



(12) 发明专利申请

(10) 申请公布号 CN 104054053 A

(43) 申请公布日 2014. 09. 17

(21) 申请号 201280067471. 3

(51) Int. Cl.

(22) 申请日 2012. 11. 13

G06F 9/46 (2006. 01)

(30) 优先权数据

13/352, 509 2012. 01. 18 US

(85) PCT国际申请进入国家阶段日

2014. 07. 18

(86) PCT国际申请的申请数据

PCT/IB2012/056371 2012. 11. 13

(87) PCT国际申请的公布数据

W02013/108091 EN 2013. 07. 25

(71) 申请人 国际商业机器公司

地址 美国纽约阿芒克

(72) 发明人 小查尔斯·盖尼 J. 马尔德

M. 法里尔 P. 伯纳德 D.W. 施米特

J.P. 库巴拉 R. 罗杰斯

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 黄剑飞

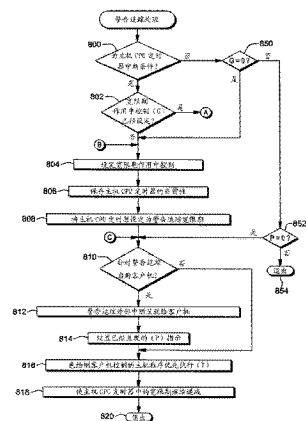
权利要求书2页 说明书25页 附图17页

(54) 发明名称

警告追踪中断设施

(57) 摘要

向程序(例如,操作系统)提供具有宽限期的警告,在该宽限期中,执行诸如清理的功能(例如,完成、停止和/或移动可分派单元)。在一个实例中,向该程序警告其正失去对其共享资源的存取。举例而言,在虚拟环境中,向客户机程序警告:其将失去其中央处理单元资源,且因此,其将执行诸如清理的功能。



1. 一种用于有助于促进计算环境中的处理的计算机程序产品,该计算机程序产品包含:

计算机可读储存介质,其可由处理电路读取且储存由该处理电路执行以用于执行一种方法的指令,该方法包含:

由第一程序将警告追踪设施已安装指示提供给第二程序,该指示指明警告追踪设施安装在该计算环境内,该警告追踪设施向该第二程序提供执行第一功能的宽限期;

由该第一程序将警告追踪通知提供给该第二程序;

基于该警告追踪通知,由该第二程序在该宽限期内启动该第一功能;及

在该宽限期之后,由该第一程序执行第二功能。

2. 根据权利要求 1 所述的计算机程序产品,其中该警告追踪通知包含中断,在该中断中,在该宽限期终止之后释放指派给该第二程序的共享资源。

3. 根据权利要求 2 所述的计算机程序产品,其中该第一程序得到对所释放的共享资源的存取以便执行该第二功能。

4. 根据权利要求 1 所述的计算机程序产品,其中该第一程序为主机程序且该第二程序为客户机程序,在提供给其上执行该客户机程序的客户机中央处理单元的时间片期间,该客户机程序存取该计算环境的共享资源,该宽限期有别于该时间片。

5. 根据权利要求 4 所述的计算机程序产品,其中该宽限期使该时间片过早地终止。

6. 根据权利要求 4 所述的计算机程序产品,其中除该时间片外,该宽限期亦提供执行该第一功能的时段。

7. 根据权利要求 6 所述的计算机程序产品,其中该宽限期取自提供给该第二程序的下一时间片。

8. 根据权利要求 1 所述的计算机程序产品,其中该第一功能包含以下之一:

完成在其上执行第二程序的处理器上执行的可分派单元;或

使该可分派单元可在该计算环境的另一处理器上重新分派。

9. 根据权利要求 1 所述的计算机程序产品,其中该方法进一步包含由该第二程序指示该第一功能在该宽限期期间的完成。

10. 根据权利要求 9 所述的计算机程序产品,其中该方法进一步包含在该第二程序下一次执行时,将该第一功能在该宽限期期间完成的指示提供给该第二程序。

11. 一种用于有助于促进计算环境中的处理的计算机系统,该计算机系统包含:

内存;及

与该内存通信的处理器,其中该计算机系统配置为执行一方法,该方法包含:

由第一程序将警告追踪设施已安装指示提供给第二程序,该指示指明警告追踪设施安装在该计算环境内,该警告追踪设施向该第二程序提供执行第一功能的宽限期;

由该第一程序将警告追踪通知提供给该第二程序;

基于该警告追踪通知,由该第二程序在该宽限期内启动该第一功能;及

在该宽限期之后,由该第一程序执行第二功能。

12. 根据权利要求 11 所述的计算机系统,其中该警告追踪通知包含中断,在该中断中,在该宽限期终止之后释放指派给该第二程序的共享资源。

13. 根据权利要求 11 所述的计算机系统,其中该第一程序为主机程序且该第二程序为

客户机程序,在提供给其上执行该客户机程序的客户机中央处理单元的一时间片期间,该客户机程序能够存取该计算环境的共享资源,该宽限期有别于该时间片。

14. 根据权利要求 13 所述的计算机系统,其中该宽限期使该时间片过早地终止。

15. 根据权利要求 13 所述的计算机系统,其中除该时间片外,该宽限期亦提供执行该第一功能的时段。

16. 根据权利要求 11 所述的计算机系统,其中该第一功能包含以下之一:

完成在其上执行第二程序的处理器上执行的可分派单元;或

使该可分派单元可在该计算环境的另一处理器上重新分派。

17. 根据权利要求 11 所述的计算机系统,其中该方法进一步包含:

由该第二程序指示该第一功能在该宽限期期间的完成;及

在该第二程序下一次执行时,将该第一功能在该宽限期期间完成的指示提供给该第二程序。

18. 一种有助于促进计算环境中的处理的方法,该方法包含:

由第一程序将警告追踪设施已安装指示提供给第二程序,该指示指明警告追踪设施安装在计算环境内,该警告追踪设施向该第二程序提供执行第一功能的宽限期;

由该第一程序将警告追踪通知提供给该第二程序;

基于该警告追踪通知,由该第二程序在该宽限期内启动该第一功能;及

在该宽限期之后,由该第一程序执行第二功能。

19. 根据权利要求 18 所述的方法,其中该第一程序为主机程序且该第二程序为客户机程序,在提供给其上执行该客户机程序的客户机中央处理单元的时间片期间,该客户机程序能够存取该计算环境的共享资源,该宽限期有别于该时间片。

20. 根据权利要求 18 所述的方法,其进一步包含:

由该第二程序指示该第一功能在该宽限期期间的完成;及

在该第二程序下一次执行时,将该第一功能在该宽限期期间完成的指示提供给该第二程序。

## 警告追踪中断设施

### 技术领域

[0001] 本发明的方面大体上涉及计算环境内的处理,且具体而言,涉及促进与共享资源相关联的处理。

### 背景技术

[0002] 资源共享的环境的一种类型为虚拟环境,该虚拟环境包括主机中央处理单元(CPU)及一个或多个客户机中央处理单元。由在主机 CPU 上执行的主机程序(例如,主机操作系统)来布建(provision)客户机 CPU(亦称为虚拟 CPU)。主机程序执行动作以从底层主机配置分配资源并将那些资源指派给客户机 CPU。

[0003] 在一个特定实施例中,当主机 CPU 进入解译执行模式时,存在客户机 CPU。在此点处,客户机操作系统(本文中亦称作客户机程序)开始虚拟化 CPU 上的执行,同时主机程序暂时中止主机 CPU 上的执行。在解译执行模式结束时,主机程序继续执行 CPU 上的执行。链接技术存在于主机与客户机之间,主机及客户机状态藉由这些连结技术而得以保存及恢复。通常,当主机程序启动客户机程序时,主机程序暂时中止,直至客户机程序返回为止。客户机 CPU 与主机 CPU 两者是一个主机 CPU 的不同模式。

[0004] 主机配置通常包括计算机系统的所有资源。这些资源包括(但不限于)中央处理单元(CPU)、主存储器及输入/输出(I/O)器件。在此系统中,多个客户机 CPU 可由单一主机 CPU 来支持。此情形藉由以下操作来实现:为每个客户机 CPU 指派称作时间片(timeslice)的一段时间以使用主机 CPU,以及接着将主 CPU 移动至另一客户机 CPU 达一时间片,等等。主机 CPU 可支持之客户机 CPU 的数目根据主机 CPU 之能力以及待指派给每个客户机 CPU 之所要能力而变化。

[0005] 客户机配置通常由两个或两个以上客户机 CPU 形成,且被称作客户机多处理(MP)配置。可藉由单独主机 CPU 之共享或甚至藉由共享单一主机 CPU 来布建每个客户机 CPU。此共享的一个属性在于:客户机 CPU 可操作达称作时间片的一段时间,且接着处于非作用中的达某一任意一段时间。非作用中时段基于以下各者而变化:由系统建立的优先权策略、将共享主机 CPU 的客户机 CPU 的总数,以及正使用的特定共享技术。

[0006] 在此客户机多处理系统中,有时称作可分派单元(DU)的程序可由客户机 CPU 上的客户机操作系统来分派,且接着在该可分派单元的执行期间,该可分派单元的主机时间片期满。此情形可使可分派单元处于以下状况下:使得可分派单元无法在客户机多处理配置的任何其他客户机 CPU 上继续,而不管任何其他客户机 CPU 的可用性。确切而言,可分派单元必须等待唯一客户机 CPU 以接收其下一时间片以便继续。基于共享技术以及客户机配置的相对优先权,可使下一时间片延迟达相当的一段时间。即使客户机配置具有能够执行可分派单元的其他客户机 CPU,可分派单元的继续也是不可能的(归因于在先前时间片期满时保存的可分派单元的客户机 CPU 的状态)。可分派单元系非作用中的,直至该精确状态可用以使客户机 CPU 继续为止。

## 发明内容

[0007] 经由提供一种用于有助于促进计算环境中的处理的计算机程序产品来克服先前技术的缺点且提供优点。该计算机程序产品包括计算机可读储存介质,该计算机可读储存介质可由处理电路读取且储存指令以供该处理电路执行以用于执行方法。该方法例如包括:由第一程序将警告追踪设施已安装指示提供给第二程序,该指示指明警告追踪设施安装在该计算环境内,该警告追踪设施将宽限期提供给该第二程序以执行第一功能;由该第一程序将警告追踪通知提供给该第二程序;基于该警告追踪通知,由该第二程序在该宽限期内启动该第一功能;以及由该第一程序在该宽限期之后执行第二功能。

[0008] 本文中亦描述且主张与本发明的一个或多个方面有关的方法以及系统。另外,本文中亦描述且可主张与本发明的一个或多个方面有关的服务。

[0009] 经由本发明的技术来实现额外特征以及优点。本发明的其他实施例以及方面在本文中加以详细描述且被视为所主张的本发明的一部分。

## 附图说明

[0010] 特别指出本发明的一个或多个方面,且在说明书完结时于申请专利范围中作为实例清楚地主张本发明的一个或多个方面。本发明的前述内容以及目标、特征以及优点自以下结合附图进行的详细描述而显而易见。

[0011] 图 1 描绘含有且使用本发明的一个或多个方面的计算环境的实施例;

[0012] 图 2 描绘含有且使用本发明的一个或多个方面的计算环境的另一实施例;

[0013] 图 3 描绘含有且使用本发明的一个或多个方面的计算环境的又一实施例;

[0014] 图 4 描绘根据本发明的方面的与警告追踪中断设施的客户机观测相关联的逻辑的实施例;

[0015] 图 5 描绘根据本发明的方面的与警告追踪中断设施的主机观测相关联的逻辑的实施例;

[0016] 图 6 描绘根据本发明的方面的与处置客户机退出的主机相关联的逻辑的实施例;

[0017] 图 7 描绘根据本发明的方面的警告追踪中断设施逻辑的概述的实施例;

[0018] 图 8A 至图 8C 描绘根据本发明的方面的与警告追踪中断设施处理相关联的逻辑的实施例;

[0019] 图 9 描绘根据本发明的方面的与接收警告追踪中断相关联的逻辑的实施例;

[0020] 图 10 描绘根据本发明的方面的所使用的诊断指令的格式的实施例;

[0021] 图 11 描绘含有本发明一个或多个方面所述的计算机程序产品的实施例;

[0022] 图 12 描绘含有且使用本发明的一个或多个方面的主计算机系统的实施例;

[0023] 图 13 描绘含有且使用本发明的一个或多个方面的计算机系统的另一个实例;

[0024] 图 14 描绘含有且使用本发明的一个或多个方面的包含计算机网络的计算机系统的另一个实例;

[0025] 图 15 描绘含有且使用本发明的一个或多个方面的计算机系统的各种组件的实施例;

[0026] 图 16A 描绘含有且使用本发明的一个或多个方面的图 15 的计算机系统的执行单元的实施例;

[0027] 图 16B 描绘含有且使用本发明的一个或多个方面的图 15 的计算机系统的分支单元的实施例；

[0028] 图 16C 描绘含有且使用本发明的一个或多个方面的图 15 的计算机系统的加载/储存单元的实施例；以及

[0029] 图 17 描绘含有且使用本发明的一个或多个方面的仿真主计算机系统的实施例。

### 具体实施方式

[0030] 根据本发明的方面,提供用于向程序(例如,操作系统)警告其具有执行功能的宽限期的能力。举例而言,对程序给予宽限期以执行清理(例如,完成、停止和/或移动可分派单元)。

[0031] 根据本发明的另一方面,向程序和/或处理器警告其将失去对资源(例如,共享资源)的存取。举例而言,将如下警告提供给与其他处理器共享资源的处理器:该处理器将失去对资源的存取。作为另一个实例,在共享处理器上执行的诸如操作系统的程序(亦即,程序与其他程序共享处理器)警告其将失去其处理器资源。

[0032] 在一个特定实施例中,提供如下能力:向在由主机 CPU 布建(provision)的客户机 CPU 上执行的客户机程序提供:关于自主机 CPU 给予客户机 CPU 的时间片的期满的警告;或关于主机对客户机的时间片的优先执行的警告。该警告提供宽限期,客户机 CPU 可使用该宽限期来执行特定功能,诸如完成可分派单元的执行、在可分派单元为可重新分派的点停止可分派单元和/或将可分派单元移动至另一客户机 CPU。

[0033] 如本文中所使用,作为实例,宽限期包括时间量、指令的数目、循环的数目等。宽限期具有可执行一个或多个功能的预定持续时间。

[0034] 参看图 1 来描述含有且使用本发明的一个或多个方面的计算环境的一个实施例。在此特定实施例中,计算环境 100 包括共享资源 104 的多个处理器 102。对每个处理器(和/或在处理器上执行的诸如操作系统的程序)给予称作时间片的某一时间量以共享资源。作为实例,资源包括中央处理单元资源、内存、输入/输出器件或接口,和/或其他资源。向能够存取资源的处理器(或在处理器上执行的程序)警告处理器的存取将结束,且因此,处理器(或程序)将执行特定动作,诸如清理、完成工作单元、停止工作单元、移动工作单元等。

[0035] 参看图 2 来描述含有且使用本发明的一个或多个方面的计算环境 200 的另一实施例。计算环境 200 系基于例如由 International Business Machines Corporation(IBM®)(Armonk, New York) 供应的 z/Architecture®。z/Architecture® 描述于 2010 年 8 月的题为“z/Architecture Principles of Operation”的 IBM® 公开案(IBM® 公开案第 SA22-7832-08 号第九版本)中,该案藉此以全文引用的方式并入本文中。在一个实例中,基于 z/Architecture® 的计算环境包括由 International Business Machines Corporation(Armonk, New York) 供应的 System z® 服务器。IBM®、z/Architecture® 以及 zSeries®;以及下文提及的 z/VM® 以及 z/OS® 为 International Business Machines

Corporation (Armonk, New York, USA) 的注册商标。本文中所使用的其他名称可为 International Business Machines Corporation 或其他公司的注册商标、商标或产品名称。

[0036] 作为一个实例,计算环境 200 包括提供虚拟机支持的中央处理器复合体 (central processor complex, CPC) 202。CPC202 例如包括一个或多个虚拟机 204 (或在另一实施例中,逻辑分割区)、一个或多个中央处理器 206、至少一个主机 208 (例如,诸如超管理器的控制程序) 以及输入 / 输出子系统 210,下文描述前述各者中的每个者。在此实例中,虚拟机以及主机包括于内存中。

[0037] 作为实例,CPC 的虚拟机支持提供操作大量虚拟机的能力,这些虚拟机各自能够主控诸如 z/VM<sup>®</sup>、z/OS<sup>®</sup> 或 Linux 的客户机操作系统 212。每个虚拟机 204 能够作为单独系统起作用。亦即,每个虚拟机可经独立地重设,主控客户机操作系统,并以不同程序操作。在虚拟机中执行的操作系统或应用程序显现为能够存取完全且完整系统,但实际上,仅系统的一部分为可用的。

[0038] CPC 的实体资源 (例如,CPU、内存、I/O 器件等) 由主机 208 拥有,且共享实体资源由主机按需要分派给客户机操作系统以满足客户机操作系统的处理需求。客户机操作系统与实体共享机器资源之间的互动由主机控制,此系由于大量客户机通常阻止主机简单地分割硬件资源并将硬件资源指派给所配置的客户机。

[0039] 中央处理器 206 为可指派给虚拟机的实体处理器资源。举例而言,虚拟机 204 包括一个或多个逻辑处理器,这些逻辑处理器中的每个者表示可经动态地分配给虚拟机的实体处理器资源 206 的全部或共享。虚拟机 204 由主机 208 管理。作为实例,主机可以在处理器 206 上执行的微码来实施,或可为在机器上执行的主机操作系统的一部分。在一个实例中,主机 208 为由 International Business Machines Corporation (Armonk, New York) 供应的处理器资源 / 系统管理器 (PR/SM)。

[0040] 输入 / 输出子系统 210 指导在器件与主存储器之间的信息的流动。输入 / 输出子系统 210 耦接至中央处理复合体,就此而言,输入 / 输出子系统 210 可为中央处理复合体的部分或与中央处理复合体分离。I/O 子系统减轻中央处理器关于与耦接至 CPC 的 I/O 器件直接通信的任务,且准许数据处理与 I/O 处理同时进行。

[0041] 在一个实施例中,主机 (例如,PR/SM) 以及处理器 (例如, System z<sup>®</sup>) 硬件 / 固件以受控合作方式彼此互动,以便处理客户机操作系统操作而不需要自 / 至客户机操作系统以及主机的控制的传送。可在无主机干预情况下经由允许针对客户机解译地执行指令的设施直接执行客户机操作。此设施提供主机可发出的指令“启动解译执行 (SIE)”,该指令指明保持客户机 (虚拟机) 状态以及控制的称作状态描述的控制区块。该指令将 CPU 置于直接处理客户机指令以及中断的解译执行模式,直至出现需要主机关注的条件为止。当发生此条件时,解译执行结束,且呈现主机中断,或 SIE 指令完成储存所遭遇的条件细节;此后一个动作被称作拦截。解译执行的一个实例描述于 1985 年 9 月的“System/370Extended Architecture/Interpretive Execution”(IBM 公开案第 SA22-7095-01 号) 中,该案藉此以全文引用的方式并入本文中。

[0042] 如本文中所使用,固件例如包括处理器的微码、毫码 (millicode) 和 / 或宏码。固

件例如包括用于较高级机器码的实施方案中的硬件层级的指令和 / 或数据结构。在一个实施例中, 固件包括例如专属码, 专属码通常作为包括信赖软件的微码或基础硬件特定的微码而递送且控制对系统硬件的操作系统存取。

