,其中例如,12位的字段定义4,096字节块的地址且通常被描述为4Kbyte (千字节) 块。

[0056] 参看图4B,用于执行分支指令的分支指令信息通常被发送至分支单元308,其在其他条件运算完成之前使用诸如分支历史表432的分支预测演算法来预测分支的结果。将在条件运算完成之前提取并推测地执行当前分支指令的目标。当条件运算完成时,将基于条件运算的条件及推测出的结果而完成或舍弃推测地执行的分支指令。典型分支指令可测试条件码且在条件码满足该分支指令的分支要求的情况下分支至目标地址,可(例如)基于包括在寄存器字段或指令的立即字段中发现的数字的若干数字来计算目标地址。在实施例中,分支单元308可使用具有多个输入寄存器电路427、428及429及输出寄存器电路430的ALU 426。分支单元308可与(例如)通用寄存器、解码/分派单元306或其他电路425通信。

[0057] 可出于包括(例如)以下各者的多种原因而中断一组指令的执行:由操作系统起始的上下文切换,导致上下文切换的程序例外或错误,导致上下文切换的I/O中断信号,或多个程序的多线程活动(在多线程环境中)。在实施例中,上下文切换动作存储关于当前在执行中的程序的状态信息,且接着载入关于正被调用的另一程序的状态信息。状态信息可存储在(例如)硬件寄存器中或存储器中。状态信息包括指向待执行的下一指令的程序计数器值、条件码、存储器转译信息及架构化寄存器内容。上下文切换活动可由硬件电路、应用程序、操作系统程序或固件码(微代码、微微代码或经授权内部码(LIC))单独地或组合地实行。

[0058] 处理器根据指令定义的方法来存取操作数。指令可使用指令的一部分的值提供立即操作数,可提供明确地指向通用寄存器或专用寄存器(例如浮点寄存器)的一个或多个寄

寄存器字段。指令可利用由运算码字段识别为操作数的多个隐含寄存器。指令可利用操作数的存储器位置。操作数的存储器位置可由寄存器、立即字段或如由z/Architecture长位移设施(long displacement facility)举例说明的寄存器与立即字段的组合提供,在该z/Architecture长位移设施中指令定义基寄存器、索引寄存器及立即字段(位移字段),其被加在一起以提供操作数在存储器中的位置。本文中的位置暗示在主存储器(主存储器)中的位置,除非另有指示。

[0059] 参看图4C,处理器使用载入/存储单元310来存取存储器。载入/存储单元310可通过经由高速缓冲存储器/存储器接口获得目标操作数在存储器中的地址及将该操作数载入架构化寄存器309或另一存储器位置中来执行载入操作,或可通过获得目标操作数在存储器中的地址及将从架构化寄存器309或另一存储器位置获得的数据存储在存储器中的目标操作数字置中来执行存储操作。载入/存储单元310可为推测性的且可以相对于指令序列而言无序的序列存取存储器;然而,载入/存储单元310对程序维持指令是有序地执行的表像。载入/存储单元310可与架构化寄存器309、解码/分派单元306、高速缓冲存储器/存储器接口或其他元件455通信,且包含各种寄存器电路、ALU 458及控制逻辑463以计算存储地址且提供管线排序以保持运算有序。一些运算可能无序,但载入/存储单元提供如本领域公知的用以使无序的运算对程序表现为已有序地执行的功能性。

[0060] 优选地,应用程序“看见”的地址常常被称为虚拟地址。虚拟地址有时被称为“逻辑地址”及“有效地址”。这些虚拟地址是虚拟的,因为虚拟地址是通过多种DAT技术中的一个(诸如图3的DAT 312)重定向至实体存储器位置,DAT技术包括(但不限于)将偏移值(offset value)作为虚拟地址的首码、经由一个或多个转译表转译虚拟地址,转译表至少单独地或组合地包括段表及页表,优选地,段表具有指向页表的输入项。在z/Architecture中,提供转译的阶层,其包括区域第一表、区域第二表、区域第三表、段表及选用的页表。常常通过利用转译后备缓冲器(TLB)来改进地址转译的性能,该转译后备缓冲器包含将虚拟地址映射至相关联实体存储器位置的输入项。输入项是在DAT 312使用转译表转译虚拟地址时创建。于是,虚拟地址的后续使用可利用快速TLB的输入项而非缓慢的顺序转译表存取。可通过包括最近最少使用(LRU)演算法的多种替换演算法管理TLB内容。

[0061] 在处理器106为多处理器系统的处理器的情况下,每一处理器有责任使经共用资源(诸如I/O、高速缓冲存储器、TLB及存储器)保持连锁以达成一致性。在实施例中,“窥探”技术将被用于维持高速缓存一致性。在窥探环境中,每一高速缓存行可被标记为处于共用状态、排他状态、有变化状态、无效状态及类似状态中的任一个中以便促进共用。

[0062] 图3的I/O单元304为处理器106提供了用于附接至外围设备(包括例如磁带、磁盘、打印机、显示器及网络)的构件。I/O单元304常常由软件驱动程序提供给计算机程序。在诸如来自IBM的z/Series的大型计算机中,频道适配器及开放式系统适配器为大型计算机的I/O单元,I/O单元提供操作系统与外围设备之间的通信。

[0063] 检测数据为与处理器106的操作有关的数据。在实施例中,对检测数据及其他系统层级量度的存取可能受限制或不可用。计算机处理器在特权状态(或监督状态)及较低特权状态(或问题状态)下操作。在特权状态中,程序可经由特权操作对所有系统资源进行存取(例如,对所有控制寄存器及监督存储器空间进行存取)。特权状态也被称为特权模式或监督模式。在计算机处理器上执行的操作系统可在特权状态中操作。较低特权状态为对系统

资源的存取受限制的非特权状态。举例而言,在较低特权状态中执行的应用程序可对控制寄存器具有有限存取权或无存取权,且仅可存取由操作系统指派给应用程序的用户存储器空间。较低特权状态通常被指派给在操作系统的控制下执行的应用程序,且不可在较低特权状态中执行特权操作。较低特权状态也被称为问题状态、问题模式或用户模式。

[0064] 对于在较低特权状态中执行的程序而言不可进行写入存取的一个这种受限资源为程序状态字(PSW)。PSW可包含待执行的下一指令的程序计数器、可由分支指令使用的条件码字段、用于指示检测被启用或停用的检测控制字段,及用以控制指令排序且确定计算机处理器的状态(包括指派给程序的特权状态)的其他信息。在多线程处理环境中,多个程序共用可用计算机处理器能力或将可用计算机处理器能力按时间截块。程序中的每一个具有上下文信息(context information),其包括相关联PSW、用于存取指派给程序的主存储器的地址转译表的原始地址、一组通用寄存器当前值、控制寄存器、浮点寄存器等。目前在作用中或控制的PSW被称为当前PSW。PSW管理目前在执行中的程序。计算机处理器具有中断能力,其准许计算机处理器回应于例外条件及外部刺激而迅速地上下文切换至另一程序。当中断出现时,计算机处理器针对特定中断类别而将当前PSW置于经指派存储位置(称为旧PSW位置)中。计算机处理器从第二经指派存储位置提取新PSW。此新上下文确定了待执行的下一程序。在实施例中,这些存储位置位于计算机处理器可存取的存储器位置中。当计算机处理器已完成处理该中断时,处置该中断的程序可重新载入包括旧PSW的旧上下文,使旧PSW又变为当前PSW,使得被中断程序可继续。

[0065] 可以明确方式(例如,当指令执行读取PSW位的部分时)或隐含方式(例如,在指令提取、操作数提取、地址产生计算、地址产生源等)参考PSW的字段。明确参考通常在运行时间执行,而隐含参考通常在指令执行期间在管线的不同阶段(即,指令提取、指令解码、运行时间及完成时间)执行。可独立于彼此来参考或更新PSW中的个别字段。

[0066] 在实施例中,通过操纵上下文,操作系统控制计算机处理资源,包括由计算机处理器启用运行时间检测。可在操作系统的执行期间以及由操作系统所执行的任何软件应用程序启用或停用运行时间检测。运行时间检测的启用/停用状态经存储为与程序相关联的PSW中的上下文信息。

[0067] 运行时间检测(RI)设施可并入实施z/Architecture的模型上。当RI设施经安装且被启用时,在程序执行期间将数据收集至CPU内的一个或多个收集缓冲器中且接着将其报告至程序缓冲器。所存储信息的每一单元被称为报告群组。报告群组的内容由多个记录组成,记录的内容表示由CPU在程序执行期间识别的事件。

[0068] 当以一配置安装运行时间检测设施时,PSW字段(RI位)启用运行时间检测。该运行时间检测控制的有效性确定了开启RI位的能力,但当RI为1时,CPU控制有效且运行时间检测经启用。运行时间检测设施可包括以下指令:载入运行时间检测控制、修改运行时间检测控制、运行时间检测发出、运行时间检测下一个、运行时间检测关闭、运行时间检测开启、存储运行时间检测控制,及测试运行时间检测控制。

[0069] 载入运行时间检测控制(LRIC)指令初始化管理运行时间检测的运行时间检测控制。修改运行时间检测控制(MRIC)指令修改最初由LRIC创建的运行时间检测控制的全部或子集。运行时间检测发出(RIEMIT)指令通过将通用寄存器的值存储在收集缓冲器中来收集该值。运行时间检测下一个(RINEXT)指令执行对在RINEXT之后的下一个顺序指令(NSI)的

导向取样。运行时间检测关闭 (RIOFF) 指令停用运行时间检测。运行时间检测开启 (RION) 指令启用运行时间检测。存储运行时间检测控制 (STRIC) 指令将该运行时间检测控制的当前值置于指定的存储位置中。测试运行时间检测控制 (TRIC) 指令检查该运行时间检测控制。在有效的情况下, 设置控制经更改指示符的状态。

[0070] 运行时间检测设施包括用于使测量警告外部中断搁置 (pending) 的能力。由运行时间检测收集且报告至程序缓冲器的信息的一些是依赖模型的, 且因此未被定义。由运行时间检测设施提供的样本及数据是用于对性能特性的统计估计, 且是实质上准确的, 且可能不可重复。举例而言, 不管取样模式如何, 都不可预测: 导致例外或与特定系统内部活动相关联的样本指令是否会导致报告群组的存储, 以及如果报告群组经存储, 包括在运行时间检测数据中的依赖模型数据是否会受影响。

[0071] 收集缓冲器是用以捕获记录的集合, 记录的内容报告在程序执行期间由处理器识别的事件。示例为: 一个或多个采纳分支的执行; 事务执行中止事件; 指令提取高速缓存未命中; 数据提取或存储高速缓存未命中; 以及RIEMIT指令的操作数。RIEMIT指令的执行通过将通用寄存器的值存储至收集缓冲器中来收集该值。可在诸如指令数据缓冲器的其他缓冲器中收集和/或存储额外数据。

[0072] 报告受报告控制控制。当样本指令经识别时, 每一报告控制启用对相应条件的检查。如果相应条件存在, 则形成且存储报告群组。当无报告控制经启用或经启用报告控制的相应条件不存在时, 不存储报告群组。可从指令数据缓冲器及其他依赖模型源获取关于样本指令所报告的数据, 且接着将其用以创建报告群组的一个或多个记录的内容, 一个这种记录为指令记录。

[0073] 可在报告群组存储器中捕获的记录类型包括: 填充、额外、开始、时间戳、指令、发出、事务执行 (TX) 中止、调用、返回和传送。当收集缓冲器中的有效记录的数目不足以填满当前报告群组大小的报告群组时, 在报告群组中使用填充记录 (filler record)。可在报告群组的额外区段中使用额外记录。开始记录为第一报告群组的第一记录。时间戳记录经存储为除第一报告群组外的每一个报告群组的记录0。当针对样本指令存储报告群组时创建指令记录作为该报告群组的最后记录。发出记录是通过RIEMIT的成功执行而创建。事务执行 (TX) 模式中止记录是通过隐含中止创建或由事务中止指令的执行而创建。调用记录是通过经分类为调用类型分支指令的分支指令的执行而创建。返回记录是通过经分类为返回指令的返回类型分支指令的执行而创建。传送记录是通过满足特定条件码准则的分支指令的执行而创建。

[0074] 图5描绘可实施于实施例中的用于处理器的运行时间检测的系统的示意图。在实施例中, 系统500包括中央处理单元 (CPU), 诸如图1的处理器106。在实施例中, 处理器106为单一处理器。在替代实施例中, 处理器106为多核心处理器的单一处理核心。在实施例中, 处理器106能够以变化的速度操作。

[0075] 在实施例中, 处理器106进一步包括寄存器510。寄存器510为能够存储数据的字以供处理器106使用的硬件寄存器。寄存器510包括用于存储可由处理器106存取的数据的位的一个或多个锁存器。寄存器510可包括 (例如) 多个通用寄存器和多个控制寄存器。处理器106另外包括与寄存器510通信的检测模块506。检测模块506为控制处理器106的检测的处理电路。检测模块506配置以直接从处理器106收集检测数据, 诸如一个或多个采纳分支的

执行路径、事务执行中止事件、各种运行时间操作数、时间戳信息等。检测模块506从处理器106收集检测数据,且将检测数据存储于收集缓冲器508中。在实施例中,收集缓冲器508为收集从检测模块506接收的数据的循环缓冲器,且当该循环缓冲器被填满时,该循环缓冲器用新数据覆盖最旧数据。

[0076] 处理器106执行一个或多个操作系统516和一个或多个应用程序518。一个或多个操作系统516和一个或多个应用程序518存储在存储器520(诸如硬驱动器、CD-ROM、闪存存储器等)中,且被载入至主存储器514中的为存储目前执行的操作系统和/或应用程序的一个或多个作用中片段、被调用页(其是按需要从存储器520载入至运行时间存储器504中)而保留的运行时间存储器504区域中。在实施例中,操作系统中的每一个作为由超管理器(未图示)管理且由处理器106执行的虚拟机来执行。

[0077] 在实施例中,处理器106将来自主存储器514中的用于目前执行的操作系统或应用程序的PSW数据512的PSW 512载入寄存器510中且设置(例如)寄存器510中的一个或多个处理器设置。在实施例中,寄存器510中的PSW包括用于启用及控制检测模块506的一个或多个位。

