



(12) 发明专利

(10) 授权公告号 CN 101946232 B

(45) 授权公告日 2014. 11. 26

(21) 申请号 200980106158. 4

(22) 申请日 2009. 02. 02

(30) 优先权数据

12/049, 984 2008. 03. 17 US

(85) PCT国际申请进入国家阶段日

2010. 08. 23

(86) PCT国际申请的申请数据

PCT/US2009/032793 2009. 02. 02

(87) PCT国际申请的公布数据

W02009/117178 EN 2009. 09. 24

(73) 专利权人 飞思卡尔半导体公司

地址 美国得克萨斯

(72) 发明人 W·C··莫耶 M·D·斯尼德尔

G·L·维森亨特

(74) 专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 陈华成

(51) Int. Cl.

G06F 9/30(2006. 01)

G06F 12/08(2006. 01)

G06F 13/00(2006. 01)

G06F 12/06(2006. 01)

G06F 9/06(2006. 01)

(56) 对比文件

CN 1869952 A, 2006. 11. 29, 全文.

CA 2549640 A1, 2006. 12. 15, 全文.

US 2005/0240820 A1, 2005. 10. 27, 全文.

US 2003/0154463 A1, 2003. 08. 14, 全文.

Yeon-Ho Im 等. Co-development of Media-processor and Source-level Debugger using Hardware Emulation-based Validation. 《6th International Conference on VLSI and CAD, IEEE》. 1999, 95-98.

审查员 狄晓斐

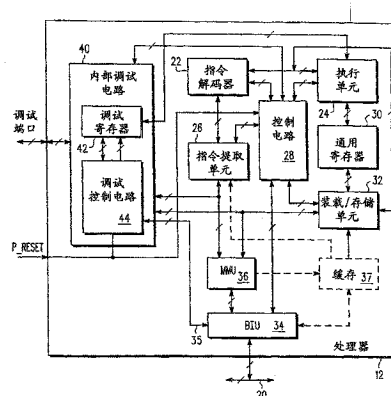
权利要求书2页 说明书9页 附图6页

(54) 发明名称

基于地址的条件调试指令的认定

(57) 摘要

一种支持基于调试指令实例在可寻址存储器中相应的地址为调试指令实例选择执行模式的处理器 (12) 实现方式, 能够提供引人注目的机制, 其中以允许指令的某些实例以调试语义来运行而通过以无操作 (NOP) 语义执行其他实例来抑制它们的方式执行调试指令。在某些实施例中, 可操作的执行语义的选择可以基于其中特定的调试指令实例所驻留的存储器页面 (458, 459) 的属性 (411)。在某些实施例中, 可以 (例如, 使用存储于边界寄存器的值) 界定地址空间部分 (561, 562) 并且将特定的调试指令实例的地址与所界定部分比较以选择适当的执行语义。在某些实施例中, 可以使用这两种评定来给特定的调试指令实例选择适当的执行语义。



1. 一种用于处理器的方法,包括:

在处理器上执行代码,所述代码至少包括相同调试指令的第一调试指令实例及第二调试指令实例;以及

基于第一和第二调试指令实例中的各实例在可由所述处理器寻址的存储器中的相应的地址为该实例选择执行模式,其中基于相应的地址,所述第一调试指令实例以无操作语义来执行,而所述第二调试指令实例以调试相关的语义来执行,

其中所述选择包括:

检查其中驻留所述调试指令的特定实例的存储器页面的调试属性;以及

至少部分地基于所述调试属性,选择所述无操作语义和所述调试相关的语义之一。

2. 根据权利要求1所述的方法,还包括:

保留与虚拟到物理的页面映射信息有关的调试属性。

3. 一种用于处理器的方法,包括:

在处理器上执行代码,所述代码至少包括相同调试指令的第一调试指令实例及第二调试指令实例;以及

基于第一和第二调试指令实例中的各实例在可由所述处理器寻址的存储器中的相应的地址为该实例选择执行模式,其中基于相应的地址,所述第一调试指令实例以无操作语义来执行,而所述第二调试指令实例以调试相关的语义来执行,

其中所述选择包括:

将所述调试指令的特定实例的存储器地址与界定所述存储器的一个或多个相应部分的一个或多个存储单元的内容进行比较;以及

至少部分基于所述存储器地址与所界定部分的对应,选择所述无操作语义和所述调试相关的语义之一。

4. 一种用于处理器的方法,包括:

在处理器上执行代码,所述代码至少包括相同调试指令的第一调试指令实例及第二调试指令实例;以及

基于第一和第二调试指令实例中的各实例在可由所述处理器寻址的存储器中的相应的地址为该实例选择执行模式,其中基于相应的地址,所述第一调试指令实例以无操作语义来执行,而所述第二调试指令实例以调试相关的语义来执行,

其中所述选择包括:

执行页面属性检查以及相对于存储器的所界定部分的比较这两者;以及

基于所述页面属性检查与所界定部分比较的组合,选择所述无操作语义和所述调试相关的语义之一。

5. 根据权利要求1所述的方法,

其中所述选择禁用了所述调试指令的其它运行的调试相关的语义。

6. 根据权利要求1、3、4中的任一项所述的方法,

其中所述选择启用了所述调试指令的其它潜在的调试相关的语义。

7. 根据权利要求1、3、4中的任一项所述的方法,

根据基于地址选择的无操作语义,对所述第一调试指令实例进行解码以产生无操作指令;以及

执行所述无操作指令。

8. 根据权利要求 1、3、4 中的任一项所述的方法,其中所述调试指令选自包括或实施一个或多个以下功能的指令集:

调试通知暂停指令;

断点或监视点指令;以及

调试异常。

基于地址的条件调试指令的认定

技术领域

[0001] 本公开内容一般地涉及数据处理系统,并且更特别地,涉及数据处理系统中调试指令的使用。

背景技术

[0002] 调试指令通常在软件开发期间使用以允许进行调试操作。一旦软件被开发并且利用调试指令检查了,这些调试指令就被去除以免在软件应用中引起所不希望的异常。但是,这些调试指令的去除可能改变系统的执行特性并且在实时应用中可能是特别有问题的。

附图说明

[0003] 通过参考附图,本发明可以被更好的理解,并且使其众多的目的、特征、及优点对本领域技术人员来说变得显而易见。

[0004] 图 1 是根据本发明的一个或多个实施例的数据处理系统的框图。

[0005] 图 2 是根据本发明的一个或多个实施例的处理器框图。

[0006] 图 3 是示出了本发明的一个或多个实施例的地址认定 (qualification) 的条件调试操作的流程图。

[0007] 图 4 示出了本发明的一个或多个实施例使用存储器分页系统 (memory paging system) 属性来促进基于地址的条件调试操作的操作。

[0008] 图 5 示出了本发明的一个或多个实施例使用分段或区域标记技术来促进基于地址的条件调试操作的操作。

[0009] 图 6 示出了本发明的一个或多个实施例的其中使用存储器分页系统和分段或区域标记技术的组合来促进基于地址的条件调试操作的操作。

[0010] 在不同的附图中使用的相同的参考符号指示相似的或相同的项目。

具体实施方式

[0011] 调试指令在各种处理器的实现方式中都得到了支持以促进条件条目进入调试暂停模式中或者促进软件调试异常 (例如,调试中断) 的产生。在任一种情况下,通过将此类调试指令设置于被评定、测试或调试其操作的那一段代码中,开发人员能够在软件控制之下以及根据执行环境的当前状态来启动调试操作。一旦软件开发和 / 或测试完成,除非去除或抑制保留在产品代码中的那些调试指令,否则它们会触发所不希望的异常及调试行为。遗憾的是,调试指令的去除往往改变系统的执行特性。也就是说,由于分支目标、页面边界及其他指令关系在调试指令去除时可能会改变,因此代码映像 (code image) 自身改变。这些改变在实时代码中可能是特别有问题的。