[0043] 在图 3 中描绘含有本发明的一个或多个方面的计算环境的另一个实例。在此实例中, 提供仿真主计算机系统 300, 仿真主计算机系统 300 仿真主机架构的主计算机 302。在仿真主计算机系统 300 中, 主机处理器 (CPU) 304 为仿真主机处理器 (或虚拟主机处理器), 且经由具有不同于供主计算机 302 的处理器使用的原生指令集架构的原生指令集架构的仿真处理器 306 来实现。仿真主计算机系统 300 具有仿真处理器 306 可存取的内存 308。在实例实施例中, 内存 308 经分割成主计算机内存部分 310 以及仿真例程部分 312。主计算机内存 310 可用于根据主计算机架构的仿真主计算机 302 的程序, 且可包括主机或超管理器 314 与执行客户机操作系统 318 的一个或多个虚拟机 316 两者 (类似于图 2 中的类似命名的组件)。

[0044] 模拟处理器 306 执行不同于模拟处理器 304 之架构的架构之经建构指令集的原生指令。例如, 自仿真例程内存 312 获得原生指令。模拟处理器 306 可藉由使用在序列与存取 / 解码例程中获得之一或多个指令来自主计算机内存 310 中的程序存取用于执行之主机指令, 该序列与存取 / 解码例程可解码所存取之主机指令以判定用于仿真所存取之主机指令之功能的原生指令执行例程。一个主机指令可为, 例如, 启动解译执行 (SIE) 指令, 主机藉由该指令而设法在虚拟机中执行客户机程序。仿真例程 312 可包括对此指令之支持, 且用于根据此 SIE 指令之定义执行虚拟机 316 中之一序列客户机指令。

[0045] 作为实例, 针对主计算机系统 302 之架构定义之其他设施可藉由经建构之设施例程来仿真, 包括诸如通用寄存器、控制寄存器、动态地址转译以及 I/O 子系统支持以及处理器高速缓存的设施。仿真例程亦可利用仿真处理器 306 中可用之功能 (诸如, 通用寄存器以及虚拟地址之动态转译) 以改良仿真例程的效能。亦可提供专用硬件以及卸除引擎以协处理器 306 仿真主计算机 302 之功能。

[0046] 根据本发明的方面, 提供可用于许多类型的计算环境中的警告追踪中断设施。尽管警告追踪中断设施可用于许多类型之环境中, 但本文中参看客户机多处理系统来描述设施的方面。如上文所描述, 在客户机多处理系统中, 客户机操作系统在客户机中央处理单元上分派可分派单元 (例如, 程序、程序代码等), 这些客户机中央处理单元由至少一个主机中央处理单元来主控。主机 CPU 为客户机 CPU 提供一时间片 (例如, 时间量或其他时段, 诸如指令之数目、循环之数目等), 可分派单元在该时间片期间执行。若在可分派单元之执行期间, 时间片期满, 则可使得可分派单元处于以下状况: 使得其无法在客户机多处理配置的任何其他客户机 CPU 上继续, 而不管任何其他客户机 CPU 的可用性。确切而言, 可分派单元必须等待唯一的客户机 CPU 来接收其下一时间片以便继续。基于所使用之特定共享技术以及客户机配置之相对优先权, 可使下一时间片延迟达相当的一段时间。即使客户机配置具有能够执行可分派单元的其他客户机 CPU, 可分派单元的继续也是不可能的 (归因于在先前时间片期满时保存的可分派单元之客户机 CPU 的状态)。可分派单元系非作用中的, 直至该精确状态可用以继续客户机 CPU 为止。

[0047] 有可能藉由授予额外时间 (或其他额外时段, 诸如额外指令、循环等) 来延长时间片, 但即使藉由此额外时间, 客户机 CPU 可延长可分派单元之执行, 而仍使可分派单元处于



相同的不可分派状况,此系由于将处于正常时间片的期满时间。

[0048] 由于主机程序并不知晓供执行任意可分派单元之任意客户机程序使用的控制以及状态,因此无法在不建立主机程序与客户机程序之间的协议之情况下进行始终对客户机操作系统授予额外时间以清理其可分派单元。在无协议之情况下,将在主处理中消耗对客户机 CPU 授予之任何额外时间,且仍有可能以相同的可分派单元卡塞状况结束。因此,根据本发明的方面,提供此协议。

[0049] 根据本发明的方面,提供包括警告之宽限期或延长时间,该警告向客户机程序指示应采取特定动作(例如,完成可分派单元或使得可分派单元为可重新分派的)。作为实例,响应于时间片之期满或在时间片期满之前响应于主机优先执行客户机而提供宽限期,以便回收处理器以用于如主机可见之某一其他较高优先权,如本文中进一步详细描述。

[0050] 作为一个实例,提供宽限期而非无条件地延长时间片。若正常时间片已完全期满,则提供宽限期,但冲销下一正常时间片,以便对各自具有时间片期望值之所有其他虚拟化客户机维持公平。若正常时间片尚未期满,则宽限期取自剩余正常时间。总之,宽限期限制给予客户机 CPU 之剩余时间(或其他时段),且自身为不可延长的。因此,客户机 CPU 无法继续执行达任意且未知时段。

[0051] 在宽限期的开始,出于清理可分派单元(例如,完成可分派单元、停止可分派单元和/或移动可分派单元)的目的而通知客户机程序。宽限期之强制执行确保:客户机 CPU 不会超出所授予的额外时段。藉以对客户机程序给予宽限期且向客户机程序通知时间(或其他时段)几乎期满(宽限期已开始)的协议系客户机程序与主机程序之间的协议,客户机程序理解该协议,藉此使得此通知为值得的。亦即,客户机程序通常将藉由在必要时使得当前可分派单元在客户机配置的另一客户机 CPU 上为可分派的(例如,移动可分派单元)来履行(honor)该通知。

[0052] 下文参看图 4 至图 10 来描述关于协定以及警告追踪中断设施(亦称作警告追踪或警告追踪设施)的其他细节。参看那些图所描述之实施例涉及具有由一个或多个主机布建之一个或多个客户机的虚拟环境。然而,本发明的一个或多个方面亦涉及其他环境,包括多个处理器和/或多个程序共享资源的非虚拟环境。

[0053] 参看图 4,描述关于如由客户机观测到之警告追踪中断设施协议的细节。如由客户机观测到的警告追踪协议包括,例如,设施已安装指示、客户机注册、通知以及主动(voluntarily)退出,在下文中描述前述各者中之每个者。

[0054] 参看图 4,客户机程序理解警告追踪协议且搜寻设施已安装指示(步骤 400)。在一个实例中,此指示为储存于控制区块(例如,“服务调用控制区块(SCCB)”)中的位,该位系使用诸如“读取 SCP 信息”命令之读取命令来观测。在判定设施已安装之后,客户机程序执行注册(步骤 402)。注册系客户机程序将关于客户机程序理解警告追踪中断设施协议的指示传达至主机程序的机制。在一个实例中,使用诊断指令来执行注册,下文进一步描述诊断指令之实例。

[0055] 在一个实施例中,自任何客户机中央处理单元启动的注册涵盖多处理配置之所有客户机中央处理单元,此系由于需要跨越客户机多处理配置之客户机 CPU 的一致行为。在客户机多处理配置中,客户机 CPU 使用相同主存储器,且假定客户机 CPU 在有时被称作单一影像的影像中操作。因此,一客户机 CPU 的注册适用于多处理环境之其他客户机 CPU。在一

个实施例中,注册为不可撤销的,且此情形有助于避免计时窗,简化开发且允许改良之可测试性。即使注册为不可撤销的,客户机程序亦可判定客户机程序是否将继续参与协议。若客户机程序如此选择,则藉由重设或已重设下文所描述之一个或多个启用指示符,客户机程序不需要参与。

[0056] 在针对警告追踪协议注册客户机程序之后,可向客户机 CPU 通知宽限期(步骤 404)。举例而言,主机 CPU 可向客户机 CPU 警告时间片之期满(或在另一个实例中,迫近期满)或其时间片之优先执行。在一特定实例中,向已注册客户机配置之 CPU 通知,例如,其正常时间片之期满,以及提供额外时段以,例如,进行清理之宽限期的开始。

[0057] 在被通知之后,客户机具有受约束时间量或其他时段、宽限期(例如,在一个特定实施例中,为 50 微秒),以使得可分派单元为可重新分派的或进行任何其他适当调整。若正常时间片已结束,则,例如,在将控制返回至主机之前,使用宽限期以使得可分派单元为可重新分派的或进行任何其他适当调整。若时间片尚未结束,则使用宽限期,且放弃时间片之任何剩余部分。发生供客户机 CPU 使用之实际时间的正常账户处理。

[0058] 在作出通知之后,客户机 CPU 处于受约束时段(例如,有限时间量),在该时段之后,非主动结束客户机 CPU 之操作。针对每个正常时间片时段仅作出一个通知。因此,客户机 CPU 仍受确保可在其他处共享基础共享主机 CPU 的最终时间控制约束,藉此在藉由主机程序提供之总体虚拟化中保留良好次序以及规则。

[0059] 可藉由使得客户机程序侦测到唯一状态之任何机制来实现通知。实例包括唯一客户机中断、可经设定之架构上定义之主存储器位置,或可用于主机与客户机两者之 I/O 外部内存器件。前述三者中之第一者需要适当客户机启用以允许中断。后两者需要不浪费宽限期之足够频繁的时段性检验。在 z/Architecture<sup>®</sup>之一特定实例中,称作警告追踪中断(WTI)之客户机中断用作通知。

[0060] 在被警告之后,客户机主动地退出其给定时间片/宽限期(步骤 406)。在被通知且使得可分派单元为可重新分派的(例如,停止可分派单元以及移动可分派单元,或完成可分派单元)之后,客户机结束当前时间片/宽限期。此退出用信号向主机程序通知:客户机实际上遵循协议。可发生客户机放弃控制因此返回至主机程序之其他原因。通常,对于使得可分派单元为可重新分派的之受约束处理,将不会发生针对任何这些外来退出之状况。若客户机 CPU 经由警告追踪中断协议在宽限期内退出,则无论何时发生该状况,均给予关于下一时间片之反馈指示。因此,客户机程序知晓其满足由宽限期强加之时间约束。

[0061] 若客户机迟延主动退出,则藉由宽限期之期满而优先执行该客户机之执行。下次以正常时间片启动客户机 CPU 时,给予反馈指示,使得客户机知晓其系迟延的。大体上,此情形可用于客户机程序中之问题判定,此系因为惯常的宽限期允许足够时间来清理以及主动退出。

[0062] 若发生外来退出,则下次在正常时间片内启动客户机 CPU 时,期望将快速地发生主动退出。同一反馈机制将向客户机程序告知:外来退出已被干预且因此提供不同信息以告知问题判定。

[0063] 主动退出是藉由使得控制自客户机程序经由主机程序传递的任何机制来实现,且该机制包括前述反馈机制。所使用之机制将定义于特定架构之警告追踪中断协议中,以便供主机程序辨识客户机请求。在一个实例中,此机制包括下文描述之诊断指令。

[0064] 除警告追踪中断协议之客户机观测外,在一实施例中,主机亦观测设施,如下文参看图 5 进一步详细描述。

[0065] 参看图 5,主机辨识设施已安装指示并将该指示反映至其客户机(步骤 500)。举例而言,主机检查控制区块(例如,SCCB)中之已安装位,且辨识警告追踪协议之已安装状态(亦即,其经设定)并知晓主机程序可如何使用该状态而对主机有利。因此,将设施指示反映至其客户机。举例而言,为了将设施反映至客户机,主机设定客户机控制区块(例如,客户机 SCCB)或客户机可存取之内存区域中之已安装位。若由于任何原因,主机程序并不希望客户机观测警告追踪中断设施协议之已安装状态且不允许客户机观测警告追踪中断设施协议之已安装状态并使用该状态,则主机程序将未经安装指示传递至客户机(例如,将客户机可见之位设定为零)。此外,在一实施例中,主机程序设定客户机 CPU 之控制,使得警告追踪协议经停用(例如,关断客户机 CPU 之状态描述中的一个或多个指明位)。

[0066] 随着客户机启动注册,主机程序接收到未经请求之注册请求,且记住客户机已注册(步骤 502)。由任何单一客户机 CPU 启动之注册请求足以注册客户机多处理配置中之所有客户机 CPU。因此,主机程序针对客户机配置之所有 CPU 启用警告追踪协议(步骤 504)。举例而言,主机程序设定客户机 CPU 之状态描述中的一个或多个指明位以针对客户机启用警告追踪中断设施。不必将注册之反馈传回至客户机。倘若即使在设施未经安装时客户机 CPU 仍尝试注册,主机将忽略请求且针对警告追踪中断设施协议将不启用客户机 CPU。

[0067] 在针对警告追踪设施注册以及启用客户机之后,客户机可接收关于协议之调用的通知(步骤 506)。此情形可在若干个情境中实现,如下文所描述。

[0068] 作为一个实例,当针对在主机 CPU(亦即,主机 CPU X)上以解译执行模式操作之客户机 CPU 启用警告追踪中断协议时,主机程序可启动来自主机 CPU Y 的协议。亦即,已由主机 CPU X 布建客户机 CPU,且主机 CPU X 目前不可用于主机程序。若主机程序不得不重新获取 CPU X,则主机程序首先使得 CPU X 退出解译执行模式。亦即,使客户机 CPU 停止,藉此退出 CPU X 之解译执行模式。在任何的任意点使客户机 CPU 停止而不允许客户机 CPU 自身主动地停止产生警告追踪中断协议意欲解决问题之风险。警告追踪中断协议藉由允许将主机程序动作变换成客户机 CPU X 中之通知而允许主机 CPU Y 请求通知(步骤 506)。由于客户机程序先前已注册,因此主机程序期望:客户机程序辨识通知且支持通知之包括主动结束执行之最后步骤的适当处置,因此使主机 CPU X 之控制返回至主机程序。一旦发生此情形,主机程序便可继续进行,而不管主机 CPU X 之使用可能已使得程序经启动。

[0069] 主机程序藉由,例如,设定任何状态、设定指示符(例如,位)或使得将未经请求的异步信号(例如,警告追踪中断)发送至客户机来通知客户机。尽管已注册,但此通知信号之接收的时间在客户机中仍为未知的。藉由注册,倘若用信号通知,客户机便已恰好同意遵守协议。

[0070] 在主机单处理器系统中,若一个主机 CPU 处于解译执行模式从而使得客户机 CPU 正操作,则不存在其他主机 CPU 来调用警告追踪中断协议。然而,即使在此状况下,当主机 CPU 在处于解译执行模式的同时辨识时间片期满时,CPU 自身仍可调用警告追踪中断协议,且可接着授予宽限期并执行通知。

[0071] 在通知之另一个实例中,当主机 CPU 使得将定义于警告追踪中断协议中之通知发送至客户机 CPU 时(归因于当主机 CPU 处于解译执行模式时,主机 CPU 辨识之内部状态改

变),发生通知。当针对警告追踪中断协议启用客户机 CPU 且主机 CPU 辨识时间片之结束时,发生此情形之实例。在用信号向客户机 CPU 通知以放弃控制之前,CPU 在内部授予宽限期以允许客户机 CPU 具有足够时间以接收信号、采取适当动作(例如,完成当前可分派单元或使得当前可分派单元为可重新分派的)且主动地结束。在内部,主机 CPU 保持状态以指示:已向客户机 CPU 通知。若客户机在宽限期内并未主动地结束,则 CPU 辨识此情形且结束客户机的执行,藉此藉由结束解译执行模式而使控制返回至主机程序。在一实施例中,客户机无法判定为何调用协议,而是仅判定其被通知清理并结束。可存在用于引起解译执行模式之结束且因此结束客户机执行的其他主机理论基础。举例而言,存在如下情境:将使所有客户机 CPU 之执行停止以便对整个客户机配置作出某一协调改变。多处理客户机配置并不具有在与客户机配置之其他 CPU 不同的规则或假定下操作的一些 CPU。此不对称性可产生不可预测之客户机结果。

[0072] 主机 CPU 接收已执行主动退出或任何其他原因之退出的客户机 CPU 之效应(例如,向主机传回 CPU 资源)(步骤 508)。若退出系归因于警告追踪中断协议定义之原因,则主机程序记住在下次启动客户机 CPU 时向客户机 CPU 提供反馈(不管可能为多久)。此反馈为肯定(“良好”)指示,假定客户机在宽限期期满之前主动退出。若退出系归因于任何其他原因,则在客户机 CPU 之下次启动时不发生警告追踪中断协议反馈。

[0073] 若客户机迟延主动退出;亦即,客户机执行动作以主动退出,但宽限期已期满,则藉由宽限期之期满而优先执行客户机 CPU 之执行。下次以正常时间片启动客户机 CPU 时,给予异常反馈指示,使得客户机知晓其已迟延其主动退出。大体上,此情形可用于客户机程序中之问题判定,此系因为惯常的宽限期允许足够时间来清理以及主动退出。

[0074] 若发生不同于警告追踪中断协议之主动退出的退出,则下次以正常时间片启动客户机 CPU 时,不包括按警告追踪协议之反馈。

[0075] 主动地退出的手段是藉由使得控制自客户机程序传递至主机程序之任何机制来实现,该机制由主机程序辨识为协议主动退出,且该机制包括上述反馈机制。在一个实例中,诊断指令用于主动退出。亦即,具有特定参数之诊断指令用以指示时间片之完成。在诊断指令由客户机程序发出并经执行之后,主机程序判定退出是否为准时的。接着,当再次启动客户机时(其为诊断之后的下一循序指令),提供指示再次启动客户机是否为准时的之条件代码。条件代码,例如,设定于用以在下一循序指令处启动客户机之客户机程序状态字(PSW)中。客户机可接着测试条件代码。

[0076] 参看图 6 进一步描述由主机进行之对于客户机主动退出的处置。最初,当客户机 CPU 停止时,控制返回至主机 CPU(步骤 600)。作出关于控制是否在宽限期内返回的判定(查询 602)。若控制在宽限期内返回,则主机程序按警告追踪中断协议观测主动客户机退出,且记住针对客户机 CPU 之下次启动之良好反馈,而不管哪一主机 CPU 可能在该时间布建客户机 CPU(步骤 604)。此系假定已安装警告追踪中断设施。若控制并未在宽限期内返回,则不记住反馈状态。然而,若客户机执行动作以主动地退出,但主动退出系在宽限期外(查询 602),则主机 CPU 上之主机程序按警告追踪中断协议观测主动客户机退出(即使客户机最终且不得不非主动地退出),且记住针对客户机 CPU 之下次启动的不良反馈,而不管哪一主机 CPU 可能在该时间布建客户机 CPU(步骤 606)。再者,此系假定已安装警告追踪设施。若否,则不记住反馈状态。

[0077] 此后,不管主机记住良好反馈抑或不良反馈,主机程序将主机 CPU 重新导向至优先执行指派(步骤 608)。亦即,将主机重新导向为现在执行使其资源(CPU)返回的一个或多个功能。

[0078] 此外,在客户机 CPU 之下次循序启动时,不管哪一主机 CPU 布建客户机 CPU,若记住反馈状态,则在启动客户机 CPU 之前设定反馈状态指示(步骤 610)。在一个实例中,在 SIE 状态描述中(例如,在状态描述之 PSW 中)设定反馈状态指示,SIE 状态描述指示下一循序指令的启动。