[0078] 一个或多个应用程序518包括经编译以在特定操作系统上执行的软件应用程序、在解译器上执行的经解译码(例如,Java)或操作系统支持线程(例如,进程管理、守护进程等)。一个或多个操作系统516和/或一个或多个应用程序518中的每一个可执行指令以触发检测模块506以使其开始或停止收集检测数据。

[0079] 在实施例中,一个或多个应用程序518中的一个执行指令,该指令已被确定为样本指令,从而在该样本指令的执行完成时创建样本点,且接着使检测模块506将应用程序的所收集数据从收集缓冲器508移动至主存储器514中的可由应用程序存取的程序缓冲器522。主存储器514可为本领域已知的任何可寻址存储器。在实施例中,主存储器514可包括有时被称为高速缓冲存储器的快速存取缓冲存储器。每一CPU可具有相关联高速缓冲存储器。在额外实施例中,主存储器514为动态随机存取存储器(DRAM)。在又一实施例中,主存储器为存储器件,诸如可由应用程序存取的计算机硬驱动器或闪存存储器。

[0080] 为了配置运行时间检测控制,处理器106支持载入运行时间检测控制(LRIC)指令。除本文予以进一步描述的特定LRIC字段外,将理解,也可定义额外字段以支持其他功能性。LRIC指令可用以载入运行时间检测且最初配置运行时间检测,且是由图5的检测模块506支持。在实施例中,检测模块506(也被称为运行时间检测模块506)实施运行时间检测控制及报告控制。运行时间检测控制的当前状态可使用存储运行时间控制(STRIC)指令从图5的寄存器510存储至主存储器514中。可作为LRIC指令的操作数载入的控制块的各种字段的定义也在本文中用以指代运行时间检测控制的相应值的状态。

[0081] 图6描绘实施例中的包括可由特权状态设置的控制的运行时间检测控制控制块(RICCB)的一部分。控制块部分600可包括除了参看图6所描述的那些值之外的额外值。可由LRIC指令来执行对控制块部分600的修改。

[0082] 该控制块部分包括有效性位602(V位)。有效性位602指示处理器中的运行时间检测控制的集合的有效性,因为控制是由LRIC指令先前设置。

[0083] 该控制块也包括S位604,其用以确定是否允许较低特权状态程序执行MRIC指令。K位606指示较低特权状态程序是否被准许在半特权状态中关于运行时间检测控制(诸如运

行时间检测控制的原始地址及极限地址)进行执行。H位608确定地址控制(即,原始地址、极限地址及当前地址)是否参考主要虚拟地址空间(primary virtual address space)或主虚拟地址空间(home virtual address space)。0位610被忽略且当作0对待。

[0084] 较低特权状态样本报告控制位612(Ps位)是与较低特权状态程序结合使用。当在较低特权状态中且运行时间检测控制中的Ps位612为零时,在运行时间检测经启用时忽略运行时间检测控制的报告控制,且因此不导致存储报告群组。当在较低特权状态中且运行时间检测控制中的Ps位612为1时,检查报告控制且根据其经定义功能来加以使用。

[0085] 监督状态样本报告控制位614(Qs位)是与监督状态程序结合使用。当在监督状态中且运行时间检测控制中的Qs位614为零时,在运行时间检测经启用时忽略运行时间检测控制的报告控制,且因此不导致存储报告群组。当在监督状态中且运行时间检测控制中的Qs位614为1时,检查报告控制且根据其经定义功能来加以使用。

[0086] 较低特权状态收集缓冲器控制位616(Pc位)控制着对图5的收集缓冲器508的更新。当在较低特权状态中且运行时间检测控制中的Pc位616为零时,在运行时间检测经启用时忽略运行时间检测控制的收集缓冲器控制,且收集缓冲器508的更新被阻止。当在较低特权状态中且运行时间检测控制中的Pc位616为1时,检查收集缓冲器控制且根据其经定义功能来加以使用。

[0087] 监督状态收集缓冲器控制位618(Qc位)控制着对收集缓冲器508的更新。当在监督状态中且运行时间检测控制中的Qc位618为零时,在运行时间检测经启用时忽略运行时间检测控制的收集缓冲器控制,且收集缓冲器508的更新被阻止。当在监督状态中且运行时间检测控制中的Qc位618为1时,检查经指示收集缓冲器控制且根据其经定义功能来加以使用。

[0088] G位620为运行时间检测暂停中断(也被称为暂停中断)的搁置控制。当G位620为零时,暂停中断不在搁置中。当G位620为1时,暂停中断在搁置中。当程序缓冲器522中的第一报告群组被写入时,G位620被设置为零。即,当运行时间检测程序缓冲器原始地址(ROA)702等于图7的运行时间检测程序缓冲器当前地址(RCA)706时,G位620被设置为零。当尝试将除了第一报告群组之外的报告群组存储在程序缓冲器522中时,G位620在运行时间检测暂停条件不存在的情况下经设置为零,且报告群组被存储。当尝试将除了第一报告群组之外的报告群组存储在程序缓冲器522中时,G位620在运行时间检测暂停条件存在的情况下经设置为1,且不存储报告群组。

[0089] U位622为用于缓冲器满中断及暂停中断的启用控制。当U位622为零时,中断请求的产生被停用,且如果该产生在搁置中,则其保持搁置。

[0090] L位624为缓冲器满中断的搁置控制。当L位624为零时,缓冲器满中断不在搁置中。当L位624为1时,缓冲器满中断在搁置中。

[0091] 密钥字段626为4位的无正负号的整数,其值被用作报告群组的存储的存储保护密钥。仅在存储密钥匹配与对存储器存取请求相关联的存取密钥时才准许报告群组的存储,且在存储密钥匹配存取密钥时或在存储密钥的提取保护位为零时准许提取。在存储密钥的四个存取控制位等于存取密钥时或在存取密钥为零时,密钥匹配。

[0092] 图7描绘当MRIC被准许以半特权模式(即,K位为1)执行时的RICCB控制块的一部分。控制块700也可用于初始化运行时间检测控制的LRIC指令的操作数。控制块700可包



括除了参看图7所描述的那些值之外的额外值。在实施例中,MRIC指令操作数的未另外指明的多个区段不可由较低特权状态程序存取。当半特权模式被准许时,运行时间检测程序缓冲器原始地址(ROA)702及运行时间检测程序缓冲器极限地址(RLA)704是由较低特权状态程序用MRIC指令设置。ROA 702为图5的程序缓冲器522的第一字节的位置。RLA 704指示程序缓冲器522的最后一个字节的位置。

[0093] 在实施例中,运行时间检测程序缓冲器当前地址(RCA)706可由MRIC指令更新。RCA 706为待存储的下一个报告群组在程序缓冲器522中的位置。RCA 706检查报告群组大小字段744(RGS字段),且影响用以形成程序缓冲器522的地址的有效位位置的数目。64位的RCA 706为字0、字1的位位置0至26-RGS和附加在右边的RGS+5二进位零。此为将存储在程序缓冲器522中的后续报告群组在图5的程序缓冲器522中的开始位置。报告群组为由检测模块506创建且随后存储在程序缓冲器522中的信息单元。在实施例中,当由RCA 706指定的RGS字段744不等于运行时间检测控制的当前报告群组大小(即,RCA 706会改变RGS字段744)时,则将RCA 706设置为ROA 702。

[0094] 剩余样本间隔计数字段742(RSIC字段)可由较低特权程序使用MRIC指令更新。RSIC字段742包括64位的无正负号的整数,其指示剩余样本间隔计数。当运行时间检测控制中的RSIC字段742的值为零或等于缩放因子(scaling factor)字段740(SF字段)中的值,且运行时间检测经启用时,则下一样本间隔为基于取样模式708(M)及SF字段740的值的完整间隔。当RSIC字段742是非零的且小于SF字段740,且运行时间检测经启用时,下一样本间隔为部分间隔。当RSIC字段742是非零的且大于SF字段740值,且运行时间检测经启用时,下一样本间隔为延长间隔。当延长间隔过期时,下一间隔是基于SF字段740值。当RSIC字段742经设置为非零值时,其经受SF字段740也经受的相同依赖模型最大限制。当RSIC字段742的原始值为零时,取样模式将支配在LRIC指令及MRIC指令的执行期间RSIC字段742是否经设置为SF字段740中的值,或该字段是否继续展示为零,直至运行时间检测经启用。

[0095] SF字段740含有64位的无正负号的整数,其值为单元的缩放因子计数。单元的尺寸是根据模式字段708(M字段)确定。当RSIC字段742中的值为零时,SF字段740提供RSIC字段742的递减至零的初始值,在递减至零时,当前指令被识别为样本指令,且从SF字段740值再新闻隔计数。SF字段740的有效值在1至 $2^{64}-1$ 的范围中。如果指定了零,则采用值一。然而,每一模型可具有SF字段740的最小值及最大值两者。最小值及最大值也可基于模式字段708而不同。如果指定了小于最小值的值,则依赖模型最小值被载入。如果指定了大于最大值的值,则依赖模型最大值被载入。

[0096] DC控制字段736为4位的无正负号的整数,其值指明与数据提取或存储高速缓存未命中相关联的高速缓存等待层级。即,样本指令遭遇数据存取高速缓存未命中。除非被另一运行时间检测控制禁止,否则尝试存储表示样本指令的报告群组,该样本指令的数据存取在数值上大于或等于由DC控制字段736的值指明的层级的高速缓存等待层级处经识别为未命中。用于数据存取的高速缓冲存储器结构及高速缓存等待层级是依赖模型的。对于具有多个操作数或长操作数的指令,其是依赖模型于哪一操作数存取(如果存在)是用于报告控制。依赖模型行为可忽略DC控制字段736的值,且因此不使用该值作为存储报告群组的原因。

[0097] IC字段734为4位的无正负号的整数,其值指明与指令提取高速缓存未命中相关联

的高速缓存等待层级。即,样本指令的提取遭遇指令提取高速缓存未命中。对于IC字段734及DC控制字段736两者而言,高速缓存等待层级为特定高速缓存层级存取与进行观测的处理器之间的距离的抽象化。等待层级取决于处理器与主存储器之间的嵌套高速缓存层级的量与在多个处理器之间共用这些高速缓存层级的方式的组合。较大等待层级通常对应于消耗较多时间的存取。IC字段734及DC控制字段736中的值可被视为高速缓存等待层级的零原点识别。举例而言,值零对应于L1高速缓冲存储器(即,最接近处理器的高速缓冲存储器)。值一因此为下一层高速缓冲存储器,其可被称为L2高速缓冲存储器,或甚至L1.5高速缓冲存储器(在一些机器中)。值2-15指明额外高速缓存等待层的逻辑渐进,直至达到主存储器,但不包括主存储器本身。一般而言,高速缓冲存储器结构不会深达十五层。因此,IC字段734及DC控制字段736中的值15被解译为特殊情况,其分别且无关于高速缓存等待层级地意味:并未出于产生报告群组的存储的目的而识别出指令提取或数据存取的高速缓存未命中。除非被另一运行时间检测控制禁止,否则尝试存储表示样本指令的报告群组,该样本指令的提取被识别在数值上大于或等于由IC字段734的值指明的层级的高速缓存等待层级处未命中。用于指令提取的高速缓冲存储器结构及高速缓存等待层级是依赖模型的。依赖模型行为可忽略IC字段734的值,且因此不使用该值作为存储报告群组的原因。

[0098] 高速缓存等待层级更动控制报告控制位732(F位)是用于非分支指令且用于分支预测控制。当运行时间检测控制中的F位732为零时,检查运行时间检测控制的高速缓存报告控制(IC字段734及DC控制字段736)且根据其经定义功能来加以使用。检查运行时间检测控制的分支预测控制(BPx<sub>n</sub>722、BPx<sub>t</sub> 724、BPt<sub>i</sub> 726及BPn<sub>i</sub> 728位)且根据其经定义功能来加以使用。当F位732为1时,这些相同控制被忽略且报告群组被存储,除非被另一控制禁止。

[0099] 数据高速缓存未命中控制位730(D位)指示报告群组是否将被存储。如果D位730为1,则可或可不将额外类型记录置于报告群组的含有关于样本指令的依赖模型数据的额外区段中。

[0100] MRIC指令包括分支预测(BP)报告控制(BPx<sub>n</sub> 722、BPx<sub>t</sub> 724、BPt<sub>i</sub> 726及BPn<sub>i</sub> 728)。如果运行时间检测控制中的BP报告控制位为零,则不检查相应条件。如果BP报告控制位为1且相应分支预测条件存在,则报告群组被存储。

[0101] BPx<sub>n</sub>位722在为1时启用对分支预测信息的检查。因此,如果样本分支被错误地预测为经采纳而实际上未被采纳,则报告群组被存储。

[0102] BPx<sub>t</sub>位724在为1时启用对分支预测信息的检查。因此,如果样本分支被错误地预测为未被采纳而实际上经采纳,则报告群组被存储。

[0103] BPt<sub>i</sub>位726在为1时启用对分支预测信息的检查。因此,如果样本分支被正确地预测为经采纳且实际上被采纳,但分支目标被错误地预测,则报告群组被存储。

[0104] BPn<sub>i</sub>位728在为1时启用对分支预测信息的检查。因此,如果样本分支被正确地预测为未被采纳且实际上未被采纳,且分支目标被错误地预测,则报告群组被存储。

[0105] 事务执行模式记录位720(X位)的启用控制控制着对事务执行模式中记录收集的收集。当运行时间检测控制中的X位720为零时,不收集事务执行模式中记录。当X位720为1时,收集事务执行模式中记录且将记录置于图5的收集缓冲器508中。如果模型未被安装有事务执行设施,则忽略X位720。

[0106] RIEMIT指令控制位718(E位)控制RIEMIT指令的执行。当在运行时间检测经启用时

运行时间检测控制中的E位718为零或被忽略且当作零来对待时,RIEMIT执行无操作(no-operation)。当E位718为1且未被以其他方式忽略时,RIEMIT经启用以执行其经定义功能。

