(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 120077369 A
(43)申请公布日 2025.05.30

(21)申请号 202380074203.2
(22)申请日 2023.09.20
(30)优先权数据
2215944.6 2022.10.27 GB
(85)PCT国际申请进入国家阶段日
2025.04.21
(86)PCT国际申请的申请数据
PCT/GB2023/052441 2023.09.20
(87)PCT国际申请的公布数据
WO2024/089383 EN 2024.05.02
(71)申请人 ARM有限公司
地址 英国剑桥
(72)发明人 约翰·迈克尔·霍利
西蒙·约翰·克拉斯克
(74)专利代理机构 北京东方亿思知识产权代理有限责任公司 11258
专利代理师 杨佳婧
(51)Int.Cl.
G06F 12/14(2006.01)

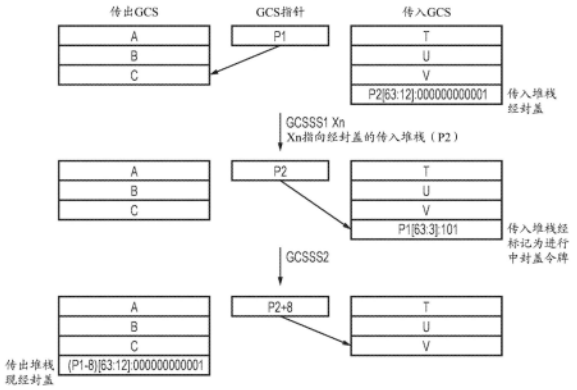
权利要求书3页 说明书22页 附图11页

(54)发明名称

堆栈指针切换有效性检查

(57)摘要

处理电路16执行与该堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作。该有效性检查操作包括验证通过存储器存取电路26响应于指定基于该传入堆栈指针值确定的地址的存储器存取请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件,该至少一堆栈封盖值有效性条件包含该传入数据值的预定部分对应于指示包括基于该传入堆栈指针值确定的该地址的地址空间的分页的给定分页地址的条件。该至少一堆栈封盖值有效性条件是无关于该传入数据值的除了该预定部分以外的另一部分是否对应于基于该传入堆栈指针值确定的该地址的子分页地址位而确定的。错误处理响应是响应于确定该传入数据值不符合该至少一堆栈封盖值有效性条件而触发的。



CN 120077369 A

1.一种装置,所述装置包括:

存储器存取电路,所述存储器存取电路用以基于堆栈指针执行对堆栈数据结构的堆栈存取;和

处理电路,所述处理电路用以执行与所述堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作,所述堆栈指针切换有效性检查操作包括验证通过所述存储器存取电路响应于指定基于所述传入堆栈指针值确定的地址的存储器存取请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件,所述至少一堆栈封盖值有效性条件包含所述传入数据值的预定部分对应于指示包括基于所述传入堆栈指针值确定的所述地址的地址空间的分页的给定分页地址的条件,其中:

所述处理电路被配置成确定所述传入数据值是否符合所述至少一堆栈封盖值有效性条件,无关于所述传入数据值的除了所述预定部分以外的另一部分是否对应于基于所述传入堆栈指针值确定的所述地址的子分页地址位;并且

所述处理电路被配置成响应于确定所述传入数据值不符合所述至少一堆栈封盖值有效性条件而触发错误处理响应。

2.根据权利要求1所述的装置,其中所述给定分页地址是指示包括基于所述传入堆栈指针值确定的虚拟地址的虚拟地址空间的分页的虚拟分页地址。

3.根据权利要求1和2中任一项所述的装置,其中所述处理电路被配置成执行与所述堆栈指针从所述传出堆栈指针值到所述传入堆栈指针值的所述切换相关联的传出堆栈封盖操作,以将有效堆栈封盖值推入到具有基于所述传出堆栈指针值选择的地址的位置,所述有效堆栈封盖值在所述有效堆栈封盖值的所述预定部分中指定指示包括基于所述传出堆栈指针值选择的所述地址的地址空间的分页的分页地址。

4.根据权利要求3所述的装置,其中在所述有效堆栈封盖值中指定的所述分页地址是指示包括基于所述传出堆栈指针值选择的虚拟地址的虚拟地址空间的分页的虚拟分页地址。

5.根据任一前述权利要求所述的装置,其中所述堆栈指针是保护控制堆栈(GCS)指针,所述保护控制堆栈指针用于控制对保护控制堆栈(GCS)数据结构的存取,以用于保护从异常或功能调用返回的返回地址。

6.根据权利要求5所述的装置,所述装置包括存储器管理电路,所述存储器管理电路用以基于与所述目标地址相关联的存储器属性数据而确定是否允许对目标地址的存取,所述存储器属性数据指定包含所述目标地址的目标存储器地址空间区域是否是用于存储所述GCS数据结构的GCS区域,其中到所述GCS区域的写入存取受限于专用类别的GCS存取指令。

7.根据权利要求6所述的装置,其中当所述存储器属性数据指定所述目标存储器地址空间区域是所述GCS区域时,所述存储器管理电路被配置成拒绝指令提取或到所述目标地址的分支。

8.根据权利要求6和7中任一项所述的装置,其中所述存储器管理电路被配置成响应于确定所述目标存储器地址空间区域是所述GCS区域,而拒绝通过除了所述专用类别的GCS存取指令以外的存储指令所触发的到所述目标地址的非GCS存储操作。

9.根据权利要求6至8中任一项所述的装置,其中所述存储器管理电路被配置成响应于确定所述目标存储器地址空间区域不是所述GCS区域,而拒绝通过所述专用类别的GCS存取

指令中的一者所触发的到所述目标地址的GCS载入/存储操作。

10.根据任一前述权利要求所述的装置,其中所述至少一堆栈封盖值有效性条件还包括所述传入数据值的最低有效部分具有不能够由任何有效指令地址的所述最低有效部分指定的位模式的条件。

11.根据前一权利要求所述的装置,其中所述处理电路被配置成响应于第一堆栈指针切换指令指定指示所述传入堆栈指针值的操作数,而执行所述堆栈指针切换有效性检查操作。

12.根据权利要求11所述的装置,所述装置包括堆栈指针寄存器,所述堆栈指针寄存器用以存储所述堆栈指针;其中:

响应于所述第一堆栈指针切换指令,当所述堆栈指针切换有效性检查操作成功时,所述处理电路被配置成将所述堆栈指针寄存器从所述传出堆栈指针值更新到所述传入堆栈指针值。

13.根据权利要求12所述的装置,其中响应于所述第一堆栈指针切换指令,所述处理电路被配置成将指定所述传出堆栈指针值的进行中令牌值推入到具有基于所述传入堆栈指针值选择的地址的位置。

14.根据权利要求13所述的装置,其中对于用以载入待在所述堆栈指针切换有效性检查操作中验证的所述传入数据值的载入操作、以及用以将所述进行中令牌值推入到具有基于所述传入堆栈指针值选择的所述地址的所述位置的存储操作,所述处理电路被配置成原子地执行所述载入操作和所述存储操作。

15.根据权利要求13和14中任一项所述的装置,其中响应于第二堆栈指针切换指令,所述处理电路被配置成:

验证通过所述存储器存取电路响应于指定基于所述堆栈指针寄存器中的目前堆栈指针确定的地址的存储器存取请求而获得的给定数据值是否是有效地形成的进行中令牌值;以及

响应于验证所述给定数据值是有效地形成的进行中令牌值:

触发有效堆栈封盖值到具有基于由所述给定数据值的一部分指定的给定堆栈指针值确定的地址的位置的写入,所述有效堆栈封盖值在所述有效堆栈封盖值的所述预定部分中指定指示包括基于所述给定堆栈指针值确定的地址的地址空间的分页的分页地址。

16.根据权利要求15所述的装置,其中响应于所述第二堆栈指针切换指令,当所述给定数据值被验证为是所述有效地形成的进行中令牌值时,所述处理电路被配置成更新所述堆栈指针寄存器中的所述目前堆栈指针,以指示将提供所述给定数据值的条目从对应堆栈数据结构去除。

17.根据权利要求15和16中任一项所述的装置,其中当所述给定数据值的最低有效部分具有不能够由任何有效指令地址的所述最低有效部分指定且不能够由符合所述至少一堆栈封盖值有效性条件的任何值指定的位模式时,所述处理电路被配置成验证所述给定数据值是所述有效地形成的进行中令牌值。

18.根据权利要求15至17中任一项所述的装置,其中所述处理电路被配置成响应于确定所述给定数据值不是有效地形成的进行中令牌值而触发错误处理响应。

19.根据任一前述权利要求所述的装置,其中所述错误处理响应包括下列中的至少一

者:

发信号通知错误;

设定错误报告指示;以及

将所述堆栈指针设定成无效值。

20.一种方法,所述方法包括:

执行与堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作,所述堆栈指针切换有效性检查操作包括验证通过存储器存取电路响应于指定基于所述传入堆栈指针值确定的地址的存储器存取请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件,所述至少一堆栈封盖值有效性条件包含所述传入数据值的预定部分对应于指示包括基于所述传入堆栈指针值确定的所述地址的地址空间的分页的给定分页地址的条件,其中:

所述传入数据值是否符合所述至少一堆栈封盖值有效性条件是无关于所述传入数据值的除了所述预定部分以外的另一部分是否对应于基于所述传入堆栈指针值确定的所述地址的子分页地址位而确定的;以及

所述方法包括响应于确定所述传入数据值不符合所述至少一堆栈封盖值有效性条件而触发错误处理响应。

21.一种包括指令的计算机程序,当所述指令由主机数据处理装置执行时,控制所述主机数据处理装置提供用于执行目标代码的指令执行环境,所述计算机程序包括:

存储器存取程序逻辑,所述存储器存取程序逻辑用以基于堆栈指针执行对堆栈数据结构的堆栈存取;和

处理程序逻辑,所述处理程序逻辑用以执行与所述堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作,所述堆栈指针切换有效性检查操作包括验证通过所述存储器存取程序逻辑响应于指定基于所述传入堆栈指针值确定的地址的请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件,所述至少一堆栈封盖值有效性条件包含所述传入数据值的预定部分对应于指示包括基于所述传入堆栈指针值确定的所述地址的地址空间的分页的给定分页地址的条件,其中:

所述处理程序逻辑被配置成确定所述传入数据值是否符合所述至少一堆栈封盖值有效性条件,无关于所述传入数据值的除了所述预定部分以外的另一部分是否对应于基于所述传入堆栈指针值确定的所述地址的子分页地址位;并且

所述处理程序逻辑被配置成响应于确定所述传入数据值不符合所述至少一堆栈封盖值有效性条件而触发错误处理响应。

22.一种存储介质,所述存储介质存储根据权利要求21所述的计算机程序。

堆栈指针切换有效性检查

[0001] 本技术涉及数据处理领域。

[0002] 堆栈指针可用以管理对存储器中的堆栈数据结构的存取。堆栈数据结构是对堆栈上的条目的存取根据后进先出(LIFO)策略受到管理的数据结构。堆栈指针提供堆栈上的项的地址可相对于其确定的地址。随着项被推入到堆栈且从堆栈弹出,堆栈指针经更新以跟踪表示堆栈的“顶部”的位置。

[0003] 至少一些示例提供一种装置,该装置包括:存储器存取电路,该存储器存取电路用以基于堆栈指针执行对堆栈数据结构的堆栈存取;和处理电路,该处理电路用以执行与该堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作,该堆栈指针切换有效性检查操作包括验证通过该存储器存取电路响应于指定基于该传入堆栈指针值确定的地址的存储器存取请求而获得的传入数据值是否符合至少一堆栈封盖值(stack cap value)有效性条件,该至少一堆栈封盖值有效性条件包含该传入数据值的预定部分对应于指示包括基于该传入堆栈指针值确定的该地址的地址空间的分页的给定分页地址的条件,其中:该处理电路被配置成确定该传入数据值是否符合该至少一堆栈封盖值有效性条件,无关于该传入数据值的除了该预定部分以外的另一部分是否对应于基于该传入堆栈指针值确定的该地址的子分页地址位;且该处理电路被配置成响应于确定该传入数据值不符合该至少一堆栈封盖值有效性条件而触发错误处理响应。

[0004] 至少一些示例提供一种方法,该方法包括:执行与堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作,该堆栈指针切换有效性检查操作包括验证通过存储器存取电路响应于指定基于该传入堆栈指针值确定的地址的存储器存取请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件,该至少一堆栈封盖值有效性条件包含该传入数据值的预定部分对应于指示包括基于该传入堆栈指针值确定的该地址的地址空间的分页的给定分页地址的条件,其中:该传入数据值是否符合该至少一堆栈封盖值有效性条件是无关于该传入数据值的除了该预定部分以外的另一部分是否对应于基于该传入堆栈指针值确定的该地址的子分页地址位而确定的;且该方法包括响应于确定该传入数据值不符合该至少一堆栈封盖值有效性条件而触发错误处理响应。