[0012] 因此,在抑制至少其中所选择的调试指令的操作时使产品代码映像能够基本上保持不变的技术是所希望的。由于给出的软件系统的后续生成以及确实甚至新近开发的系统和现有系统的新部署,都可能会再使用其中保留了此类调试指令的代码,如果调试指令的

抑制能够影响某些调试指令实例（例如，驻留于稳定的代码中的那些调试指令实例）而没有影响其他的调试指令实例，那么这将会是所希望的。

[0013] 已经发现，支持基于其在可寻址的存储器中的各自地址选择用于调试指令实例的执行模式的处理器的实现方式能够提供以下引人注目的机制（attractive mechanism），其以允许指令的某些实例以调试语义来运行而以无操作（NOP）的语义执行其他实例来抑制其他实例的方式执行调试指令。在某些实施例中，对运行的执行语义的选择可以基于其中驻留有特定的调试指令实例的存储器页面的属性。在某些实施例中，可以（例如，使用存储于边界寄存器（bounding register）的值）界定部分地址空间并且将特定的调试指令实例的地址与所界定部分进行比较以选择适当的执行语义。在某些实施例中，可以使用这两种评定来为特定的调试指令实例选择适当的执行语义。

[0014] 具体地，我们关注某些说明性的处理器设计、存储器的组织、调试型指令及调试环境。例如，在某些实施例中，实施调试通知暂停（dnh）指令的处理器适合于根据 dnh 指令的实例驻留于其中的存储器地址或区域来给该实例提供选择性的执行语义。在某些实施例中，存储器管理单元依照存储器页面、分段或区域的匹配（或失配）标准注释了指令实例（例如，在它们被从存储器中取出时）并且基于该注释来选择执行语义。在某些实施例中，解码逻辑被用来给由处理器所实施的 NOP 型指令提供代替通常为执行没有给其选择 NOP 语义的调试指令所提供的那些信号或操作码的信号或操作码。当然，在此所描述的技术对于其他调试指令有着广泛的应用以及并广泛用于其他处理器设计中，但是本领域技术人员会根据所说明的内容来了解及理解。

[0015] 因此，鉴于上述内容，但并不限于调试型指令、基础的处理器或系统的架构、地址匹配标准以及选择可以在本发明的实施例中使用的适当的执行语义的机制的范围，我们描述了某些说明性的实施例。

[0016] 一般地，处理器和条件调试

[0017] 图 1 示出了与本发明的某些实施例一致的数据处理系统 10。通常，数据处理系统 10 可以在单个集成电路上或者在多个集成电路上实施。在某些情况下，可以将数据处理系统 10 实施为片上系统。在所示出的配置中，数据处理系统 10 包括处理器 12、外部调试电路 14、I/O 模块 16，及存储器 18。数据处理系统 10 的组件使用任何适合的技术来互连以及互操作。为了简单起见，我们示出了主要的功能块之间经由总线 20 的互连，但本领域技术人员会认识到在没有脱离本发明的情况下可以使用各种互连技术和拓扑的任一种。

[0018] 典型地，处理器 12 的实现方式包括用于存储将由处理器执行的指令的取指缓冲器或其他设备、解码器和顺序逻辑、一个或多个执行单元，以及连同适合的数据、指令及控制通路的寄存器存储单元。在任意给定的时间，与处理器 12 所执行的计算一致的，程序代码（例如，指令）和数据的单元驻留在存储器 18 内，在一个或多个缓存级内和 / 或在处理器存储单元（例如取指缓冲器、寄存器等）中。通常，多种存储器体系中的任一种都可以使用，包括分离或混合在存储器或缓存内的指令和数据的设计。存储器 18（或其任意部分）可以位于与处理器相同的集成电路上，可以位于与处理器 12 不同的集成电路上或者可以跨多个集成电路。另外，存储器 18 可以包括任意适合类型的存储单元，例如，只读存储器（ROM）、随机存取存储器（RAM）、非易失性存储器（例如，闪存）等。外部调试电路 14 可以包含在与处理器相同的集成电路上，或者可以作为独立于任意集成电路或含有处理器 12 的

片上系统的分离系统来实施。

[0019] 图 2 是对应于数据处理系统 10(参见图 1)的处理器 12(现在称为处理器 12)的处理器实例的框图。处理器 12 包括指令解码器 22、执行单元 24、指令提取单元 26、控制电路 28、通用寄存器 30、装载 / 存储单元 32、总线接口单元 (BIU) 34 及内部调试电路 40。处理器 12 经由耦接至 BIU 34 的总线 20 与数据处理系统的其他组件通信。典型地,其他组件包括存储器(可能包括缓存)和 I/O 组件,如先前参考图 1 所示出的。

[0020] 注意,在所示出的实施例中,存储器管理单元 (MMU) 36 协调地址转换(例如,依照所实施的分段或分页的存储器模型)并且提供方便的位置来实施在此所描述的条件调试控制中的某一些。因此,在以下的描述中,我们例示了包括 MMU 设备的某些设计。不过,基于这里的描述,本领域技术人员会认识到沿着从存储器到执行单元 24 的指令通路的其他组件(例如,指令提取单元 26 和指令解码器 22)可以宿存(host)类似的条件调试控制部分或者所描述的特定控制部分,特别是在可以省略 MMU 的实施例中更是如此。图 2 还示出了可以连同某些实施例中的处理器 12 一起提供(例如,在片上)的可选的缓存 37。可选的缓存 37 作为一体化的数据 / 指令缓存而示出,但本领域技术人员会认识到如果需要,可以提供分离的数据和指令部分,并且某些实施例可以只将数据部分或者只将指令部分与处理器 12 共同定位在一起。

[0021] 内部调试电路 40 监测处理器 12 内部的活动并且,响应于根据存在于调试寄存器 42 之内或者处理器 12 内部的其他地方的所存储的调试配置信息而检测到一个或多个预定的条件,可以产生一个或多个数据断点事件、指令断点事件、指令执行事件(例如分支或陷阱捕获事件)、指令完成事件等。内部调试电路 40 可以经由图 2 中所示出的调试端口耦接至外部调试单元,例如 IEEE ISTO-5001 兼容的 Nexus™ 调试单元。外部调试单元可以包括在图 1 中所示出的全部或部分的外部调试电路 14。Nexus™ 是 Freescale Semiconductor, Inc., Austin, Texas 的商标。调试端口可以是串行接口,例如 JTAG,或者可以实施为并行端口、串行和并行端口的结合,或者实施为以太网端口。在某些实施例中,处理器 12 可以使用适当的互连或信号发送机制将内部调试电路连接到片上其他调试电路。