[0107] J位746在为零时指定:不管掩码值如何,条件分支(BC)指令在其他类型分支类别中。如果J位746为1,则指定掩码15的BC指令在返回类型分支类别中。当BC指令指定掩码1-14时,其不受J位746影响且始终在其他类型分支类别中。当在返回类型分支类别中时,R位716控制至图5的收集缓冲器508中的包括。当在其他类型分支类别中时,B位748控制至收集缓冲器508中的包括。其他类型分支类别也可被指示为传送类型分支类别。

[0108] 指令地址码位714(C位)控制调用类型分支的启用。如果运行时间检测控制中的C位714为1且指令为调用类型分支,则收集缓冲器508被更新。如果对调用类型分支及返回类型分支两者的依赖模型检测被组合,则C位714对两个类型起作用,且R位716无效。

[0109] R位716为返回类型分支的启用控制。如果运行时间检测控制中的R位716为1且指令为返回类型分支,则收集缓冲器508被更新。

[0110] B位748为对除了调用类型分支及返回类型分支以外的分支的启用控制。如果运行时间检测控制中的B位748为1且指令为由运行时间检测识别的其他类型分支,则收集缓冲器508被更新。

[0111] 最大地址已超出位712(MAE位)在经设置为1时指示已存储了具有经设置为1的C位714的一个或多个报告群组。一旦MAE位712经设置为1,继续运行时间检测的执行不会将该位设置回至零。将MAE位712指定为零的LRIC指令或MRIC指令的执行会将MAE位712设置为零。

[0112] 运行时间检测下一个(RINEXT)控制位710(N位)控制着对运行时间检测下一个指令(其控制样本指令的执行)的启用。当运行时间检测控制中的N位710为零或被忽略且当作零来对待时,RINEXT执行无操作。当N位710为1且未被以其他方式忽略时,RINEXT被启用以执行其经定义功能。

[0113] 模式字段708(M字段)为4位的无正负号的整数,其在运行时间检测控制中的值指定用于运行时间检测控制的取样模式。所支持的取样模式可包括基于对CPU循环进行计数、对指令进行计数的取样,或可回应于诸如RINEXT的样本指令而定向至样本。

[0114] 报告群组大小字段744(RGS)为3位的无正负号的整数,其值指定报告群组的记录的数目( $R_{RG}$ )。报告群组中的记录的数目可从两个记录(包括开始/时间戳记录及指令最后记录)变化至多达256个记录。在实施例中,上限可为依赖模型的。置于报告群组中的16字节记录的数目为 $2^{(RGS+1)}$ 。

[0115] 主要CPU能力抑制控制位738(Y位)及次要CPU能力抑制控制位739(Z位)被共同称为抑制控制。对报告群组的存储的抑制意味着不执行存储尝试。当配置中的所有CPU的CPU能力相同时,抑制控制无效且不发生抑制。在一配置中,如果一个CPU的CPU能力不同于另一CPU的能力,则抑制控制生效,且将至少一个CPU称为以CPU能力或主要CPU能力操作,而将至少一其他CPU称为以次要CPU能力操作。主要CPU能力及次要CPU能力是不同操作速度。当Y位738及Z位739均为零时,抑制不发生。当Y位738为零且Z位739为1时,在CPU(例如,处理器106)以次要CPU能力操作的情况下发生抑制。当Y位738为1且Z位739为零时,在CPU(例如,处理器106)以主要CPU能力操作的情况下发生抑制。当Y位738及Z位739均为1时,发生抑制。

[0116] 图7的以上字段及位是字段的放置及命名的示例,且在本文中为清楚起见而提供。

将理解,在其他实施例中,可仅使用字段的子集,字段可呈任何次序或在任何位置中,和/或可用不同名称来表示。

[0117] 当运行时间检测经安装且启用时,可在收集缓冲器508中捕获许多事件及数据。收集缓冲器508被用以捕获记录的集合,记录的内容报告在程序执行期间由处理器106识别的事件。示例为:一个或多个采纳分支的执行,事务执行中止事件,高速缓存未命中,及运行时间检测发出指令的操作数。IC控制字段734及DC控制字段736设置层级,程序可对在该层级采取用以改进指令或数据预取行为的某一校正动作有兴趣。RIEMIT指令的执行通过将通用寄存器的值存储至收集缓冲器508中来收集该值。额外数据可被收集和/或存储在其他缓冲器中,其他缓冲器诸如用以收集依赖模型样本指令数据以建构运行时间检测指令记录的指令数据缓冲器(IDB)(未描绘)。

[0118] 所收集的运行时间检测信息是以取样方式报告。对来自指令流的指令进行取样。经取样的指令被称为样本指令。当运行时间检测经启用时,用于确定样本指令的若干模式被定义如下。在循环计数模式中,计数为SF 740或RSIC 742(其任一个被用以提供用于当前间隔的计数)中所指定的CPU循环的数目。回应于与取样模式相关联的事件调整该计数。举例而言,当处理器106处在操作状态中时,该计数可被递减。当该计数递减至诸如零的临限值时,当前指令被识别为样本指令,且将该计数重新初始化至SF 740值且该计数开始随下一循环递减。当样本指令的执行完成时,执行报告(如果适当)。

[0119] 在指令计数模式中,在SF 740或RSIC 742(其任一个被用以提供用于当前间隔的计数)中指定计数。针对由单一操作单元组成的指令,在作为用以调整计数的事件的指令完成时递减计数。当计数递减至诸如零的临限值时,指令为样本指令。对于由多个操作单元组成的指令,可按以下方式中的一个递减计数:

[0120] a. 针对可中断指令,直到部分完成的所有操作单元表示一个已计数单元,针对该已计数单元,计数递减。

[0121] b. 针对可中断指令,从最近的部分完成起直到最终完成的所有操作单元表示一个已计数单元,针对该已计数单元,计数递减。

[0122] c. 针对在执行由指令的参数指定的处理的CPU确定子部分之后完成的指令,完成表示一个已计数单元,针对该已计数单元,计数递减。

[0123] d. 针对在执行多个操作单元之后完成而不在以上类别a-c中的指令,最后一个操作单元的完成表示一个已计数单元,针对该已计数单元,计数递减。

[0124] 当针对指令的任何已计数单元而言计数递减至零时,指令为样本指令。当达到诸如零的临限值时,将计数重新初始化至SF 740值,且计数开始如以上a-d中所描述的倒数计数(count down)。在计数模式的所有情况下,在样本指令的最后一个操作单元完成之后发生报告(如果适当)。

[0125] 在导向取样模式中,当N位710为1且RINEXT指令被成功执行时,导向取样发生。样本指令为RINEXT指令的后的下一顺序指令(NSI)。如果下一顺序指令为执行类型指令,则样本指令为该执行类型指令的目标指令。当处于循环计数或指令技术模式中时,可发生导向取样。计数取样结合导向取样及其所导致动作中的任一个继续且不另外受影响,以下情况除外:如果根据计数取样确定的样本指令与由导向取样确定的指令相同,则不存储两个报告群组。

[0126] 不管取样模式如何,当样本指令是由RINEXT指令的执行识别时,存储报告群组。然而,运行时间检测控制Y 738、Z 739、Qs 614及Ps 612继续有效。

[0127] 循环计数及指令计数取样各自基于内部系统事件及例外条件而确定会经受一定量的可变性的适当间隔。倒数计数(countdown)在运行时间检测从停用转变至启用时开始。视可插入于RINEXT与NSI的完成之间的任何事件而定,导向取样经受较少量的可变性。值得注意,中断可导致被视为NSI的指令不再为NSI。

[0128] 与模式无关,取样识别样本指令。一旦样本指令经识别,收集便在该样本指令的执行完成时停止,且开始报告。接着应用管理报告的各种报告控制。在使报告群组的存储在搁置中时,收集继续。

[0129] 当不在事务执行模式中时,在样本指令的执行完成时,报告群组的存储变为在搁置中。当在事务执行模式中时,在样本指令的执行完成时,报告群组的存储被延后,直至事务结束,且接着变为在搁置中。当报告群组的存储被延后或在搁置中时,如果识别出以下中断中的任一个,则报告群组可被清除:1) 程序中断;2) 紧急机器检查中断;3) 重新启动中断;及4) 监督调用中断。

[0130] 任何搁置中的I/O、外部及可抑制的机器检查中断保持搁置,直至报告群组已被存储或运行时间检测控制确定报告群将不被存储。

[0131] 每一模式可或可不允许报告控制的不同集合。当取样模式为指令计数或循环计数,但导向取样也被使用时,可能通过多个取样方法来识别出相同样本指令。当此情况发生,且待使用的报告控制根据取样模式而不同时,应用与导向取样相关联的报告控制。

[0132] 归因于可能出现的非同步及非请求系统事件,用以对特定指令进行取样的间隔的精确确定通常不可行。实情为,可将RINEXT指令用以较接近地指明样本指令。

[0133] 当在循环计数模式或者指令计数时,可以与从指令计数或循环计数样本识别样本指令的很接近地发出RINEXT指令。相关联的报告群组的内容是好像样本指令被识别为RINEXT指令的NSI,而不是好像应用了样本指令的循环计数或指令计数识别。

[0134] 如果以下例外条件中的任何一个或多个得到满足,则RINEXT的执行可作为无操作来执行:

[0135] 1. 运行时间检测控制是无效的。

[0136] 2. 在问题状态中,当前运行时间检测控制的Ps 612为零,从而指示问题状态报告不被准许。

[0137] 3. 在监督状态中,当前运行时间检测控制的Qs 614为零,从而指示监督状态报告不被准许。

[0138] 4. 当前运行时间检测控制的N位710为零,从而指示RINEXT指令本身不被准许。

[0139] 5. 存储被抑制。

[0140] 6. 当前PSW中的字段指示运行时间检测被停用。

[0141] 7. 依赖模型临限值将被超过。在一时间段中RINEXT已被发出的次数已超过依赖模型极限。

[0142] 8. 存在程序缓冲器满条件。

[0143] 9. 存在运行时间检测暂停条件。

[0144] 10. 下一顺序指令为开始解译执行指令。

[0145] 11. 下一顺序指令为监督调用指令。

[0146] 转而参看图8,大体上展示了收集缓冲器508的实施例。如先前所描述,当在程序执行期间启用运行时间检测时,运行时间检测数据是收集于处理器106内。在实施例中,在处理器106内收集数据的位置为收集缓冲器508,且视情况为指令数据缓冲器。在实施例中,收集缓冲器508为处理器106的内部缓冲器,其用以存储(save)所收集的最近记录。当检测到样本触发点时,将记录从收集缓冲器508复制至程序缓冲器522中以作为将写入至程序缓冲器522的报告群组的部分。在实施例中,以非破坏性方式从收集缓冲器复制记录。

[0147] 收集缓冲器508可被称为“硬件收集缓冲器”,因为收集缓冲器508位于处理器中,且在实施例中经实施为表示给定事件的指令地址802及事件后设数据804的寄存器对的阵列。在实施例中,指令数据缓冲器也由寄存器对的阵列来实施。事件的示例为采纳分支,针对该采纳分支,寄存器对可保存该分支的指令地址,且该后设数据可保存分支的目标以及关于分支的历史行为的信息。在实施例中,寄存器对经排序,且随着事件出现于指令流中而被顺序地更新。维持计数器以指示阵列中的最近经更新输入项的索引。在实施例中,收集缓冲器508为循环缓冲器,且当收集缓冲器508被填满时,下一事件覆写阵列中的第一输入项,且在发生后续事件时阵列的寄存器对的顺序更新重新开始。因而,假设阵列CB[0]至CB[N-1]及指示最新更新索引的计数器i,所捕获的事件的追踪可由序列CB[i]、CB[i-1]、……、CB[1]、CB[0]、CB[N-1]、CB[N-2]、……、CB[i+1]来表示。在另一实施例中,使用两个指针:指向缓冲器中的最旧输入项的头指针,及指向缓冲器中的最新输入项的尾/当前指针。

[0148] 在收集缓冲器508中顺序地捕获表示处理器106在任何给定执行点处的状态的事件。将收集缓冲器508用以捕获记录的集合,记录的内容报告在程序执行期间由处理器106识别的事件(例如,一个或多个采纳分支的执行,事务执行中止事件,RIEMIT指令的操作数等)。在实施例中,经识别的事件视图7中所展示的RICCB的内容而定。图8中所展示的收集缓冲器508的实施例中的输入项包括事件指令地址802及其他相关的事件后设数据804。事件后设数据804的示例包括(但不限于):采纳分支的指令地址及该分支的目标(包括关于分支的历史行为的某一信息);RIEMIT指令的指令地址及各别寄存器值;及事务中止指令的地址及各别事务复原进入点。

[0149] 图8中所展示的收集缓冲器508的实施例能够存储多达32个输入项(即,关于32个事件的信息),其中由64个位(例如,位0:63)指定每一指令地址802且由64个位(例如,位64:127)指定事件后设数据804。收集缓冲器的大小( $R_{CB}$ )为依赖模型计数,其表示记录的数目。在图8中所展示的收集缓冲器508的实施例中,收集缓冲器的字节大小为16字节记录大小的倍数。在实施例中,收集缓冲器的大小为大于或等于模型的最大报告群组的计数( $R_{RG}$ )与报告群组中的未从收集缓冲器获取的记录的计数( $R_{NC}$ )之间的差的记录数目。因此,在实施例中,收集缓冲器的大小经表示为: $R_{CB} \geq (R_{RG} - R_{NC})$ 。

[0150] 在实施例中,收集缓冲器508及指令数据缓冲器(如果使用指令数据缓冲器)的内容被清除或以其他方式受以下事件影响:(1)中断;(2)开启及关闭运行时间检测设施的PSW位(例如,位24)从1变至0;及(3)当运行时间检测设施处于事务执行模式中时识别出样本指令时(在此情况下,对收集数据缓冲器508及指令数据缓冲器的进一步更新停止且在事务结束时继续),此时,报告群组的存储在搁置中且收集缓冲器508及指令数据缓冲器被清除。