[0005] 至少一些示例提供一种包括指令的计算机程序,当这些指令由主机数据处理装置执行时,控制该主机数据处理装置提供用于执行目标代码的指令执行环境,该计算机程序包括:存储器存取程序逻辑,该存储器存取程序逻辑用以基于堆栈指针执行对堆栈数据结构的堆栈存取;和处理程序逻辑,该处理程序逻辑用以执行与该堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作,该堆栈指针切换有效性检查操作包括验证通过该存储器存取程序逻辑响应于指定基于该传入堆栈指针值确定的地址的请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件,该至少一堆栈封盖值有效性条件包含该传入数据值的预定部分对应于指示包括基于该传入堆栈指针值确定的该地址的地址空间的分页的给定分页地址的条件,其中:该处理程序逻辑被配置成确定该传入数据值是否符合该至少一堆栈封盖值有效性条件,无关于该传入数据值的除了该预定部分以外的另一部分是否对应于基于该传入堆栈指针值确定的该地址的子分页地

址位;且该处理程序逻辑被配置成响应于确定该传入数据值不符合该至少一堆栈封盖值有效性条件而触发错误处理响应。

[0006] 本技术的进一步方面、特征和优点将由于结合附图阅读的以下示例描述而显而易见,在这些附图中:

[0007] 图1示出数据处理装置的示例;

[0008] 图2示出功能调用的示例;

[0009] 图3示出保护控制堆栈(GCS)推入和弹出操作的示例;

[0010] 图4示出GCS载入/存储操作的存取权限检查;

[0011] 图5示出非GCS载入/存储操作的存取权限检查;

[0012] 图6示出指令提取或分支操作的权限检查;

[0013] 图7示出控制堆栈切换以将堆栈指针从传出堆栈指针值切换到传入堆栈指针值的方法;

[0014] 图8示出第一堆栈指针切换指令的处理;

[0015] 图9示出第二堆栈指针切换指令的处理;

[0016] 图10示出堆栈指针切换;

[0017] 图11示出GCS记录的示例编码;并且

[0018] 图12示出模拟示例。

[0019] 一种装置包括存储器存取电路,该存储器存取电路用以基于堆栈指针执行对堆栈数据结构的堆栈存取。软件的不同部分可与存储器中的不同堆栈结构相关联,因此当在软件的那些部分之间切换时,可能需要将对应堆栈指针从与传出软件相关联的传出堆栈指针值切换到与传入软件相关联的传入堆栈指针值。然而,对于一些堆栈来说,重要的可为堆栈指针无法切换到不打算作为传入堆栈结构的任意位置。例如,当将项推入到堆栈上并从堆栈弹出项时,一些堆栈可与某些安全性措施相关联。如果堆栈指针可切换到任意软件选择的地址,那么这可绕过那些安全性措施,因为其可导致存取尚未受那些措施保护的数据的未来堆栈存取。

[0020] 因此,可能需要提供用于监管堆栈指针的切换的技术。一种方法可为将对堆栈指针的更新捕捉到更高特权的执行状态,使得更高特权的软件可检查传入堆栈指针值并确定是否可允许更新。然而,可能需要支持快速软件线程切换,而不需要调用核心或其它更高特权的软件。此类线程切换可能频繁地发生,因此在堆栈指针的每次切换时便调用核心的性能负担可能被视为太高。

[0021] 在下文讨论的示例中,处理电路支持执行与该堆栈指针从传出堆栈指针值到传入堆栈指针值的切换相关联的堆栈指针切换有效性检查操作。该堆栈指针切换有效性检查操作包括验证通过该存储器存取电路响应于指定基于该传入堆栈指针值确定的地址的存储器存取请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件。该至少一堆栈封盖值有效性条件包含该传入数据值的预定部分对应于指示包括基于该传入堆栈指针值确定的该地址的地址空间的分页的给定分页地址的条件。该处理电路确定该传入数据值是否符合该至少一堆栈封盖值有效性条件,无关于该传入数据值的除了该预定部分以外的另一部分是否对应于基于该传入堆栈指针值确定的该地址的子分页地址位。错误处理响应是响应于确定该传入数据值不符合该至少一堆栈封盖值有效性条件而触发的。

[0022] 因此,传入数据值是通过存储器存取电路从具有基于传入堆栈指针值确定的存储器地址的位置获得。例如,如果传入堆栈指针值确实表示先前建立的堆栈结构,那么传入数据值可从应代表堆栈的顶部的位置获得。为了通过有效性检查,传入数据值需要在预定部分中指定给定分页地址,该给定分页地址指示包括提供传入数据值本身的位置的地址的地址空间的分页。因此,软件需要确保非作用中堆栈在相对于稍后将用以参照堆栈的传入堆栈指针的位置处具有指定在存储该值的地址空间中的分页的分页地址的堆栈封盖值。这提供安全性措施以用于滤除提供传入堆栈指针值的错误值(其不对应于待切换到的所预期的堆栈结构)的一些错误操作,因为相对不可能的是由错误传入堆栈指针值表示的位置将刚好含有指定该位置本身的分页地址的数据值。

[0023] 可能认为,可通过要求以下而进一步提升安全性:为了成为有效值,传入数据值应指定提供该值的位置的具体地址,包含区分相同分页中的不同地址的子分页地址位。这可进一步提升安全性,因为到存储该指针本身的位置的相同地址的指针将为极罕见的,使得在将随机其它地址指定为传入堆栈指针值时意外地通过堆栈指针切换有效性检查操作的可能性将为极低的。

[0024] 然而,发明人认知到,在实践中,对于安全性来说,可为足够的是指示用以识别可有效地切换到的堆栈结构的堆栈封盖值中的分页地址。通过确定至少一堆栈封盖值有效性条件是否由传入数据值满足,无论传入数据值的任何部分是否对应于从传入堆栈指针值确定的地址的子分页地址位,这释放提供用于其它目的的堆栈封盖值的数据字的位。这可为特别有用的,因为对于一些使用情况来说,可能需要支持可分配到堆栈上的数个不同堆栈记录类型。如果在检查堆栈封盖值的有效性时要求子分页地址位有意义,那么除了该地址的若干较低位(其可能不需要给定任何数据/指令地址对准限制来明确地指定)以外,将需要该数据字的大部分位以指定堆栈封盖值中的地址,进而剩下非常少的位可用于编码堆栈记录类型。相较之下,通过要求有效堆栈封盖值指定存储堆栈封盖值的位置的分页地址,但不要求指定子分页地址位,这释放更大量位以用于编码堆栈记录类型,其可极有用于支持处理器架构的未来扩增。

[0025] 该给定分页地址可为指示包括基于该传入堆栈指针值确定的虚拟地址的虚拟地址空间的分页的虚拟分页地址。为了在堆栈指针的切换时为有效,通过要求传入数据值(通过基于传入堆栈指针值的存储器存取电路而获得)提供存储该传入数据值的位置的虚拟分页地址,这还提供传入堆栈是以在先前存取堆栈时使用的相同虚拟到物理地址转换映射来存取的健全性检查(sanity check),其可帮助检测基于攻击者定义将不同虚拟地址映射到存储堆栈的位置的物理地址上的混淆虚拟到物理地址转换映射的攻击。

[0026] 该处理电路可支持执行与该堆栈指针从该传出堆栈指针值到该传入堆栈指针值的该切换相关联的传出堆栈封盖操作,以将有效堆栈封盖值推入到具有基于该传出堆栈指针值选择的地址的位置,该有效堆栈封盖值在该有效堆栈封盖值的该预定部分中指定指示包括基于该传出堆栈指针值选择的该地址的地址空间的分页的分页地址。传出堆栈封盖操作与堆栈指针切换有效性检查操作互补,其中其在传出堆栈上设定有效堆栈封盖值,该有效堆栈封盖值在该堆栈稍后在稍后堆栈切换操作中切换回作为传入堆栈时预期将通过该堆栈指针切换有效性检查操作。

[0027] 传出堆栈封盖操作不一定取决于传入堆栈指针在堆栈指针切换有效性检查操作

中符合至少一堆栈封盖值有效性条件。即使传入堆栈指针是错误的,导致基于传入堆栈指针值载入的传入数据值在堆栈指针切换有效性检查操作的检查中失败,但使用有效堆栈封盖值来封盖传出堆栈仍可为安全的。在如下文描述的任一情况下,在一些具体实施中,传出堆栈封盖操作可通过到堆栈指针切换有效性检查操作的不同指令触发,因此实施用于触发传出堆栈封盖操作的指令的功能的电路逻辑可独立于任何有效性检查而定义,且可简单地实行传出堆栈封盖操作。即使响应于触发该堆栈指针切换有效性检查操作的指令而检测到错误,这可不必然停止执行用于触发传出堆栈封盖操作的指令(响应在堆栈指针切换有效性检查操作中所识别的错误的时序可变化,且可不需要立即发信号通知错误,如下文进一步解释)。

[0028] 针对该传出堆栈封盖操作的在该有效堆栈封盖值中指定的该分页地址可为指示包括基于该传出堆栈指针值选择的虚拟地址的虚拟地址空间的分页的虚拟分页地址。

[0029] 上文描述的技术可用于以下任何使用情况:其中堆栈数据结构是由处理电路使用以维持线程特定数据,其中不同的软件线程可分配不同堆栈数据结构,使得可能需要在线程切换上将堆栈指针从传出堆栈指针切换到传入堆栈指针,以改变哪些堆栈是主动的。该技术可特别有用于堆栈上的数据是敏感的、或否则需要管控堆栈上的数据使用的某些安全性措施的使用情况,使得在堆栈指针切换时存在风险,即,切换到不表示已经历那些安全性措施的堆栈的存储器中的任意地址可能产生可危害所执行的软件代码的错误结果的风险。

[0030] 然而,该技术可为有用的一个具体情境是其中该堆栈指针是保护控制堆栈(GCS)指针,该保护控制堆栈指针用于控制对保护控制堆栈(GCS)数据结构的存取,以用于保护从异常或功能调用返回的返回地址。此GCS数据结构可用作抵抗返回导向编程(ROP)攻击的防御措施,这些返回导向编程攻击是旨在通过改变在功能返回或异常返回时使用的返回地址(一般通过在功能调用或异常的嵌套序列期间将其存储在存储器中时改变返回地址)导致不正确控制流程的常见攻击形式。可在存储器的区域中建立经保护GCS数据结构,其具有限制在GCS数据结构中写入数据的能力的至少一防御措施,相对于正常存储器区域提供一些额外保护,从而使得攻击者可使未预期写入到GCS数据结构的指令改变保存于GCS数据结构上的返回状态较不可能。来自GCS数据结构的经保护返回状态信息可用以直接地控制异常/功能返回,或与由来自其它来源的软件(例如,存储器中分开的软件管理堆栈结构)获得的异常/功能返回信息比较,以检查异常/功能返回信息在触发对应的异常/功能返回之前是否能安全使用。然而,虽然在某些安全性措施可用以防止对GCS数据结构的错误更新,但如果攻击者可通过将用以存取GCS数据结构的堆栈指针切换到指向由攻击者控制的存储器区域的传入堆栈指针值而轻易地绕过这些安全性措施,使得错误返回状态接着在受害者软件后续尝试存取GCS数据结构时返回,那么那些措施可为无效的。此外,一些攻击可基于将传入堆栈指针切换到正确GCS数据结构,但切换到该GCS数据结构上的错误位置(例如不表示堆栈的目前“顶部”的位置)可产生稍后异常/功能返回表现不正确的风险,因为其使用打算用于不同异常/功能返回的返回状态。

[0031] 因此,堆栈指针切换有效性检查操作可特别有用于在支持使用GCS数据结构的架构中使用,因为其实现传入GCS数据结构是否能安全使用的健全性检查,而不需要每一堆栈指针切换被捕捉到较高特权层级。此外,使用分页地址而非完整地址的堆栈封盖值的更有效编码是有用的,因为其释放大量堆栈记录编码以用于其它目的。由于GCS数据结构可用以

保护数据免于受恶意攻击者窜改,因此提供用于额外类型的堆栈记录的编码空间可为有用的,以允许使用GCS数据结构保护其它类型的信息(不仅是功能/异常返回状态信息)。

[0032] 可在由处理电路使用的指令集架构中支持数个安全性措施,以保护GCS数据结构免于窜改。该装置可具有存储器管理电路,其用以基于与该目标地址相关联的存储器属性数据而确定是否允许对目标地址的存取,该存储器属性数据指定包含该目标地址的目标存储器地址空间区域是否是用于存储该GCS数据结构的GCS区域,其中到该GCS区域的写入存取受限于专用类别的GCS存取指令。因此,GCS数据结构可分配在专用类型的存储器区域(由存储器管理电路视为不同于用于一般数据或程序代码的正常存储器的不同类型的存储器)。通过限制可写入到存储器的此GCS区域的指令类别,这减少攻击者在尝试修改保存于GCS数据结构上的返回状态时可利用的攻击面。