[0022] 如所示的,内部调试电路 40 包括调试寄存器 42 和调试控制电路 44。调试寄存器 42 典型地包括按字段分组的用于控制各种调试相关事件的位,包括指令断点、数据断点、监视点以及与调试关联的其他消息。通常,这些调试资源可以在处理器 12 和外部调试电路 14 之间共享,并且在任何情况下都可以被用来建立根据在此所描述的地址、条件调试控制选择性地启用(或禁用)的断言状态(predicate state)(例如存储器页面属性、标记的分段或区域,和 / 或地址标识符的结合)。

[0023] 通常,调试寄存器 42 给可以用于实施指令和 / 或数据访问断点和监视点事件一个或多个地址比较值、地址范围和数据匹配值以及其他调试控制标准提供存储。这些地址和数据值,连同各种控制标准一起,被用来确定处理器 12 何时访问一个或多个预定的指令地址或数据地址以便产生断点或监视点事件,这能够促使处理器 12 在内部调试模式有效时开始用于调试异常的异常处理,或者促使处理器 12 在外部调试模式有效时进入其中处理器 12 响应由外部调试电路 14 通过内部调试单元 40 的调试端口提供的命令的调试暂停模式。

[0024] 举例来说,但不限于所举实例,调试寄存器 42 可以包括适合于存储调试配置信息

的调试控制寄存器,指令地址比较寄存器和数据地址比较寄存器,以及调试状态寄存器,调试计数器和数据值比较寄存器。通常,调试寄存器 42 可以是用户软件编程模型的可见部分。当一个或多个启用计数的事件发生时,可以将调试计数器配置成向下计数。当计数值达到零时,可以发出调试计数事件的信号并且若已启用则可以生成调试中断。数据值比较寄存器可以存储用于数据比较目的的数据值(例如在条件断点的实现方式中)。

[0025] 在内部调试模式中,寄存器资源由软件管理,并且不需要使用外部调试电路。软件可以通过使用作为软件指令的移入和移出专用寄存器指令的数据移动来为执行基于软件的调试活动而初始化个体调试寄存器来配置寄存器。启用的调试事件触发软件调试中断。然后,软件中断处理器可以执行由数据处理系统 10 的软件编程器确定的各种所期望的活动。

[0026] 在外部调试模式中,可以给外部调试电路 14 分配调试寄存器 42 的共用调试寄存器的所有权,以及当所配置的调试事件发生时,处理器 12 可以进入暂停状态并且等待由外部调试电路 14 所提供的命令。当外部调试模式启用时,软件不再具有对共用调试资源的控制。在某些实施例中,调试寄存器 42 包括位于调试寄存器 42 之内或者位于处理器 12 内部的其他地方的外部调试控制寄存器,其中外部调试控制寄存器可以不是用户软件编程模型的一部分。也就是说,在处理器 12 上执行的软件可能不具有对外部调试控制寄存器的可见性。外部调试电路 14 可以直接地经由调试端口(如在图 2 中所示的)访问共用的调试资源以及任何专用的外部调试资源,其中所述调试端口可以实施为例如 JTAG TAP 端口。在一个实施例中,可以将调试寄存器 42 和外部调试控制寄存器映射为 JTAG 数据寄存器,其中寄存器选择编码包含于各种 JTAG 指令的一个或多个字段之内,所述 JTAG 指令提供了由调试器通过 JTAG IR 和 DR 操作对寄存器的读和写访问。

[0027] 举例来说,但并不限于所举实例,我们现在总结条件调试型指令的两个实例,其给定实例的运行可以(使用在此所描述的技术)根据实例在可由处理器寻址的存储器中出现的地址来认定。实例是由处理器内核(例如可从 Freescale Semiconductor 购得的 e700/e500 和 e200Core 系列的内核)实施的调试通知暂停(dnh)指令和断点指令编码,其用于成本敏感的、嵌入式的实时应用(例如在 MPC5000 系列的自动微控制器(MCU)中)。

[0028] dnh 指令提供了在指令于非暂停模式的内核上的执行与外部调试设备随后的操作之间的桥路。特别地,dnh 指令允许软件将内核从运行状态转换成暂停状态(如果由外部调试器启用了的话),并且以消息和在指令本身内编码的位两者来通知外部调试器。如典型实施的那样,dnh 指令是其中其运行时间的效果由可以设置于调试控制寄存器(参见例如,图 2,调试寄存器 42)内的某些指示的状态所确定的条件调试指令。例如,在一种实现方式中,如果 DNH_NOP 指示由外部调试设备或由软件所设置并且 DNH_EN 指示被清除,则 dnh 指令实例作为 nop 来执行。如果 DNH_WPT 指示被设置了,则将监视点的信号发送到 CPU 的外部。

[0029] 粗粒(coarse grain)控制机制(例如,DNH_NOP 和 DNH_EN 指示的适当使用)允许在需要时将 dnh 指令作为 nop 来处理,而不是暂停处理器使其处于调试状态或者捕获非法指令异常。这种控制允许 dnh 指令在代码开发之后保持嵌入于应用的代码映像中,然而通过简单地充当 nop 操作仍然是无害的。通过这种方式,代码映像本身不需要改变,并且由于分支目标、页面边界及其他指令关系保持不变,因而提供了可预测的执行。但是,dnh 型指

令的全部（乃至整个分组的）实例的粗粒控制可以不提供充分的粒度以允许将在稳定的代码中的 dnh 指令实例作为 nop 来处理，然而同时允许另一 dnh 指令实例作为暂停或监视点来运行。

[0030] 注意，在 2007 年 10 月 12 日提交的，题目为“DEBUGINSTRUCTION FOR USE IN A DATA PROCESSING SYSTEM”并且发明人为 Moyer、Snyder 和 Whisenhunt 的共同拥有的美国申请 No. 11/871,847 公开了先前描述的 dnh 指令的变体 dnh2，其中 dnh2 指令的格式允许将个体 dnh2 指令实例与有限数量的控制分组中的一个关联起来，使得特定分组的控制机制（例如，特定分组 GROUPx.DNH_NOP 和 GROUPx.DNH_EN 指示的使用）可以被用来给每个分组确定适当的执行语义。虽然作为说明我们关注 dnh 指令，但是基于这里的描述，本领域技术人员会意识到可以将在此所描述的地址认定的调试控制应用于实施指令变体（例如 dnh2 指令）的设计。

[0031] 类似地，断点 (bkpt) 指令在本发明的某些实施例中能够用作提供了地址认定的调试指令。在某些指令集的实现方式中，断点指令可以不由指令集架构 (ISA) 明确定义，但是可以实施为非法指令的操作码，该非法指令的操作码在执行时产生可以分派给调试设备来处理的异常或故障。在某些 ISA 和实现方式中，可以定义和提供 bkpt 指令操作码（或某些相似的指令编码）。为了描述清晰，但并不限制或损失一般性，我们谈及断点指令，好像断点指令是所定义的指令集编码（例如，bkpt 指令）那样。