[0151] 在诸如图1B中所展示的模拟主机计算机系统的实施例中,收集缓冲器508使用寄

存器和/或存储器实施。在该实施例中,如果存在,则可选指令数据缓冲器也使用寄存器和/或存储器实施。

[0152] 在实施例中,额外能力可影响数据收集,且可看作提供额外数据收集点,而不实质上干扰先前所描述的正常指令计数或循环计数取样。这些能力包括RIEMIT指令的执行,RIEMIT指令通过将通用寄存器的值存储至收集缓冲器508中来收集该值。另外,先前所描述的运行时间检测控制中的数据收集控制位可用以定制所收集数据的类型,例如,E、C、R及B控制位。以此方式,所收集数据的类型是可程序化的。

[0153] 在实施例中,实施指令数据缓冲器以收集用以建构运行时间检测指令记录的依赖模型样本指令数据。指令数据缓冲器从指令收集数据以便在该指令被识别为样本指令时使数据可用。在实施例中,该指令数据缓冲器为处理器中的硬件缓冲器/存储位置,关于可变为触发器以作为样本点的指令的信息是存储在该硬件缓冲器/存储位置中,使得在登出过程期间,该信息可与来自收集缓冲器508的数据一起被写出。类似于收集缓冲器508,该指令数据缓冲器包括指令地址及与该指令相关联的后设数据。指令数据缓冲器中的后设数据常常为依赖机器的,且可包括(但不限于):高速缓存未命中相关的信息,及分支预测相关的信息。

[0154] 根据实施例,所收集的其他数据可不来自于收集缓冲器508且可不来自于指令数据缓冲器。示例包括用以形成以下各者的部分的数据:(1)报告群组的第一记录:时间戳或开始记录;及(2)额外类型的记录,其可针对每一个报告群组而创建且因此不存储在收集缓冲器508中,这些记录(当存在时)可放置于报告群组的额外或依赖机器部分中。这些系统产生的记录在本文中被称为“系统信息记录”。

[0155] 图9描述了在样本点处被存储到程序缓冲器522的报告群组900的高级示例。在记录中的报告群组的大小被表示为 $R_{RG}$ ,等于 $2^{(R_{GS}+1)}$ ,其中, $R_{GS}$ 是作为指数的报告群组大小。当在报告群组中使用,可以或不非破坏性地复制从除了收集缓冲器508之外的位置复制的模型相关的数量的记录( $R_{NC}$ )。在图9的示例中, $R_{RG}=8$ , $R_{GS}=2$ ,并且 $R_{NC}=4$ 。在图9中所示的示例报告群组900包括报头区段902、主体区段904、额外记录区段906和页脚区段908。

[0156] 报头区段902可以包括开始记录或时间标记记录,用于存储状态、跟踪和/或定时信息。对于在程序缓冲器中存储的第一报告群组在报头区段902中存储开始记录(即,当 $R_{CA}706$ 等于 $R_{OA}702$ 时)。在一个实施例中,开始记录包括:记录类型字段“02”;报告群组数量(NRG)字段,用于指示在程序缓冲器中当前存储了多少报告群组; $R_{GS}$ 字段,用于指示报告群组的大小;停止(S)字段,用于指示程序缓冲器522是否满了;暂停(H)字段,用于指示是否暂停了运行时间检测;以及,日时(TOD)时钟时间段,用于指示何时写入开始记录。在一个实施例中,在开始记录中的字段的至少一个子集出自RI控制块(例如, $R_{ICCB}$ )。时间标记记录的一个实施例具有记录类型“03”,并且包括用于指示何时存储记录的TOD时钟字段。在一个实施例中,对于除了第一报告群组之外的每一个报告群组在报头区段902中存储时间标记记录。

[0157] 报告群组的主体区段904可以包括用于从收集缓冲器508样本的事件和信息的多种记录。事件和信息可以表示例如由发送指令捕获的状态信息、事务执行异常终止、调用、返回、分支和填充。

[0158] 在一个实施例中,在成功地执行RIEMIT指令时,在收集缓冲器508中建立和存储发

送记录。发送记录的一个实施例包括：记录类型字段“10”；指令地址码字段，用于指示在发送记录中如何表示当前PSW的指令地址比特位置；指令地址字段，其根据编址模式（例如，64、31或24比特）来变化，并且如果RIEMIT是执行类型指令的目标则包含RIEMIT指令或执行类型指令的指令地址，以及，发送数据字段，用于存储来自RIEMIT指令指定的一般寄存器的数据。

[0159] 在一个实施例中，通过隐含的异常终止或通过事务异常终止指令的执行来建立和在收集缓冲器508中存储事务执行模式异常终止记录。异常终止记录的一个实施例包括：记录类型字段“11”、指令地址码字段，用于指示如何在事务执行异常终止记录中表示当前PSW的指令地址比特位置；指令地址字段，其根据编址模式（例如，64、31或24比特）来变化，并且如果异常终止指令是执行类型指令的目标则包含异常终止指令或执行类型指令的指令地址，以及，用于与异常终止相关联的任何模型相关数据的字段。

[0160] 在一个实施例中，通过调用类型分支指令来建立调用记录，该调用类型分支指令例如是：当R2字段非零时的BRANCH AND SAVE（分支和存储）（BASR）；BRANCH AND SAVE（分支和存储）（BAS）；BRANCH RELATIVE AND SAVE LONG；BRANCH RELATIVE AND SAVE；当R2字段是非零时的BRANCH AND LINK（BALR）；BRANCH AND LINK（BAL）；以及，当R2字段是非零时的BRANCH AND SAVE AND SET MODE。调用记录的一个实施例包括：记录类型字段“12”、指令地址码字段，用于指示如何在调用记录中表示当前PSW的指令地址比特位置；指令地址字段，其根据编址模式（例如，64、31或24比特）来变化，并且如果分支指令是执行类型指令的目标则包含分支指令或执行类型指令的指令地址，以及，良好作为的字段，用于指示是否周期地预测了分支；以及，目标地址字段，其包含分支目标地址（也被称为“被调用位置”）。

[0161] 返回记录和传送记录可以具有与调用记录相同的格式。在一个实施例中，返回记录具有记录类型字段“13”并且通过诸如当R2字段非零时的BRANCH ON CONDITION（BCR）的返回类型分支指令的执行而被建立。对于返回记录，如果分支是执行类型指令的目标，则指令地址字段包含分支指令或执行类型指令的地址，并且目标地址字段包含返回位置。

[0162] 在实施例中，传送记录具有为“14”的记录类型字段且是通过诸如以下各者的返回类型分支指令的执行而创建：a. 当R2字段为非零且掩码在1-14的范围中时，BRANCH ON CONDITION（BCR）；b. 当J位为零或掩码在1-14的范围中时，BRANCH ON CONDITION（BC）；c. BRANCH ON COUNT（BCT, BCTR, BCTG, BCTGR）；d. BRANCH ON INDEX HIGH（BXH, BXHG）；e. BRANCH ON INDEX LOW OR EQUAL（BXLE, BXLEG）；f. BRANCH RELATIVE ON CONDITION（BRC）；g. BRANCH RELATIVE ON CONDITIONLONG（BRCL）；h. BRANCH RELATIVE ON COUNT（BRCT, BRCTG）；i. BRANCH RELATIVE ON COUNT HIGH（BRCTH）；j. BRANCH RELATIVE ON INDEX HIGH（BRXH, BRXHG）；k. BRANCH RELATIVE ON INDEX LOW OR EQUAL（BRXLE, BRXLG）；l. COMPARE AND BRANCH（CRB, CGRB）；m. COMPARE AND BRANCH RELATIVE（CRJ, CGRJ）；n. COMPARE IMMEDIATE AND BRANCH（CIB, CGIB）；o. COMPARE IMMEDIATE AND BRANCH RELATIVE（CIJ, CGIJ）；p. COMPARE LOGICAL AND BRANCH（CLRB, CLGRB）；q. COMPARE LOGICAL AND BRANCH RELATIVE（CLRJ, CLGRJ）；r. COMPARE LOGICAL IMMEDIATE AND BRANCH（CLIB, CLGIB）；及s. COMPARE LOGICAL IMMEDIATE AND BRANCH RELATIVE（CLIJ, CLGIJ）。传送记录是在分支经采纳时创建。对于传送记录，指令地址字段含有分支指令或执行类型指令（如果分支为执行类型指令的目标）的地址，且目标地址字段含有返回位置。



[0163] 当收集缓冲器508中的有效记录的数目不足以填满具有当前RGS的报告群组时,在报告群组中使用填充记录。填充记录的实施例包括“00”的记录类型字段,以指示该记录为填充记录,且剩余字节是未定义的。

[0164] 额外记录区段906(在存在时)可含有依赖模型记录。在实施例中,除了记录类型经设置为“01”以指示该记录为额外记录且该额外记录的剩余字节含有依赖模型数据之外,额外记录的格式类似于填充记录。

[0165] 页脚区段908可以包括包含关于样本指令的执行的指令记录。当对于样本指令存储报告群组时,建立指令记录。指令记录的一个实施例包括:记录类型字段“04”、指令地址码字段,用于指示如何在指令记录中表示当前PSW的指令地址比特位置;指令地址字段,其根据编址模式(例如,64、31或24比特)来变化,并且如果样本指令是执行类型指令的目标则包含样本指令或执行类型指令的指令地址,以及,指令数据缓冲器(IDB)字段,其包含从IDB收集的任何模型相关数据。

[0166] 图10描述了用于实现运行时间检测设施的一个实施例的处理流1000。处理流1000可以被处理器106实现。处理流1000可以替代地被图1B的模拟处理器29实现。为了容易说明,在此相对于处理器106来描述处理流1000。在块1002处,处理器106执行诸如应用518的应用的指令流。在块1004,例如通过位于在图5中所示的指令模块506中的硬件来捕获基于执行的指令的全部或子集的运行时间检测数据。在诸如在图5中所示的收集缓冲器508的位于处理器上的硬件收集缓冲器508中存储运行时间检测数据。在块1006处(例如,通过位于处理器106上的硬件)检测运行时间检测样本点触发器,并且在块1008处,将硬件收集缓冲器508的内容复制到诸如程序缓冲器522的程序缓冲器内。程序缓冲器52位于被应用可范围的地址空间中。在一个实施例中,应用518在较少特权的状态中执行,并且对于硬件收集缓冲器522的访问仅是通过管理状态,并且因此,应用518不访问硬件收集缓冲器522的内容。

[0167] 在一个实施例中,以对于块1002透明的方式来执行块1004-1006。如在此使用,术语“透明”指的是完全独立于块1002的块1004-1006的操作,并且如此一来,在使能(即,在执行)块1004和1006的捕获和检测的同时,它们对于由块1002在执行的指令的性能或执行的功能行为没有负担。在一个实施例中,与块1002同时(或在时间上重叠地)执行块1004和1006,块1002执行应用518的指令流。对于此的优点是执行检测不影响应用518的性能。调用块1008通常对于块1002具有显著的性能影响。然而,与由块1006提供的可编程样本机制相结合的由块1004提供的透明缓冲机制允许如何调用块1008的受控管理,因此减轻了由块1008观察到的执行开销。

[0168] 运行时间检测样本点触发器和被捕获和在收集缓冲器508中存储的数据的类型两者是例如基于参考图6上述的RICCB的内容而可编程的。

[0169] 如上所述,实施例可以计算机实施程序及用于实践那些程序的装置的形式体现。实施例可包括在计算机可读/可用介质1102上的如图11中所描绘的计算机程序产品1100,该计算机可读/可用介质1102具有含有作为制品体现于有形介质中的指令的计算机程序代码逻辑1104。计算机可读/可用介质1102的示例性制品可包括软盘、CD-ROM、硬驱动器、通用串行总线(USB)闪速驱动器或任何其他计算机可读存储介质,其中,当计算机程序代码逻辑1104被载入至计算机中且由计算机执行时,该计算机变为用于实践本发明的装置。实施例包括计算机程序代码逻辑1104,该计算机程序代码逻辑(例如)存储在存储介质中、经载入

至计算机中和/或由计算机执行,或经由某一传输介质(诸如,经由电布线或缆线、经由光纤或经由电磁辐射)进行传输,其中,当计算机程序代码逻辑1104被载入至计算机中且由计算机执行时,该计算机变为用于实践本发明的装置。当实施于通用微处理器上时,计算机程序代码逻辑1104的多个段配置该微处理器以创建特定逻辑电路。

[0170] 技术效果和益处包括用于被管理的运行时间的基于硬件的运行时间检测设施。在此所述的运行时间检测设施可以不仅用在实验室环境中或用于离线分析,而且用在在运行时间时的程序内的现场软件环境中并且在程序控制下。实施例提供了与基于软件的检测作比较减小的用于收集检测数据的开销,并且因此可以收集更多的程序概要分析数据,导致更精确的程序概要分析数据以引导重新编译和其他形式的优化(例如,数据重新组织)。另外,由在此所述的实施例提供的硬件事件的可获得性也可以用于引导诸如垃圾收集的其他被管理的运行时间设施。

[0171] 本文中所使用的术语仅用于描述特定实施例的目的且不欲限制本发明。如本文中所示,除非上下文另有清晰指示,否则单数形式“一”及“该”意欲也包括复数形式。将进一步理解,术语“包含”在用于本说明书中时指定所陈述的特征、整数、步骤、操作、元件和/或组件的存在,但不排除一个或多个其他特征、整数、步骤、操作、元件、组件和/或其群组的存在或添加。

[0172] 以下权利要求中的所有构件或步骤加功能元件的相应结构、材料、动作及等效物意欲包括用于连同如具体所主张的其他所主张元件一起执行功能的任何结构、材料或动作。已出于说明及描述目的呈现了本发明的描述,但该描述不欲为详尽的或限于呈所揭示的形式。在不脱离本发明的范畴及精神的情况下,许多修改及变化对于一般本领域普通技术人员将显而易见。选择并描述实施例以便最佳地解释本发明的原理及实际应用,且使其他一般本领域普通技术人员能够理解本发明的各种实施例,实施例具有适合于所设想的特定用途的各种修改。