[0033] 该存储器管理电路可响应于确定该目标存储器地址空间区域是该GCS区域,而拒绝通过除了该专用类别的GCS存取指令以外的存储指令所触发的到该目标地址的非GCS存储操作。通过对某些GCS存取类型的存储指令限制写入到GCS区域的能力,其它更通用存储指令无法窜改GCS数据结构的内容,从而为GCS数据结构中存储的经保护返回状态信息提供更大安全性保证。这减少攻击者尝试安装ROP攻击时可利用的攻击面。

[0034] 类似地,该存储器管理电路可响应于确定该目标存储器地址空间区域不是该GCS区域,而拒绝通过该专用类别的GCS存取指令中的一者所触发的到该目标地址的GCS载入/存储操作。因此,如果存取是由GCS存取类型的指令触发,那么可拒绝对于未指定为GCS数据结构的存储器区域的存取。这避免GCS存取类型的指令被错误使用于存取不打算用于存储GCS数据结构的存储器的区域,且给出GCS读取将是到不可能已被非GCS存取指令修改的存储器区域的信心,以防御抵抗ROP攻击。堆栈指针切换有效性检查操作的传入数据值的载入可被视为GCS载入操作,使得如果该地址由存储器属性数据识别为对应于除了GCS区域类型以外的存储器区域类型,那么其失败。

[0035] 在一些示例中,当该存储器属性数据指定该目标存储器地址空间区域是该GCS区域时,该存储器管理电路还可拒绝指令提取或到该目标地址的分支。因此,存储在GCS区域中的数据无法被视为可执行指令(尽管其可被视为指示存储可执行指令的存储器中的另一位置的地址指针)。鉴于GCS区域的此性质,可极有用的是设定有效堆栈封盖值(其在执行堆栈切换操作时预期在经分配作为GCS区域类型的区域中的传入堆栈上)以指定含有存储堆栈封盖值本身的位置的地址的相同分页的分页地址。这减少通过在存取GCS存储器区域中的GCS数据结构时意外地将堆栈封盖值解释为异常返回地址或功能返回地址所导致的不正确控制流程的风险,如果攻击者危害受害者代码中的控制流程使得在堆栈封盖值处于堆栈的顶部时触发功能返回或异常返回,那么其可能是风险。即使攻击者确实导致对堆栈的此错误存取,从而导致堆栈封盖值被返回作为用于异常返回或功能返回的经保护返回地址,对应返回操作将导致识别错误(直接在堆栈存取或返回操作时识别,或稍后在尝试从由被错误地解译为返回地址的堆栈封盖值所表示的地址提取指令时识别),因为当堆栈封盖值指向其本身的分页且此预期是分配作为GCS区域类型的分页时,不允许该分支或指令提取。

[0036] 指定分页地址的传入数据值的条件可不是传入数据值被要求通过以满足堆栈指针切换有效性检查操作中的检查的唯一条件。还可施加其它条件。

[0037] 例如,在一些具体实施中,该至少一堆栈封盖值有效性条件还包括该传入数据值

的最低有效部分具有不能够由任何有效指令地址的最低有效部分指定的位模式的条件。更具体来说,至少一堆栈封盖值有效性条件可包含最低有效部分具有用以表明有效堆栈封盖值的特定位模式的条件,其中该位模式不能够由任何有效指令地址的最低有效部分指定。

[0038] 例如,此特定位模式可具有位值为1的至少一位。对于一些处理器架构,指令编码可具有特定数目的位,且可存在任何指令存储在对准到在指令大小的倍数下的地址边界的地址处的要求。例如,指令可具有32位编码,且因此可能需要存储在对准到4字节地址边界的地址处(对于具有不同指令大小的架构来说,地址边界可处于不同的粒度)。如果尝试执行指令提取或到未对准地址的分支,那么可发信号通知错误。因此,通过编码堆栈封盖值,使得其最低有效位具有不能够由任何有效指令地址的最低有效部分指定的非零位模式(因为如果其被视为指令地址,那么该模式将使该值成为未对准指令地址),这可提供用于检测使用堆栈封盖值作为返回地址来执行异常返回或功能返回的错误尝试的另一措施,因此改善安全性。

[0039] 因此,分配到非作用中堆栈上的“堆栈封盖值”(预期在切换回到该堆栈时被验证)可视为“封盖”(或“密封”)该堆栈,其在意义上是于该堆栈的顶部处设有堆栈封盖值以防止将堆栈用于控制返回操作,直到封盖已去除为止。

[0040] 传入堆栈指针值可独立于传出堆栈指针值而定义,使得从传出堆栈指针值到传入堆栈指针值的切换可表示到完全不同堆栈结构的切换,而非仅是在将条目推入到堆栈上或从堆栈弹出条目时执行的到目前堆栈指针的增量或减量。

[0041] 该处理电路可响应于第一堆栈指针切换指令指定指示该传入堆栈指针值的操作数,而执行该堆栈指针切换有效性检查操作。因此,传入堆栈指针值可定义为第一堆栈指针切换指令的任意指定的操作数(例如,通过在通过第一堆栈指针切换指令指定的通用寄存器中指定传入堆栈指针值)。

[0042] 该装置可具有堆栈指针寄存器,该堆栈指针寄存器用以存储该堆栈指针。堆栈指针切换的传出堆栈指针值可为在执行第一堆栈指针切换指令时针对堆栈指针寄存器指定的目前值。

[0043] 响应于第一堆栈指针切换指令,当堆栈指针切换有效性检查操作成功时(即,基于传入堆栈指针值获得的传入数据值符合至少一堆栈封盖值有效性条件,且满足任何其它检查(取决于特定具体实施)),该处理电路可将堆栈指针寄存器从传出堆栈指针值更新到传入堆栈指针值。

[0044] 在一些示例中,第一堆栈指针切换指令可为在处理电路的最低特权执行状态下执行的软件(一般是用户层级软件)可用的使堆栈指针切换的唯一机制。在最低特权执行状态下执行的软件可能不具有到堆栈指针寄存器的直接写入存取(可能的是软件可仍能够在更高特权的执行状态下执行时直接写入到堆栈指针寄存器)。通过限制用户层级软件造成堆栈指针切换的能力,这改善安全性。由于该软件必须使用第一堆栈指针切换指令以使该堆栈指针切换,因此这强制堆栈指针切换有效性检查操作执行。例如,当目前特权层级具有比某一阈值特权层级低的特权时,由该软件尝试执行请求到堆栈指针寄存器的更新的系统寄存器指令可被拒绝。

[0045] 在一些具体实施中,第一堆栈指针切换指令还可触发上文提及的传出堆栈封盖操作。因此,响应于第一堆栈指针切换指令,在执行第一堆栈指针切换指令时针对堆栈指针寄

存器指定的目前值被使用为传出堆栈指针值,且形成指定对应于传出堆栈指针值的(虚拟)分页地址的有效堆栈封盖值,并将该有效堆栈封盖值写入到对应于传出堆栈指针值的地址处的位置。

[0046] 然而,在实践中,在相同指令中执行堆栈指针切换有效性检查操作和传出堆栈封盖操作两者可能需要转换两个不同目标地址:待基于传入堆栈指针值载入的传入数据值的载入地址、以及待基于传出堆栈指针值存储的有效堆栈封盖值的存储地址。一些电路具体实施可能不能够响应于相同指令而处理两个不同地址转换,因此需要在一个指令中执行堆栈指针切换有效性检查操作和传出堆栈封盖操作两者(虽然是可能的)的指令集架构可对微架构电路设计者产生显著的设计挑战。

[0047] 因此,为了简化电路设计具体实施,可为有用的是定义用于执行堆栈切换的两个分开架构指令:第一堆栈指针切换指令,该第一堆栈指针切换指令触发该堆栈指针切换有效性检查操作,且如果该有效性检查操作成功,那么更新该堆栈指针寄存器以将该堆栈指针切换到该传入堆栈指针值;和第二堆栈指针切换指令,该第二堆栈指针切换指令用以执行该传出堆栈封盖操作。

[0048] 然而,使用两个指令方法,传出堆栈指针值(响应于第一堆栈指针切换指令而在堆栈指针寄存器中取代)应在第一堆栈指针切换指令与第二堆栈指针切换指令之间被“记住”。从架构观点,可能并不期望消耗架构寄存器以保留传出堆栈指针值,因为这可增加寄存器压力,且如果架构寄存器经分配以存储传出堆栈指针值以用作为第二堆栈指针切换指令的操作数,那么还可能并不期望将传出堆栈指针值暴露到在切换堆栈指针之后执行的软件,这可能是风险。

[0049] 因此,指令集架构可定义第一堆栈指针切换指令在架构上将传出堆栈指针值写入到传入堆栈(暂时地),使得传出堆栈指针值通过将其写入到存储器地址空间而保留。这不排除一些微架构具体实施,其能够使用微架构机制(例如寄存器重新命名或存储缓冲),以为第二堆栈指针切换指令提供对通过第一堆栈指针切换指令在名义上写入到存储器的传出堆栈指针值的更快存取,这快于第二堆栈指针切换指令存取存储器实际上将有的情况(例如,可使用处理电路硬件内的缓冲,因为预期在被第一指令存储相对不久之后第二指令会频繁地需要传出堆栈指针)。尽管如此,此类微架构电路机制是任选的且从架构(软件可见)观点,这些指令可经处理,就像第一堆栈指针切换指令将传出堆栈指针值写入到存储器,且第二堆栈指针切换指令将其从存储器读取。

[0050] 因此,响应于第一堆栈指针切换指令,该处理电路可将指定传出堆栈指针值的进行中令牌值推入到具有基于传入堆栈指针值选择的地址的位置。这暂时地使传出堆栈指针值在架构上可见的存储器地址空间中可存取,使得可接受盖写堆栈指针寄存器中的传出堆栈指针值,以将堆栈指针切换到传入堆栈指针值,即使传出堆栈尚未被封盖。

[0051] 对于用以载入待在该堆栈指针切换有效性检查操作中验证的该传入数据值的载入操作、以及用以将该进行中令牌值推入到具有基于该传入堆栈指针值选择的该地址的该位置的存储操作,该处理电路可原子地执行该载入操作和该存储操作。因此,提供机构以防止介入到基于正执行的载入和存储之间的传入堆栈指针值选择的地址的存取,或者如果此类介入存取是可能的,那么检测其并校正此原子性(atomicity)的缺乏(例如,通过触发管线清除且重新执行第一堆栈指针切换指令)。可使用用以强制载入与存储之间的原子性的

任何已知原子存取机制。

[0052] 响应于第二堆栈指针切换指令,该处理电路可验证通过该存储器存取电路响应于指定基于该堆栈指针寄存器中的目前堆栈指针确定的地址的存储器存取请求而获得的给定数据值是否是有效地形成的进行中令牌值;以及响应于验证该给定数据值是有效地形成的进行中令牌值:触发有效堆栈封盖值到具有基于由该给定数据值的一部分指定的给定堆栈指针值确定的地址的位置的写入,有效堆栈封盖值在该有效堆栈封盖值的该预定部分中指定指示包括基于该给定堆栈指针值确定的地址的地址空间的分页的分页地址。在一些示例中,基于给定堆栈指针值确定的地址可为从给定堆栈指针值以对应于一个堆栈条目的大小的量的地址偏移。如果第二堆栈指针切换指令在第一堆栈指针切换指令之后执行,那么预期基于目前堆栈指针获得的给定数据值(在第二堆栈指针切换指令时)应为先前通过第一堆栈指针切换指令写入到传入堆栈数据结构的进行中令牌值。然而,应理解,处理电路的硬件电路逻辑不可能知道第二堆栈指针切换指令实际上是否跟着第一堆栈指针切换指令,即,硬件电路逻辑只是实施“黑箱”电路逻辑,其基于其定义的操作数而实施第二堆栈指针切换指令的功能,无关于在该指令之前软件选择哪些其它指令执行。因此,在硬件中不存在将强制第二堆栈指针切换指令的操作数确实与先前用于第一堆栈指针切换指令的操作数对应的电路。

[0053] 因此,第二堆栈指针切换指令检查基于堆栈指针寄存器中的目前堆栈指针获得的给定数据值对应于有效地形成的进行中令牌值,且如果有效,那么使用由该进行中令牌值提供的给定堆栈指针值(预期但不保证是先前通过先前第一堆栈指针切换指令切换远离的传出堆栈结构的地址)以存取对应的堆栈结构,并将有效堆栈封盖值推入到该结构。同样地,形成有效堆栈封盖值,使得其在预定部分中指定指示包括由给定堆栈指针值指示的地址的地址空间的分页的分页地址。这确保当软件稍后希望将对应堆栈切换回作为堆栈指针切换操作的传入堆栈时,针对堆栈指针切换有效性检查操作执行的稍后检查将通过。

[0054] 响应于第二堆栈指针切换指令,当给定数据值经验证为是有效地形成的进行中令牌值时,该处理电路被配置成更新堆栈指针寄存器中的目前堆栈指针,以指示将提供给定数据值的条目从对应堆栈数据结构去除。例如,用以指示将条目去除的更新可为与将在从堆栈弹出条目时执行的对目前堆栈指针的相同类型更新(例如,增量或减量,取决于堆栈在推入/弹出时增长/缩减的方向为何)。此更新意味进行中令牌值将不再在对堆栈的后续弹出存取时存取。不必实际地从存储器删除进行中令牌值,因为到堆栈指针的更新意味其将不在基于堆栈指针的后续存取时存取。如果存在后续推入操作以将新条目推入到堆栈,那么进行中令牌值稍后可在任何情况下被盖写。