[0079] 参看图 7 至图 9 来描述关于与警告追踪中断设施相关联之处理的其他细节。详言之,图 7 描绘与警告追踪中断设施处理之概述相关联之逻辑的一实施例;图 8A 至图 8C 提供根据本发明的方面的警告追踪中断设施处理之细节;且图 9 描绘与接收警告追踪中断相关联之逻辑的一实施例。

[0080] 参看图 7,最初客户机程序(例如,客户机操作系统)辨识已安装警告追踪中断设施(步骤 700)。在一实施例中,此情形是藉由客户机程序观测位于,例如,指明控制区块中之已安装设施指示符,例如位,来实现。若客户机操作系统具有支持以参与警告追踪中断设施,则客户机操作系统辨识警告追踪中断设施已安装指示符,且接着指示其参与协定的能力。在一个实例中,此情形包括注册客户机之参与警告追踪处理的意图(步骤 702)。如本文中所述,在一个实例中,经由诊断指令来执行注册。在注册时,客户机操作系统向主机 CPU 与主机程序两者指示客户机操作系统知晓如何处置警告追踪中断(WTI),警告追踪中断(WTI)为向客户机提供如下警告之明确中断:例如,客户机将失去对其共享资源,例如,客户机 CPU,的存取,且,例如客户机应关于其当前正执行之可分派单元采取动作。在一实施例中,注册为 WTI 之接收的先决条件。若并未针对警告追踪中断设施注册客户机,则在客户机之时间片期满后,便不供应宽限期,且使客户机 CPU 脱离解译执行模式。

[0081] 在一实施例中,即使已注册,客户机程序仍具有停用 WTI 之呈现的两个机制。举例而言,可将例如程序状态字(PSW)中的所选定位设定为零,此情形停用包括 WTI 之所有外部中断的呈现;或可将指定控制寄存器(例如,CRO)中的位设定为零以仅停用 WTI。当两个位为一时,启用 WTI 之呈现。若 WTI 之呈现保持停用达整个 WTI 宽限期,则客户机之执行在不利用 WTI 之情况下结束,该情形构成非主动退出。

[0082] 在客户机 CPU 的解译执行期间,若客户机 CPU 在内部辨识主机 CPU 定时器外部中断条件(例如,期满之时间片)或由主机程序请求之优先执行(查询 704),则内部 CPU 处理在主机接收到控制之前判定是否将执行警告追踪中断处理(查询 706)。亦即,内部 CPU 处理检查针对警告追踪处理启用客户机,且因此判定警告追踪处理应包括于待执行之处理中。若将不执行警告追踪中断处理,则客户机之解译执行结束(步骤 708),且控制返回至主机程序(步骤 710)。返回至查询 706,然而,若将执行警告追踪中断处理,则如下文进一步详细描述,执行处理(步骤 712)。

[0083] 参看图 8A 至图 8C 来描述警告追踪中断处理之其他细节的实施例。在此处理中,使用包括以下各者之若干个控制指示符:

[0084] 警告追踪中断设施宽限期作用中内部控制(例如,G 位),其并非架构上可见,但供内部 CPU 逻辑使用;

[0085] 警告追踪中断(WTI)呈现内部控制(例如,P 位),其在为 1 时指示已将 WTI 呈现

给客户机,且在为 0 时指示尚未呈现 WTI。类似于警告追踪中断设施宽限期作用中内部控制,WTI 呈现内部控制并非架构上可见,但供内部 CPU 逻辑使用;

[0086] 客户机控制的主机程序优先执行(例如,T 位),其例如为客户机 CPU 状态描述中的警告追踪干预请求指示符;以及

[0087] 在 E 指示符为一时,存在外部中断之启用。在一个实例中,E 指示符为当前程序状态字(PSW)内的位。

[0088] 参看图 8A,在一个实例中,辨识主机 CPU 定时器中断条件(例如,期满之时间片),或辨识警告追踪干预请求(例如,主机希望早早地传回 CPU 资源;亦即在时间片结束之前传回 CPU 资源)。若辨识出主机 CPU 定时器中断条件(查询 800),则作出关于宽限期作用中控制指示符是否经设定(例如,G 等于 1?)的判定(查询 802)。若 G 未经设定,则将 G 指示符设定为例如 1(步骤 804),且警告追踪中断设施宽限期将开始。接着保存主机 CPU 定时器之当前值(所保存值在本文中被称为原始值)(步骤 806),且将主机 CPU 定时器设定为警告追踪宽限期(例如,50 微秒)(步骤 808)。

[0089] 此后,作出关于是否针对警告追踪中断启用客户机之判定(查询 810)。在一实施例中,若客户机层级 2 为作用中的从而指示一客户机已启动另一客户机,则客户机 2 关于客户机 1 中断而退出解译执行模式,且取消客户机 1“启动解译执行”指令。因此,处理在此点处系作为客户机 1。若客户机 2 并非作用中的,则处理仅以客户机 1 继续。若针对 WTI 启用客户机,则将警告追踪外部中断(WTI)呈现给客户机(步骤 812)。在一个实例中,在需要时,此中断包括特定中断码,该特定中断码经呈现,从而指示其具有宽限期以便执行一个或多个功能(例如,清理)。

[0090] 另外,将 P 设定为 1,从而指示 WTI 已经呈现(步骤 814)。又,使用互锁更新功能将 T 位设定为 1(若最初已使用干预请求,则 T 位可能已为 1)(步骤 816)。使主机 CPU 定时器中之宽限期继续递减,而不管是否呈现了 WTI(步骤 818)。接着退出此处理(步骤 820)。在一个实例中,退出此程序之指明指示:CPU 已完成警告追踪中断设施之当前处理,且 CPU 正返回至如由 CPU 之当前状态指明的其他处理。

[0091] 返回至查询 810,若针对警告追踪中断未启用客户机,则处理进行至步骤 816。在此例子中,针对 WTI 未启用客户机,因此无法将 WTI 呈现给客户机。然而,将 T 位设定为未决的,使得稍后在针对 WTI 确实启用客户机时可侦测到 T 位。

[0092] 返回至查询 800,若并非主机 CPU 定时器中断条件,则辨识警告追踪中断请求(亦即,主机优先执行)。亦即,客户机之状态描述之干预请求字段中的 T 位为 1。因此,作出关于 G 指示符是否经设定的判定(查询 850)。若 G 指示符未经设定(例如,0),则处理以步骤 804 继续。在此情形下,T 等于 1 之条件为启动 WTI 程序的初始原因。然而,若 G 位经设定,则作出关于 P 是否经设定的判定(查询 852)。若 P 未经设定(例如,等于 0),则处理在呈现 WTI 之尝试中以步骤 810 继续。然而,若 P 经设定(例如,不等于 0),则在警告追踪设施宽限期开始之后 T 等于 1 的发现无影响,且退出程序(步骤 854)。

[0093] 返回至查询 802,若 G 经设定(例如,等于 1),则客户机 CPU 已在宽限期中执行,且主机 CPU 定时器之期满指示宽限期已期满。因此,先前已启动 WTI 循环,且宽限期已期满。因此,参看图 8B,将先前保存之原始主机 CPU 定时器值减小达在宽限期期间实际使用之时间量,且接着将其加载至主机 CPU 定时器中(步骤 860)。退出解译执行模式(步骤 862),且

将主机 CPU 定时器外部中断呈现给主机（步骤 864）（此退出为非主动客户机退出之形式）。

[0094] 除以上内容外,可经由可针对 WTI 启用 CPU 之某些指令来启动 WTI 分析。举例而言,参看图 8C,如本文中所描述,最初,可针对 WTI 启用 CPU 之包括以下各者的若干个指令执行监视:例如,可设定 PSW 中之指明位的加载 PSW(扩展)指令以及接着储存或系统屏蔽指令,以及可设定控制寄存器中之所选定位的“加载控制”。举例而言,可针对中断启用之指令检查用于潜在警告追踪处理的 T 位。若  $T = 0$  (查询 880),则不存在 WTI,且退出此程序(步骤 884)。然而,若  $T = 1$ ,则处理以查询 822 继续。

[0095] 在查询 882 处,作出关于 P 是否经设定(例如,等于 1)之判定。若 P 经设定,则退出此程序(步骤 884),此系由于先前已侦测到启用。然而,若 P 未经设定(例如,不等于 1),则作出关于 G 是否经设定(例如,等于 1)的另一判定(查询 886)。若 G 未经设定,则处理以步骤 804 继续(图 8A)。然而,若 G 经设定(例如,等于 1)(步骤 886(图 8C)),则处理以图 8A 中之查询 810 继续(步骤 888),且退出处理。

[0096] 参看图 9 来描述警告追踪中断之处理的其他细节。当客户机程序接收到警告追踪中断时,客户机程序执行客户机程序将执行之任何功能(例如,OS 功能)以便例如使得可分派工作单元为可重新分派的(步骤 900)。举例而言,客户机在特定点处停止可分派单元,保存可分派单元之状态,且将可分派单元移动至另一客户机 CPU,或藉由提供状态信息使得可分派单元能够被移动,等等。客户机操作系统藉由向主机程序发出警告追踪清理完成信号而用信号通知客户机操作系统已结束(又称作主动退出)(步骤 902)。此信号可为使得客户机操作放弃剩余时间片的任何机制。然而,该信号由主机程序辨识为协议之清理部分。在一个实例中,使用诊断指令之清理已完成功能。

[0097] 若客户机程序在宽限期期满之前发出清理已完成信号(查询 904),则主机程序记住客户机 CPU 及时退出(步骤 906)。此退出系主动退出。当下次启动客户机 CPU 时,将信号之准时本质指示回至客户机 CPU(步骤 908)。在一个实例中,客户机继续执行 PSW 经设定以指示成功条件代码(例如,条件代码 0)。

[0098] 返回至查询 904,若客户机程序由于任何原因而花费过长时间,则宽限期借助于使宽限期递减至零之主机 CPU 定时器而期满,因此将主机 CPU 定时器外部中断条件呈现给 CPU。在此状况下,CPU 辨识:客户机已在宽限期中,且并不授予另一宽限期。确切而言,客户机执行停止,且控制藉由接收外部中断而回复至主机程序。主机程序辨识:客户机 CPU 之此结束为非主动客户机退出。

[0099] 在下次客户机 CPU 启动时,客户机操作系统可接着发出清理已完成信号,虽然现在已过迟。主机程序不再期望等待清理已完成信号之接收。因此,在下次启动客户机 CPU 时,将信号之过迟本质指示回至客户机 CPU(步骤 912)。在一个实例中,客户机继续执行 PSW 经标示以指示在下次启动时客户机将观察到的迟条件。过迟诊断指令之发出有时被称作陈旧诊断指令,此系因为过迟诊断指令先前错过在宽限期内退出且接着由于不良原因稍后退出。

[0100] 在一个实例中,在客户机 CPU 之新启动后,客户机程序便可针对是否已在宽限期内发出信号而检查协议之信号继续执行部分。客户机程序可使用此信息来调查:客户机程序可能因何原因而延迟且作出改良以改良关于未来更及时之统计。

[0101] 在一实施例中,当针对所有外部中断停用客户机时,可启用外部中断之若干个指

令执行监视。当针对外部中断启用客户机时,检验 WTI 启用。在此点上,若 WTI 经启用且 P 位为 0,则将 WTI 呈现给客户机 CPU。

[0102] 如上文所提及,在一实施例中,诊断功能用以指示清理已完成或针对警告追踪中断设施注册。对于清理已完成,诊断功能在以清理参数发出并执行时用信号通知:发出 CPU 已执行了与警告追踪外部中断之接收相关联的任何所要处理。当执行完成时,设定条件代码,从而指示是否在警告追踪中断之后允许用于清理之模型相依时间间隔内发出完成。

[0103] 关于寄存器功能,诊断功能在以寄存器参数发出并执行时用信号通知:发出配置理解警告追踪中断。当执行完成时,设定成功条件代码。注册状态藉由系统重设来清除。

[0104] 参看图 10 来描述诊断指令之格式的一实施例。在一实施例中,诊断指令 1000 包括:指示诊断功能之作业码 1002;第一寄存器字段 1004 ( $R_1$ );第二寄存器字段 1006 ( $R_2$ );通用寄存器字段 1008 ( $B_2$ );以及移位字段 1010 ( $D_2$ )。在一个实例中,将  $D_2$  字段之内容添加至通用寄存器  $B_2$  之内容。结果并不用以寻址数据,而是将某些位(例如,位 48 至 63)用作操作码扩展。当操作码扩展为预定值时,指定警告追踪清理已完成,且放弃时间片。

[0105] 在一个实例中, $R_2$  字段未经使用,且其含有零。另外,通用寄存器  $R_1$  之指定位未经使用且将含有零,且通用寄存器  $R_1$  之指定位(例如,位 63)在为零时指定清理已完成功能,且在为 1 时指定寄存器功能。

[0106] 在使用共享实体 CPU 之逻辑分割区中,此功能可藉由允许将逻辑 CPU 正执行所在之实体 CPU 指派给另一逻辑 CPU 而改良系统效能。

[0107] 除诊断外,在 WTI 宽容间隔期间的任何其他 SIE 退出(无论何原因)类似地恢复递减达所花费之宽限期时间量的主机 CPU 定时器之原始值。

[0108] 本文中详细描述的内容为警告追踪中断设施,在一实施例中,该警告追踪中断设施提供一机制,可藉由该机制将警告追踪外部中断呈现给具有共享 CPU 资源(诸如,逻辑分割区)之配置中的 CPU。控制程序可使用警告追踪外部中断作为信号以使得当前正执行之可分派单元在该配置上之不同 CPU 上为可分派的。

[0109] 在一实施例中,于时间片中在实体处理器上执行之逻辑(客户机)处理器接收到指示宽限期之警告信号,例如,在逻辑处理器将被中断(自可共享之实体处理器解除分配)之前的时间量,从而使得能够完成由逻辑处理器进行之工作或将工作移动至另一逻辑处理器。作为实例,用信号向客户机 CPU 通知:其时间片已期满,且客户机 CPU 应优先执行当前可分派工作单元(DU)以使得当前可分派工作单元在另一客户机 CPU 上为可重新分派的。在一个实例中,警告信号为一中断,该中断具有指示该中断为 WTI 的中断码。在另一个实例中,中断码包括关于对宽限期给予之时间量或其他时段的信息。

[0110] 在一实施例中,警告追踪中断设施可用于非虚拟环境以及虚拟环境中,在这些环境中,一个程序和/或处理器与一个或多个其他程序和/或处理器共享资源(例如,CPU 资源或其他资源)。

[0111] 在环境为虚拟环境的一实施例中,自客户机之观点而言:

[0112] 1. 客户机程序观测警告追踪中断协议设施之已安装条件。

[0113] 2. 客户机程序针对警告追踪中断协议注册。

[0114] 3. 客户机 CPU 根据特定架构(例如,共享内存指示、共享 I/O 器件指示、中断)接收警告追踪通知。



[0115] 4. 在客户机 CPU 上执行的客户机程序根据已接收通知之客户机程序的本质来执行可适用处理（期望通知之处理为按操作系统独特的）。

[0116] 5. 客户机 CPU 根据警告追踪协议主动技术放弃控制。

[0117] 6. 在客户机 CPU 下次启动时，客户机程序可根据警告追踪协议观测反馈。

[0118] 另外，在一实施例中，自主机的观点而言：

[0119] A. 客户机程序观测警告追踪中断协议设施之已安装条件。

[0120] 1. 主机程序获取警告追踪中断协议设施已安装指示。

[0121] 2. 主机程序持久地记住警告追踪中断协议之已安装状态。

[0122] 3. 主机程序将警告追踪协议已安装状态指示给每个客户机配置。

[0123] 4. 主机程序停用所有未注册客户机 CPU 中之警告追踪协议。

[0124] 5. 主机程序做好准备以辨识来自每个客户机配置之客户机警告追踪注册请求。

[0125] B. 主机程序辨识来自客户机之警告追踪注册请求。

[0126] 1. 主机程序持久地记住客户机配置理解警告追踪协定。

[0127] 2. 主机程序针对警告追踪协议启用客户机。

[0128] C. 在正常客户机 CPU X 操作期间，使用客户机 CPU X 之优先执行重新获取对应主机 CPU X。

[0129] 1. CPU Y 中之主机程序用信号向客户机 CPU X 发通知。

[0130] a. 根据警告追踪协议，CPU X 经由共享内存位置之更新、共享 I/O 器件之更新或对客户机 CPU X 之中断而将通知传播至客户机 CPU X。

[0131] D. 客户机 CPU X 停止，从而使控制返回至主机 CPU X。

[0132] 1. 若在宽限期内，则 CPU X 上之主机程序按警告追踪协议观测主动客户机退出，且记住针对客户机 CPU X 之下次启动的良好反馈，而不管哪一主机 CPU 可能在该时间布建客户机 CPU X。

[0133] a. 若在宽限期内，但客户机 CPU X 之退出并非根据警告追踪协议，则不记住反馈状态。

[0134] 2. 若不在宽限期内，则 CPU X 上之主机程序按警告追踪协议观测主动客户机退出，且记住针对客户机 CPU X 之下次启动的不良反馈，而不管哪一主机 CPU 可能在该时间布建客户机 CPU X。

[0135] a. 若不在宽限期内，但客户机 CPU X 之退出并非根据警告追踪协议，则不记住反馈状态。

[0136] 3. 主机 CPU X 上之主机程序将 CPU X 重新导向为优先执行指派。

[0137] E. 若记住反馈状态，则不管哪一主机 CPU 布建客户机 CPU X，客户机 CPU X 之下次循序启动在启动客户机 CPU X 之前根据警告追踪协议设定反馈指示。

[0138] 在一实施例中，客户机配置之客户机处理器接收唯一中断，该中断系针对计算机架构而定义，且中断之含义系警告追踪中断。该中断指示将中断识别为警告追踪中断的特定码。该中断隐含被称作宽限期之相对较短之时间间隔，从而逐渐引导至结束客户机处理器执行。

[0139] 在宽限期期间，在一个实例中，名义上期望客户机程序使得当前可分派工作单元在另一客户机处理器上为可重新分派的，因此避免卡塞于当前客户机处理器上、等待其来

自主机之下次正常时间片启动。

[0140] 在一个实例中,客户机处理器之每次主机程序启动授予相对较短之时间间隔仅一次。自例如客户机处理器正执行所在之现有时间间隔来授予时间间隔。由于自正常剩余时间片来分配所授予之时间间隔,因此时间间隔本质上并非借用时间,而是使用来自当前时间间隔之受约束时间量来确保客户机处理器在相对较短之一段时间内确实被优先执行。

[0141] 在当前时间片已期满之另一个实例中,除客户机处理器正执行所在之现有时间间隔外,亦将时间间隔授予为额外时间。由主机程序对照将由客户机处理器消耗的期望客户机处理器接下来执行之期望的下一循序正常时间间隔来考虑所授予之时间间隔。仍意欲确保:客户机处理器在相对较短之一段时间内确实被优先执行。

[0142] 在一个实例中,可产生对警告追踪事件之中断请求以向程序告知:程序正接近共享 CPU 上的当前执行间隔的结束。中断请求为在注册配置且针对警告追踪中断设施启用配置时产生的未决条件类型。

[0143] 程序(例如,主机以及客户机)之间的合作处理使程序(例如,客户机操作系统)间的资源共享(例如,CPU)优化。一或多项方面以相同 CPU 利用提供例如更好之响应时间。另外,在超管理器未分派之前释放系统串行化。

[0144] 在另一实施例中,本发明的一个或多个方面可供来自操作系统之请求使用以使个别线程继续改良时间敏感性工作之经过时间。亦即,线程可请求额外时间或被提供额外时间以执行功能。