[0032] 如前所述，粗粒控制机制允许在需要时将 bkpt 指令作为 nop 来处理，而不是暂停处理器使其处于调试状态。这种控制允许 bkpt 指令保持嵌入于代码开发后的应用的代码映像内，然而通过简单地充当 nop 操作而仍然是无害的。如前所述，由于分支目标、页面边界以及其他指令关系保持不变，可以提供可预测的执行。但是，同样如前所述，bkpt 指令的全部（或一组）实例的粗粒控制可以不提供足够的粒度 (granularity) 以允许将在稳定的代码中的 bkpt 指令实例作为 nop 来处理，然而同时允许另一 bkpt 指令实例作为断点来操作。

[0033] 因此，我们描述了以下技术，其中可以将调试指令实例通过地址的认定用来特殊化 dnh、bkpt 和 / 或其他调试型指令的特定实例的行为。根据这里的描述，本领域技术人员会想到实施相似的或类似的调试指令的处理器和系统的广泛应用。

[0034] 图 3 是示出了根据本发明的一个或多个实施例的地址认定的条件调试技术的建立和操作的流程图。在附图的左侧，对于地址认定的条件调试示出了建立 301 序列。例如，在某些实施例中，考虑 (311) 将个体代码块用于禁用（或启用）调试型指令实例，例如其中所出现的 dnh、dnh2、bkpt 等。作为一般命题 (general proposition)，代码块可以具有任意适合的粒度并且可以对应于函数、过程、模块、指令的子序列等。使用由地址认定器 (qualifier) 的存储单元 310 提供的任何粒度，建立序列 301 存储用于改变个体调试指令实例的运行或行为的对应的页面框的标识符、分段标识符、区域标识符（例如，作为基址 / 掩码对）和 / 或任何其他基于地址的认定器。

[0035] 在处理器（例如处理器 12（回想图 2））上的代码执行 (302) 期间，典型地从存储器或缓存中取出 (321) 每条指令（或指令组）以为指令（或该组指令）依照程序序列的解码以及最终的执行作准备。对于调试指令实例，相对于存储单元 310 中基于地址的认定器来检查 (322) 特定的存储器地址，其中特定的实例在存储器中在特定的存储器地址处找

到。如果在存储单元 310 中所指示的存储器地址的页面、分段或范围匹配 (323) 调试型指令实例的特定存储器地址,则对于该指令实例抑制 (324) 调试执行语义。典型地,由指令提取或解码单元 (参见例如图 2,指令提取单元 26,指令解码器 22) 将匹配的指令实例替换为 nop 指令操作码,但是如果需要还可以使用其他的抑制机制。最终,相应的指令 (在调试执行语义被抑制时) 被执行 (325)。

[0036] 图 4 示出了本发明的一个或多个实施例的操作,其中存储器分页系统属性被用来促进基于特定的调试指令实例的存储器地址的条件调试操作。如前面所解释的,基于地址的认定器以基于由调试硬件和 / 或软件所提供的指示的存储单元来表示。更特别地,在图 4 的说明中,将调试相关的页面框属性提供 (405) 并编码于作为页面转换 410 集体示出的一个或多个分页存储器管理存储单元 (例如,页表、转换后备缓冲器 (TLB) 条目等) 中。本领域技术人员会想到 MMU 450 以及,更一般地,分页存储器管理系统的多种适合的实现方式中的任一种。为了清晰起见,但并不限于适合的实现方式的范围,我们只是说明了简化的分页系统中与我们的技术的说明有关的那些方面。

[0037] 常规地,MMU 450 提供了虚拟地址 (例如,指令提取地址 427) 到可寻址存储单元内的物理地址 (例如,与存储器和 / 或缓存 420 内的位置 471 对应的物理地址 499) 的转换。与存储器分页系统的说明一致的,可寻址的存储单元被视为处于页面框水平的粒度,从而使用页面转换 410 将与虚拟地址 498 对应的页面框映射到物理地址空间中的相应页面 (例如,物理页面 458)。同样常规地,对页面属性 411 被编码为与映射 412 相关联。例如,在某些存储器分页系统的实现方式中,可以将直写 (W)、缓存禁止 (I)、存储器相干性要求 (M)、受保护 (G) 及字节序 (E, endianness) 属性指定为在与个体转换关联的页面属性 411 内的构成指示。为了支持我们的地址认定的条件调试技术,我们将附加的调试禁用 (D) 属性添加到页面属性 411,这可以 (在某些实现方式中) 使用在分页系统实现方式中特别定义的属性或者 (在其他实现方式中) 使用另外未分配的用户可定义的页面属性来提供。无论是什么特定的编码,从与存储于所寻址的位置 471 的特定调试型指令实例的提取有关的页面转换 410 取回的特定的调试禁用 (D) 指示 413 被用来确定所提取的指令实例的执行语义。

[0038] 在图 4 的说明中,虚拟到物理的页面转换 418 和 419 对应于物理地址空间内相应的物理页面 458 和 459。因此,基于页面转换 410 中相应的调试禁用 (D) 指示集 (405),从页面 458 和 459 内的物理地址提取的调试型指令实例以 nop 语义来执行。为了说明的目的,我们假定,在所示出的实施例中,集合调试禁用 (D) 指示 413 指出相应的调试型指令应当以 nop 语义来执行。当然,可以使用其他编码含义。

[0039] 通常,nop 语义可以用多种方式中的任一种来提供。例如,MMU450 可以注释 (452) 所提取的调试型指令实例 (或者调试型指令实例出现于其中的缓存线路或其他提取单元) 以为了后面的替换 (或者过载) 而在沿着指令通路的便利点 (convenient point) 处标记它。特别地,这样注释的调试型指令实例的操作码可以用在提取单元 426 (例如,在 453) 或者在指令解码器 422 (例如,在 454) 的 nop 指令操作码来代替 (或过载)。作为选择,MMU 450 能够自己替换或过载 nop 指令操作码,但这假定指令集知识的级别一般不提供于存储器管理单元中。

[0040] 图 5 示出了本发明的一个或多个实施例的操作,其中分段或区域标记技术被用来促进基于特定的调试指令实例的存储器地址的条件调试操作。如前所述,基于地址的认定

器在存储单元中基于由调试硬件和 / 或软件所提供 (505) 的指示来表示。更特别地,在图 5 的说明中,地址基址 / 掩码对 (511... 512) 被用来标记在存储器和 / 或缓存 518 中表示的可寻址的存储单元的相应分段或区域 (561、562)。使用比较器集 (516... 517) 将指令提取地址 527 与地址基址 / 掩码对比较并且根据匹配来提供引起相应的调试型指令实例的注释 (552) 和 / 或替换 / 过载 (在沿着指令通路的便利点处) 的指示 513。

[0041] 在图 5 的说明中,从可寻址的存储器内的位置 571 取回的指令实例的调试型执行语义根据指令提取地址 527 与一个或多个地址基址 / 掩码对 (511... 512) 的对应关系来抑制。与前面的描述一致的,我们假定地址基址 / 掩码对指出相应的调试型指令将以 nop 语义来执行,但是也可以使用其他的编码含义。如前所述,沿着指令通路的便利点包括提取单元或指令解码器。