[0055] 响应于第二堆栈指针切换指令,当给定数据值的最低有效部分具有不能够由任何有效指令地址的最低有效部分指定且不能够由符合至少一堆栈封盖值有效性条件的任何值指定的位模式时,该处理电路可验证给定数据值是有效地形成的进行中令牌值。同样地,可有用的是进行中令牌值具有对应于未对准指令地址的最低有效部分,使得使用进行中令牌值作为返回地址的任何尝试将导致如上文讨论的未对准错误。此外,可能需要针对有效堆栈封盖值的最低有效部分和有效进行中令牌值具有不同编码,以避免攻击者使用错误控制流程绕过保护(例如,这允许仅执行第一/第二堆栈指针切换指令的一个而非两个一起执行的错误被检测)。

[0056] 响应于第二堆栈指针切换指令,该处理电路可响应于确定给定数据值不是有效地形成的进行中令牌值而触发错误处理响应。

[0057] 对于当传入数据值在堆栈指针切换有效性检查操作的至少一堆栈封盖值有效性条件失败时的错误处理响应、以及当给定数据值不是有效地形成的进行中令牌时响应于第二堆栈指针切换指令的错误处理响应两者来说,该错误处理响应可以数个方式实施。例如,错误处理响应可包括下列中的至少一者:发信号通知错误;设定错误报告指示;以及将该堆栈指针设定成无效值。在一些具体实施中,错误处理响应可直接使处理被暂停(例如,在检测到错误时,发信号通知错误或异常)。其它示例可使用使错误被通知的更间接手段,使得错误本身可不发生直到稍后发生。例如,通过将堆栈指针设定成无法为有效存储器地址的无效值,基于无效堆栈指针而存取存储器的后续尝试可能触发错误。这有时可比直接基于第一/第二堆栈指针切换指令的检查触发错误而更轻易地在硬件中实施。例如,其中较高地址位具有除了所有为0或所有为1以外的值的地址空间的保留部分可经保留,使得其无法指定为有效地址,因此如果需要错误处理响应,那么通过将堆栈指针设定成此保留部分中的地址,这可导致稍后的存储器存取错误,使得仍可避免错误处理。应理解,所采取的具体错误处理响应可在上文描述的通用技术的不同具体实施之间变化。此外,在一些情况下,针对失败的堆栈封盖值有效性检查所采取的错误处理响应可不同于当给定数据值不是有效地形成的进行中令牌时针对第二堆栈指针切换指令所采取的错误处理响应。

[0058] 上文讨论的技术可实施在数据处理装置内,该数据处理装置具有针对实施上文讨论的存储器存取电路和处理电路而提供的硬件电路。

[0059] 然而,相同技术还可实施在计算机程序内,该计算机程序是在主机数据处理装置上执行以提供用于目标代码的执行的指令执行环境。即使主机数据处理装置本身不支持该架构,此计算机程序可控制主机数据处理装置以模拟其将提供在实际支持根据某个指令集架构的目标代码的硬件装置上的架构环境。计算机程序可具有存储器存取程序逻辑和处理程序逻辑,其控制主机数据处理装置以仿真上文所讨论的特征,包含针对堆栈指针切换有效性检查操作的支持。存储器存取程序逻辑执行基于堆栈指针执行堆栈存取。处理程序逻辑执行堆栈指针切换有效性检查操作,其包含基于传入堆栈指针值载入的传入数据值的预定部分是否指定包括从传入堆栈指针值衍生的地址的分页的给定分页地址(在经模拟地址空间中,而非主机数据处理装置的主机地址空间)的检查。因此,当需要堆栈指针从传出堆栈指针值切换到传入堆栈指针值的目标代码在由主机数据处理装置上执行的模拟计算机程序所提供的指令执行环境中执行时,即使主机数据处理装置本身在硬件中不支持堆栈指针切换有效性检查操作,也可达成如上文所论述的相同功能。

[0060] 例如,当针对一个指令集架构所编写的程序码是在支持不同指令集架构的主机处理器上执行时,此模拟程序可为有用的。再者,由于软件在模拟执行环境上的执行可使软件测试得以与支持新架构的硬件设备的进行中开发平行,模拟可允许针对较新版本的指令集架构的软件开发在支持该新架构版本的处理硬件就绪之前开始。模拟程序可存储在存储介质上,该存储介质可为非暂态存储介质。

[0061] 图1示意地示出数据处理装置2的示例。数据处理装置具有包含若干个管线级的处理管线4。在此示例中,管线级包含用于从指令高速缓冲存储器8提取指令的提取级6;用于解码提取程序指令的解码级10,以产生待由管线的其余级处理的微操作(经解码指令);用

于检查微操作所需的操作数在寄存器文件14中是否可用,并发布用于将给定微操作的所需操作数执行一次的微操作是可用的发布级12;用于执行对应于通过处理读自寄存器文件14的操作数以产生结果值的微操作的数据处理操作的执行级16;和用于将处理的结果写回到寄存器文件14的写回级18。应了解此仅为可能的管线架构的一个示例,且其它系统可具有额外级或不同的级配置。例如,在乱序处理器中,可包含用于将由程序指令或微操作指定的架构寄存器映射到识别寄存器文件14中的物理寄存器的物理寄存器说明符的寄存器重命名级。在一些示例中,在由解码级10解码的程序指令与由执行级所处理的对应微操作之间可有一对一关系。在程序指令与微操作之间还可能有一对多或多对一关系,使得,例如,单一程序指令可分成两个或更多个微操作,或者两个或更多个程序指令可融合以作为单一微操作来处理。

[0062] 执行级16(处理电路的示例)包含若干个处理单元,以用于执行不同类别的处理操作。例如,执行单元可包含用于对读自寄存器14的标量操作数进行算术或逻辑运算的标量算术/逻辑单元(ALU)20;用于对浮点值进行运算的浮点单元22;用于评估分支操作的结果并调整该程序计数器的分支单元24,该程序计数器据此表示目前的执行点;和用于执行载入/存储操作以存取存储器系统8、30、32、34中的数据的载入/存储单元26。载入/存储单元是存储器存取电路的示例。提供存储器管理单元(MMU)28,其是存储器管理电路的示例,以用于执行由载入/存储单元26基于数据存取指令的操作数所指定的虚拟地址与识别存储器系统中数据的存储位置的物理地址之间的地址转换。MMU具有转换后备缓冲区(TLB)29,其用于从存储在存储器系统中的分页表高速缓存地址转换数据,其中分页表的分页表条目定义地址转换映射和存取权限,其例如管控是否允许管线上执行的给定程序从给定存储器区域读取或写入数据或执行指令。当分页表结构被遍历以定位对应于所需地址的分页表条目时,MMU 28可具有在分页表走访期间请求存储器存取的电路。存储器管理单元是存储器管理电路的示例。

[0063] 在此示例中,存储器系统包含一级数据高速缓冲存储器30、一级指令高速缓冲存储器8、共享二级高速缓冲存储器32和主系统存储器34。将理解此仅为可能的存储器层次结构的一个示例,并可提供高速缓冲存储器的其它布置。在执行级16中示出的处理单元20到26的具体类型仅为一个示例,且其它具体实施可具有不同组的处理单元或可包含相同类型的处理单元的多个示例,使得可平行地处理多个相同类型的微操作。应理解图1仅为可能的处理器管线具体实施的一些组件的简化表示,且处理器可包含为了简洁起见而未示出的许多其它元件。虽然图1示出具有对存储器34存取的单一处理器核心,该装置2还可具有一个或多个另外的处理器核心,其共享对存储器34的存取,其中每一核心具有相应的高速缓冲存储器8、30、32。

[0064] 图2示出调用功能(为了易于参见,标示为fn1)以及从功能返回的示例。功能(也称为过程)是可从程序的另一部分调用的指令序列,且当完成时其将处理返回到该功能被调用的程序流程的部分。相同功能可从程序中的多个不同位置被调用,且因此在调用功能时存储功能返回地址,使得功能返回可区分程序流程应返回到哪个地址。

[0065] 例如,如图2所示,具有连结指令BLR的分支可在功能将被调用的该点(由地址#add1表示)处执行,以使程序流程分支到使用具有连结指令的分支的操作数所指定的分支目标地址#add2处的指令。具有连结指令的分支还导致处理电路设定连结寄存器(用于跟踪

功能返回地址的指定寄存器)到在具有连结指令的分支之后的下一指令的地址(在此示例中,功能返回地址是#add1+4)。在已采取分支之后,在功能代码中执行数个指令(例如,LD、MUL、ADD等),且当功能完成时,执行返回分支指令RET,其使分支到存储在连结寄存器中的返回地址指示的指令。

[0066] 如果无其它功能从fn1内被调用,且在到达fn1的结束处的返回分支之前无异常发生,那么在连结寄存器中的地址仍应相同如fn1被调用时所设定的一般。

[0067] 然而,通常,由背景程序代码所调用的第一功能fn1本身可以嵌套方式调用另一功能(就是说fn2),且在此情况中,对fn2的功能调用将盖写存储在连结寄存器中的返回地址,且因此在调用另一功能之前,第一功能fn1的功能代码应包含指令,以将来自连结寄存器的返回地址存储到存储器中的数据结构(例如,堆栈结构,以后进先出(LIFO)方式操作),且在从fn2返回之后,fn1的功能代码应将返回地址恢复到执行返回分支之前的连结寄存器。用于保存并恢复功能返回状态(例如返回地址)的职责将通常在于软件(可不存在用于保存返回地址的架构上强制硬件机构)。

[0068] 然而,当功能返回地址存储在存储器中时,可能易于受到攻击者修改该数据的攻击,例如使用在另一处理器核心上执行的另一线程,或通过中断该调用功能且同时执行其它程序代码,其盖写存储在存储器中的该返回地址。替代地,该攻击者可执行一些指令,其目标在于修改将该返回地址从存储器恢复到寄存器的该指令的地址操作数,使得从存储器载入的该数据不同于在调用嵌套功能之前原本存储到存储器的返回地址。如果攻击者可导致返回分支以分支到程序流程中并非功能调用分支之后的指令的点,那么攻击者可能够导致软件不正确动作,且可能够绕过某些安全保护或导致执行非所需操作。

[0069] 功能调用是操作的示例,其产生返回状态信息,返回状态信息提供关于处理电路稍后将被恢复到的状态的信息。返回状态信息可被捕获的另一情境可为采取异常时,其中提供在硬件中的异常处理电路,或软件异常处理器可捕获异常返回状态信息,例如指示在从处理异常返回之后将执行的指令的地址的异常返回地址,和/或指示在从异常返回之后处理器将执行的模式或执行状态的存储的处理器状态信息。例如,存储的处理器状态信息可指示采取异常的异常层级,以及关于采取异常时处理器的操作状态的其它信息。如同功能调用,异常可被嵌套,且因此当采取另一异常时,针对异常捕获的异常返回状态可存储到存储器(以硬件自动地,或通过软件异常处理器),且因此当其存储在存储器中时可能容易受攻击者窜改。这些类型的攻击可称为返回导向编程(ROP)攻击。可能期望提供针对此类攻击的架构反制措施。

[0070] 图3示出用于使用在存储器中称为“保护控制堆栈(GCS)”的经保护数据结构40来保护免于ROP攻击的方法。在存储器地址空间内的GCS数据结构的位置可通过软件选择,但硬件提供经设计以保护GCS数据结构免于受恶意攻击者窜改的架构特征。

[0071] 如图1所示,寄存器14可包含控制寄存器,该控制寄存器包含用于存储堆栈指针的一个或多个经保护控制堆栈指针(GCSPR)寄存器36,该堆栈指针指示GCS数据结构上的地址。在一些示例中,GCS指针寄存器可为堆积的寄存器组,分别提供用于至少两个执行状态(例如,异常层级),以实现在不同执行状态下操作的软件参照存储器内的不同GCS结构,而不需要在执行状态的每一转变之后重新编程共享的堆栈指针寄存器。其它示例可使用单一GCS指针寄存器,且软件可在执行状态之间的转变时更新存储在GCS指针寄存器中的堆栈指

针。

[0072] 如图3所示,GCS数据结构40存储在由用于控制地址转换以及存取权限检查的存储器管理单元(MMU)28所使用的分页表的相关联分页表条目直接或间接指定的存储器属性所指定为存储器的GCS区域的存储器区域中。GCS区域属性可直接在包括GCS数据结构的至少部分的存储器区域的对应分页表条目的编码内指定,或可间接参照在由该分页表条目所参照的寄存器内。