[0145] 如本领域技术人员将了解,本发明的一个或多个方面可体现为系统、方法或计算机程序产品。因此,本发明的一个或多个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、微码等)或组合软件与硬件方面之实施例的形式,这些实施例在本文中皆可大体上被称作“电路”、“模块”或“系统”。此外,本发明的一个或多个方面可采用以一个或多个计算机可读媒体体现所述的计算机程序产品之形式,该一个或多个计算机可读媒体上体现有计算机可读程序代码。

[0146] 可利用一个或多个计算机可读媒体的任何组合。计算机可读媒体可为计算机可读储存介质。举例而言,计算机可读储存介质可为(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,或前述各者之任何合适组合。计算机可读储存介质之更特定实例(非详尽列表)包括以下各者:具有一或多条导线之电连接、携带型计算机磁盘、硬盘、随机存取内存(RAM)、只读存储器(ROM)、可抹除可程序化只读存储器(EPROM 或闪存)、光纤、携带型光盘只读存储器(CD-ROM)、光学储存器件、磁性储存器件,或前述各者之任何合适组合。在此文件之内容脉络中,计算机可读储存介质可为可含有或储存供指令执行系统、装置或器件使用或结合指令执行系统、装置或器件而使用之程序的任何有形媒体。

[0147] 现参看图 11,在一个实例中,计算机程序产品 1100 例如包括一个或多个非暂时性计算机可读储存介质 1102 以在上面储存计算机可读程序代码构件或逻辑 1104,从而提供并促进本发明的一个或多个方面。

[0148] 可使用适当媒体来传输体现于计算机可读媒体上之程序代码,适当媒体包括(但不限于)无线媒体、有线媒体、光缆、RF 等或前述各者之任何合适组合。

[0149] 可以一种或多种程序设计语言的任何组合来撰写用于进行本发明的一个或多个方面之操作的计算机程序码,该一或多种程序设计语言包括诸如 Java、Smalltalk、C++ 或

其类似者之面向对象程序设计语言,以及诸如“C”程序设计语言、组译器或类似程序设计语言之习知程序性程序设计语言。程序代码可完全在用户计算机上执行,部分地在用户计算机上执行,作为独立软件包而执行,部分地在用户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境中,远程计算机可经由任何类型之网络(包括局域网(LAN)或广域网(WAN))连接至用户计算机,或可连接至外部计算机(例如,使用因特网服务提供商经由因特网)。

[0150] 本文中参看根据本发明之实施例之方法、装置(系统)以及计算机程序产品的流程图说明和/或方块图来描述本发明的一个或多个方面。应理解,可藉由计算机程序指令来实施这些流程图说明和/或方块图之每个区块以及这些流程图说明和/或方块图中之区块的组合。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之一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之该处理器而执行之指令产生用于实施在该或这些流程图和/或方块图区块中所指定之功能/动作之构件。

[0151] 亦可将这些计算机程序指令储存于一计算机可读媒体中,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起作用,使得储存于该计算机可读媒体中之指令产生一制造对象,该制造对象包括实施在该或这些流程图和/或方块图区块中所指定之功能/动作的指令。

[0152] 亦可将这些计算机程序指令加载至计算机、其他可程序化数据处理装置或其他器件上,以使得在该计算机、其他可程序化装置或其他器件上执行一系列操作步骤以产生一计算机实施程序,使得在该计算机或其他可程序化装置上执行之指令提供用于实施在该或这些流程图和/或方块图区块中所指定之功能/动作的程序。

[0153] 诸图中的流程图以及方块图说明根据本发明的一个或多个方面的各种实施例的系统、方法以及计算机程序产品之可能实施方案的架构、功能性以及操作。就此而言,流程图或方块图中的每个区块可表示程序代码之一模块、区段或部分,其包含用于实施指定之逻辑功能的一个或多个可执行指令。亦应注意,在一些替代实施方案中,区块中所注释的功能可不以诸图中所注释之次序发生。举例而言,取决于所涉及之功能性,连续展示之两个区块实际上可实质上同时执行,或这些区块有时可以相反次序执行。亦应注意,可藉由执行指定之功能或动作的基于专用硬件之系统或专用硬件与计算机指令之组合来实施方块图和/或流程图说明之每个区块以及方块图和/或流程图说明中之区块的组合。

[0154] 除以上内容外,本发明的一个或多个方面亦可由供应客户环境之管理的服务提供商来提供、供应、部署、管理、服务等。举例而言,服务提供商可为一个或多个客户产生、维护、支持等执行本发明的一个或多个方面的计算机程序码和/或计算机底层结构。作为回报,作为实例,服务提供商可在订用和/或收费协议下自客户收取付款。或者或另外,服务提供商可自广告内容销售至一个或多个第三方而收取付款。

[0155] 在本发明的一个方面中,应用程序可经部署以用于执行本发明的一个或多个方面。作为一个实例,应用程序之部署包含提供可操作以执行本发明的一个或多个方面的计算机底层结构。

[0156] 作为本发明的另一方面,可部署一计算机底层结构,其包含将计算机可读程序代码整合至计算系统中,其中与计算系统组合之程序代码能够执行本发明的一或多项方面。

[0157] 作为本发明的又一方面,可提供用于整合计算机底层结构之程序,其包含将计算机可读程序代码整合至计算机系统中。计算机系统包含计算机可读媒体,其中计算机媒体包含本发明的一个或多个方面。与计算机系统组合之程序代码能够执行本发明的一或多项方面。

[0158] 尽管上文描述了各种实施例,但这些实施例仅为实例。举例而言,其他架构的计算环境可含有并使用本发明的一个或多个方面。另外,宽限期可不同于时间量,诸如指令或循环之数目或任何其他可量化值。可在不偏离本发明之精神的情况下进行许多改变和/或添加。

[0159] 另外,其他类型的计算环境可受益于本发明的一个或多个方面。作为实例,适用于储存和/或执行程序代码之数据处理系统为可用的,其包括直接或经由系统总线间接耦接至内存组件的至少两个处理器。内存组件例如包括在程序代码之实际执行期间所使用之本端内存、大容量储存器,以及高速缓存,该高速缓存对至少某一程序代码提供临时储存以便减少在执行期间必须自大容量储存器撷取程序代码的次数。

[0160] 输入/输出或 I/O 器件(包括但不限于)键盘、显示器、指针器件、DASD、磁带、CD、DVD、随身碟(thumb drive)以及其他内存媒体等)可直接或经由介入 I/O 控制器耦接至系统。网络配接器亦可耦接至系统以使得数据处理系统能够经由介入之私有或公用网络耦接至其他数据处理系统或远程打印机或储存器件。调制解调器、电缆调制解调器以及以太网网络卡仅为可用类型之网络配接器中的少数几种。

[0161] 下文描述可含有和/或使用本发明的一个或多个方面的计算环境之其他实例。

[0162] 参看图 12,描绘实施本发明的一个或多个方面之主计算机系统 5000 的代表性组件。代表性主计算机 5000 包含与计算机内存(亦即,中央储存器)5002 通信之一个或多个 CPU5001,以及用于与其他计算机或 SAN 以及其类似者通信之至储存介质器件 5011 以及网络 5010 的 I/O 接口。CPU5001 符合具有经建构指令集以及经建构功能性之架构。CPU5001 可具有用于将程序地址(虚拟地址)变换成内存之实地址的动态地址转译(DAT)5003。DAT 通常包括转译后备缓冲器(TLB)5007,其用于对转译进行快取使得对计算机内存 5002 之区块之稍后存取并不需要地址转译之延迟。通常,高速缓存 5009 用于计算机内存 5002 与处理器 5001 之间。高速缓存 5009 可为阶层式高速缓存,从而具有可用于一个以上 CPU 之大高速缓存以及大高速缓存与每个 CPU 之间的较小之较快速(较低层级)高速缓存。在一些实施方案中,将较低层级高速缓存分裂以提供单独的低层级高速缓存以用于指令提取以及数据存取。在一实施例中,经由高速缓存 5009 藉由指令提取单元 5004 自内存 5002 提取指令。在指令解码单元 5006 中解码指令,且将指令分派(在一些实施例中,藉由其他指令)给指令执行单元 5008。通常,使用若干个执行单元 5008,例如算术执行单元、浮点执行单元以及分支指令执行单元。指令系由执行单元执行,从而在需要时自指令指定之寄存器或内存存取操作数。若将自内存 5002 存取(加载或储存)操作数,则加载/储存单元 5005 通常在正执行之指令之控制下处置存取。可在硬件电路中或在内部微码(固件)中或藉由两者之组合执行指令。

[0163] 如所注释,计算机系统包括本端(或主)储存器中之信息,以及寻址、保护,以及参考和改变记录。寻址之一些方面包括地址之格式、地址空间之概念、地址之各种类型,以及一类型之地址经转译至另一类型之地址的方式。主储存器中之一些储存器包括永久指派之

储存地址。主储存器为系统提供数据之可直接寻址之快速存取储存。数据与程序两者在其可得以处理之前将被加载至主储存器中（自输入器件）。

[0164] 主储存器可包括有时称作高速缓存之一个或多个较小之较快速存取缓冲储存器。高速缓存通常与 CPU 或 I/O 处理器实体地相关联。除效能外，相异储存介质之实体构造以及使用的效应大体上为程序不可观测的。

[0165] 可维护用于指令以及用于数据操作数的单独高速缓存。高速缓存内之信息以连续字节维持于称作快取区块或快取列（或简称为列）之整体边界上。模型可提供传回快取列之以字节为单位之大小的“擷取高速缓存属性”指令。模型亦可提供“预取数据”以及“预取数据相对长度”指令，前述两个指令实现储存器至数据或指令高速缓存中之预取或数据自高速缓存的释放。

[0166] 储存器被视为长水平位串。对于大多数操作而言，对储存器之存取以自左至右序列进行。将位串再分为八个位之单元。八位单元被称作字节，字节系所有信息格式之基本建置区块。储存器中之每个字节位置藉由一唯一非负整数来识别，该非负整数系该字节位置之地址，或简称为字节地址。邻近字节位置具有连续地址，在左侧上以 0 开始且以自左至右之序列进行。地址为无正负号之二进制整数，且为 24 个、31 个或 64 个位。

[0167] 在储存器与 CPU 或信道子系统之间一次一个字节或字节群组地传输信息。除非以其他方式指定，否则例如在 **z/Architecture**<sup>®</sup> 中，储存器中的字节群组是藉由群组之最左侧字节来寻址。群组中之字节之数目由待执行之操作隐含地或显式地指定。当用于 CPU 操作中时，字节群组被称作字段。在每个字节群组中，例如在 **z/Architecture**<sup>®</sup> 中，以自左至右之序列对位编号。在 **z/Architecture**<sup>®</sup> 中，最左侧位有时被称作“高位序”位，且最右侧位被称作“低位序”位。然而，位编号并非储存地址。可仅寻址字节。为了对储存器中之字节的个别位进行操作，存取整个字节。字节中之位自左至右编号为 0 至 7（例如，在 **z/Architecture**<sup>®</sup> 中）。地址中之位对于 24 位地址可编号为 8 至 31 或 40 至 63，或对于 31 位地址可编号为 1 至 31 或 33 至 63；地址中之位对于 64 位地址编号为 0 至 63。在多个字节之任何其他固定长度格式内，构成格式之位自 0 开始连续编号。出于错误侦测目的，且较佳为了校正，可将一个或多个检查位与每个字节或与字节群组一起来传输。这些检查位藉由机器自动产生，且不可直接受程控。以字节之数目来表达储存容量。当储存操作数字段之长度是藉由指令之操作码隐含时，字段据称为具有固定长度，该固定长度可为一个、两个、四个、八个或十六个字节。可针对一些指令隐含较大字段。当储存操作数字段之长度未经隐含而是经显式陈述时，字段据称为具有可变长度。可变长度操作数之长度可以一字节之增量（或藉由一些指令，以两个字节之倍数或其他倍数）变化。当将信息置放于储存器中时，替换指明字段中所包括的仅那些字节位置之内容，即使至储存器之实体路径的宽度可大于正储存之字段的长度亦如此。

[0168] 信息之某些单元将在储存器中的整体边界处。当储存地址为以字节为单位的长度的倍数时，边界对于信息单元而言被称作整体。对整体边界上之 2 个、4 个、8 个以及 16 个字节之字段给予专用名称。半字组 (halfword) 为二字节边界上之两个连续字节的群组，且为指令之基本建置区块。字组为四字节边界上之四个连续字节的群组。双字组为八字节边界上之八个连续字节的群组。四倍字组 (quadword) 为 16 字节边界上之 16 个连续字节的

群组。当储存地址指明半字组、字组、双字组以及四倍字组时，地址之二进制表示分别含有一个、两个、三个或四个最右侧零位。指令将在二字节整体边界上。大多数指令之储存操作数并不具有边界对准要求。

[0169] 在实施用于指令以及数据操作数之单独高速缓存的器件上，若将程序储存至随后提取指令所自之快取列上，则可经历显著延迟，而不管储存器是否变更随后提取的指令。

[0170] 在一实施例中，可藉由软件（有时指代经授权内码、固件、微码、毫码、皮码（pico-code）及其类似者，前述各者中之任一者将与本发明的一个或多个方面一致）来实践本发明。参看图 12，体现本发明的一个或多个方面的软件程序代码可由主机系统 5000 之处理器 5001 自诸如 CD-ROM 光驱、磁带机或硬盘机之长期储存介质器件 5011 存取。软件程序代码可体现于多种已知媒体中之任一者上以供数据处理系统使用，诸如磁盘、硬盘机或 CD-ROM。程序代码可散布于这些媒体上，或可自计算机内存 5002 散布至用户或经由网络 5010 自一计算机系统之储存器散布至其他计算机系统以供这些其他系统之用户使用。

[0171] 软件程序代码包括控制各种计算机组件以及一个或多个应用程序之功能以及互动的操作系统。程序代码通常自储存介质器件 5011 分页至程序代码可用于由处理器 5001 进行处理的相对较高速计算机储存器 5002。用于将软件程序代码体现于内存中、实体媒体上和 / 或经由网络散布软件程序代码的技术以及方法为熟知的，且本文中将进一步加以论述。程序代码在产生并储存于有形媒体（包括但不限于电子内存模块（RAM）、闪存、光盘（CD）、DVD、磁带及其类似者）上时常常被称作“计算机程序产品”。计算机程序产品媒体通常可由较佳计算机系统中之处理电路来读取以供处理电路执行。

[0172] 图 13 说明可实践本发明的一个或多个方面的代表性工作站或服务器硬件系统。图 13 之系统 5020 包含包括可选周边器件之代表性底层计算机系统 5021，诸如个人计算机、工作站或服务器。底层计算机系统 5021 包括一个或多个处理器 5026，以及一总线，该总线用以根据已知技术在处理器 5026 与系统 5021 之其他组件之间连接以及实现处理器 5026 与系统 5021 之其他组件之间的通信。举例而言，总线将处理器 5026 连接至内存 5025 以及长期储存器 5027，长期储存器 5027 可包括硬盘机（包括，例如，磁性媒体、CD、DVD 以及闪存中的任一者）或磁带机。系统 5021 可能亦包括用户接口配接器，其经由总线将微处理器 5026 连接至一个或多个接口器件（诸如，键盘 5024、鼠标 5023、打印机 / 扫描仪 5030 和 / 或其他接口器件），这些接口器件可为诸如触敏式屏幕、数字化键入板（entry pad）等之任何用户接口器件。总线亦经由显示器配接器将诸如 LCD 屏幕或监视器之显示器件 5022 连接至微处理器 5026。

[0173] 系统 5021 可借助于能够与网络 5029 通信（5028）之网络配接器与其他计算机或计算机之网络通信。实例网络配接器为通信频道、符记环、以太网或调制解调器。或者，系统 5021 可使用诸如蜂巢式数字封包数据（CDPD）卡之无线接口进行通信。系统 5021 可与局域网络（LAN）或广域网（WAN）中之这些其他计算机相关联，或系统 5021 可为具有另一计算机之客户端 / 服务器配置中的客户端，等等。所有这些配置以及适当通信硬件和软件为此项技术中知晓的。

[0174] 图 14 说明可实践本发明的一个或多个方面的数据处理网络 5040。数据处理网络 5040 可包括多个别网络，诸如无线网络和有线网络，这些网络中之每个者可包括多个别工作站 5041、5042、5043、5044。另外，如本领域技术人员将了解，可包括一个或多个 LAN，其中

LAN 可包含耦接至主机处理器之多个智能型工作站。

[0175] 仍参看图 14, 网络亦可包括大型计算机或服务器, 诸如网关计算机 (客户端服务器 5046) 或应用程序服务器 (可存取数据储存库且亦可直接自工作站 5045 存取的远程服务器 5048)。网关计算机 5046 充当至每个别网络中之入口点。当将一网络连接协议连接至另一网络连接协议时, 需要网关。网关 5046 可较佳借助于通信链路耦接至另一网络 (例如, 因特网 5047)。网关 5046 亦可使用通信链路直接耦接至一个或多个工作站 5041、5042、5043、5044。可利用可购自 International Business Machines Corporation 之 IBM eServer™ System z® 服务器来实施网关计算机。

[0176] 同时参看图 13 和图 14, 可体现本发明的一个或多个方面的软件程序设计码可由系统 5020 之处理器 5026 自诸如 CD-ROM 光驱或硬盘机之长期储存介质 5027 存取。软件程序设计码可体现于多种已知媒体中之任一者上以供数据处理系统使用, 诸如磁盘、硬盘机或 CD-ROM。程序代码可散布于这些媒体上, 或可自内存散布至用户 5050、5051 或经由网络自一计算机系统之储存器散布至其他计算机系统以供这些其他系统之用户使用。

[0177] 或者, 程序设计码可体现于内存 5025 中, 且使用处理器总线由处理器 5026 来存取。此程序设计码包括控制各种计算机组件和一个或多个应用程序 5032 之功能和互动的操作系统。程序代码通常自储存介质 5027 分页至程序代码可用于由处理器 5026 进行处理的高速内存 5025。用于将软件程序设计码体现于内存中、实体媒体上和 / 或经由网络散布软件程序代码的技术和方法为熟知的, 且本文中将进一步加以论述。程序代码在产生并储存于有形媒体 (包括但不限于电子内存模块 (RAM)、闪存、光盘 (CD)、DVD、磁带以及其类似者) 上时常常被称作“计算机程序产品”。计算机程序产品媒体通常可由较佳计算机系统中之处理电路来读取以供处理电路执行。

[0178] 最易于可用于处理器之高速缓存 (通常比处理器之其他高速缓存快且小) 为最低 (L1 或层级一) 高速缓存, 且主储存器 (主存储器) 为最高层级高速缓存 (若存在 3 个层级, 则为 L3)。最低层级高速缓存常常被分成保持待执行之机器指令的指令高速缓存 (I- 高速缓存), 以及保持数据操作数之数据高速缓存 (D- 高速缓存)。

[0179] 参看图 15, 针对处理器 5026 描绘例示性处理器实施例。通常, 高速缓存 5053 之一或多个层级用以缓冲存储器区块以便改良处理器效能。高速缓存 5053 为保持最可能使用之内存数据之快取列的高速缓冲器。典型快取列为 64 个、128 个或 256 个字节之内存数据。除用于对数据进行快取外, 单独高速缓存亦常常用于对指令进行快取。高速缓存连贯性 (内存和高速缓存中之列之复本的不同步) 常常藉由此项技术中熟知之各种“窥探”算法来提供。处理器系统之主存储器储存器 5025 常常被称作高速缓存。在具有 4 个层级之高速缓存 5053 的处理器系统中, 主存储器 5025 有时被称作层级 5 (L5) 高速缓存, 此系由于其通常为较快速的, 且仅保持可用于计算机系统之非挥发性储存器 (DASD、磁带等) 之一部分。主存储器 5025 对由操作系统页入和页出主存储器 5025 之数据页“进行快取”。