[0173] 如本领域普通技术人员将了解,本发明的方面可体现为系统、方法或计算机程序产品。相应地,本发明的方面可采用完全硬件实施例、完全软件实施例(包括固件、常驻软件、微代码等)或组合软件方面与硬件方面的实施例的形式,实施例在本文中都可通称为“电路”、“模块”或“系统”。此外,本发明的方面可采用体现于一个或多个计算机可读介质中的计算机程序产品的形式,该一个或多个计算机可读介质具有体现于其上的计算机可读程序代码。

[0174] 可利用一个或多个计算机可读介质的任何组合。计算机可读介质可为计算机可读信号介质或计算机可读存储介质。计算机可读存储介质可为(例如,但不限于)电子、磁性、光学、电磁、红外线或半导体系统、装置或器件,或前述各者的任何合适组合。计算机可读存储介质的更特定示例(非详尽清单)将包括以下各者:具有一或多条导线的电连接件、携带型计算机磁片、硬碟、随机存取存储器(RAM)、唯读存储器(ROM)、可抹除可程序化唯读存储器(EPROM或闪速存储器)、光纤、携带型致密盘唯读存储器(CD-ROM)、光学存储器件、磁性存储器件或前述各者的任何合适组合。在本文件的上下文中,计算机可读存储介质可为可含有或存储供指令执行系统、装置或器件使用或结合指令执行系统、装置或器件使用的程序的任何有形介质。

[0175] 计算机可读信号介质可包括传播的数据信号,该传播的数据信号具有体现于其中

(例如,在基频中或作为载波的部分)的计算机可读程序代码。此传播信号可采用多种形式中的任一个,包括(但不限于)电磁、光学或其任何合适组合。计算机可读信号介质可为并非计算机可读存储介质且可传达、传播或输送供指令执行系统、装置或器件使用或结合指令执行系统、装置或器件使用的程序的任何计算机可读介质。

[0176] 可使用任何适当介质来传输体现于计算机可读介质上的程序代码,适当介质包括(但不限于)无线、有线、光纤缆线、RF等或前述各者的任何合适组合。

[0177] 可以一或多种程序设计语言的任何组合来撰写用于执行本发明的方面的操作的计算机程序代码,该一或多种程序设计语言包括物件导向式程序设计语言(诸如,Java、Smalltalk、C++或其类似者)及习知程序性程序设计语言(诸如,“C”程序设计语言或类似程序设计语言)。程序代码可完全在用户的计算机上执行,部分地在用户的计算机上执行,作为独立套装软件而执行,部分地在用户的计算机上执行且部分地在远程计算机上执行,或完全在远程计算机或服务器上执行。在完全在远程计算机或服务器上执行的情形中,远程计算机可经由任何类型的网络(包括局域网(LAN)或广域网络(WAN))连接至用户的计算机,或可连接至外部计算机(例如,使用因特网服务提供者,经由因特网)。

[0178] 上文中参考根据本发明的实施例的方法、装置(系统)及计算机程序产品的流程图说明和/或示意图来描述本发明的方面。将理解,可通过计算机程序指令来实施流程图说明和/或方块图中的每一块及流程图说明和/或方块图中的块的组合。可将这些计算机程序指令提供至通用计算机、专用计算机或其他可程序化数据处理装置的处理器以产生一机器,以使得经由该计算机或其他可程序化数据处理装置的处理器执行的指令产生用于实施该或流程图和/或方块图块中所指定的功能/动作的构件。

[0179] 也可将这些计算机程序指令存储在计算机可读介质中,其可指导计算机、其他可程序化数据处理装置或其他器件以特定方式发挥作用,使得存储在该计算机可读介质中的指令产生一制品,该制品包括实施该或流程图和/或方块图块中所指定的功能/动作的指令。

[0180] 也可将计算机程序指令载入至计算机、其他可程序化数据处理装置或其他器件上,以使一系列操作步骤在该计算机、其他可程序化装置或其他器件上执行以产生计算机实施的程序,使得在该计算机或其他可程序化装置上执行的指令提供用于实施该或流程图和/或方块图块中所指定的功能/动作的程序。

[0181] 如上所述,实施例可以计算机实施的程序及用于实践那些程序的装置的形式体现。在多个实施例中,本发明系体现于由一个或多个网络元件执行的计算机程序代码中。实施例包括在计算机可用介质上的计算机程序产品,该计算机可用介质具有含有体现于作为制品的有形介质中的指令的计算机程序代码逻辑。计算机可用介质的示例性制品可包括软碟、CD-ROM、硬驱动器、通用流总线(USB)随身碟或任何其他计算机可读存储介质,其中,当计算机程序代码逻辑被载入至计算机中且由计算机执行时,该计算机变为用于实践本发明的装置。实施例包括计算机程序代码逻辑,该计算机程序代码逻辑(例如)存储在存储介质中、载入至计算机中和/或由计算机执行,或经由某一传输介质(诸如,经由电布线或缆线、经由光纤或经由电磁辐射)进行传输,其中,当计算机程序代码逻辑被载入至计算机中且由计算机执行时,该计算机变为用于实践本发明的装置。当实施于一通用微处理器上时,计算机程序代码逻辑的多个段配置该微处理器以创建特定逻辑电路。

[0182] 图中的流程图及方块图说明根据本发明的各种实施例的系统、方法及计算机程序产品的可能实施的架构、功能性及操作。就此而言,流程图或方块图中的每一块可表示程序代码的模块、段或部分,其包含用于实施指定逻辑功能的一个或多个可执行指令。也应注意,在一些替代实施中,块中所提到的功能可不以诸图中所提到的次序发生。举例而言,视所涉及的功能性而定,连续展示的两个块实际上可实质上同时执行,或块有时可以相反次序执行。也应注意,可通过执行指定功能或动作的基于专用硬件的系统或专用硬件与计算机指令的组合来实施方块图和/或流程图说明的每一块及方块图和/或流程图说明中的块的组合。