[0042] 建立于前面的实例上,图 6 示出了本发明的一个或多个实施例的操作,其中存储器分页系统和标记技术的组合被用来促进基于地址的条件调试操作。如前所述,基于地址的认定器在存储单元中基于由调试硬件和 / 或软件提供 (605) 的指示来表示。但是,在图 6 的说明中,页面框属性和地址基址 / 掩码对两者都提供了。如前所述,将页面框属性编码于作为页面转换 610 集中示出的一个或多个分页存储器管理存储单元 (例如,页表、转换后备缓冲器 (TLB) 条目等) 中。特别地,其条目 618 (其与指令提取地址 627 对应) 提供了将页面 658 作为在位置 671 (以及所提取的调试指令) 所驻留的可寻址的存储器中的页面标识的转换。

[0043] 在图 6 的说明中,地址基址 / 掩码对 (例如,地址基址 / 掩码对 611) 被用来标记在页面 658 内的相应区域 661 并且被用来进一步界定可寻址的存储单元中调试型执行语义将被抑制的部分。特别地,图 6 示出了 (i) 在指令提取地址 627 与地址基址 / 掩码对 611 的内容之间的比较 (616) 结果以及 (ii) 从与存储于所寻址的位置 671 的特定的调试指令实例的相应取回有关的页面转换 610 取回的调试禁用 (D) 指示 613 的 AND (与) 组合 (参见逻辑 699)。因此,所示出的配置提供了用于将要抑制其调试执行语义的存储器位置的集合缩小到出现于具有调试禁用 (D) 指示的页面内的存储器位置的子集的机制。

[0044] 当然,页面和区域指示器的其他组合是可能的且可预见的。例如,在一种变体中,NOT-AND 组合能够被用来排除与另外落在已经对调试禁用 (D) 指示编码的页面范围之内的可寻址位置的地址基址 / 掩码对 611 匹配的可寻址位置。类似地,在另一种变体中,OR 组合能够被用来包括与地址基址 / 掩码对 611 匹配的或者落在已经对调试禁用 (D) 指示编码的页面范围之内的可寻址位置。而且,虽然我们示出了两个输入的简单的、单级的组合,但是如果需要也可以使用更大数量的输入以及更复杂的组合方案。在每一种情况下,都给对应的调试型指令实例执行注释 (例如,652) 和 / 或替换 / 过载 (在沿着指令通路的便利点处)。

[0045] 参考图 6 的特别说明,从可寻址存储器中的位置 671 取回的指令实例的调试型执行语义基于指令提取地址 627 与 (i) 其中出现了集合调试禁用 (D) 指示 613 的转换 610 的条目 (618) 以及 (ii) 地址基址 / 掩码对 (611) 的内容这两者的对应来抑制。与上文一致的,我们假定调试禁用 (D) 指示和地址基址 / 掩码对覆盖了将以 nop 语义执行的调试型指令实例,但也可以使用其他编码含义。同样如前所述,以 nop 操作码替换 / 过载的沿着指令通路的便利点包括提取单元或指令解码器。

[0046] 其他实施例

[0047] 进程标识符 (PID) 或地址空间标识符 (ASID) 通常使用于处理器的实现方式中并且能够被用来给每个执行进程提供其自身唯一的虚拟地址空间。因此,在本发明的某些实施例中,该 PID 或 ASID 字段值的全部或部分可以用于建立希望调试型或 nop 的执行语义的一组地址的比较。例如,PID 或 ASID 字段值的此类使用可以包括(或排除)落在与处于(或来自)指定了特定的执行语义的集合中的特定的进程或线程关联的虚拟地址空间之内的那些地址。

[0048] 在本发明的某些实施例中,可以使用虚拟化技术。虚拟化通常建立不同的地址空间中彼此独立操作的逻辑分区。例如,在某些多线程处理器的实现方式中,可以将每个虚拟处理器分配给逻辑分区。类似地,在多处理器系统中,可以将每个物理处理器分配给逻辑分区。通常,逻辑分区的分配可以通过分配分区 ID(LPID) 值来实现。分区 ID 值形成了虚拟地址的扩展并且在地址转换期间能够被用来相对于每个 TLB 条目的逻辑分区值进行匹配。因此,在本发明的某些实施例中,可以包含 LPID 字段值的全部或部分作为范围比较逻辑的输入以使得一个或多个虚拟地址空间或分区进一步规范。

[0049] 通常,地址空间比较(基于 PID、ASID 或 LPID 字段值的使用或者基于某一虚拟化设备的使用)可以单独使用或者可以结合参考图 3-6 所说明的或所描述的查找或比较来使用。可以论证,相似的功能能够通过在与地址空间对应的那些页面上标记适当的页面属性来提供;但是,标记/不标记页面可能是耗时的或不便的。因此,在某些实施例中,上述替代方案可能是有吸引力的。

[0050] 尽管本发明在此参考具体的实施例来描述,但是在没有脱离下面的权利要求中所阐述的本发明的范围的情况下能够进行各种修改和改变。例如,虽然我们已经针对某些说明性的调试指令和存储器架构描述了技术,但是我们的技术不必限于此。同样,在一种实施例中,调试指令的种类允许将调试指令分成独立的分组,其中每个分组具有对执行时所采用的动作的独立控制,由此提供附加的调试灵活性。这些分组可以被独立控制,以及所引起的动作可以由硬件或软件调试器动态地修改。同样可以使用本发明的技术来给分组界定的调试指令提供地址认定。

[0051] 本发明的实施例可以使用多种不同的信息处理系统中的任一种来实施。因此,虽然图 1 和 2,连同它们的附随描述一起涉及了示例性的数据处理系统和处理器架构,但是这些示例性的架构仅仅是说明性的。当然,为了讨论的目的已经简化这里的架构的描述并且本领域技术人员会认识到所说明的在逻辑块或组件之间的边界只是说明性的以及替代的实施例可以合并逻辑块或电路元件和/或将功能的替代分解施加到各种逻辑块或电路元件之上。

[0052] 实现本发明的物品、系统及装置大部分包括本领域技术人员所知道的且在此在功能上描述的电子组件、电路和/或代码(例如,软件、固件和/或微代码)。因此,对组件、电路及代码的细节在出于清晰、具体以及促进对本发明的基础概念的了解和理解目的所需的细节水平上进行解释。在某些情况下,对本领域已知的特征、结构、组件或实施技术采用一般化描述以便避免对本发明的教导造成困惑或扰乱。

[0053] 通常,术语“程序”和/或“程序代码”在此被用来描述用于在计算机系统上执行所设计的指令的序列或集合。同样地,此类术语可以包括或包含设计为在计算机系统上执

行的子程序、函数、过程、对象方法、软件方法的实现方式、接口或对象、可执行应用程序、小程序、小服务程序、源代码、目标代码或中间代码、共享库和 / 或动态装载 / 连接库和 / 或其他指令序列号或指令组。

[0054] 在此所描述的程序代码的全部或部分, 以及任何在此所描述的信息处理系统的软件实施功能, 可以由信息处理系统的元件来访问或接收, 例如, 从计算机可读的介质中或者经由其他系统访问或接收。通常, 计算机可读的介质可以永久地、可移动地或远程地耦接至信息处理系统。计算机可读的介质例如可以包括, 但不限于, 任意数量的下列设备: 磁存储介质, 包括盘和带存储介质; 光学存储介质, 例如光盘介质 (例如, CD-ROM、CD-R 等) 和数字视频盘存储介质; 非易失性存储器存储介质, 包括基于半导体的存储器单元, 例如闪速存储器、EEPROM、EPROM、ROM; 铁磁数字存储器; MRAM; 易失性存储介质, 包括寄存器、缓冲器或缓存、主存储器、RAM 等; 以及数据传输的相关介质, 包括经由计算机网络的传输装置、点对点通信设备, 以及载波或信号, 以上仅列举几例。