[0180] 程序计数器 (指令计数器) 5061 追踪待执行之当前指令的地址。z/Architecture® 处理器中之程序计数器为 64 个位, 且可截断为 31 或 24 个位以支持先前寻址限制。程序计数器通常体现于计算机之程序状态字 (PSW) 中, 使得程序计数器在内容脉络切换期间持续。因此, 具有程序计数器值之进行中程序可藉由例如操作系统来中断 (自程序环境至操

作系统环境之内容脉络切换)。程序之 PSW 在程序并非作用中的时维持程序计数器值,且在操作系统正执行时使用操作系统之程序计数器(在 PSW 中)。通常,使程序计数器递增达等于当前指令之字节之数目的量。精简指令集计算(RISC)指令之长度通常为固定的,而复杂指令集计算(CISC)指令之长度通常为可变的。IBM z/Architecture<sup>®</sup>之指令为具有 2 个、4 个或 6 个字节之长度的 CISC 指令。举例而言,藉由内容脉络切换操作或分支指令之分支符记操作来修改程序计数器 5061。在内容脉络切换操作中,将当前程序计数器值连同关于正执行之程序的其他状态信息(诸如,条件代码)一起保存于程序状态字中,且加载新程序计数器值从而指向待执行之新程序模块的指令。执行分支符记操作以便藉由将分支指令之结果加载至程序计数器 5061 中来准许程序作出决策或在程序内循环。

[0181] 通常,指令提取单元 5055 用以代表处理器 5026 提取指令。提取单元在内容脉络切换之后提取“接下来的循序指令”、分支符记指令之目标指令或程序之最初指令。现代指令提取单元常常使用预取技术来基于可能使用预取指令之可能性而推测性地预取指令。举例而言,提取单元可提取包括下一循序指令的指令之 16 个字节和其他循序指令之额外字节。

[0182] 接着由处理器 5026 执行所提取指令。在一实施例中,将所提取指令传递至提取单元之分派单元 5056。分派单元解码指令,且将关于经解码指令之信息转递至适当单元 5057、5058、5060。执行单元 5057 通常将自指令提取单元 5055 接收关于经解码算术指令之信息,且将根据指令之作业码对操作数执行算术运算。较佳将操作数自内存 5025、经建构之寄存器 5059 或自正执行之指令的实时字段提供至执行单元 5057。执行之结果在储存时储存于内存 5025、寄存器 5059 中或其他机器硬件(诸如,控制寄存器、PSW 寄存器及其类似者)中。

[0183] 处理器 5026 通常具有一个或多个单元 5057、5058、5060 以用于执行指令之功能。参看图 16A,执行单元 5057 可借助于介接逻辑 5071 与经建构通用寄存器 5059、解码/分派单元 5056、加载储存单元 5060 和其他处理器单元 5065 通信。执行单元 5057 可使用若干个寄存器电路 5067、5068、5069 以保持算术逻辑单元(ALU)5066 将进行运算的信息。ALU 执行诸如加法、减法、乘法和除法之算术运算以及诸如与(and)、或(or)和互斥或(XOR)、旋转和移位的逻辑功能。较佳地,ALU 支持设计相依性之专用运算。举例而言,其他电路可提供包括条件代码和修复支持逻辑的其他经建构设施 5072。通常,ALU 运算之结果保持于输出寄存器电路 5070 中,输出寄存器电路 5070 可将结果转递至多种其他处理功能。存在处理器单元之许多配置,但本发明描述仅意欲提供对一实施例之代表性理解。

[0184] 例如,加法(ADD)指令将在具有算术和逻辑功能性的执行单元 5057 中执行,而例如浮点指令将在具有专用浮点能力的浮点执行中执行。较佳地,执行单元藉由对藉由指令识别的操作数执行作业码定义之功能而对操作数进行运算。举例而言,加法(ADD)指令可由执行单元 5057 对在藉由指令之寄存器字段识别的两个寄存器 5059 中发现之操作数执行。

[0185] 执行单元 5057 对两个操作数执行算术加法,且将结果储存于第三操作数中,其中第三操作数可为第三寄存器或两个源寄存器中的一者。执行单元较佳利用算术逻辑单元(ALU)5066,算术逻辑单元(ALU)5066 能够执行诸如移位、旋转、与(And)、或(Or)和互斥或(XOR)之多种逻辑功能,以及包括加法、减法、乘法、除法中之任一者的多种代数函数。一些 ALU5066 经设计以用于纯量运算且一些 ALU5066 经设计以用于浮点。取决于架构,数据可为



大端序 (Big Endian) (其中最低有效字节处于最高字节地址) 或小端序 (Little Endian) (其中最低有效字节处于最低字节地址)。IBM **z/Architecture**<sup>®</sup> 系大端序。取决于架构, 带正负号之字段可为正负号和量值 (1 之补码或 2 之补码)。2 之补码系有利的, 此系因为: ALU 并不需要设计减法能力, 此系由于 2 之补码中的负值或正值仅需要 ALU 内的加法。举例而言, 常常以速记法来描述数字, 其中 12 位字段定义 4,096 字节区块之地址, 且通常经描述为 4Kbyte (千字节) 区块。

[0186] 参看图 16B, 用于执行分支指令之分支指令信息通常发送至分支单元 5058, 分支单元 5058 常常使用诸如分支历史表 5082 之分支预测算法来在其他条件运算完成之前预测分支之结果。将提取当前分支指令之目标, 且在条件运算完成之前推测性地执行当前分支指令之目标。当完成条件运算时, 基于条件运算之条件和所推测结果而完成或舍弃推测性执行之分支指令。典型分支指令可测试条件代码, 且在条件代码满足分支指令之分支要求情况下分支至目标地址, 可基于例如在寄存器字段或指令之实时字段中发现之若干数字 (包括一) 而计算目标地址。分支单元 5058 可使用 ALU5074, ALU5074 具有多个输入寄存器电路 5075、5076、5077 和一输出寄存器电路 5080。举例而言, 分支单元 5058 可与通用寄存器 5059、解码分派单元 5056 或其他电路 5073 通信。

[0187] 举例而言, 一群指令之执行可由于多种原因而被中断, 这些原因包括由操作系统启动之内容脉络切换、引起内容脉络切换之程序异常或错误、引起内容脉络切换之 I/O 中断信号, 或多个程序之多线程活动 (在多线程化环境中)。较佳地, 内容脉络切换动作保存关于当前正执行之程序的信息, 且接着加载关于正调用之另一程序的状态信息。举例而言, 状态信息可保存于硬件寄存器中或内存中。状态信息较佳包含指向待执行之下一指令的程序计数器值、条件代码、内存转译信息和经建构之寄存器内容。内容脉络切换活动可单独或组合地藉由硬件电路、应用程序、操作系统程序或固件码 (微码、皮码或经授权内码 (LIC)) 来训练。

[0188] 处理器根据指令定义的方法来存取操作数。指令可使用指令之一部分的值来提供实时运算元, 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) 的一个或多个寄存器字段。指令可利用藉由作业码字段识别之隐含寄存器作为操作数。指令可将内存位置用于操作数。操作数之内存位置可由寄存器、实时字段或寄存器与实时字段之组合来提供, 如藉由 **z/Architecture**<sup>®</sup> 长位移设施例证, 其中指令定义基础寄存器、索引寄存器和实时字段 (位移字段), 前述三者相加在一起以提供例如操作数在内存中的地址。除非以其他方式指示, 否则位置于本文中通常隐含主存储器 (主存储器) 中的位置。

[0189] 参看图 16C, 处理器使用加载 / 储存单元 5060 来存取存储器。加载 / 储存单元 5060 可藉由获得目标操作数在内存 5053 中之地址且将操作数加载于寄存器 5059 或另一内存 5053 之位置而执行加载操作, 或可藉由获得目标操作数在内存 5053 中之地址且将自寄存器 5059 或另一内存 5053 之位置获得的数据储存于内存 5053 中之目标操作数位置中而执行储存操作。加载 / 储存单元 5060 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内存, 然而, 加载 / 储存单元 5060 对于程序维持按次序执行指令的显现。加载 / 储存单元 5060 可与通用寄存器 5059、解码 / 分派单元 5056、高速缓存 / 内存接口 5053 或其他组件 5083 通信, 且包含各种寄存器电路、ALU5085 和控制逻辑 5090 以计算储存地址并提

供管线定序以保持操作按次序。一些操作可为无序的,但加载/储存单元提供使得无序操作对于程序显现为已按次序执行的功能性,如此项技术中所熟知的。

[0190] 较佳地,应用程序“看见”的地址常常被称作虚拟地址。虚拟地址有时被称作“逻辑地址”和“有效地址”。这些虚拟地址为虚拟的在于:其藉由多种动态地址转译(DAT)技术中之一者而重新导向至物理内存位置,这些动态地址转译(DAT)技术包括(但不限于)简单地对虚拟地址加偏移值作为前缀、经由一个或多个转译表转译虚拟地址,转译表较佳单独或组合地包含至少一个段表和一页表,较佳地,段表具有指向页表之条目。在 **z/Architecture**<sup>®</sup> 中,提供转译阶层,包括区第一表、区第二表、区第三表、段表和可选之页表。常常藉由利用转译后备缓冲器(TLB)来改良地址转译之效能,该转译后备缓冲器(TLB)包含将虚拟地址映像至相关联物理内存位置的条目。当DAT使用转译表来转译虚拟地址时,产生这些条目。虚拟地址之后续使用可接着利用快速TLB之条目而非缓慢循序转译表存取。可藉由包括最近最少使用(LRU)之多种替换算法来管理TLB内容。

[0191] 在处理器为多处理器系统之处理器的状况下,每个处理器具有保持诸如I/O、高速缓存、TLB和内存之共享资源互锁以达成一致性的责任。通常,“窥探”技术将用于维持高速缓存之一致性中。在窥探环境中,可将每个快取列标示为处于共享状态、互斥状态、改变之状态、无效状态和其类似者中之任一者以便促进共享。

[0192] 举例而言,I/O单元5054(图15)为处理器提供用于附加至包括磁带、光盘、打印机、显示器和网络之周边器件的构件。I/O单元常常由软件驱动程序呈现给计算机程序。在诸如可购自**IBM**<sup>®</sup>之**System z**<sup>®</sup>的大型计算机中,信道配接器和开放系统配接器为大型计算机之提供操作系统与周边器件之间的通信之I/O单元。

[0193] 另外,其他类型的计算环境可受益于本发明的一个或多个方面。作为实例,如本文中提及,环境可包括一仿真器(例如,软件或其他仿真机制),在该仿真器中仿真特定架构(包括例如指令执行、诸如地址转译的经建构之功能,和经建构之寄存器)或其子集(例如,在具有处理器和内存的原生计算机系统上)。在此环境中,仿真器之一个或多个模拟函式可实施本发明的一个或多个方面,即使执行该仿真器之计算机可具有不同于正模拟之能力的架构亦如此。作为一个实例,在仿真模式中,解码特定指令或正仿真之操作,且建置适当模拟函式以实施个别指令或操作。

[0194] 在一模拟环境中,主计算机例如包括:内存,其储存指令和数据;指令提取单元,其自内存提取指令且视情况,提供所提取之指令的本端缓冲;指令解码单元,其接收所提取之指令且判定已提取的指令之类型;以及指令执行单元,其执行这些指令。执行可包括:将数据自内存加载至寄存器中;将数据自寄存器储存回至内存;或执行某一类型之算术或逻辑运算(如由解码单元判定)。在一个实例中,以软件来实施每个单元。举例而言,将由这些单元执行之操作实施为仿真器软件内之一个或多个子例程。

[0195] 更特定而言,在大型计算机中,常常借助于编译应用程序而供程序设计师(现今通常为“C”程序设计师)使用经建构之机器指令。储存于储存介质中之这些指令可原生地在**z/Architecture**<sup>®</sup> **IBM**<sup>®</sup>服务器中或者在执行其他架构之机器中执行。可在现有和未来之**IBM**<sup>®</sup>大型计算机服务器中和在**IBM**<sup>®</sup>之其他机器(例如,电力系统(Power

System) 服务器和 System x<sup>®</sup> (服务器) 上仿真指令。可于在使用由 IBM<sup>®</sup>、Intel<sup>®</sup>、AMD<sup>™</sup> 及其他制造商制造之硬件的广泛多种机器上执行 Linux 的机器中执行指令。除在依据 z/Architecture<sup>®</sup> 之该硬件上执行外, 亦可使用 Linux 以及使用由 Hercules、UMX 或 FSI (Fundamental Software, Inc) 进行之模拟的机器, 其中执行大体上处于仿真模式。在仿真模式中, 由原生处理器执行仿真软件以仿真仿真处理器的架构。

[0196] 原生 (native) 处理器通常执行仿真软件, 该仿真软件包含固件或原生操作系统以执行仿真处理器的模拟。仿真软件负责提取并执行仿真处理器架构之指令。仿真软件维护仿真程序计数器以追踪指令边界。仿真软件一次可提取一个或多个仿真机器指令, 并将该一个或多个仿真机器指令转换成一群对应原生物理指令以供原生处理器执行。可对这些经转换之指令进行快取, 使得可实现较快速转换。尽管如此, 仿真软件仍维护仿真处理器架构之架构规则以便确保针对仿真处理器撰写之操作系统和应用程序正确地操作。此外, 仿真软件将提供藉由仿真处理器架构识别之资源, 使得经设计以在模拟处理器上执行之操作系统或应用程序可在具有仿真软件之原生处理器上执行, 这些资源包括 (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之动态地址转译功能、中断机制、内容脉络切换机制、当日时间 (TOD) 时钟及至 I/O 子系统之经建构接口。

[0197] 解码正仿真之指定指令, 且调用子例程以执行个别指令之功能。仿真仿真处理器之功能的仿真软件功能系例如按以下各者来实施: “C” 子例程或驱动程序, 或在理解较佳实施例之描述之后将在本领域技术人员之技艺内的提供用于指定硬件之驱动程序的某一其他方法。包括 (但不限于) 以下各者之各种软件及硬件仿真专利说明用以达成针对不同机器进行架构之指令格式用于可用于本领域技术人员的目标机器的仿真的多种已知方式: Beausoleil 等人之题为 “Multiprocessor for Hardware Emulation” 之美国专利证书第 5, 551, 013 号; 及 Scalzi 等人之题为 “Preprocessing of Stored Target Routines for Emulating Incompatible Instructions on a Target Processor” 的美国专利证书第 6, 009, 261 号; 及 Davidian 等人之题为 “Decoding Guest Instruction to Directly Access Emulation Routines that Emulate the Guest Instructions” 的美国专利证书第 5, 574, 873 号; 及 Gorishek 等人之题为 “Symmetrical Multiprocessing Bus and Chipset Used for Coprocessor Support Allowing Non-Native Code to Run in aSystem” 的美国专利证书第 6, 308, 255 号; 及 Lethin 等人之题为 “Dynamic Optimizing Object Code Translator for Architecture Emulation and Dynamic Optimizing Object Code Translation Method” 的美国专利证书第 6, 463, 582 号; 及 Eric Traut 之题为 “Method for Emulating Guest Instructions on a Host Computer Through Dynamic Recompile of Host Instructions” 的美国专利证书第 5, 790, 825 号, 前述专利证书中之每个者藉此以其全文引用之方式并入本文中; 及许多其他专利证书。

[0198] 在图 17 中, 提供仿真主计算机系统 5092 之实例, 其仿真主机架构之主计算机系统 5000'。在仿真主计算机系统 5092 中, 主机处理器 (CPU) 5091 为仿真主机处理器 (或虚拟主机处理器), 且包含具有不同于主计算机 5000' 之处理器 5091 之原生指令集架构的原生指令集架构之仿真处理器 5093。仿真主计算机系统 5092 具有仿真处理器 5093 可存取的内 存 5094。在实例实施例中, 内存 5094 经分割成主计算机内存 5096 部分及仿真例程 5097 部

分。主计算机内存 5096 可用于根据主计算机架构之仿真主计算机 5092 的程序。模拟处理器 5093 执行不同于模拟处理器 5091 之架构的架构之经建构指令集之原生指令,这些原生指令系自仿真例程内存 5097 获得,且可藉由使用在序列与存取 / 解码例程中获得之一个或多个指令来自主计算机内存 5096 中之程序存取主机指令以供执行,序列与存取 / 解码例程可解码所存取之主机指令以判定用于仿真所存取之主机指令之功能的原生指令执行例程。举例而言,针对主计算机系统 5000' 之架构定义之其他设施可藉由经建构之设施例程来仿真,包括诸如通用寄存器、控制寄存器、动态地址转译及 I/O 子系统支持及处理器高速缓存的设施。仿真例程亦可利用在仿真处理器 5093 中可用之功能(诸如,通用寄存器及虚拟地址之动态转译)以改良仿真例程的效能。亦可提供专用硬件及卸除引擎以协处理器 5093 仿真主计算机 5000' 之功能。

[0199] 本文所使用之术语仅为了描述特定实施例的目的,且并不意欲为本发明之限制。如本文中所使用,除非上下文另有清楚地指示,否则单数形式“一”及“该”意欲亦包括复数形式。应进一步理解,当术语“包含”用于此说明书中时,其指定所陈述之特征、整数、步骤、操作、组件和 / 或组件之存在,但并不排除一个或多个其他特征、整数、步骤、操作、组件、组件和 / 或其群组之存在或添加。

[0200] 以下申请专利范围中之所有构件或步骤加功能组件的对应结构、材料、动作及等效物(若有的话)意欲包括用于结合如特别主张之其他所主张组件执行功能的任何结构、材料或动作。出于说明及描述之目的已呈现了本发明的一个或多个方面的描述,但该描述并不意欲为详尽的或以所揭示形式限制本发明。在不偏离本发明之范畴及精神之情况下,许多修改及变化对于一般本领域技术人员将为显而易见的。选择及描述实施例以便最好地解释本发明之原理及实务应用,且使得其他一般本领域技术人员能够针对具有如适合于所预期之特定用途的各种修改之各种实施例来理解本发明。