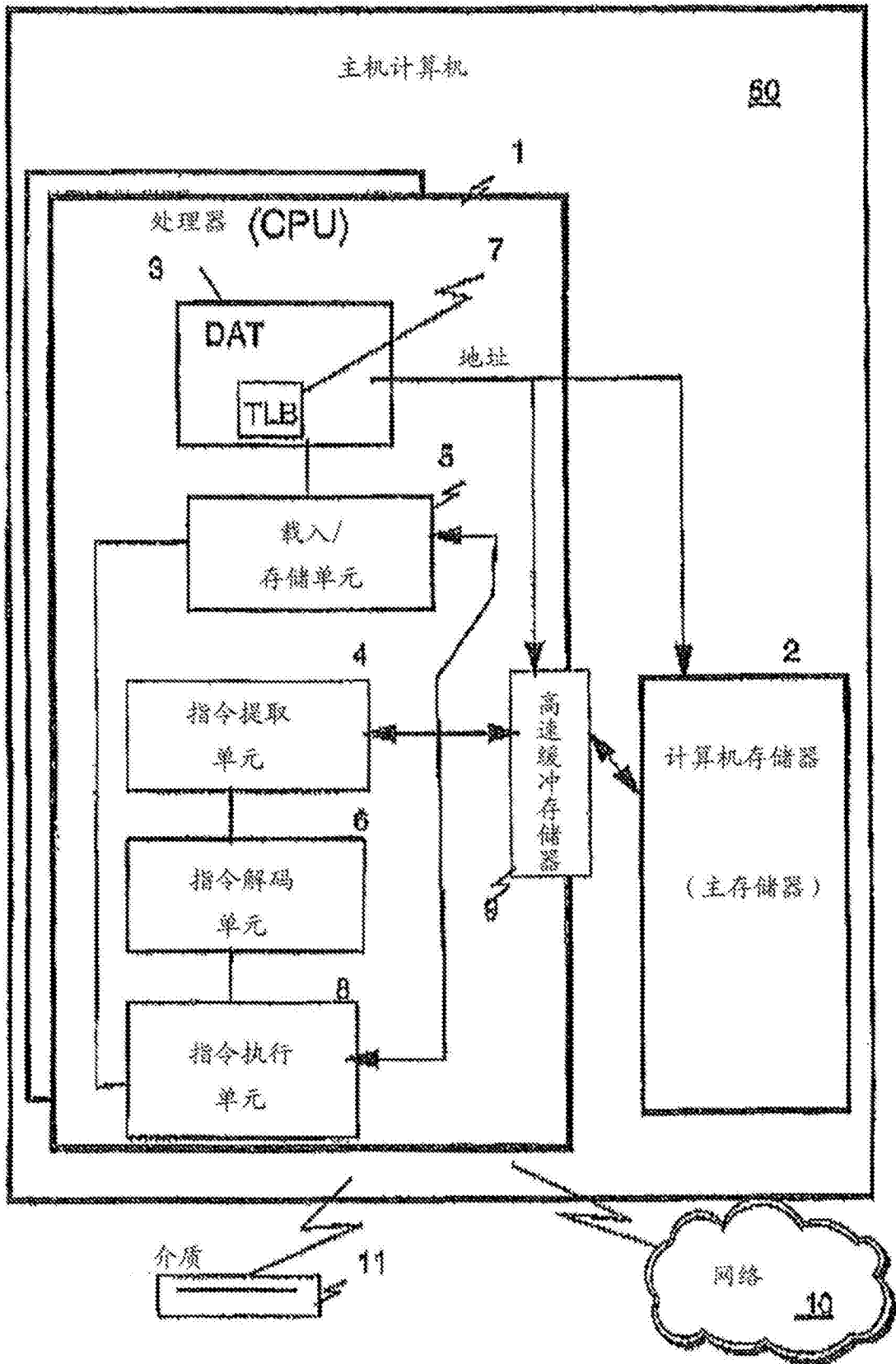


图1A

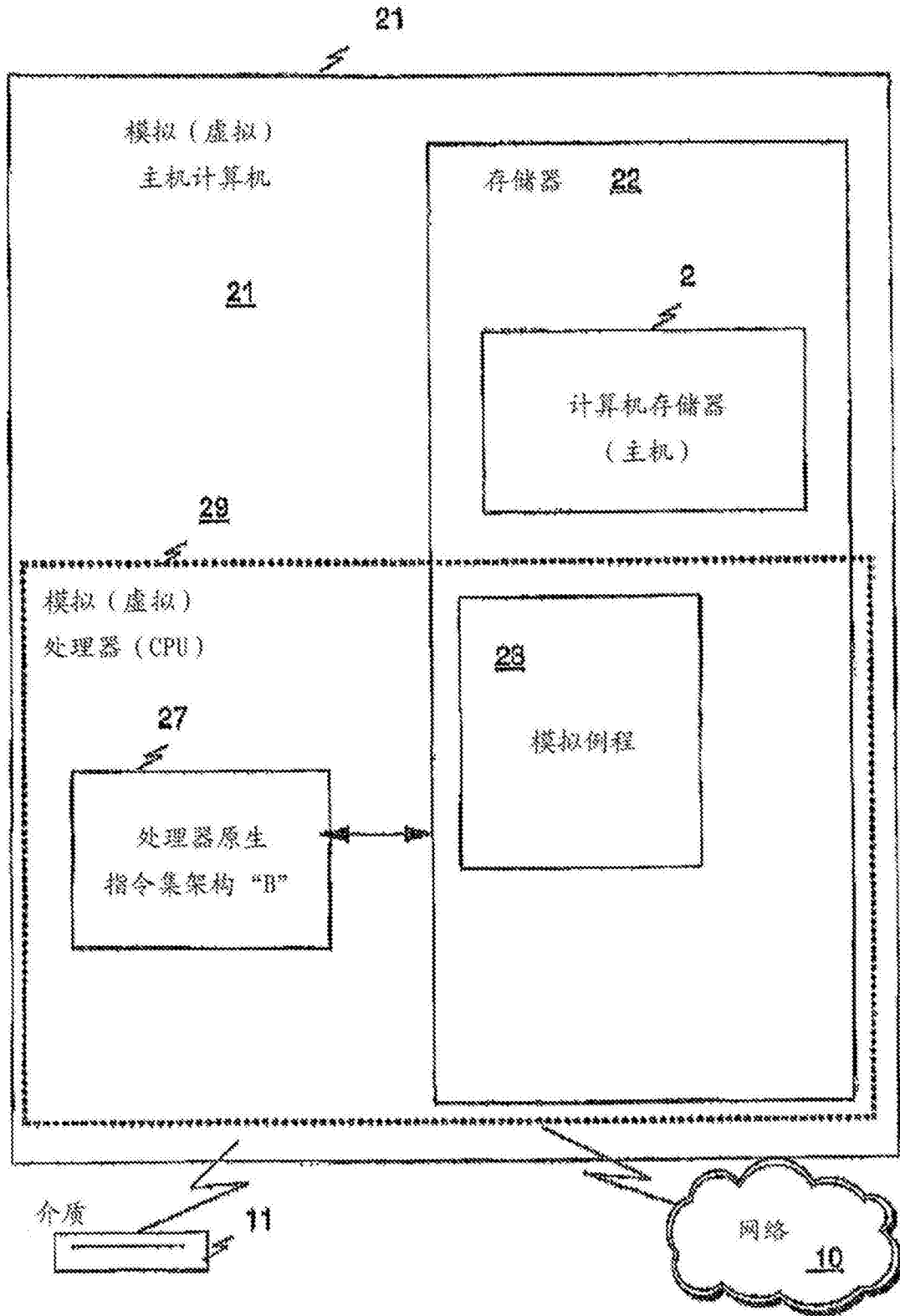


图1B

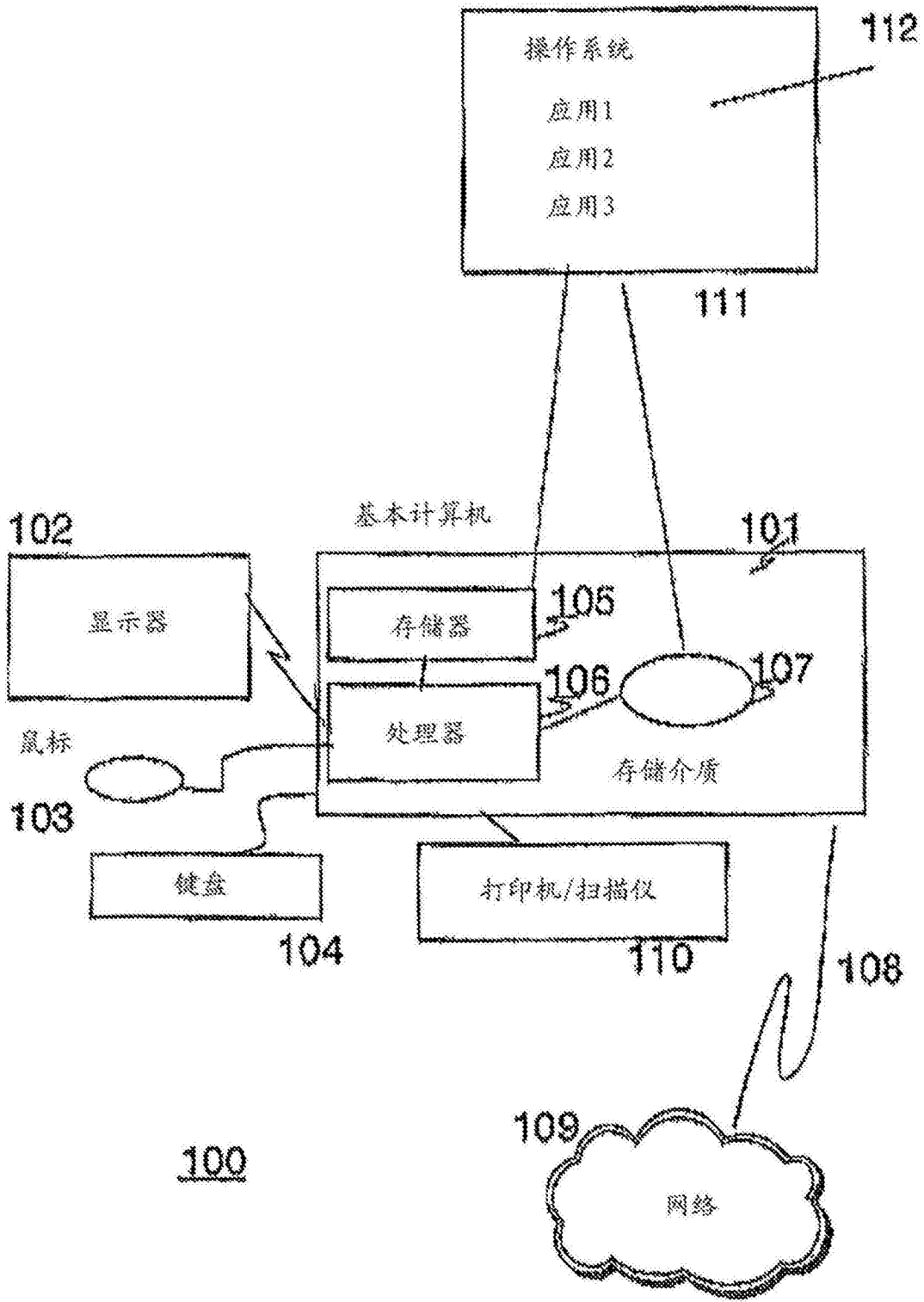


图1C

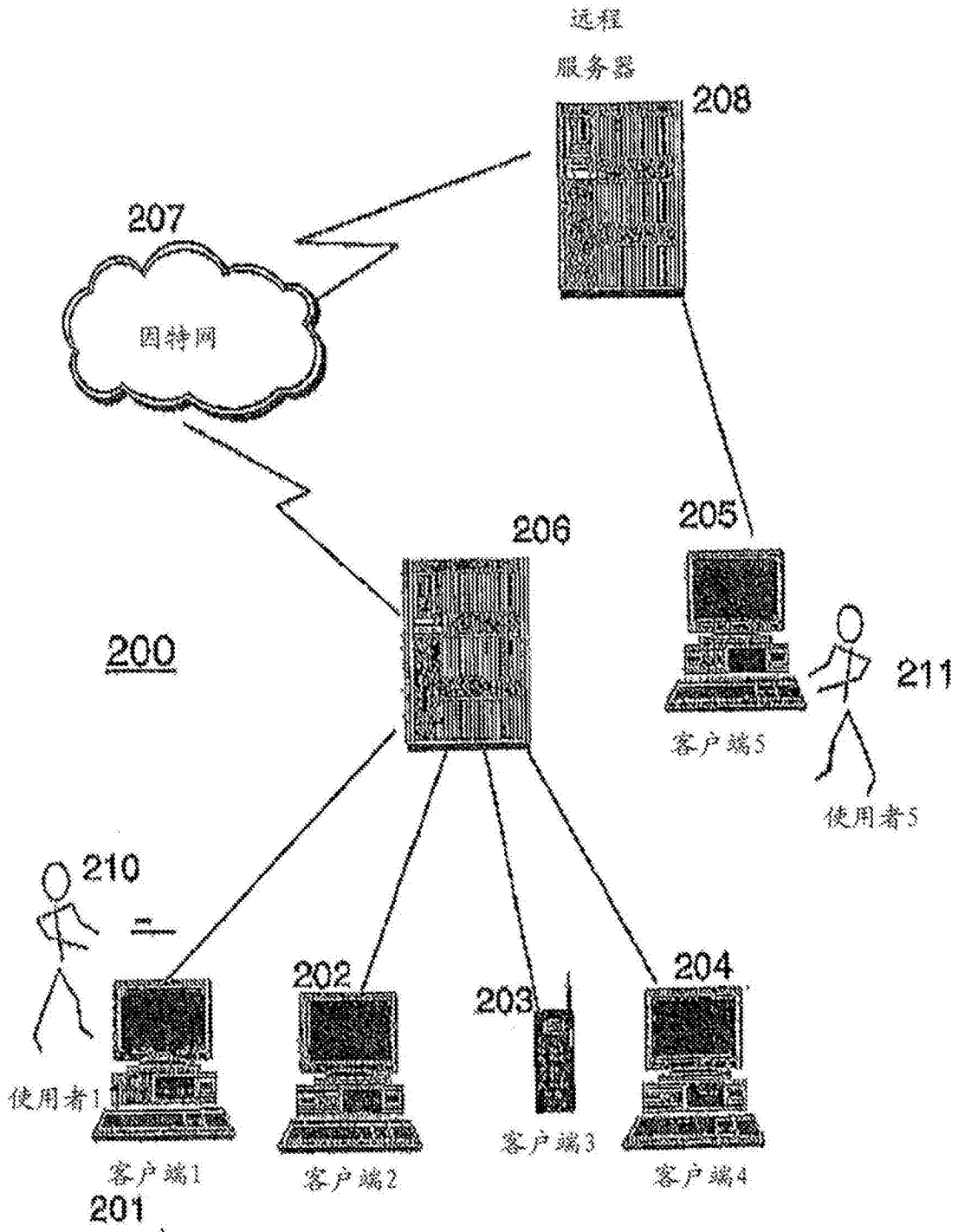


图2



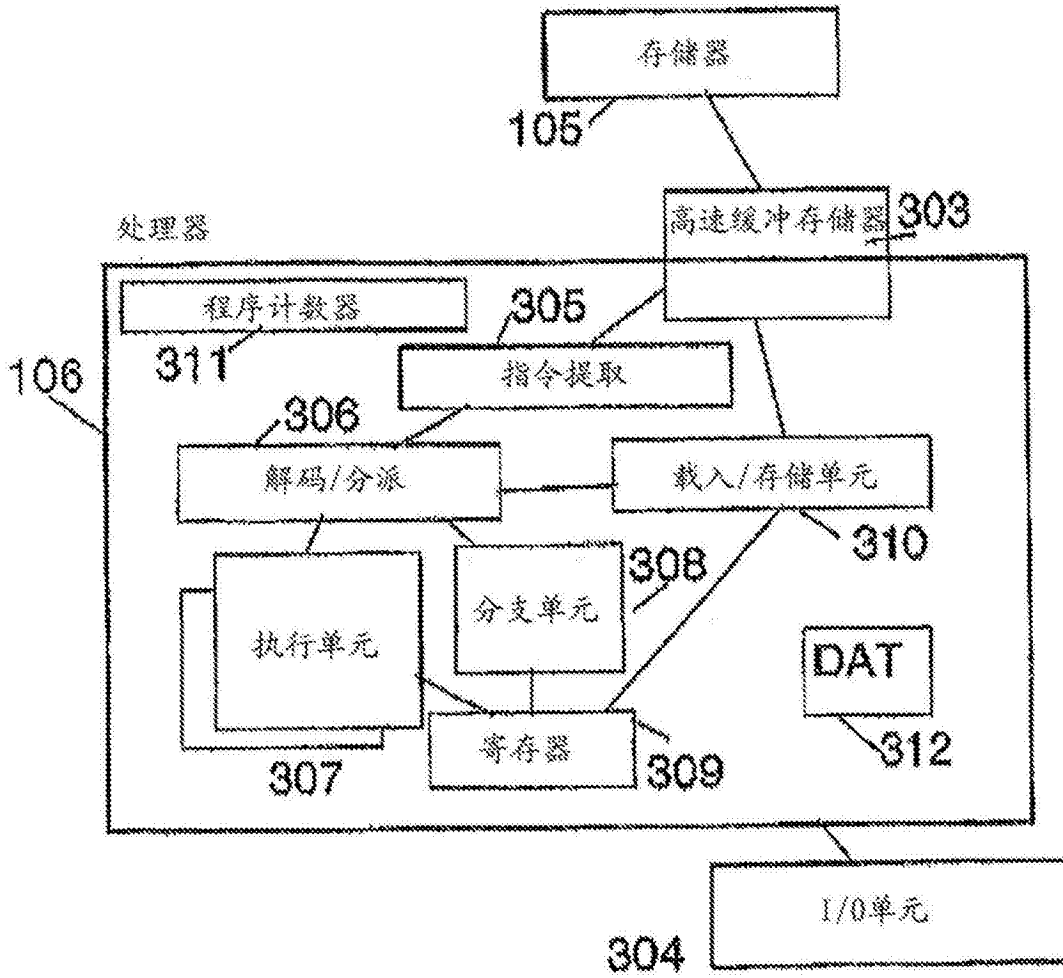


图3

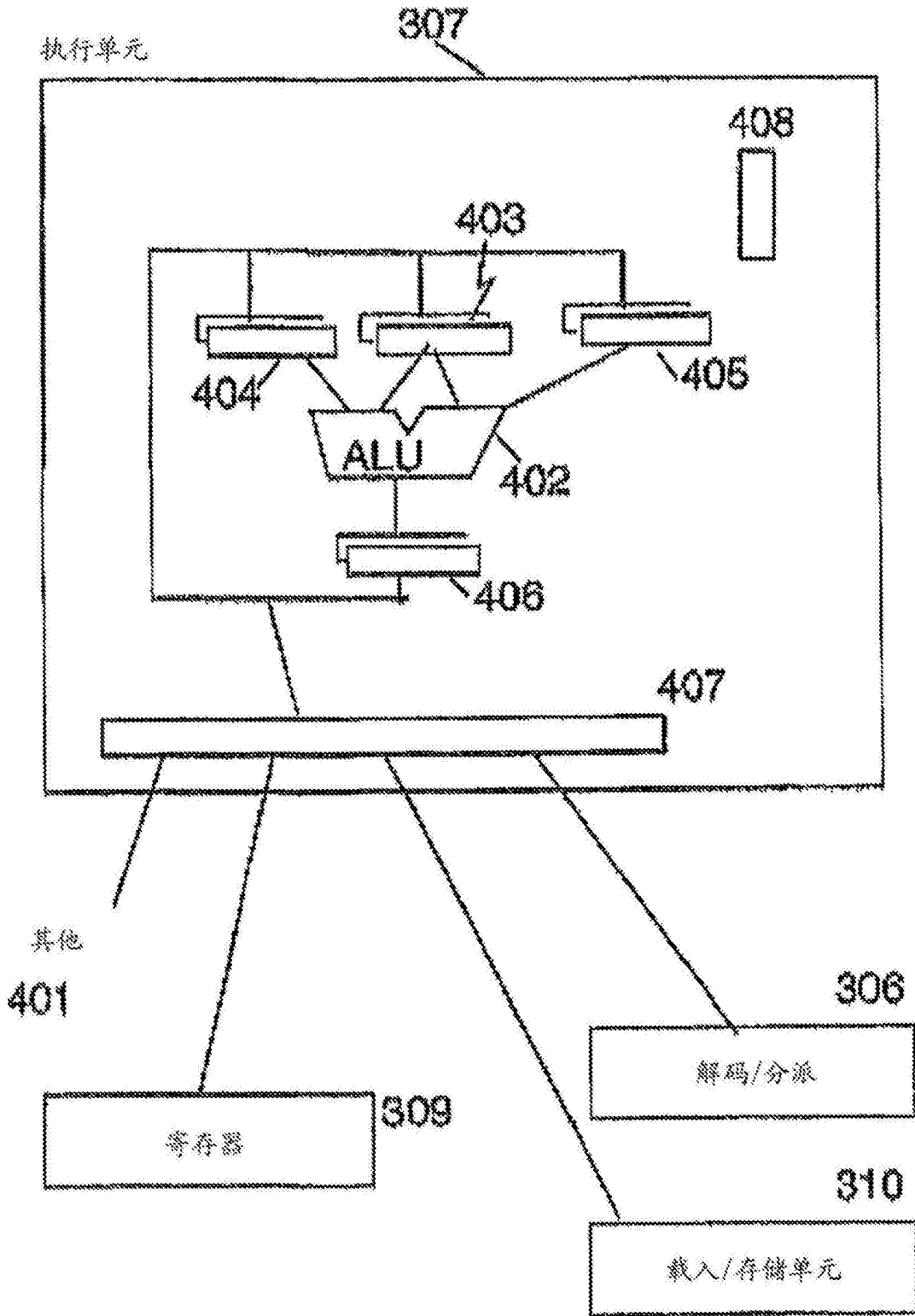


图4A