[0073] 当存储器区域被识别为GCS区域时,那么当执行GCS存取指令的特定子集时,到该区域的写入存取是限于在由处理电路16触发的写入请求。由软件使用用于一般存储操作而不打算存取GCS结构的通用存储指令并不视为GCS存取指令的限制子集中的一个。MMU 28仍可允许使用通用载入指令读取GCS结构,该通用载入指令导致发布并非GCS存储器存取请求的读取请求。当存储器存取请求请求对GCS区域的存取时,该请求是写入请求,且该请求不是由GCS存取指令的限制子集的一个触发的GCS存储器存取请求,那么该存储器存取请求被拒绝且发信号通知错误。GCS存取指令的子集可包含至少一GCS推入指令,其导致返回状态信息(例如来自连结寄存器的功能返回地址,或异常返回地址或采取异常时被捕获的经存储的处理器状态)被推入到使用GCS指针寄存器58中指示的堆栈指针所确定的GCS结构上的位置。GCS推入指令还导致堆栈指针以取决于推入到GCS的堆栈帧的大小的量前行(例如,如果GCS经管理为升序堆栈,那么以堆栈帧的大小来增量堆栈指针;或如果GCS经管理为递减堆栈,那么以堆栈帧的大小来减量堆栈指针)。GCS存取指令还可包含至少一种形式的GCS弹出指令,其从GCS结构弹出经保护返回信息。以及返回从堆栈弹出的返回指令,GCS弹出指令还导致堆栈指针在堆栈指针针对GCS推入指令调整的方向的相对方向上调整(例如,如果GCS经管理为升序堆栈,那么以堆栈帧的大小来减量堆栈指针;或如果GCS经管理为递减堆栈,那么以堆栈帧的大小来增量堆栈指针)。如下文所述,GCS存取指令还可包含GCS堆栈指针切换指令GCSSS1、GCSSS2,其可允许将特殊用途值写入到GCS存储器区域中的堆栈上,以用于保护该堆栈免于GCSPR 36中的堆栈指针不适当切换的目的。

[0074] GCS存取指令可不被允许存取未由分页表属性指定为GCS区域类型的存储器区域。因此,当由该存取的目标存储器区域未标记为GCS区域类型时,如果尝试执行GCS存取(包含针对GCS堆栈指针切换指令GCSSS1、GCSSS2的存储器存取),那么可发信号通知错误。通过禁止使用GCS存取指令存取非GCS区域,这不鼓励程序设计者使用GCS存取指令,除非真的打算是GCS存取,以减少可用于攻击者的攻击面。此外,这给出由GCS弹出指令存取或由GCS堆栈指针切换指令GCSSS1、GCSSS2中的一个验证的数据不能够由非GCS指令修改的信心。

[0075] GCS结构与由软件使用的任何数据结构分开,以将存储的返回状态信息维持在存储器中,以处理功能调用或异常的嵌套。因此,当功能调用或异常嵌套时,GCS结构不打算消除软件本身跟踪返回状态信息的保存和恢复的需求(返回状态的软件触发保存可以如在不支持上文所讨论的GCS保护的架构措施的处理器上相同的方式继续)。替代地,GCS结构提供经保护存储器的区域,其被保护免于危害的程序代码的窜改,其可用以提供用于验证打算由软件使用以从功能调用或异常的处理返回的返回状态信息的信息。

[0076] 在一些具体实施中,使经保护返回状态信息从GCS结构弹出的GCS弹出指令还可导致处理电路16比较弹出的返回状态与存储在寄存器中的目前返回状态信息(例如,用于功能返回的连结寄存器,或异常返回地址寄存器和/或用于异常返回的经存储的处理器状态

寄存器),且如果从GCS结构40弹出的返回状态信息与软件打算用于功能/异常返回的预期返回状态信息之间存在失配,那么发信号通知错误。因此,软件可通过包含功能调用/返回或异常进入/返回时执行的程序代码内的GCS推入和GCS弹出指令的示例而受保护免于窜改。

[0077] 其它具体实施可限定用于验证期望的返回状态信息是否有效的单独指令,不同于从GCS结构40弹出返回状态信息的指令。

[0078] 替代地,GCS弹出指令可直接将来自GCS的经保护返回状态弹出到用以指定异常返回或功能返回的返回状态的一个或多个寄存器(或可与异常/功能返回指令组合,以弹出经保护返回状态并且使用该状态用于控制异常/功能返回),在此情况中,不必实行验证是否软件提供的期望的返回状态信息是有效,因为在此一具体实施中,GCS保护的返回状态是直接用以控制异常/功能返回。例如,对于功能返回地址的GCS保护,可将功能返回地址直接弹出到连结寄存器,取代软件可基于其自己的经管理堆栈结构而放置在其中的任何软件管理的功能返回地址。

[0079] 此外,还可支持存取指令的其它类型的GCS。当启用GCS模式时(控制寄存器中的控制状态可控制是否GCS模式启用),一些指令(其在停用GCS的使用的模式中具有其它功能)在执行时,可导致处理电路16执行额外功能(例如额外的GCS模式特定安全性检查)。

[0080] 一般来说,通过提供用于定义用于GCS结构40的GCS存储器区域类型的架构支持,且将到GCS区域类型的写入存取限制到GCS存取指令的有限子集(其可能不被允许存取除了GCS区域类型以外的存储器区域),这减少攻击者可用于尝试窜改存储在GCS结构40上的经保护返回状态信息的攻击面。

[0081] 图4是示出GCS载入/存储操作的存取权限检查的流程图。GCS载入/存储操作是由GCS存取指令的类别的一个触发的载入/存储操作(例如,GCS存取指令包含GCS推入和弹出指令,以及GCS堆栈指针切换指令GCSSS1、GCSSS2)。在步骤110,处理电路16基于寄存器36中的GCS指针而确定GCS载入/存储操作的目标地址。在步骤112,存储器管理单元28在其TLB 29中查找目标地址,以获得GCS载入/存储操作的目标地址的存储器属性。在步骤114,MMU 28确定目标地址是否对应于GCS存储器区域,其是用于存储GCS数据结构的专用类型的存储器区域。如果目标地址不对应于GCS存储器区域类型,那么在步骤116,拒绝GCS载入/存储操作。发信号通知错误,其可中断正执行的处理且使异常处理器处理错误的原因。通过抑制到未标记为GCS存储器区域类型的区域的GCS存取,这防止GCS载入/存储指令被错误使用于存取非GCS存储器,且还意味由GCS载入操作返回的经保护返回状态可为受信任的,因为其不可能已被非GCS指令窜改。

[0082] 如果在步骤114,确定目标地址对应于GCS存储器区域,那么在步骤118,MMU 28确定任何其它存取权限检查是否通过。这些检查可检查其它属性,例如指示读取请求和写入请求是否允许分别为存储器区域的读取/写入权限信息的其它属性,或者定义处理器2的执行状态子集的属性,其中该区域被允许存取。如果任何其它存取权限检查失败,那么在步骤116,拒绝GCS载入/存储操作,且发信号通知错误。由处理器在发生错误时所设定的错误类型信息可取决于错误的原因是否为对非GCS存储器区域的GCS存取、或另一类型的存取权限违反而不同。如果所有任何其它存取权限检查皆通过,那么在步骤120,允许GCS载入/存储操作。

[0083] 图5示出针对非GCS载入/存储操作(由除了GCS存取指令类别以外的指令所触发的载入/存储)执行的类似存取权限检查。步骤130、132和134类似于图4的步骤110、112、114。相较于图4,在图5的步骤134,相较于GCS载入/存储操作,非GCS载入/存储操作对于目标地址是否对应于GCS存储器区域的检查的响应是相反方向,其中当目标地址对应于GCS存储器区域时,那么拒绝非GCS存储操作,而如果目标地址不对应于GCS存储器区域,那么拒绝GCS存储操作。

[0084] 因此,如果在步骤134,确定目标地址对应于GCS存储器区域,且在步骤135,确定目前载入/存储操作是非GCS存储操作,那么在步骤136,拒绝非GCS载入/存储操作,且发信号通知错误。取决于在步骤138执行的任何其它存取权限检查的结果,即使非GCS载入操作的目标是GCS存储器区域,这些非GCS载入操作仍可能被允许。如果任何其它存取权限检查失败,那么在步骤136,再次拒绝非GCS载入/存储操作。否则,如果目标地址不对应于GCS存储器区域(在步骤134的否)、或非GCS操作是载入操作(在步骤135的否),且在步骤138,任何其它存取权限检查(与GCS存取检查无关)通过,那么在步骤140,允许非GCS载入/存储操作。

[0085] 图6是示出指令提取或分支操作的权限检查的流程图。在步骤150,提取级6请求提取与目标地址相关联的指令、或检测到到目标地址处的指令的分支(可基于通过执行级16的分支单元24执行分支指令、或基于通过与提取级6相关联的分支预测器所作出的未来分支预测而检测到该分支)。响应于指令提取或分支,在步骤152,MMU 28在其TLB 29中查找目标地址以识别对应于目标地址的存储器属性数据(其中如果存在TLB未命中(miss),那么执行到存储器的分页表走访),并确定目标地址是否位于由存储器属性数据识别为GCS区域的区域中。如果目标地址位于GCS区域中,那么在步骤154,拒绝指令提取或分支,且发信号通知错误。这防止存储GCS结构上的数据被视为可能产生不可预测结果的风险的可执行指令。

[0086] 如果目标地址不位于GCS区域中,那么在原则上可执行从该地址载入的指令,取决于在步骤156执行的任何其它权限检查。例如,这些检查可包含基于分页表的检查,其检查内容是在目标地址的存储器属性数据中指示的执行权限数据。如果那些其它权限检查通过,那么在步骤158,可允许指令提取或分支。如果其它权限检查失败,那么在步骤154,再次拒绝指令提取或分支,且发信号通知错误。

[0087] 因此,GCS存储器区域受限于提供数据值(例如地址指针和其它信息)且无法提供可执行指令码,因为尝试从其提取指令、或将指令分支到其中,GCS区域将触发错误。

[0088] 上文描述的措施可有用于保护GCS数据结构上的内容在特定程序内处理功能返回和异常返回时免于受恶意攻击者窜改。

[0089] 然而,对控制流程另一可能的风险可为当寄存器36中的GCS指针从与传出堆栈数据结构相关联的传出堆栈指针值切换到与传入堆栈数据结构相关联的传入堆栈指针值时。当在使用不同GCS数据结构的线程之间切换时,此类切换可为常见的(且被允许),但可能为攻击者尝试使用的途径,以在尝试暴露对攻击者不应可存取的对受害者程序可存取的敏感信息时导致不正确处理。例如,如果堆栈指针切换的传入堆栈指针值被导致成为不对应于该传入线程的正确GCS数据结构的地址,那么可能存在不适当控制流程被触发的风险,特别是如果针对传入堆栈指针值指定的地址刚好是在存储器的GCS区域内的地址(例如,指向与不同线程相关联的GCS结构上的返回状态的地址)。即使传入堆栈指针值是在该传入线程的正确GCS数据结构内的地址,如果传入堆栈指针值设定成除了表示堆栈的目前“顶部”的位

置以外的位置,那么可能仍存在错误的风险,因为这可能对正被用作不同异常/功能返回的返回地址的一个异常/功能返回导致有效返回状态,这仍导致不正确信息。

[0090] 传入堆栈指针值中的此类错误可能由于编程错误而意外出现,或通过攻击者而恶意地出现,该攻击者危害执行码以导致将寄存器36中的GCS指针的值切换以指向不正确位置(例如,在受攻击者控制的GCS数据结构中的位置)(在GCS区域中,因此不被图4的检查抓到)的指令被执行,该不正确位置已预先填入经设计以蒙骗受害者软件程序进行分支到不正确码序列中的返回地址。

[0091] 用于减轻免于受传入堆栈指针值的此类错误设定的一种方法可为需要使到GCSPR36的每一写入被捕捉到更高特权的执行状态,使得更高特权的程序(例如操作系统或管理程序)可在更新GCSPR 36之前验证堆栈指针的新值是否安全。然而,这将不利地影响性能。GCS指针的切换在应用程序层级线程之间切换时是非常常见的并确保快速切换且因此确保高性能,可能需要允许此类GCS指针切换在无需调用更高特权的程序的情况下进行。

[0092] 还可能并不期望允许对GCS指针寄存器36的直接存取,以降低上文描述的形式的攻击风险。

[0093] 因此,可提供一些指令以用于控制GCS指针寄存器36中的堆栈指针值的切换,其可实施用于检测传入堆栈指针值的不适当设定的一些健全性检查。这些GCS指针切换指令被视为是GCS存取指令类别的成员,使得到由这些指令触发的GCS区域的写入可通过图4和图5所示的检查。

[0094] 当在不同线程之间切换时,软件堆栈(使用与GCS指针分开的堆栈指针跟踪)通过软件切换,且还可切换目前保护控制堆栈。为了确保软件无法切换到传入GCS上的任何任意位置,当从该GCS切换而出时堆栈封盖值经添加到传出GCS,且当堆栈指针上的传入堆栈切换时,该封盖值在切换到该GCS时验证。