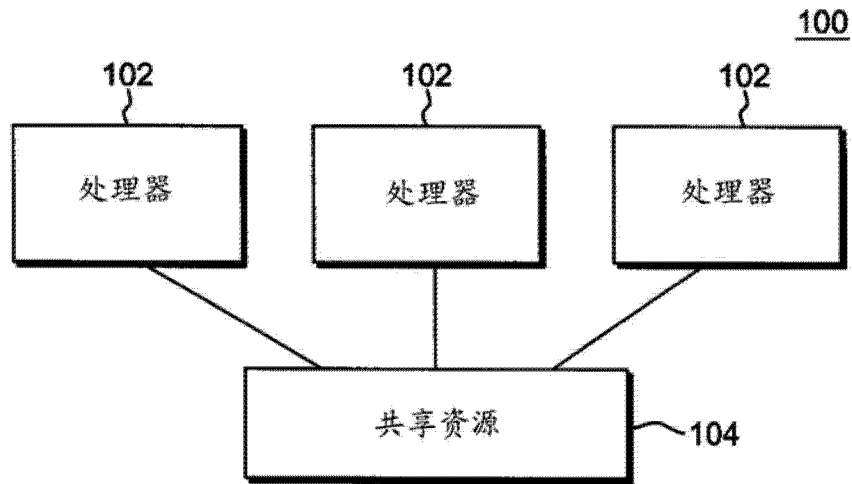


图 1

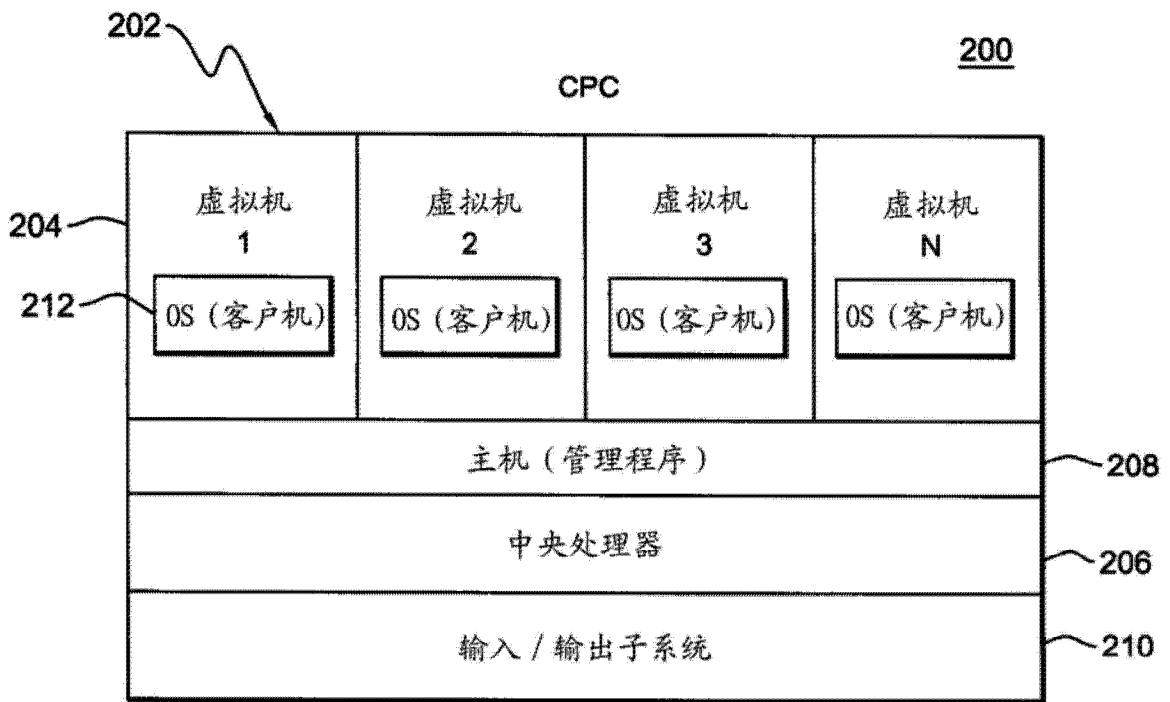


图 2

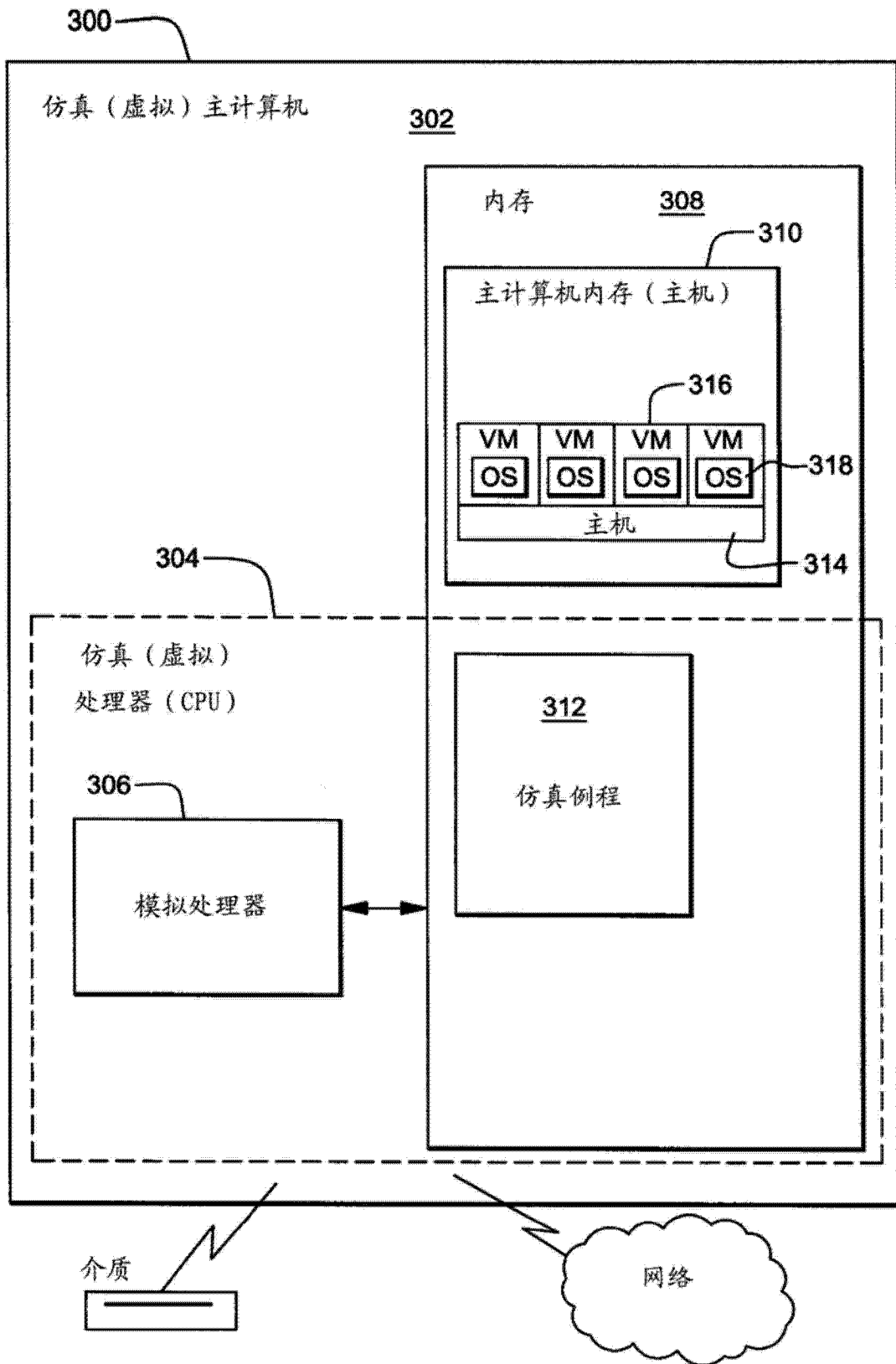


图 3

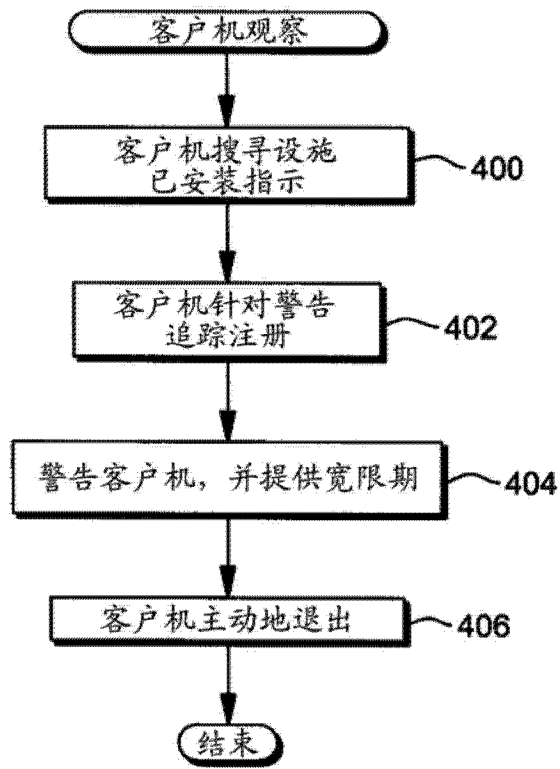


图 4

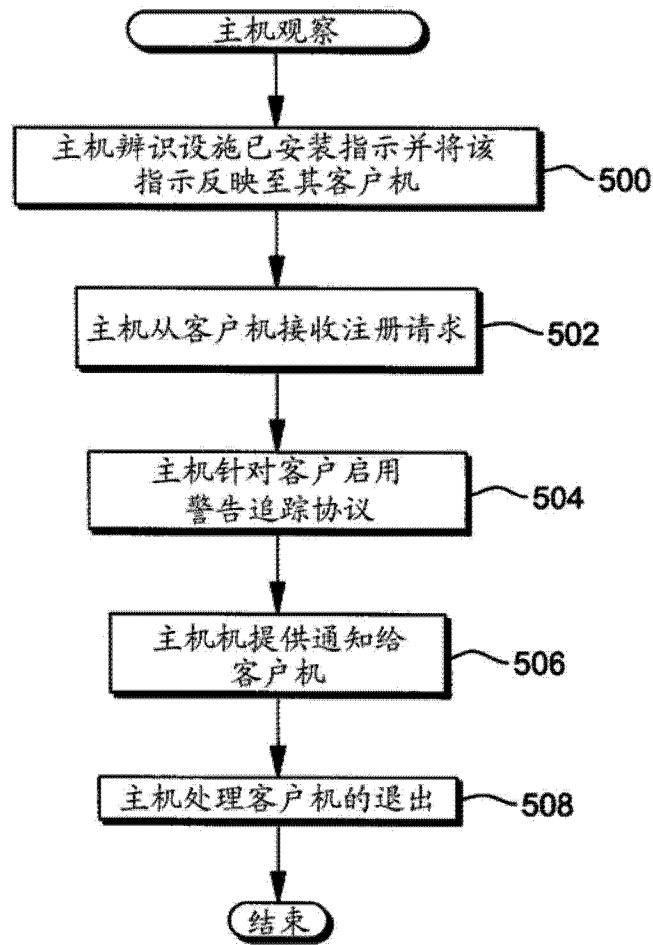


图 5



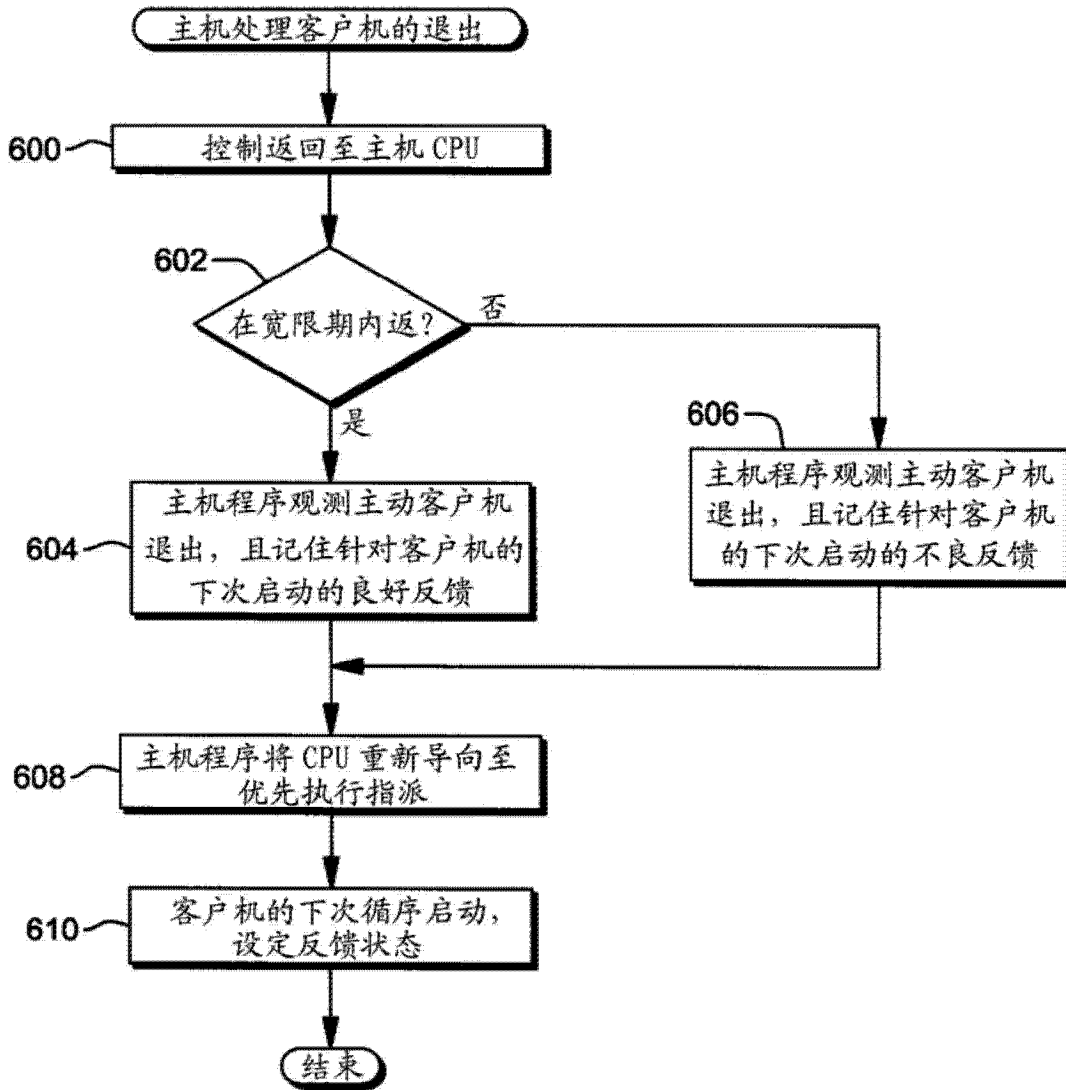


图 6

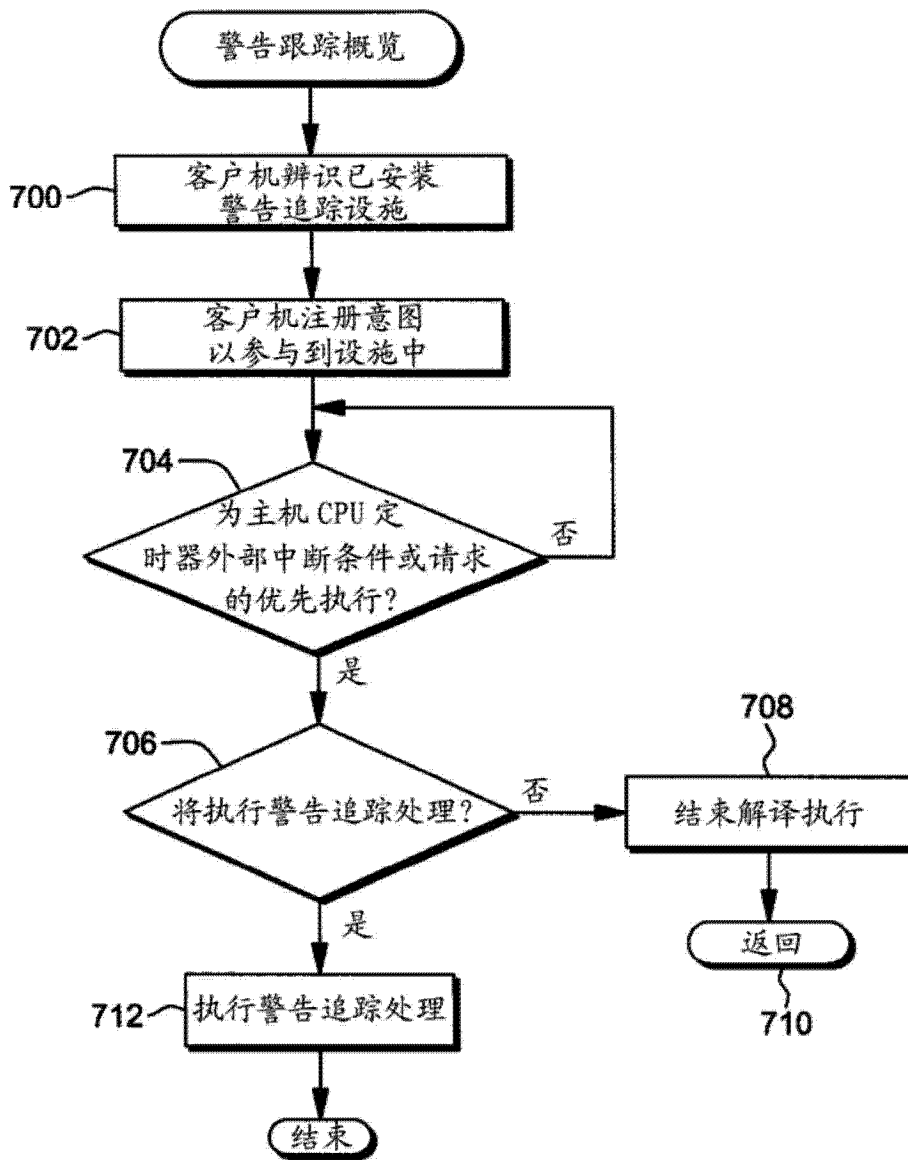


图 7