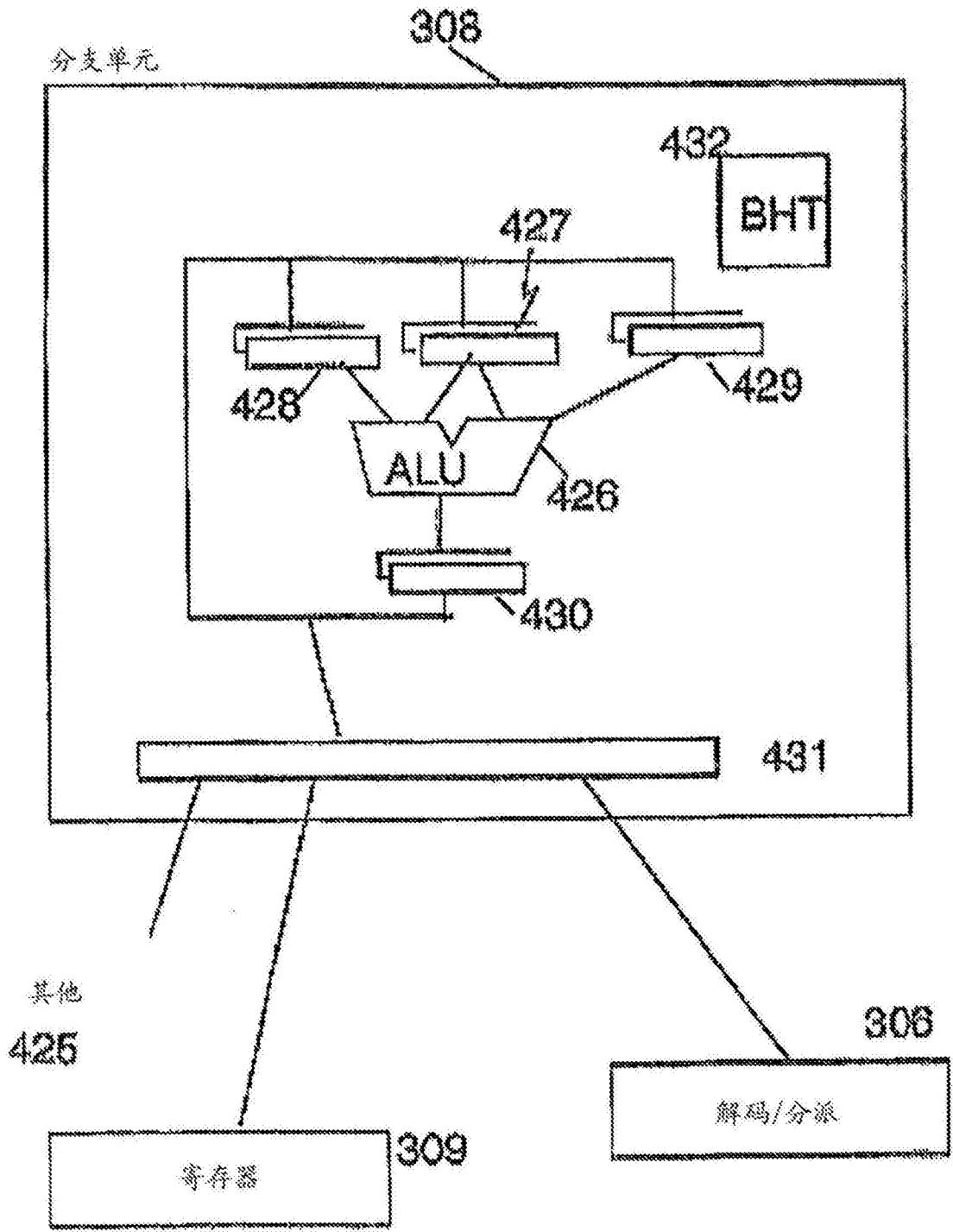


图4B

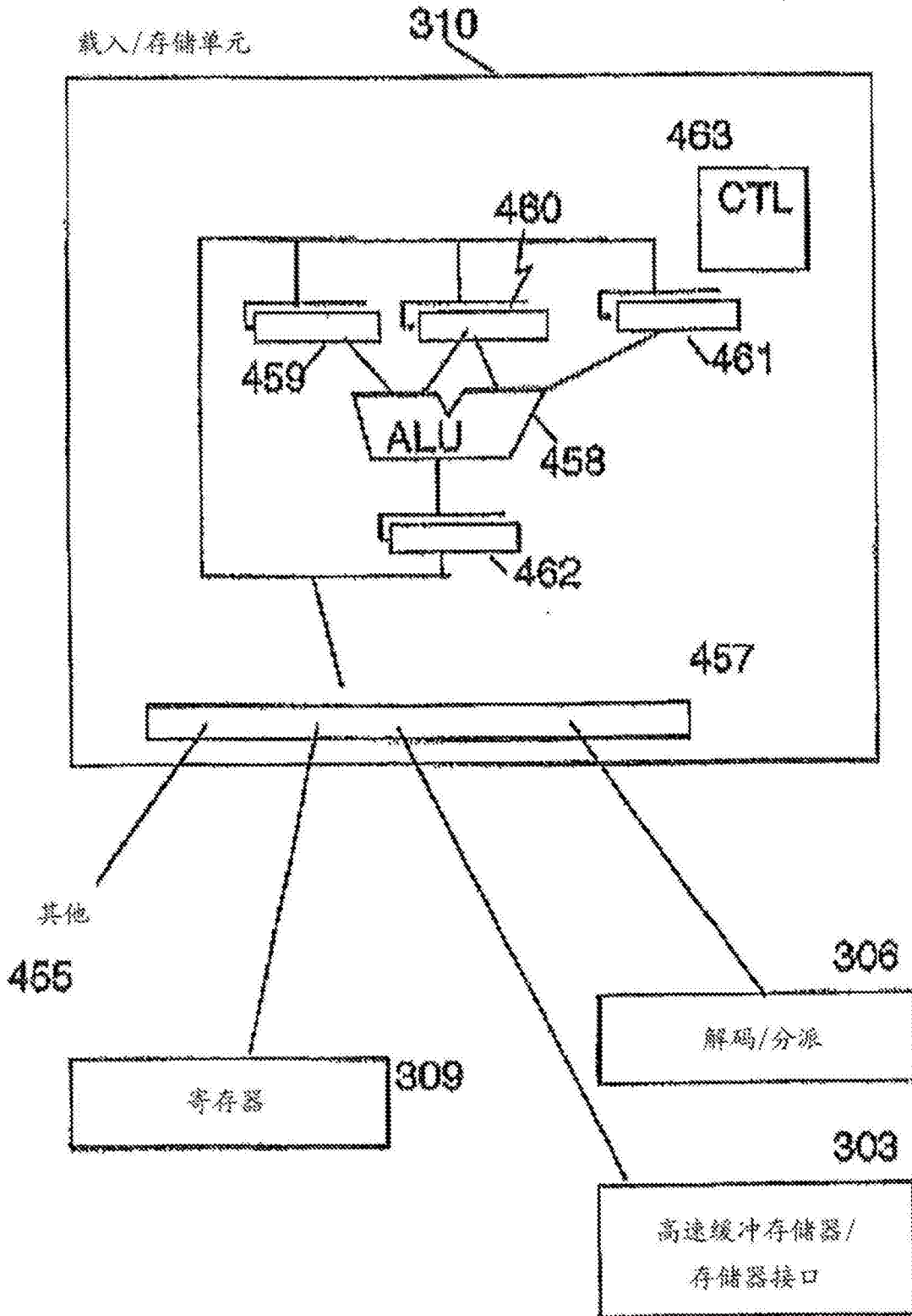


图4C

500

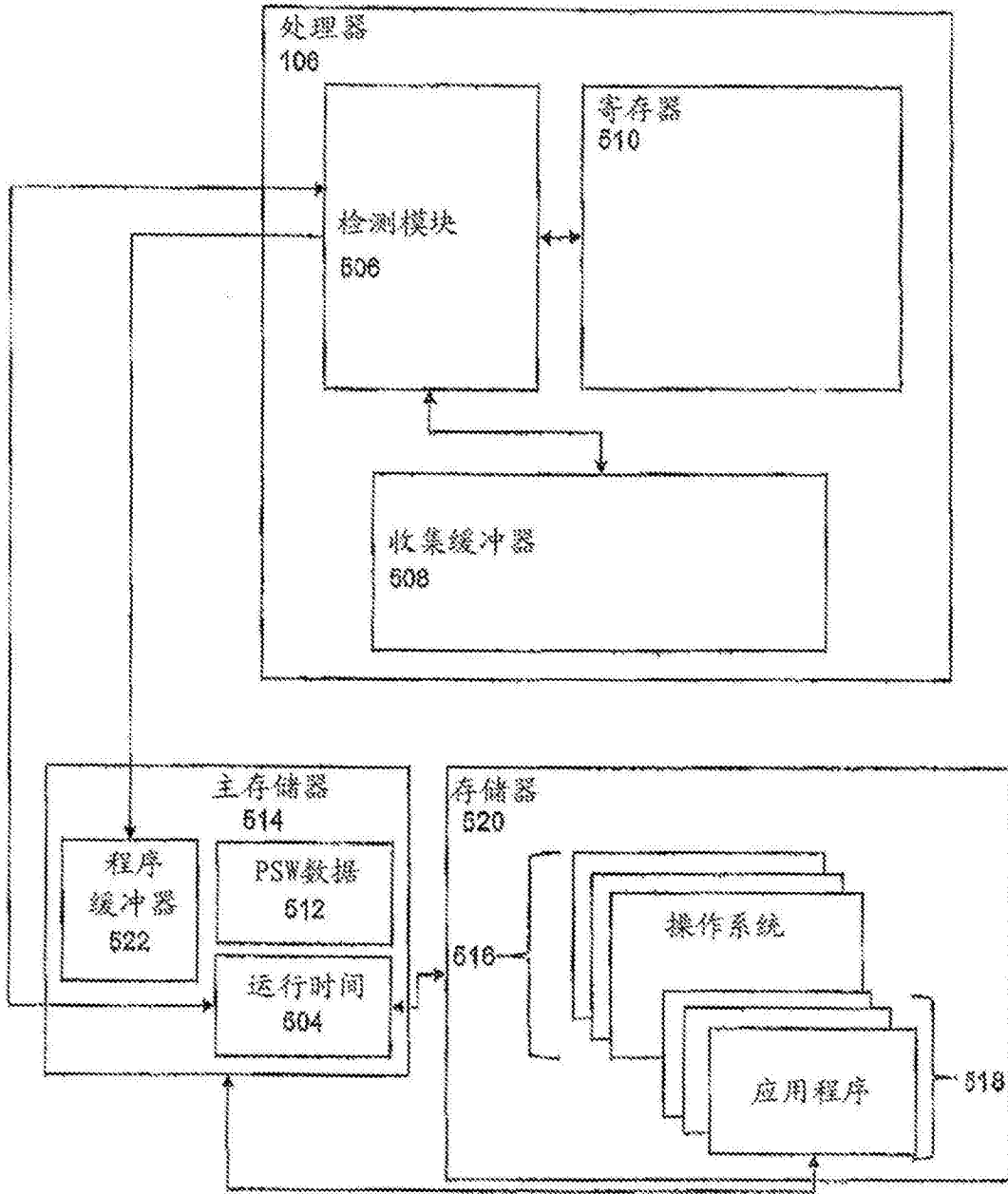


图5

600

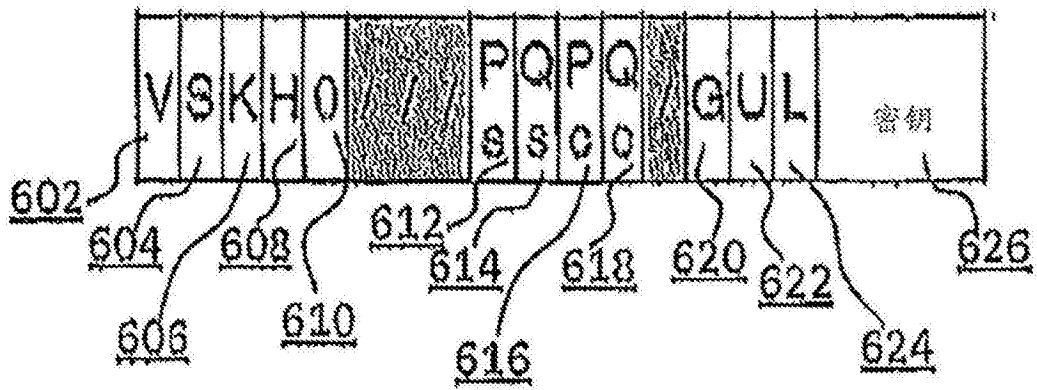


图6

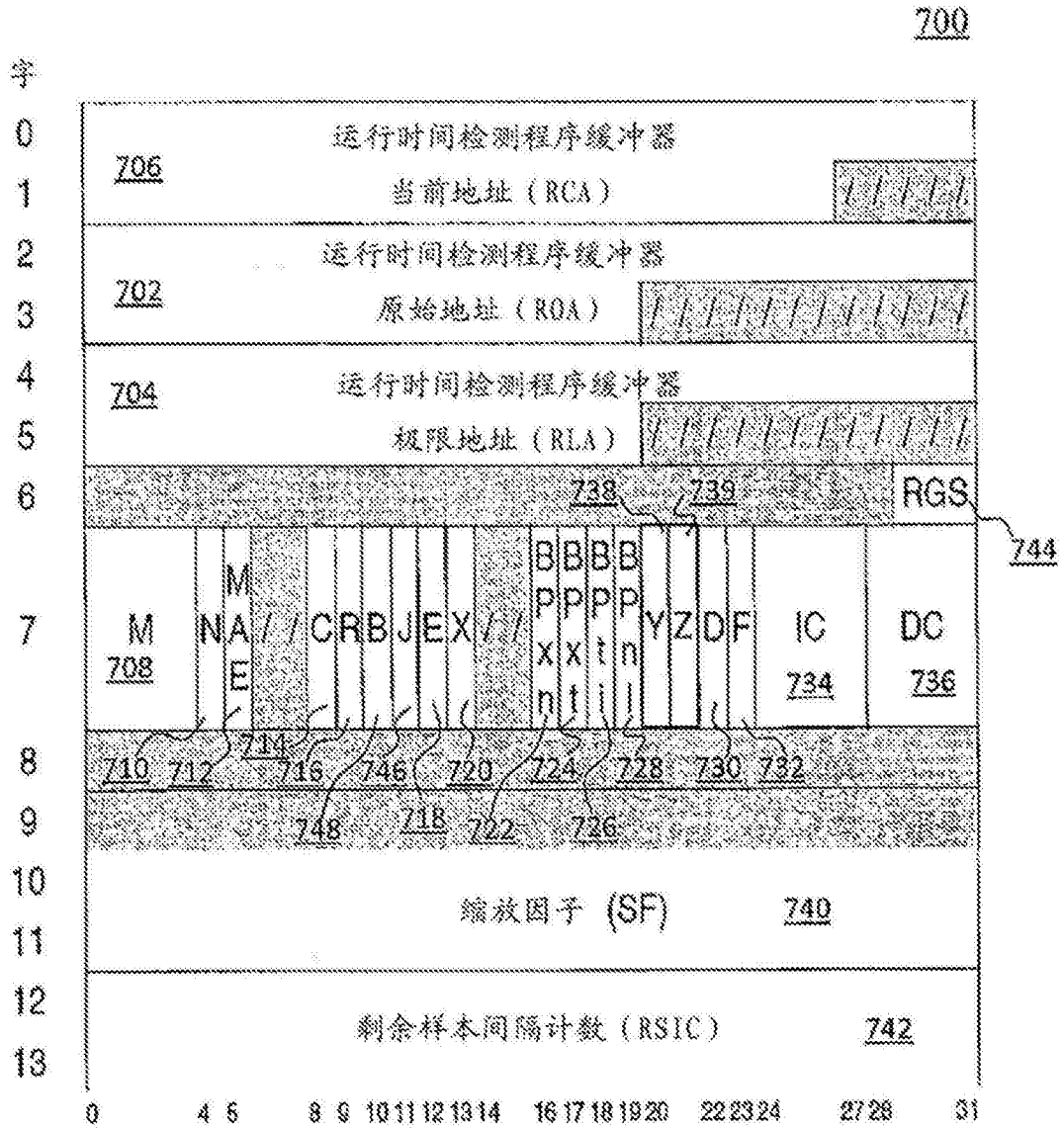


图7

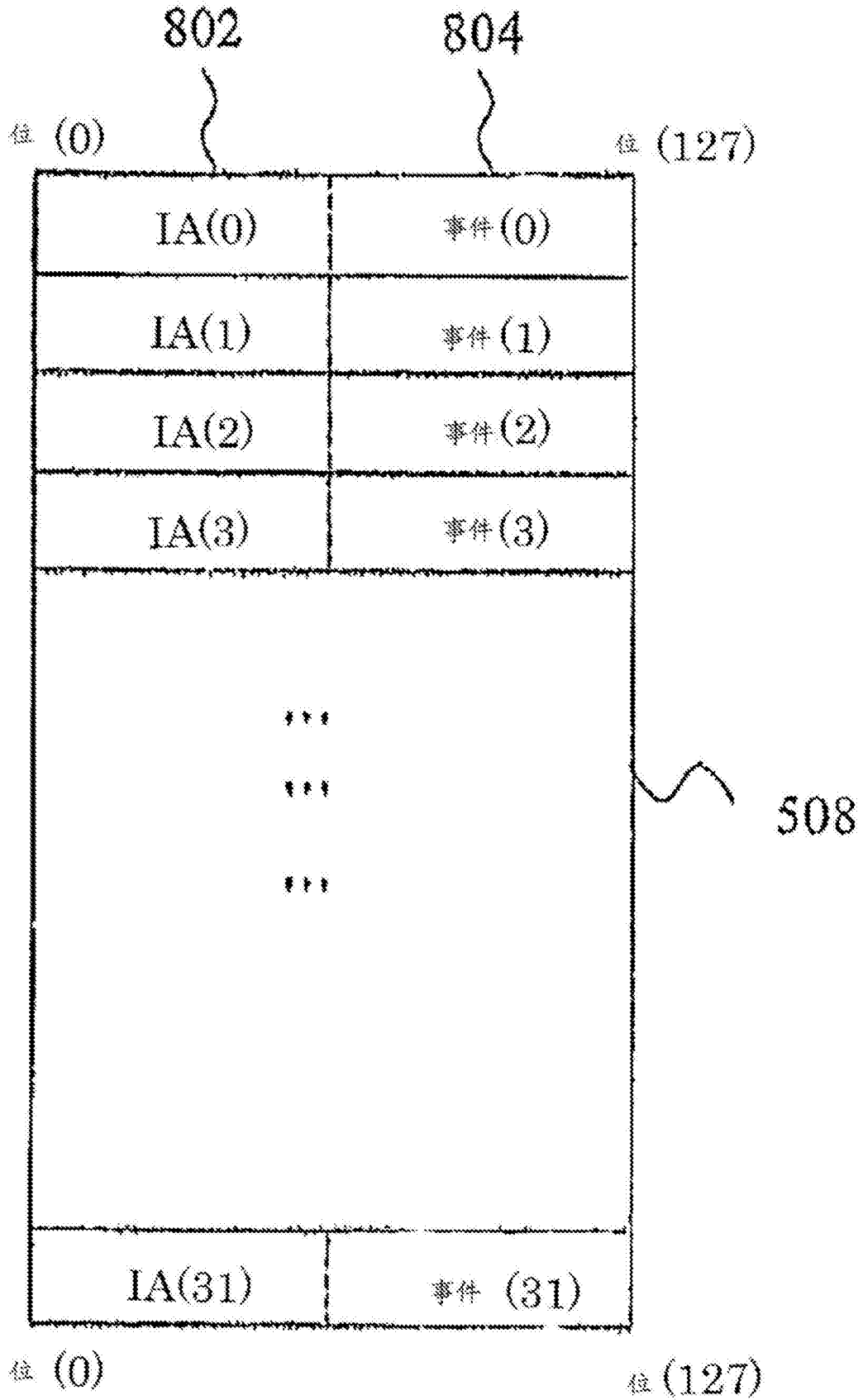