[0095] 该封盖值经设计以可与任何过程返回值区分,这意味在切换堆栈时,我们可自信我们仅切换到传入GCS的顶部。例如,封盖值的较低地址位可设定成无法由任何有效指令地址指定的非零位模式,因为指令对准需求可能需要将所有有效指令地址在指令长度的粒度下对准到地址边界(例如,对于32位指令编码来说,对准地址边界可为4字节间隔)。这意味任何有效过程返回值对于较低位应具有0,且因此封盖值可通过在较低位中具有至少一个1而区分。

[0096] 封盖值还含有我们在切换到GCS时检查的封盖位置的地址。此地址提供:

[0097] 1)我们正使用GCS的PA映射的正确VA返回到预期GCS的健全性检查,通过将封盖的地址与从封盖载入的值比较。

[0098] 2)安全性检查,以防止封盖(不正确地)用作过程返回的部分的情况。如果我们尝试分支到保持在封盖中的值,那么这隐含地为*数据*地址,因为其位于GCS区域中(因此如果存在该地址处执行指令的尝试,那么图6所示的检查将触发错误),且当处理器尝试从此位置执行时,其将不具有执行权限且将采用异常。

[0099] 我们观察到如果保持在封盖中的地址位于与封盖的位置相同的存储器分页中,那么上文第2点中的安全性检查继续维持,因为地址仍将是*数据*分页(GCS存储器区域,其是使用由存储器地址空间的分页的粒度定义的分页表条目直接或间接指定的存储器属性所设定)。如果我们仅存储封盖的位置的位[63:12],那么这允许我们存储较少的地址位,因为

这确保我们位于与封盖位置相同的4KB分页。因此,尝试使用该封盖作为返回地址仍将导致权限错误。

[0100] 健全性检查#1还部分保留,其中我们确信我们按预期正切换到相同GCS,尽管仅切换到相同GCS内的位置(但在任何情况下,仍不可能在该GCS先前切换而出时有效地切换到并非堆栈的顶部的该GCS中的位置,因为切换到不在该堆栈的顶部处的位置将导致除了有效封盖值以外的信息被返回,从而导致无效堆栈封盖值有效性检查,且因此执行错误处理响应)。

[0101] 因此,封盖值仅需要指定分页地址,且不需要指定子分页地址位。在切换到传入堆栈时检查封盖值的有效性时,传入堆栈指针的子分页地址位不需要检查以检测封盖值是否有效。

[0102] 这是有用的,因为其允许GCS中的记录的格式具有更多保留位,从而允许该架构的未来扩增。这是非常有用的,因为预期GCS可用以保护除了功能/异常返回状态信息以外的广泛范围的其它信息,且因此具有编码其它记录类型的编码空间可为有用的。

[0103] 图7示出在将在GCSPR 36中的堆栈指针从传出堆栈指针值切换到传入堆栈指针值时执行的步骤(其中传入堆栈指针值是无关于传出堆栈指针值,即,这不仅仅是在推入到堆栈或从堆栈弹出时所执行的目前堆栈指针的增量或减量,而是堆栈指针的任意切换以指定提供作为触发该切换的指令的操作数的地址)。

[0104] 在步骤200,处理电路16执行堆栈指针切换有效性检查操作。这包含发布存储器存取请求,以从基于传入堆栈指针值确定的地址X载入传入数据值。在许多情况下,地址X可为由传入堆栈指针值本身所指示的地址。然而,还将可行的是,传入数据值的地址X通过将偏移施加到由传入堆栈指针值指示的地址而确定,取决于标记为堆栈的“顶部”的位置相对于寄存器36中的堆栈指针值表示的方式。用于从地址X载入传入数据值的存储器存取请求是GCS载入存取,因此如果地址X不对应于GCS存储器,那么可能失败。

[0105] 如果软件正确地运作,预期基于传入堆栈指针值而从地址X获得的传入数据值应是先前在从GCS指针的先前切换上的该GCS结构切换而出时将已放置在传入GCS结构上的堆栈封盖值。然而,处理电路16并不知道(尚未)传入堆栈指针值是否实际上对应于先前存取的GCS结构,因此还可能是传入数据值可能不对应于有效堆栈封盖值。

[0106] 因此,在步骤200,处理电路16执行堆栈指针切换有效性检查操作,以检查传入数据值是否符合至少一堆栈封盖值有效性条件。至少一堆栈封盖值有效性条件至少包含传入数据值的预定部分对应于指示包括地址X的地址空间的分页的给定分页地址,无关于传入数据值的另一部分是否对应于地址X的子分页地址位。更具体来说,传入数据值的该预定部分必须对应于对应于虚拟地址X的虚拟分页地址,使得此检查可检查用以基于传入堆栈指针值存取堆栈的虚拟到物理地址映射是否仍与先前在该堆栈被存取时相同。通过不考虑用于验证至少一堆栈封盖值有效性条件的地址X的子分页地址位,这释放GCS堆栈记录中显著量的编码空间以用于其它目的,如将在下文关于图11更详细地讨论。如下文讨论的图8的步骤224所示,至少一堆栈封盖值有效性条件还可对传入数据值的编码施加其它要求。

[0107] 在步骤202,确定传入数据值是否符合由处理电路16施加的至少一堆栈封盖值有效性条件的每一个。需要满足的至少一堆栈封盖值有效性条件至少包含传入数据值指定存储传入数据值的位置的虚拟分页地址。还可施加一个或多个额外堆栈值有效性条件(例如,

传入数据值的较低部分是表示堆栈封盖值的令牌位模式)。如果不符合任何堆栈封盖值有效性条件,那么在步骤204,触发错误处理响应。例如,此响应可发信号通知错误、在错误报告寄存器中设定错误标志、或将GCSPR 36中的堆栈指针设定成在无法针对任何有效存储器地址而指定的保留地址范围中的无效地址(如果存在任何尝试对该保留范围中的地址执行载入/存储操作或指令提取,那么MMU可触发错误)。

[0108] 如果确定传入数据值不符合所施加的至少一堆栈封盖值有效性条件的每一个时,那么在步骤206,处理电路16允许堆栈指针从传出堆栈指针值切换到传入堆栈指针值(因此传入堆栈指针值被写入到GCSPR 36)。

[0109] 此外,在步骤208,处理电路16执行传出堆栈封盖操作以将有效堆栈封盖值推入到具有基于传出堆栈指针值选择的地址Y的位置。有效堆栈封盖值在将于步骤200检查的预定部分中指定指示包括(虚拟)地址Y的地址空间的分页的(虚拟)分页地址)。因此,这确保传出堆栈经密封(以防止基于对堆栈的顶部的存取触发返回,因为堆栈封盖值具有将对该地址的指令提取或分支导致错误的值),且具有在对应堆栈指针值经指定为用于堆栈切换的传入堆栈指针时将稍后允许有效切换回到该堆栈的适当封盖值。

[0110] 在一些具体实施中,图7所示的传出堆栈封盖操作可响应于还触发堆栈指针切换有效性检查操作的相同指令而执行。

[0111] 然而,图8和图9示出特定示例,其中传出堆栈封盖操作是响应于与触发堆栈指针切换有效性检查操作的指令不同的指令而执行。将这些操作分成分开的指令可大幅简化电路具体实施,因为其避免两个不同存储器地址在相同指令中由MMU 28转换的需求。图8示出用以执行堆栈指针切换有效性检查操作且实施GCSPR36中的堆栈指针的切换的第一堆栈指针切换指令(GCSSS1)的功能。图9示出用以执行传出堆栈封盖操作的第二堆栈指针切换指令(GCSSS2)的功能。

[0112] 如图8的步骤220所示,第一堆栈指针切换指令的操作数是在GCSPR 36中指定的传出堆栈指针值以及在通用寄存器Xn中指定的传入堆栈指针值,该通用寄存器Xn的寄存器指定器是在第一堆栈指针切换指令的指令编码内编码。因此,软件可选择任意通用寄存器以用于定义传入堆栈指针,且先前的指令可以任何任意软件特定的方式设定该传入堆栈指针。GCSPR36不需在GCSSS1指令的编码中明确地编码,因为其是该操作的隐含操作数。

[0113] 响应于GCSSS1指令由解码级10解码且经发布以供管线4的发布级12执行,在步骤222,处理电路16控制存储器存取电路(载入/存储单元26)以发布请求从对应于基于传入堆栈指针值(衍生自寄存器Xn)确定的地址的位置将传入数据值载入的载入存储器存取请求。在一个特定具体实施中,由载入存储器存取请求所指定的地址等于传入堆栈指针值,但其它示例可通过施加偏移到传入堆栈指针值而衍生载入地址。此载入存储器存取请求取决于图4所示的检查,因为其为GCS载入操作,因为其由GCSSS1指令触发。假设载入操作通过这些检查,传入数据值返回,且在步骤224,处理电路16检查(i)传入数据值的预定部分是否对应于用于步骤222的载入的虚拟地址的分页地址部分,且(ii)传入数据值的最低有效部分是否是识别有效堆栈封盖值的特定位模式,该特定位模式不能够由任何有效指令地址的最低有效部分指定。如果这些检查的任一个失败,那么传入数据值不是有效堆栈封盖值,且在步骤226,触发错误处理响应(其可为上文针对图7的步骤204所提及的错误处理响应的类型的任一个)。

[0114] 如果在步骤224的检查通过(传入数据值的预定部分对应于用于在步骤222的载入的虚拟地址的分页地址部分,且传入数据值的最低有效部分具有识别有效堆栈封盖值的特定位模式),即,传入数据值符合至少一堆栈封盖值有效性条件,那么在步骤228,处理电路16控制存储器存取电路26以发布存储存储器存取请求,以将进行中令牌值存储到具有基于传入堆栈指针选择的地址的存储器位置。进行中令牌值指定传出堆栈指针值,且使其最低有效部分设定成不能够由任何有效指令地址的最低有效部分指定的另一位模式,其不同于有效堆栈封盖值的最低有效部分的编码。可能不会预期对于切换堆栈指针的指令使用传出堆栈指针作为操作数是必要的,因为将预期可行的是只要使用GCSPR 36中的传入堆栈指针值盖写传出堆栈指针。然而,通过将传出堆栈指针写入到传入堆栈(在传入堆栈已通过执行步骤224的堆栈封盖值有效性检查验证之后),这帮助支持堆栈指针切换操作被分在两个指令GCSSS1与GCSSS2之间的具体实施,因为其使传出堆栈指针值能够被保存,使得其可用于第二堆栈指针切换指令GCSSS2,而不需要其在GCSPR 36中被盖写之后消耗架构寄存器以用于保留传出堆栈指针值。

[0115] 在步骤230,在步骤224的检查通过的情况下,更新GCSPR 36以指定传入堆栈指针值(因此传出堆栈指针值被盖写)。

[0116] 如图8所示,在步骤222的载入和在步骤228的存储是原子地执行的,使得通过软件执行GCSSS1指令、以及另一线程存取基于传入堆栈指针确定的相同地址所见的结果是与如果在步骤222的载入和在步骤228的存储在不具有任何介入写入到该地址发生在步骤222的载入与在步骤228的存储之间的情况下连续地执行而将发生的结果一致。可能存在可用以确保此原子性的数个不同技术。例如,可使用基于锁定的方法以锁定对相关存储器位置的存取,使得在载入与存储之间的周期中没有到该存储器位置的其它写入可被允许。替代地,由其它线程(例如,在不同处理器核心执行)触发的冲突写入操作仍可能在介入周期中被允许,但如果检测到此写入,那么可提供机制以检测此类冲突写入操作并中止由堆栈指针切换指令执行的处理(例如,清除管线并回绕到包含GCSSS1指令的线程中的先前执行点,以导致GCSSS1指令稍后被重新执行)。载入和存储的原子处理通过减少由GCSSS1指令检查的状态在该检查完成之前改变(其可能有错误的风险)的机会,来帮助改善安全性。

[0117] 图9示出第二堆栈指针切换指令(GCSSS2)的处理。如步骤250所示,指令的操作数是在GCSPR 36中指定的目前值。预期在GCSSS2指令跟着GCSSS1指令的常见使用情况中,GCSPR36中的值应是基于GCSSS1指令的Xn操作数指定的传入堆栈指针值。然而,在实践中,处理器的硬件不可能检查GCSSS2指令是实际上跟着GCSSS1指令,因此虽然下文GCSSS2的伪代码是指如“传入”堆栈指针值的GCSSS2指令的操作数,但更大致来说,其在存储于堆栈指针寄存器36中的目前堆栈指针上操作,而不考虑这是否实际上是与先前GCSSS1指令的传入堆栈指针值相同的值。

[0118] 响应于GCSSS2指令由解码级10解码且经发布以供管线4的发布级12执行,在步骤252,处理电路16控制存储器存取电路26以发布请求从对应于基于从GCSPR 36获得的目前堆栈指针确定的地址的位置将给定数据值载入的载入存储器存取请求。在一个特定具体实施中,由载入存储器存取请求所指定的地址等于来自GCSPR 36的目前堆栈指针值,但其它示例可通过施加偏移到目前堆栈指针值而衍生载入地址。此载入存储器存取请求取决于图4所示的检查,因为其为GCS载入操作,因为其由GCSSS2指令触发。假设载入操作通过这些检