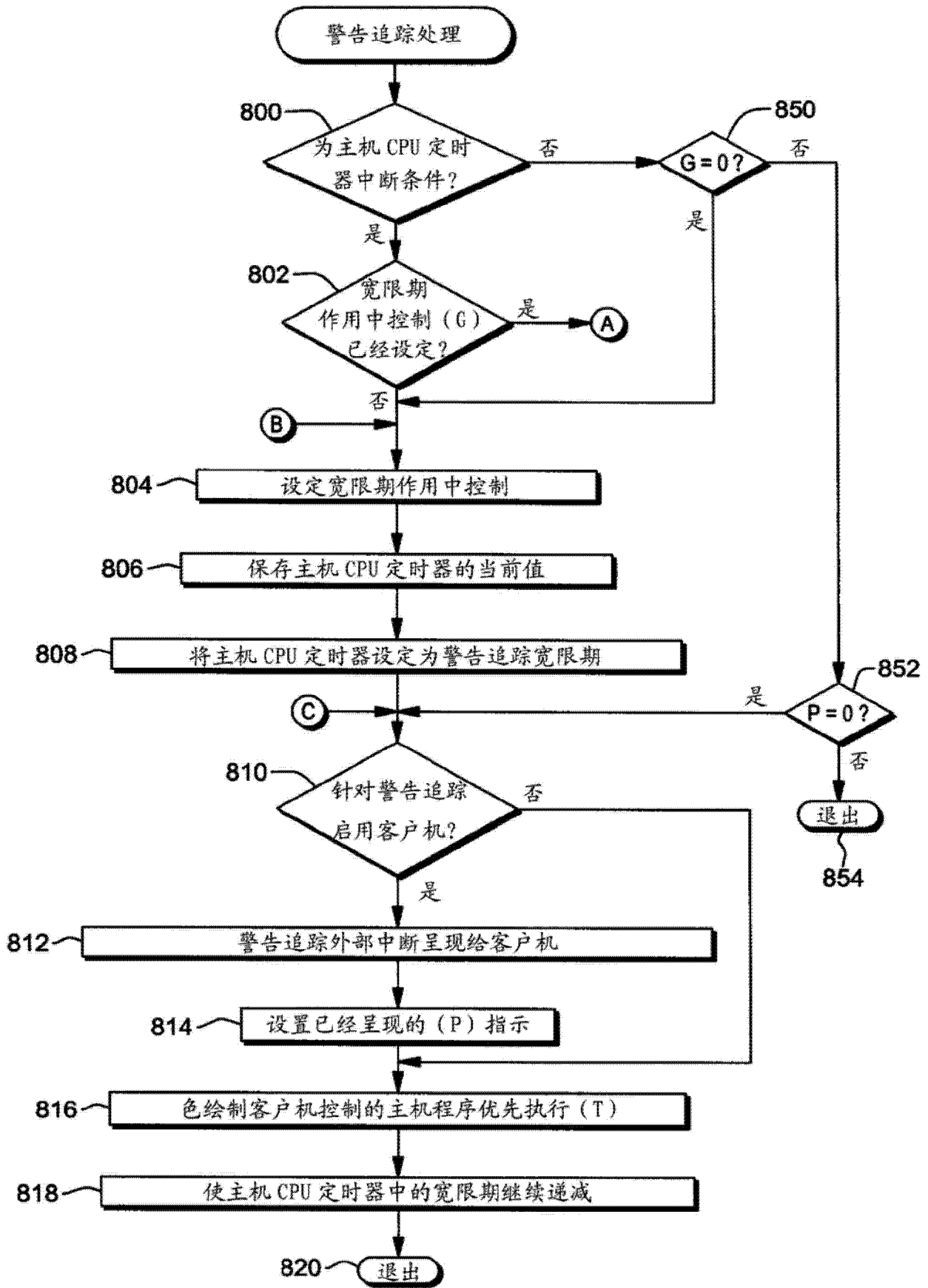


图 8A

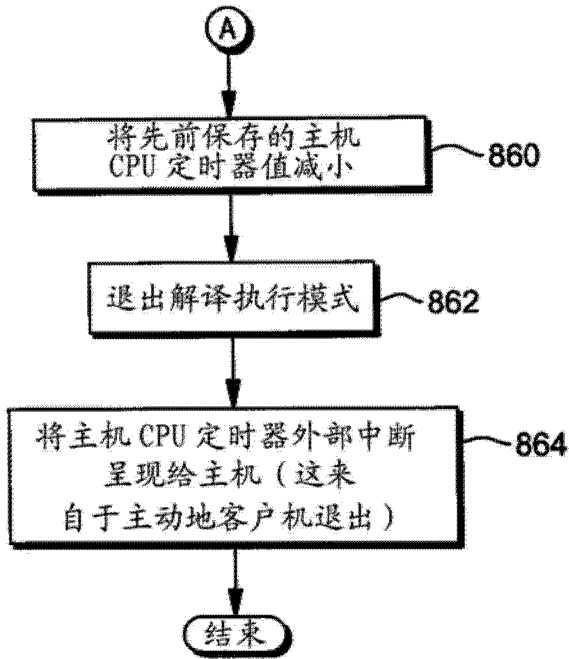


图 8B

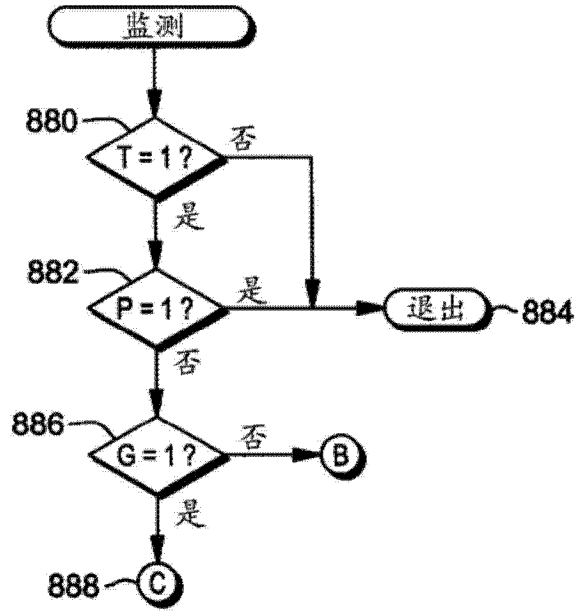


图 8C

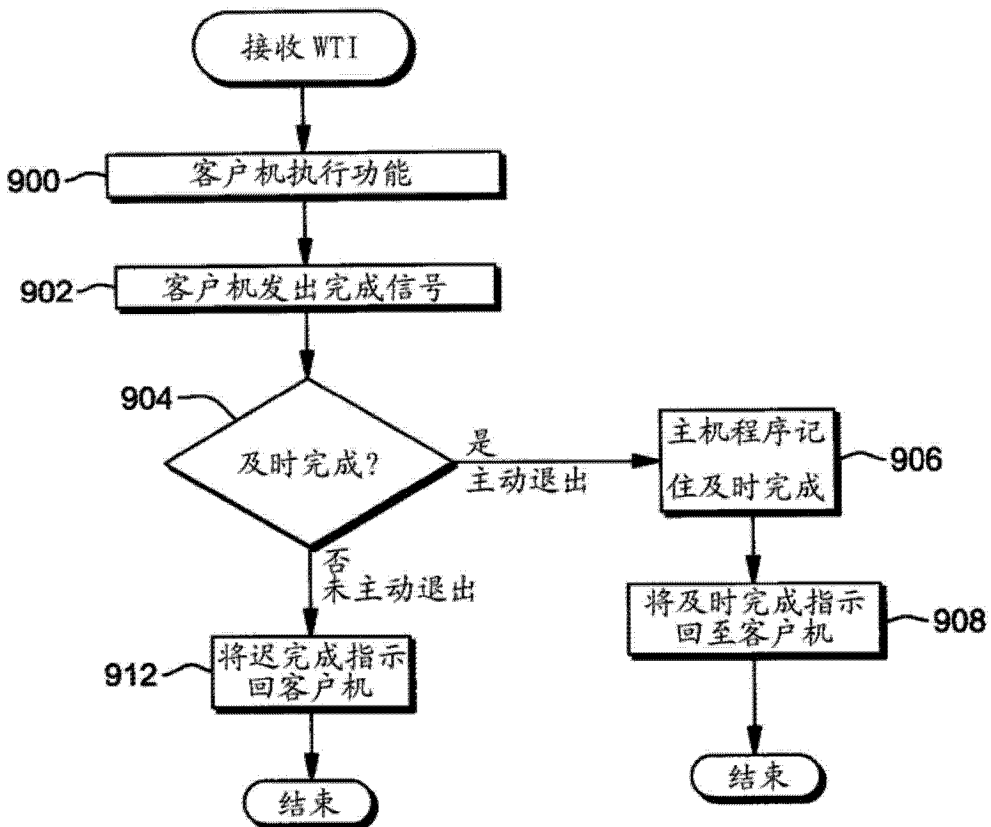


图 9

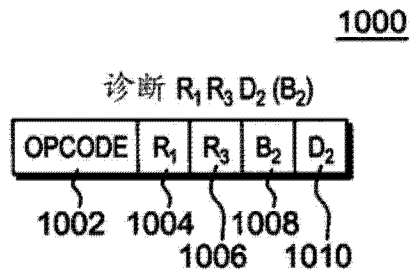


图 10

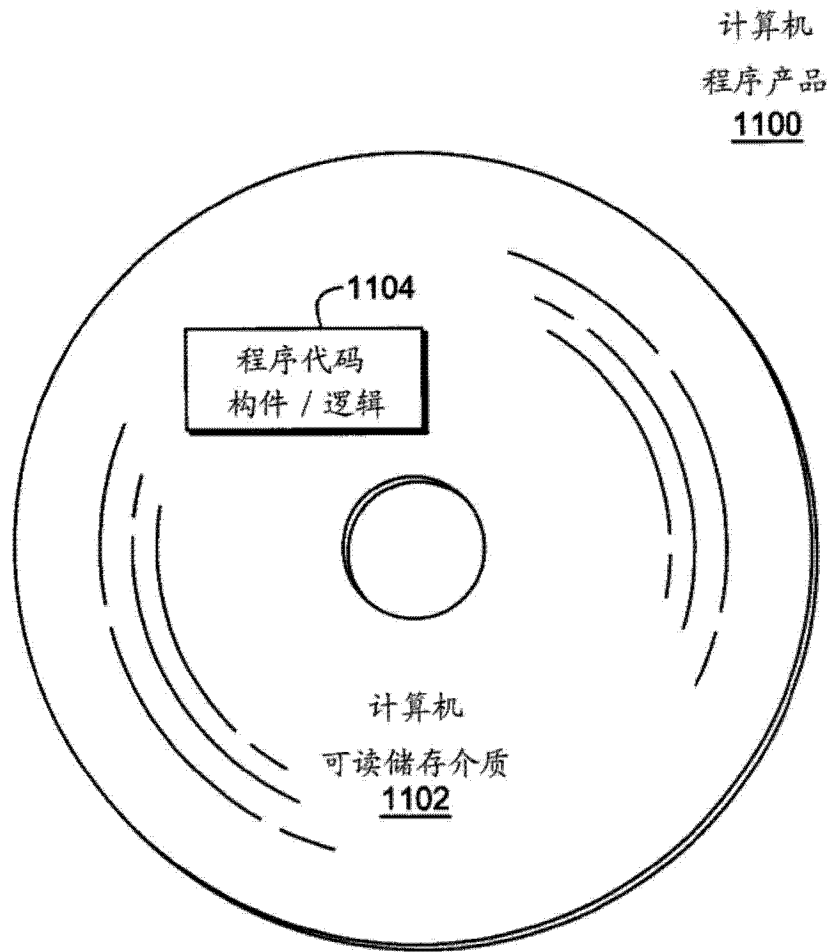


图 11

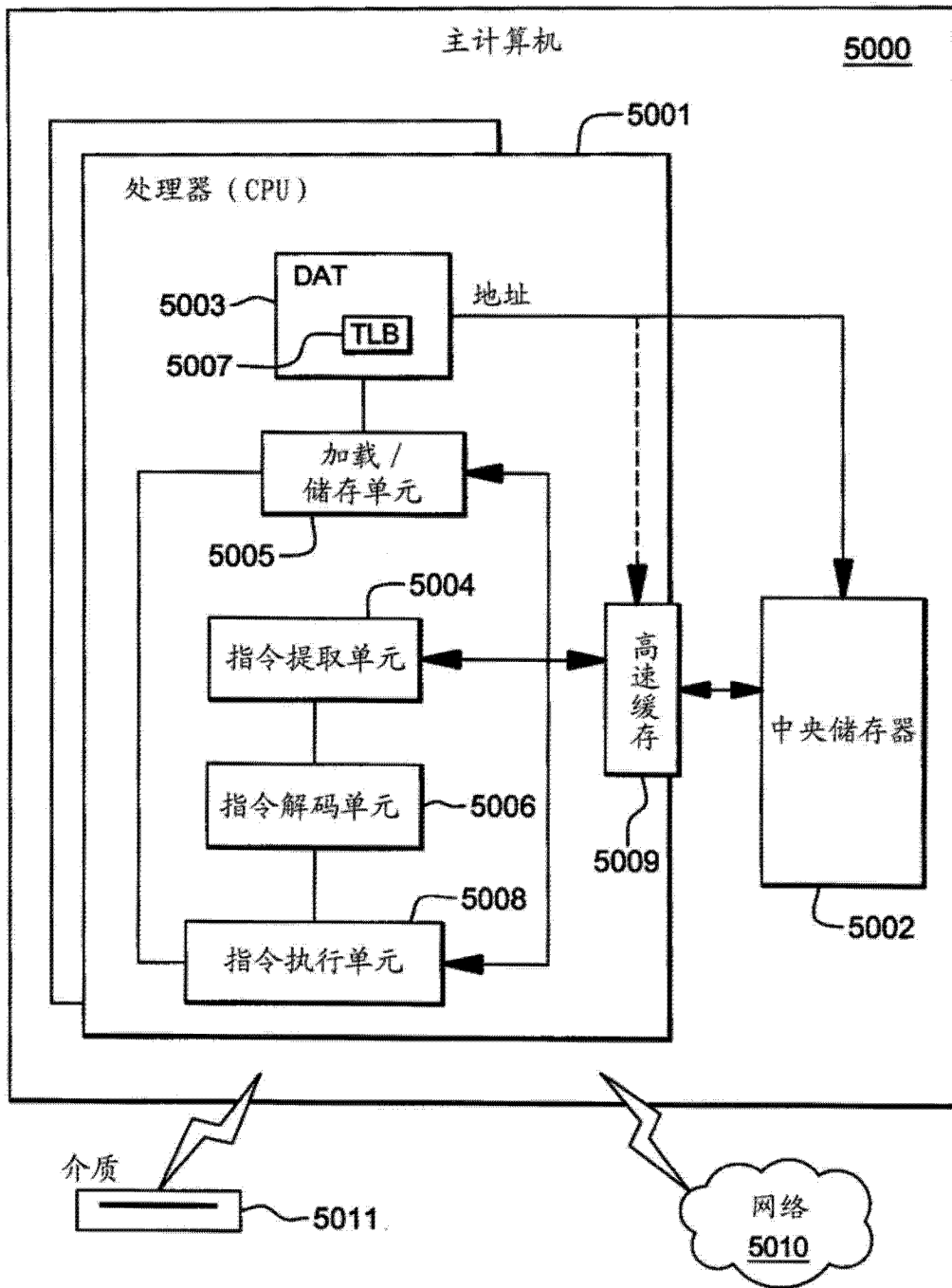


图 12