[0055] 最后, 说明书和附图应当被看作是说明性的而非限制性的, 并且与这里的描述一致的, 范围广泛的变体、修改及外延伸是可预见的。在此针对具体实施例所描述的任何益处、优点或问题的解决方案并非意图被看作是任一或全部权利要求的关键的、必需的或本质的特征或元素。

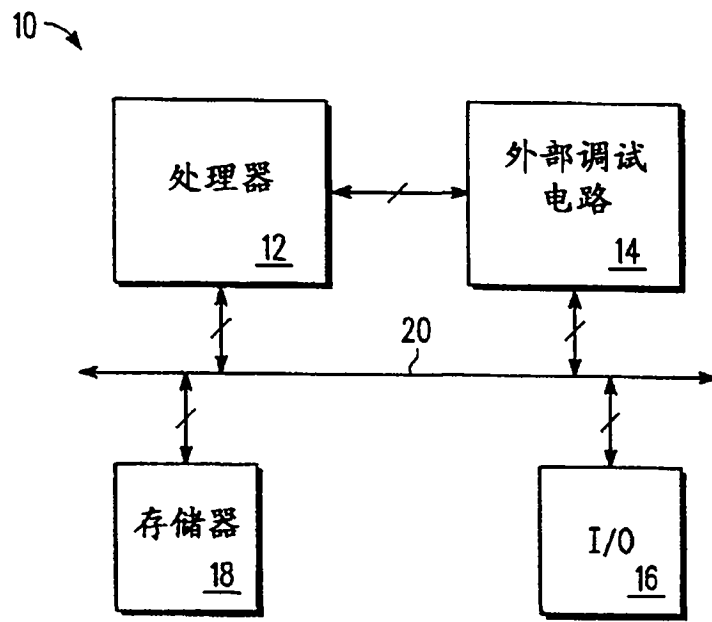


图 1

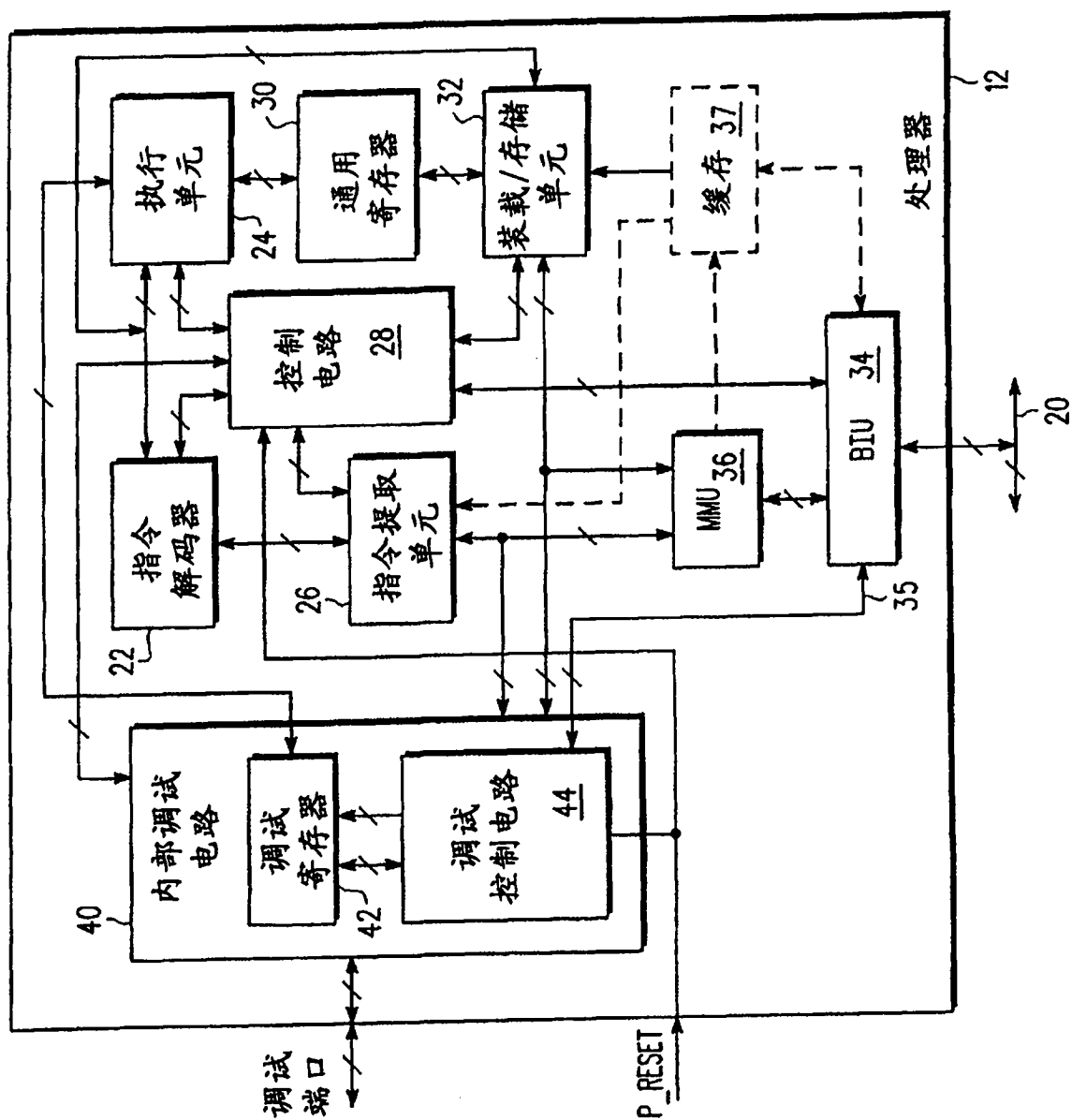


图 2

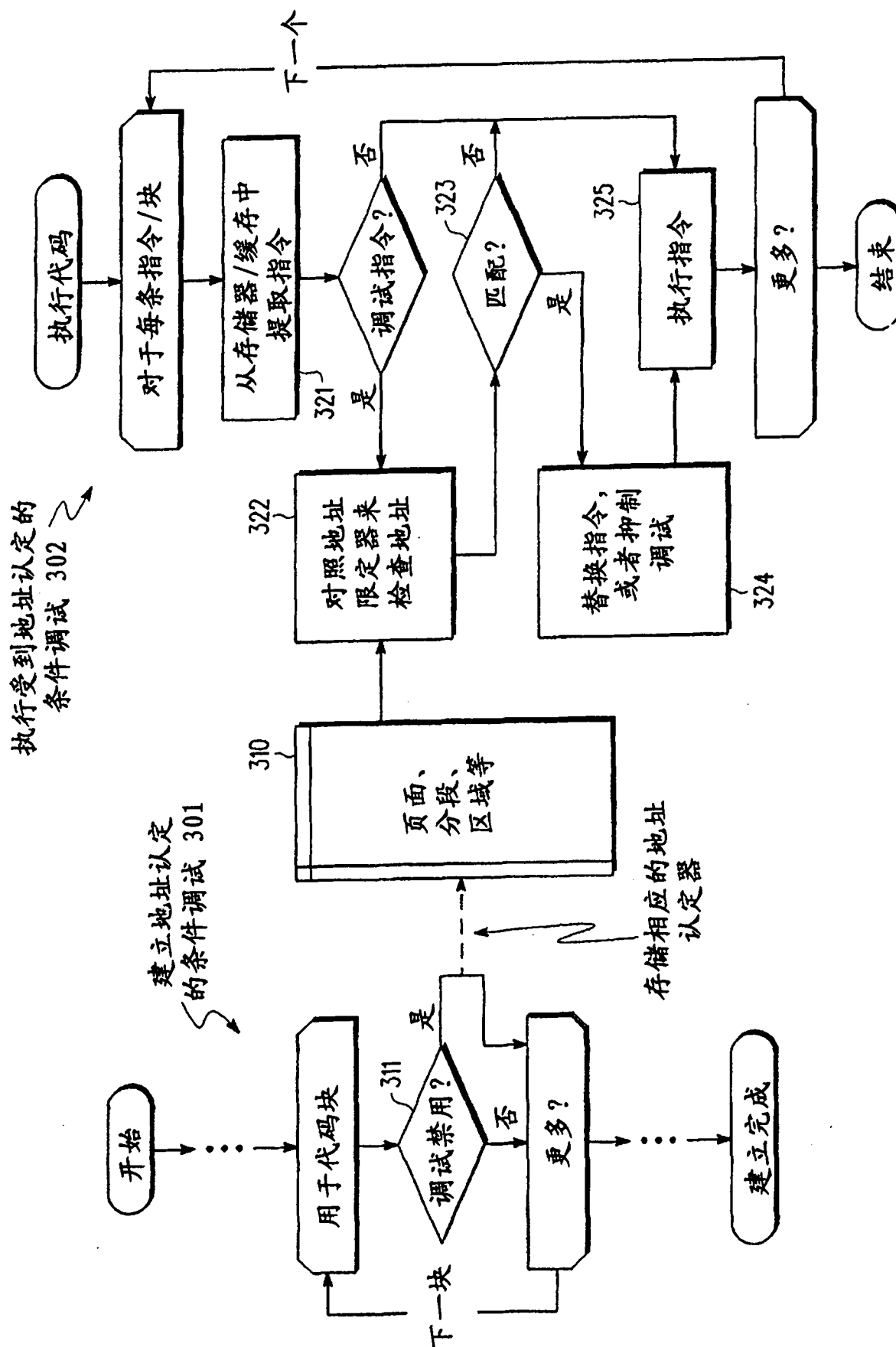


图 3