查,给定数据值返回,且在步骤254,处理电路16检查给定数据值是否是有效地形成的进行中令牌。例如,当给定数据值的最低有效位部分匹配用以编码进行中令牌的特定位模式时,处理电路16确定给定数据值是有效地形成的进行中令牌。此位模式不同于有效堆栈封盖值的最低有效部分处的位模式,因此不能够由符合至少一堆栈封盖值有效性条件的任何值指定。此外,此位模式不能够通过任何有效指令地址所指定(例如,因为其在较低位位置中是非零,其比表示指令对准边界的有效度(significance)的位更低有效)。如果给定数据值不是有效地形成的进行中令牌,那么在步骤256,触发错误处理响应(同样地,此可为先前提及的错误处理响应类型的任一个)。

[0119] 如果给定数据值是有效地形成的进行中令牌,那么在步骤258,处理电路16控制存储器存取电路26以发布存储存储器存取请求,以将有效堆栈封盖值写入到具有基于在步骤252载入的给定数据值的部分中指定的给定堆栈指针值确定的地址Y的位置。在一些示例中,地址Y可为相对于给定堆栈指针值以对应于一个堆栈条目的大小的量的偏移,以反映有效堆栈封盖值到堆栈的顶部上的添加意味对应堆栈的堆栈指针将从其先前值(由给定堆栈指针值表示)增量或减量,类似于将新的堆栈条目推入到该堆栈的任何其它堆栈推入操作。因此,在步骤258的存储操作是到与在步骤252从其载入给定数据值的地址不同的地址Y。如果GCSSS2指令并未跟着先前GCSSS1指令,那么在步骤258的存储是到基于GCSSS1指令的传出堆栈指针的传出堆栈结构,而在步骤252的存储是来自传入堆栈结构。有效堆栈封盖值在图8的步骤224针对分页地址检查的预定部分内指定包括虚拟地址Y的地址空间的分页的(虚拟)分页地址,该虚拟地址Y是由从在步骤252载入的给定数据值获得的给定堆栈指针值所指定。此在传出堆栈上设定堆栈封盖值,使得在稍后将堆栈指针切换回传出堆栈(现用作为传入堆栈)时,该稍后堆栈指针切换可通过在图8的步骤224针对GCSSS1指令执行的检查。

[0120] 此外,在图9的步骤260,处理电路响应于GCSSS2指令而更新在GCSPR36中的堆栈指针,以指示将提供给定数据值的条目从堆栈去除。此更新可为以对应于一个堆栈记录的大小的量增量或减量目前堆栈指针。是否施加增量或减量取决于堆栈是否经管理为递减堆栈或升序堆栈(两个选项皆可能,即,对于递减堆栈来说堆栈指针在推入上减量且在弹出上增量,且对于升序堆栈来说反之亦然)。

[0121] 用于表示GCSSS1和GCSSS2指令的功能性的示例伪代码如下所示。应理解,虽然此示出为在硬件处理器中的伪代码,但其将使用实施对应功能的硬件电路逻辑门而实施。

[0122]
```
gcsss1(bits(64) Xn)

bits(64) outgoing_pointer, incoming_pointer, incoming_value,
      in_progress_entry;
outgoing_pointer = GCSPR;
incoming_pointer = Xn;

incoming_value = mem[incoming_pointer];
```

```
if incoming_value == incoming_pointer[63:12]:'000000000001' then
            // valid entry, as incoming value is a validly formed stack cap
            //value
      in_progress_entry = outgoing_pointer[63:3]:'101';  // in_progress_token
      mem[incoming_pointer] = in_progress_entry;  // atomic with earlier load
      GCSPR = incoming_pointer[63:3]:'000'; //switch GCS pointer
else
     assert(FALSE);  // take error handling response

gcsss2()
bits(64) outgoing_pointer, incoming_pointer, outgoing_value;
incoming_pointer = GCSPR;

outgoing_value = mem[incoming_pointer];

if outgoing_value[2:0] == '101' then       // valid in_progress token

      outgoing_pointer[63:0] = (UInt(outgoing_value[63:3]:'000') - 8)<63:0>;
      outgoing_value = outgoing_pointer[63:12]:'000000000001';
       // set up valid entry on outgoing stack, including the stack cap value

      mem[outgoing_pointer] = outgoing_value;
      GCSPR = incoming_pointer + 8;
      // nudge the incoming GCS up one data word, to represent removal of
      //in-progress token
      return outgoing_pointer;
else
      assert(FALSE); // take error handling response.
```

[0123]

[0124] 当然,用于有效堆栈封盖值和进行中令牌的具体编码可与伪代码中示出的那些不同。

[0125] 图10示意地示出基于上文所讨论的操作的堆栈切换的示例。图10的顶部部分示出当在GCSPR 36中的传出GCS指针P1目前指向传出GCS的顶部时的初始状态,且传入GCS(与指针P2相关联)先前已通过在堆栈的顶部存储有效堆栈封盖值而封盖。

[0126] 如图10的中间部分所示,当执行GCSSS1指令将指示指针P2的操作数指定到传入GCS时,堆栈指针切换有效性检查操作通过,因为传入堆栈指针值的分页地址部分P2匹配从对应于传入堆栈指针值的传入GCS上的位置所获得的值的预定部分P2[63:12]。因此,传入堆栈标记为指定传出堆栈指针P1[63:3]的进行中令牌值,且GCSPR 36经更新以指示传入堆栈指针P2。

[0127] 如图10的下部分所示,当执行GCSSS2指令时,传入GCS上的进行中令牌(基于在GCSPR 36中的目前堆栈指针值P2而存取)经验证并且其通过检查,且因此传出堆栈通过将有效堆栈封盖值写入到基于从进行中封盖值令牌获得的传出堆栈指针P1[63:3]所寻址的传出堆栈上的位置而封盖,该有效堆栈封盖值指定包括使用有效堆栈封盖值写入的位置的分页的分页地址部分。此外,在GCSPR 36中的目前堆栈指针经更新,以表示进行中令牌从传入GCS的去除(在此示例中通过加8,但其它示例可使用减量或可具有不同大小的堆栈记录,且因此具有不同增量/减量大小)。

[0128] 图11示出可放置在GCS堆栈上的GCS记录的不同格式的示例:

[0129] ●过程(功能)返回地址记录300,其提供62个位的过程返回地址。

[0130] 0b00的较低2个位表明此为过程返回记录(因为这些位是0,地址是4字节对准,且因此可为对准到地址边界的32位指令编码的有效指令地址)。

[0131] ●堆栈封盖值记录302,其提供52个位的分页地址,且对于较低12个位具有

0b000000000001的值。因为较低位是0b01,如果解译为指令地址,那么此将为未对准地址,因此将触发未对准错误。

[0132] ●进行中令牌记录304,其提供61个位的堆栈指针值(用以表示如上文提及的在堆栈指针切换期间的传出堆栈指针),使用此编码存在对于此传出堆栈指针为双字对准的需求,使得堆栈指针地址的第62个位隐含地为0。记录304的较低3个位具有编码0b101以表明此为进行中令牌。

[0133] ●异常记录306,其使较低5个位设定为0b01001且所有更有效位是0。当在堆栈上检测到异常记录306时,经定义数目的随后记录(其中完整64位可用于记录异常状态信息)提供异常返回状态的项,例如,异常返回地址、与正传回的处理相关联的经存储处理器状态等。

[0134] 此编码意味存在数个经保留编码保持空闲,可用于未来扩增:

[0135] ●编码308,其使较低3个位设定为0b001(与堆栈封盖记录302和异常记录306相同),但其使下9个位设定为除了0b000000000或0b000000001以外的任何者。这些编码允许编码至少$2^9-2=510$种额外记录类型(通过在位[11:3]中设定不同“记录类型”指示符),各自支持在记录的位[63:12]处的剩余52个位中的其它类型信息的记录。此额外的编码空间因堆栈封盖值记录302指定分页地址而非完整地址而释放。

[0136] ●编码310,其使较低2个位[1:0]设定为0b10或0b11,其支持在剩余位[63:2]中指定的完整62位值。

[0137] 图12示出可使用的模拟器具体实施。虽然稍早所述的实施方案以用于操作支持所关注技术的特定处理硬件的装置和方法来实施本发明,但还可能根据本文所述的实施方案提供指令执行环境,其是透过使用计算机程序实施。此类计算机程序常称为模拟器,因为其提供硬件架构的基于软件的具体实施。模拟器计算机程序的种类包含仿真器、虚拟机、模型和二进制转换器(包含动态二进制转换器)。一般来说,模拟器具体实施可在任选地运行主机操作系统1320、支持模拟器程序1310的主机处理器1330上运行。在一些布置中,在硬件与所提供的指令执行环境和/或相同的主机处理器上提供的多个相异指令执行环境之间可有多层模拟。历史上,已需要强大的处理器来提供模拟器具体实施,其以合理速度执行,但此种方法在某些情况下可为有正当理由的,例如当因为兼容性或再使用原因此需要执行另一处理器原生的程序代码时。例如,模拟器具体实施可提供具有不为主机处理器硬件所支持的额外功能性的指令执行环境,或提供一般与不同的硬件架构相关联的指令执行环境。模拟的综述是于“一些有效的架构模拟技术(Some Efficient Architecture Simulation Techniques)”中给出,Robert Bedichek,Winter 1990USENIX Conference,页数53到63。

[0138] 在先前已参照特定硬件架构或特征来描述实施方案的情况下,在模拟实施方案中,可通过合适的软件架构或特征提供等效功能。例如,可在模拟实施方案中将特定电路实施为计算机程序逻辑。类似地,存储器硬件(例如寄存器或高速缓冲存储器)可在模拟实施方案中实施为软件数据结构。于先前描述实施方案中提及的硬件元件的一个或多个存在于主机硬件(例如,主机处理器1330)上的布置中,一些模拟实施方案可在适当时利用主机硬件。

[0139] 模拟器程序1310可存储在计算机可读存储介质(其可为非暂态介质)上,并提供程序接口(指令执行环境)给目标代码1300(其可包含应用程序、操作系统和管理程序),该程

序接口与通过模拟器程序1310模型化的硬件架构的接口相同。因此,目标代码1300的包含如上文描述的堆栈指针切换指令GCSSS1、GCSSS2的程序指令可在指令执行环境内使用模拟器程序1310执行,使得实际上不具有上文讨论的装置2的硬件特征的主机计算机1330可仿真这些特征。类似地,如上文针对MMU 28和载入/存储单元26所讨论的各种存储器管理检查功能以及到存储器的存取的触发(包含支持GCS存储器区域类型),可使用模拟器程序1310的存储器存取程序逻辑1318来仿真。

[0140] 因此,模拟器程序1310可具有处理程序逻辑1312,其模拟上文描述的处理电路4的状态。例如,处理程序逻辑1312可响应于目标代码1300的模拟执行期间发生的事件而模拟执行状态的转变,并执行处理操作。指令解码程序逻辑1314(其可视为处理程序逻辑的部分)解码目标代码1300的指令,且将这些指令映射到主机装置1330的原生指令集中的对应指令集。寄存器仿真程序逻辑1316例如通过存取主机装置1330的寄存器或存储器1332中的数据,而将由目标代码所请求的寄存器存取映射到对主机装置1330的主机硬件上维持的对应数据结构的存取(例如,当指令(例如GCS推入/弹出指令或GCSSS1、GCSSS2指令)需要对GCS指针寄存器的存取而执行时,对表示经模拟GCS指针寄存器的主机位置的存取可通过寄存器仿真程序逻辑1316管理)。存储器存取程序逻辑1318实施地址转换、分页表遍历以及以对应于在上文的硬件实施方案中描述的MMU 28的方式存取控制检查,但还具有映射模拟物理地址(基于针对目标代码1300所限定的页表而通过地址转换获得)的额外功能到用以存取主机存储器1332的主机虚拟地址。这些主机虚拟地址可本身使用由主机所支持的标准地址转换机构来转换成主机物理地址(将主机虚拟地址转换成主机物理地址是在受到模拟器程序1310控制的范围之外)。

[0141] 在本申请中,用语“被配置成...”是用以意指装置的元件具有能够实行该经定义操作的配置。在此上下文中,“配置”意指硬件或软件的互连的布置或方式。例如,该装置可具有专用硬件,其提供经定义的操作,或者处理器或其它处理设备可经编程以执行该功能。“被配置成”并不意味着装置元件需要以任何方式改变以提供所定义的操作。

[0142] 在本申请中,以短语“的至少一个”前缀的特征列表意味着这些特征的任何一个或多个可个别地或组合地提供。例如,“[A]、[B]和[C]的至少一个”涵盖下列选项的任一个:单独A(不具有B或C)、单独B(不具有A或C)、单独C(不具有A或B)、A和B的组合(不具有C)、A和C的组合(不具有B)、B和C的组合(不具有A)、或A、B和C的组合。

[0143] 虽然本文已参照附图详细地描述本发明的说明性实施方案,应了解本发明不限于那些精确实施方案,且本领域的技术人员可于其中实行各种变化与修改,而不脱离如随附权利要求书所定义的本发明的范围。