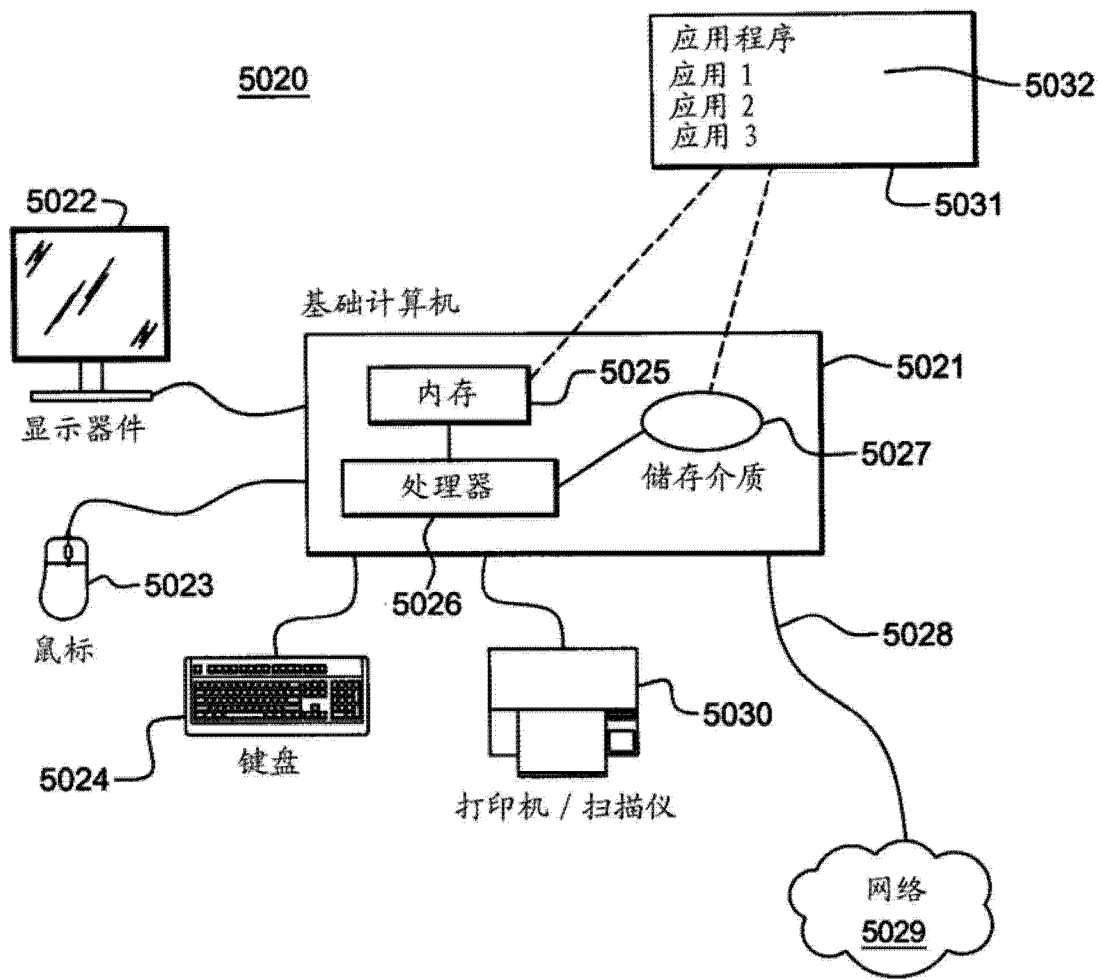


图 13

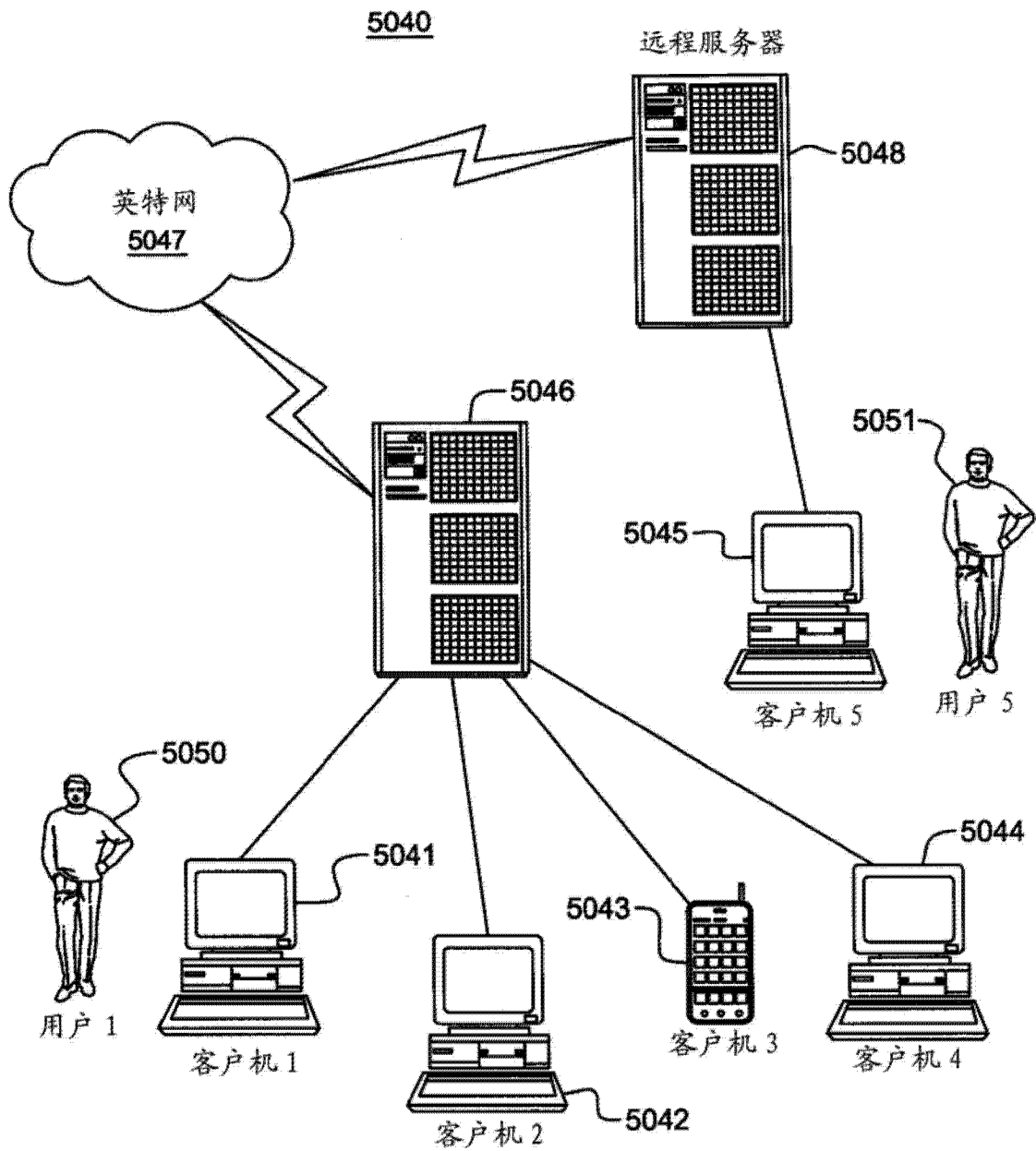


图 14



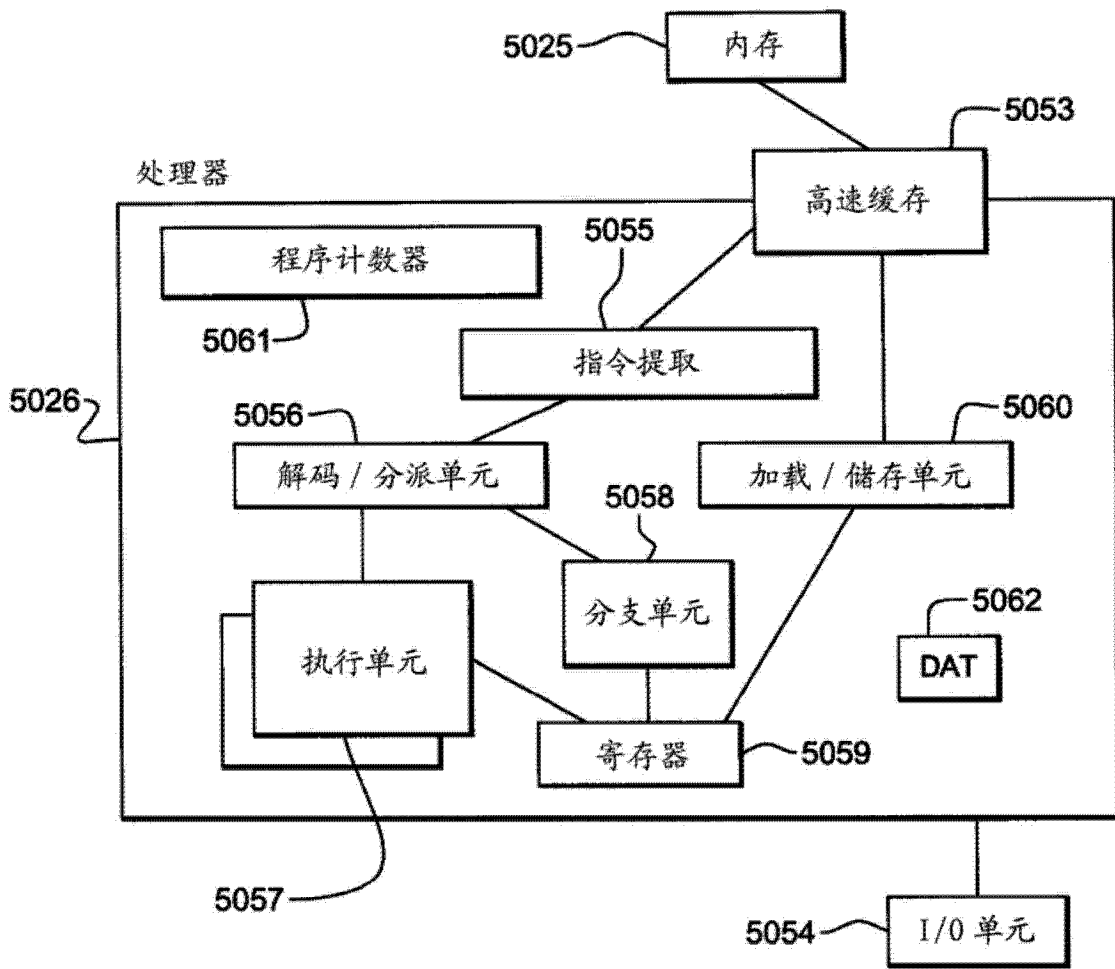


图 15

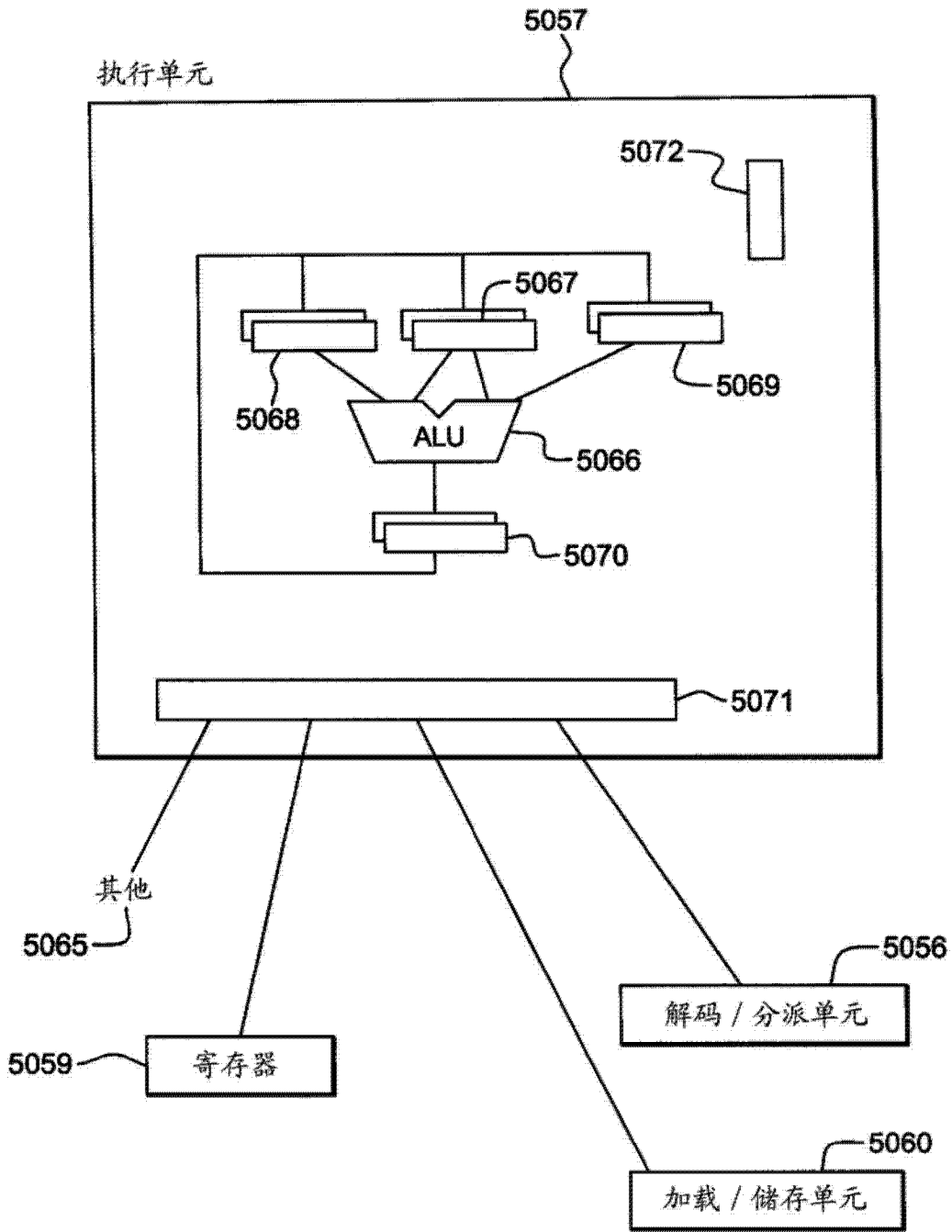


图 16A

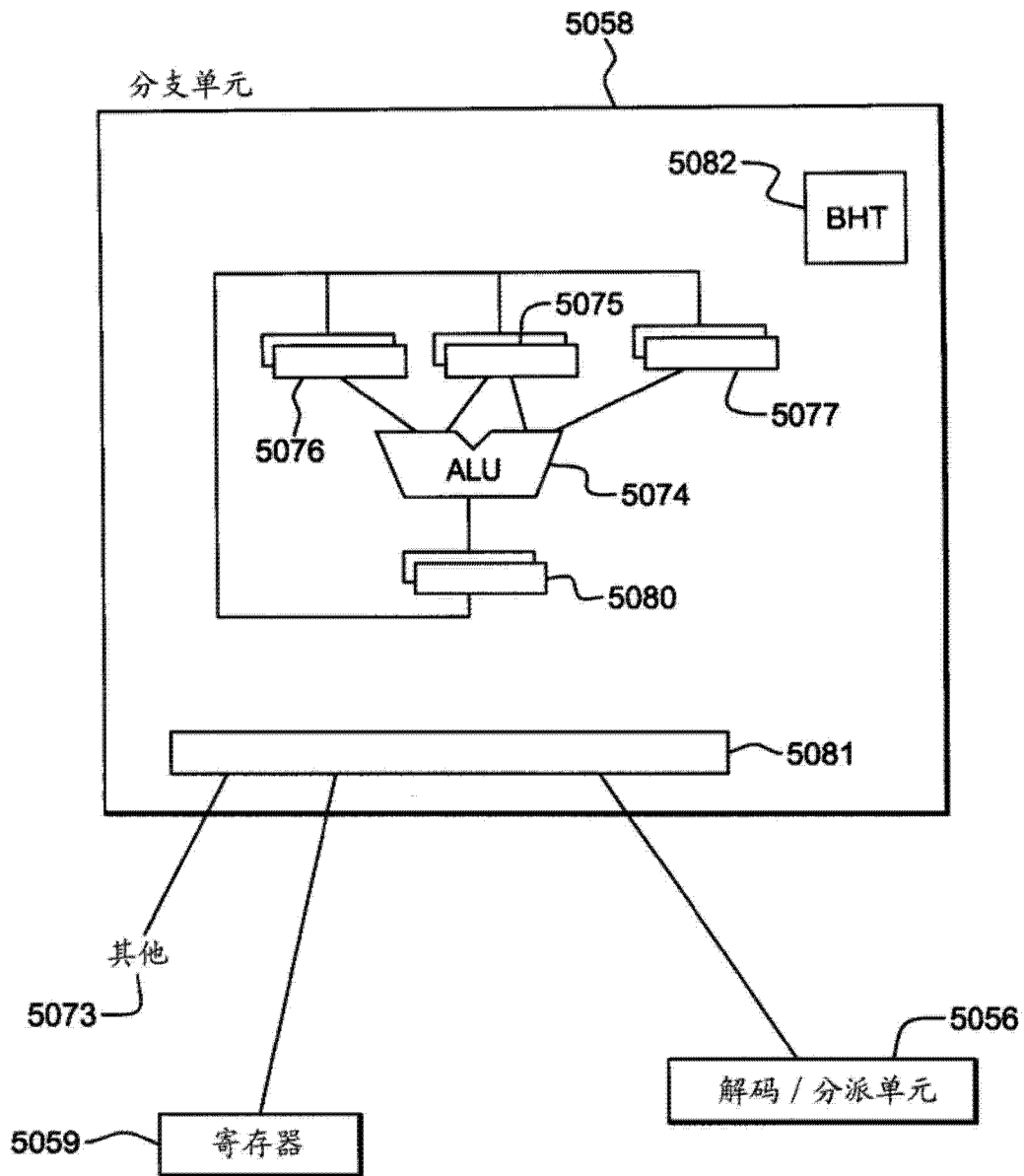


图 16B

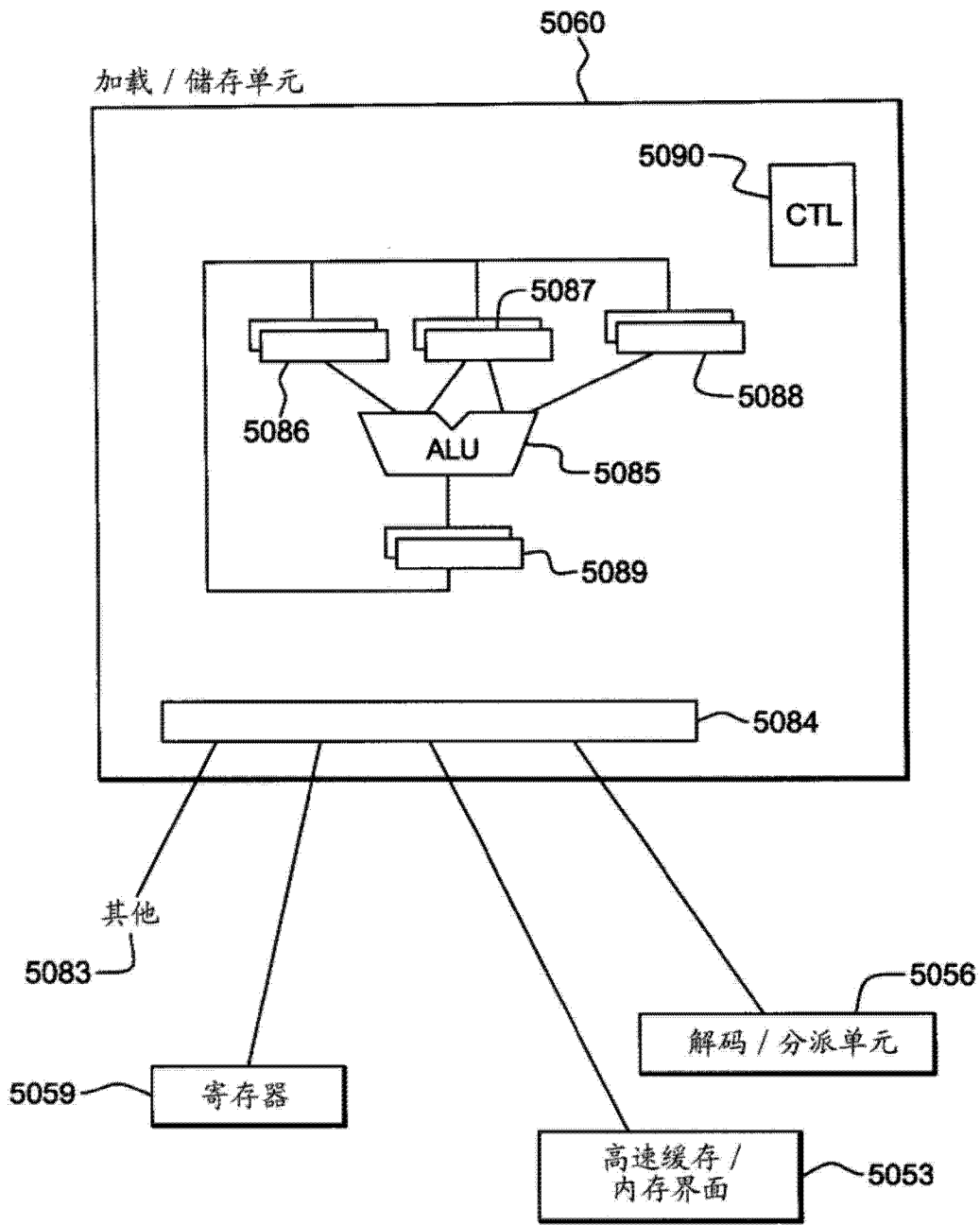


图 16C

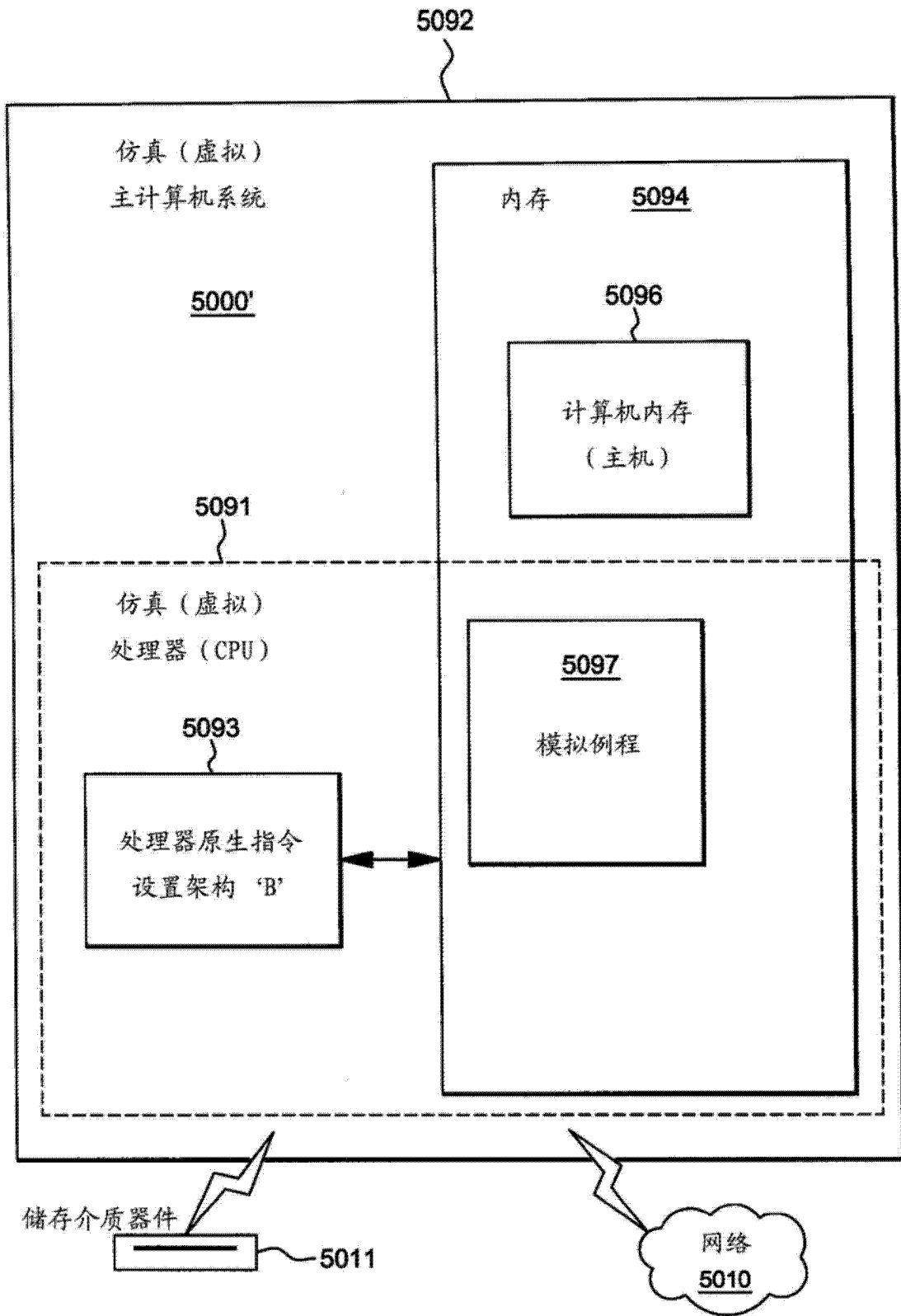


图 17