图8



900

记录号码	记录类型	区段
0	开始, 时间戳	报头 902
1	发出, TX中止, 呼叫, 返回, 分支, 填充	主体 904
2		
3		
$R_{RG}-R_{NC} (=4)$		
$R_{RG}-R_{NC}+1 (=5)$	额外, 依赖模型	额外 906
$R_{RG}-2 (=6)$		
$R_{RG}-1$	指令	页脚 908

图9

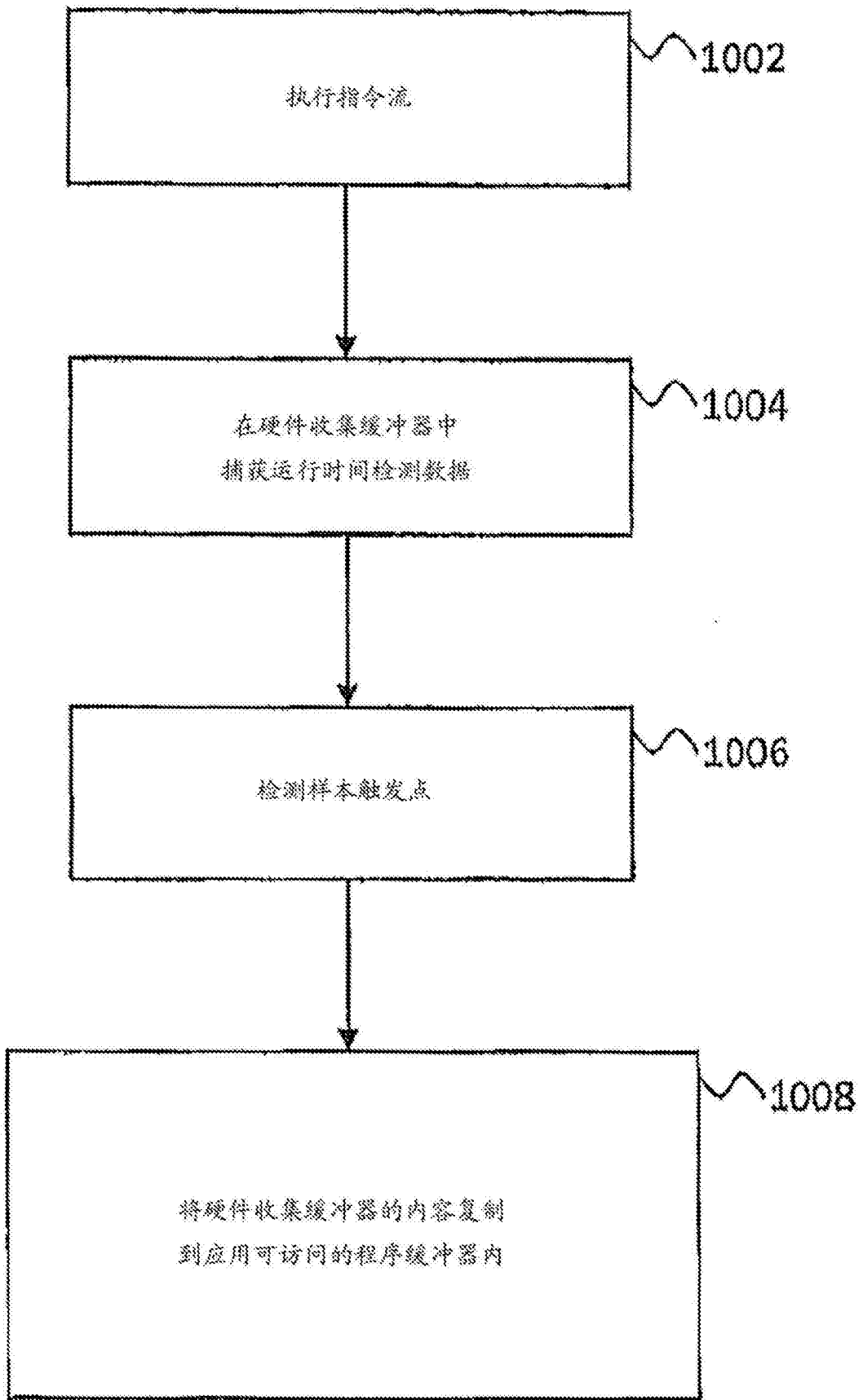


图10

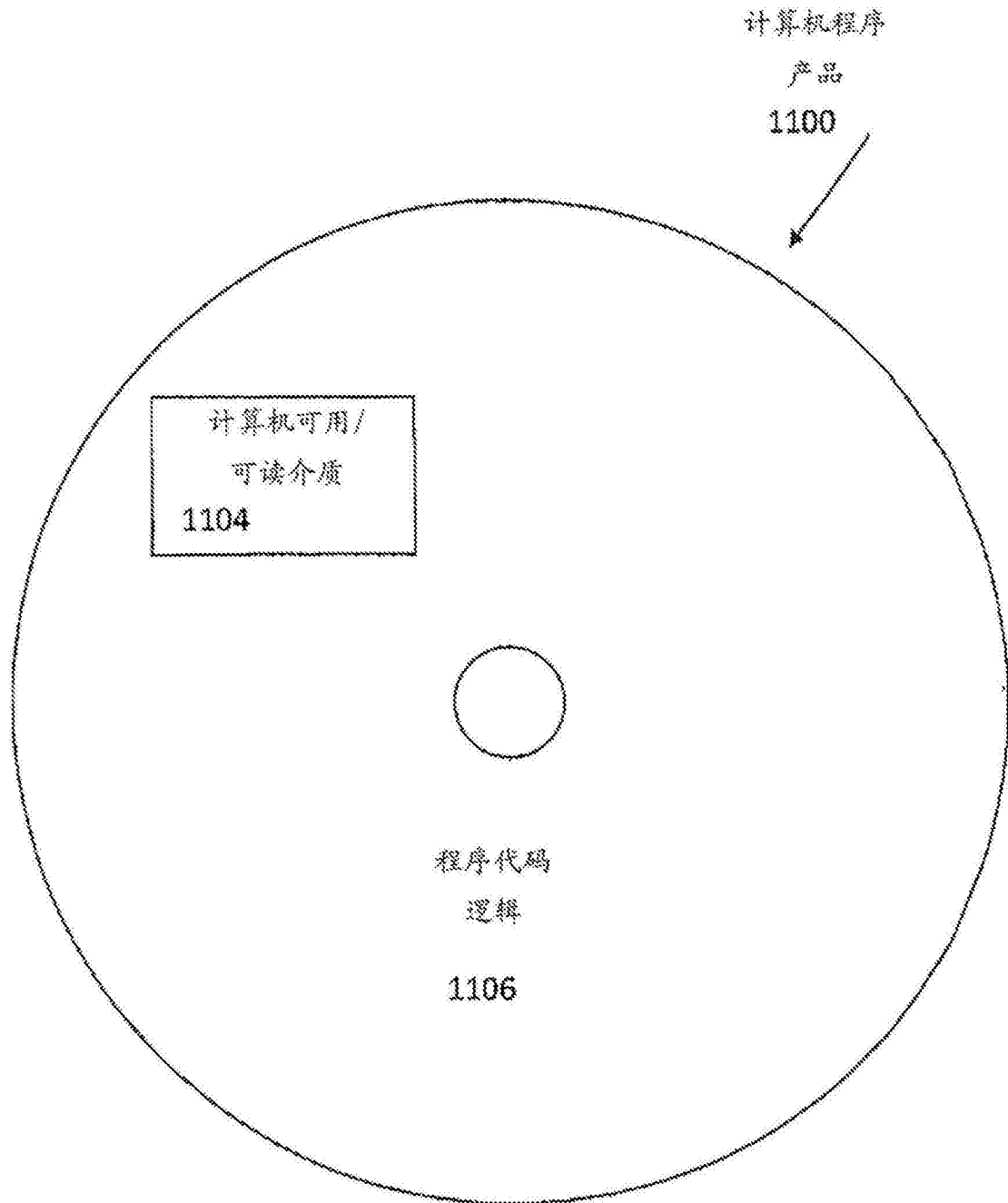


图11