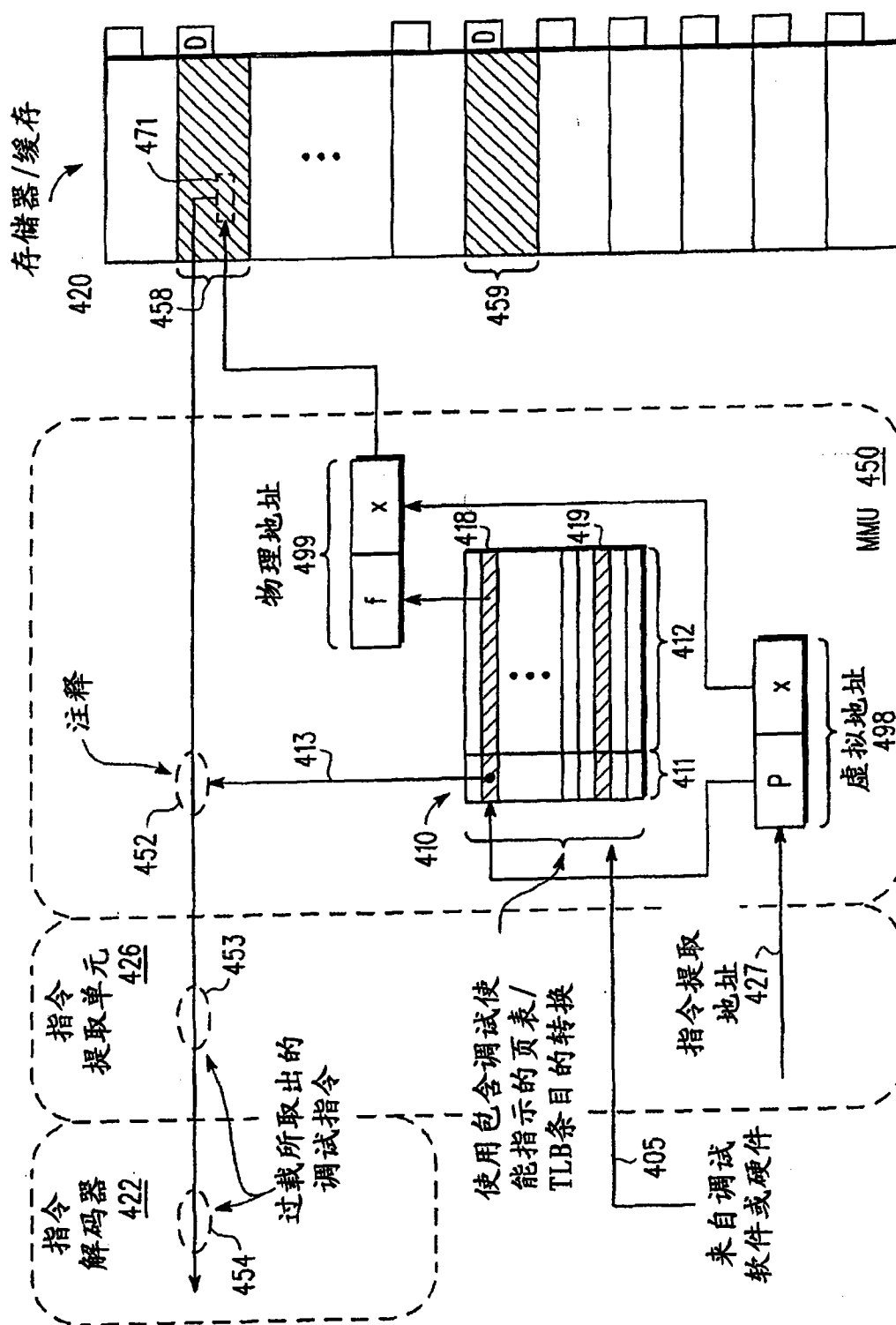


图 4

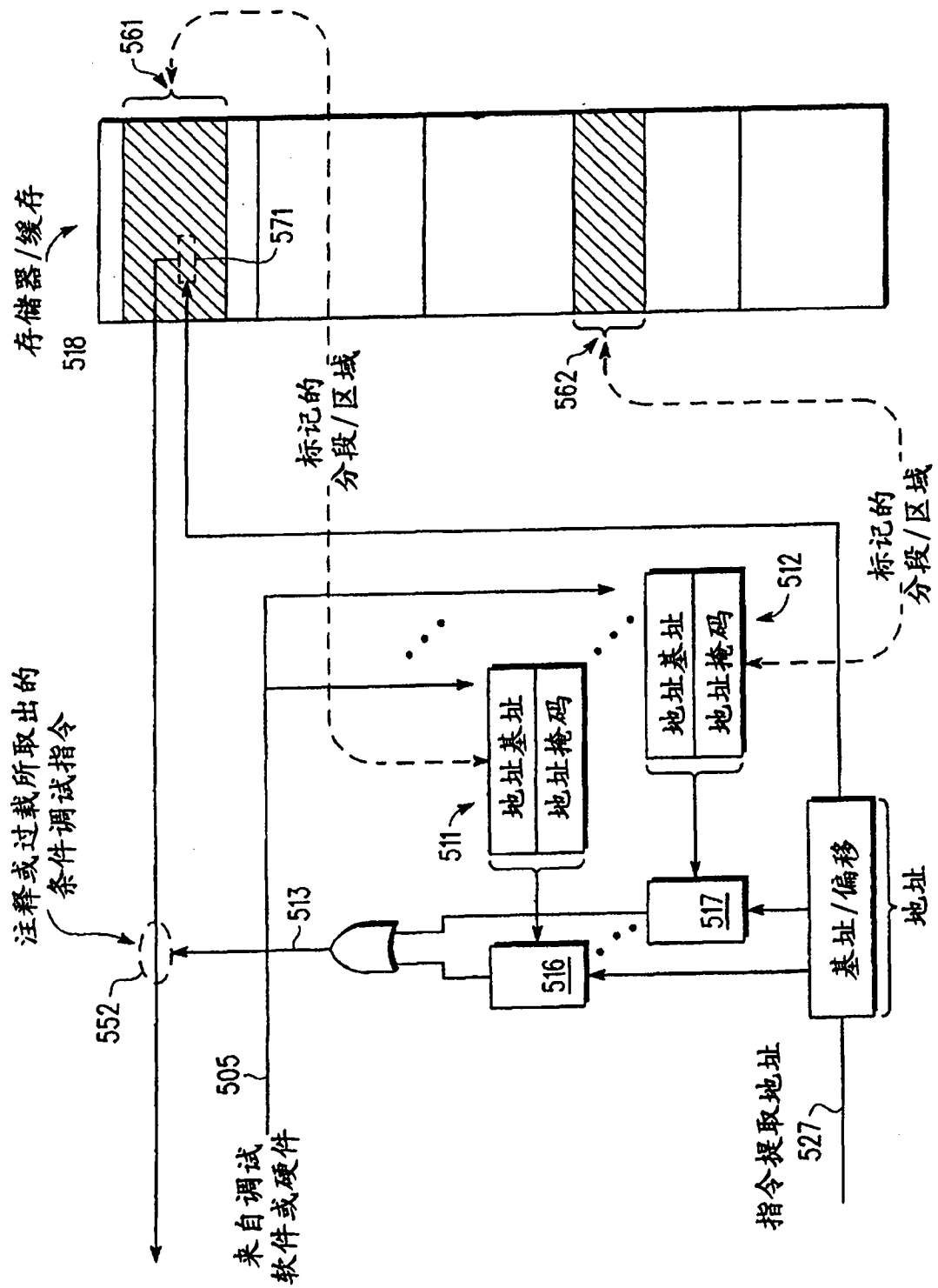


图 5

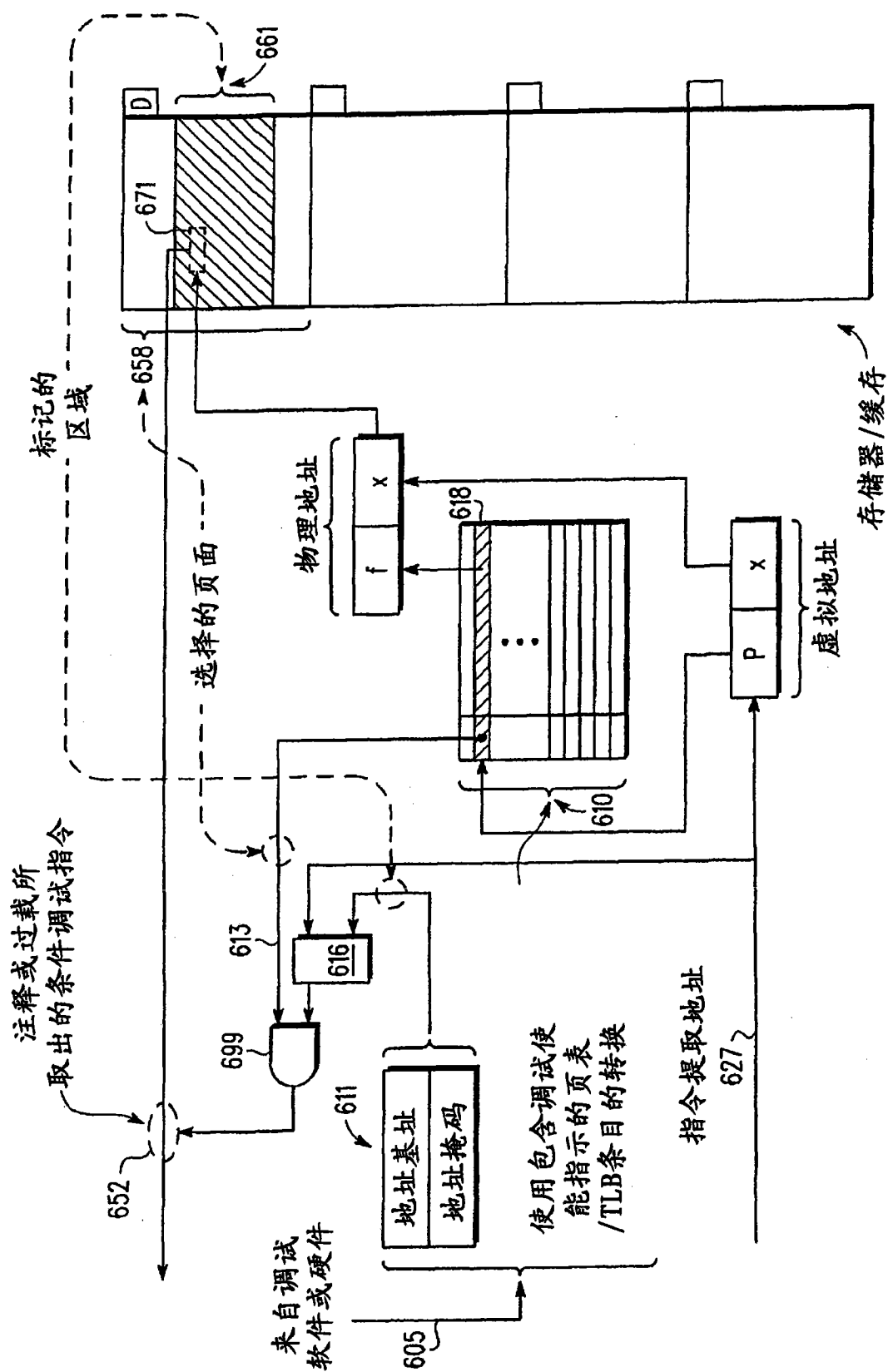


图 6