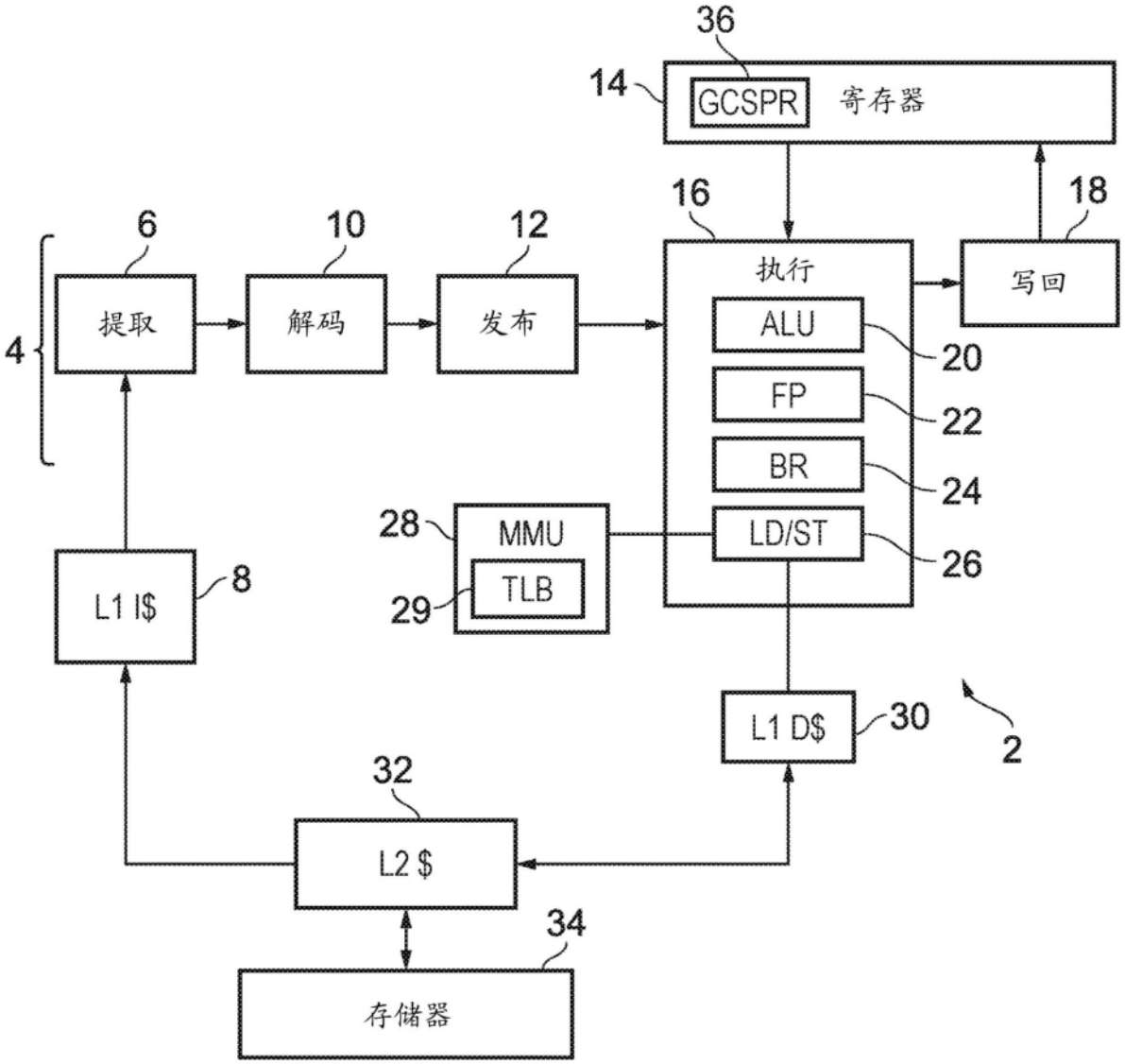

36
14
GCSPR
寄存器
16
18
6
10
12
执行
写回
4
提取
解码
发布
ALU
20
FP
22
BR
24
28
MMU
LD/ST
26
29
TLB
L1 I$
8
L1 D$
30
2
32
L2 $
34
存储器

图1

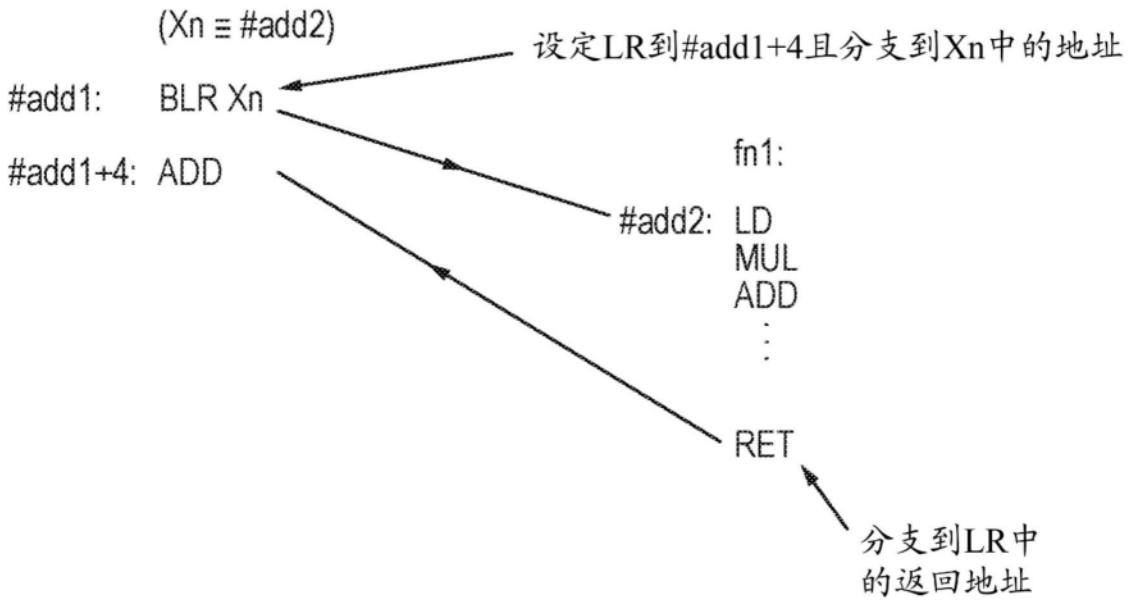

(Xn ≡ #add2)
设定LR到#add1+4且分支到Xn中的地址
#add1: BLR Xn
#add1+4: ADD
fn1:
#add2: LD
MUL
ADD
RET
分支到LR中的返回地址

图2

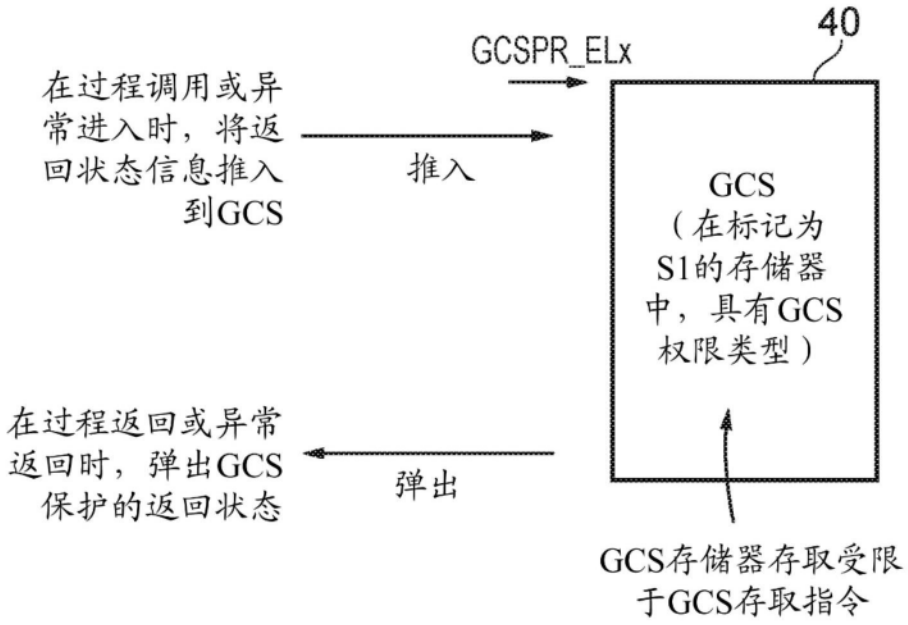

GCSPR_ELx
40
在过程调用或异常进入时，将返回状态信息推入到GCS
推入
GCS（在标记为S1的存储器中，具有GCS权限类型）
在过程返回或异常返回时，弹出GCS保护的返回状态
弹出
GCS存储器存取受限于GCS存取指令

图3

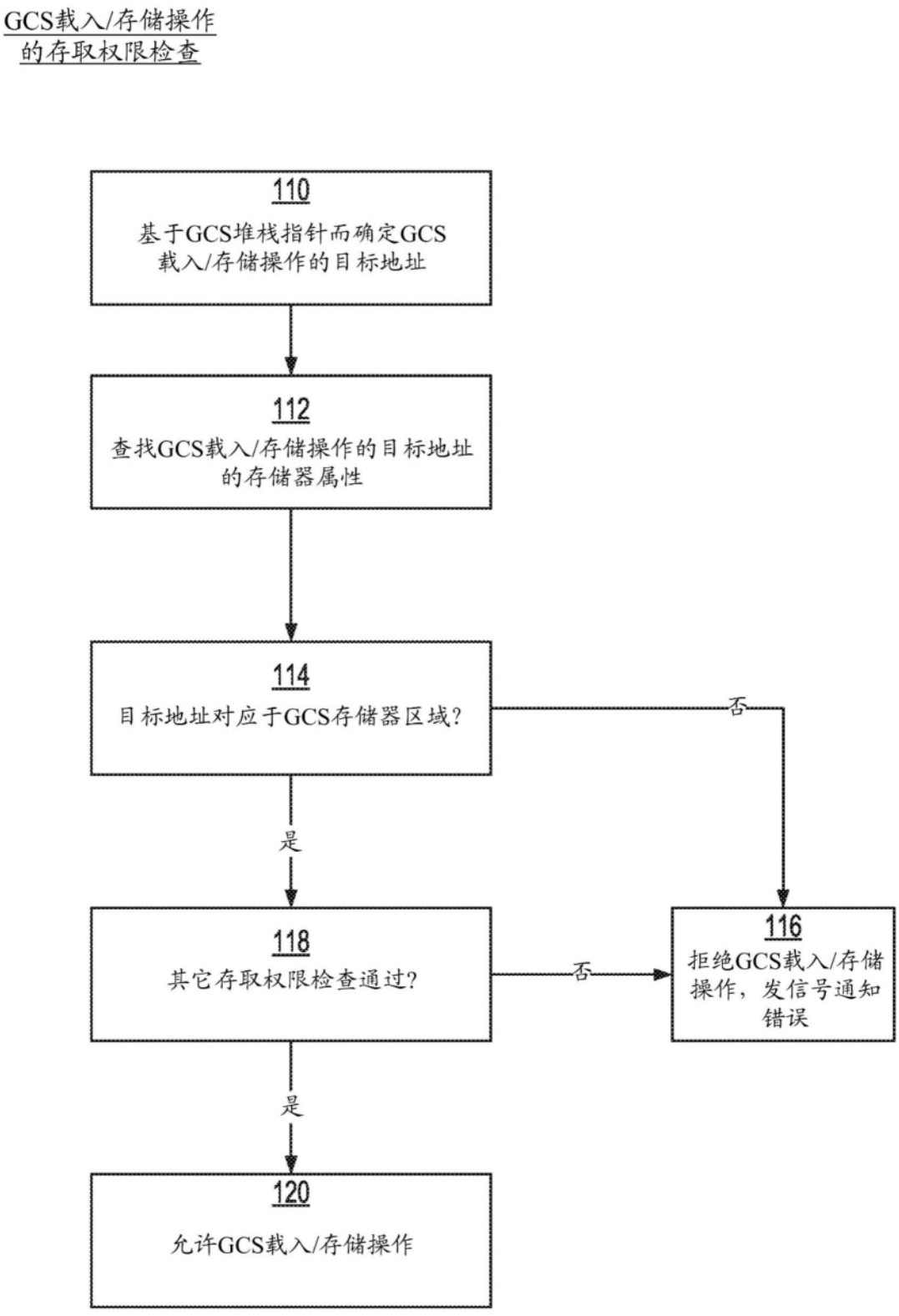

GCS载入/存储操作的存取权限检查
110
基于GCS堆栈指针而确定GCS载入/存储操作的目标地址
112
查找GCS载入/存储操作的目标地址的存储器属性
114
目标地址对应于GCS存储器区域?
否
是
118
其它存取权限检查通过?
否
116
拒绝GCS载入/存储操作，发信号通知错误
是
120
允许GCS载入/存储操作

图4

非GCS载入/存储操作的
存取权限检查

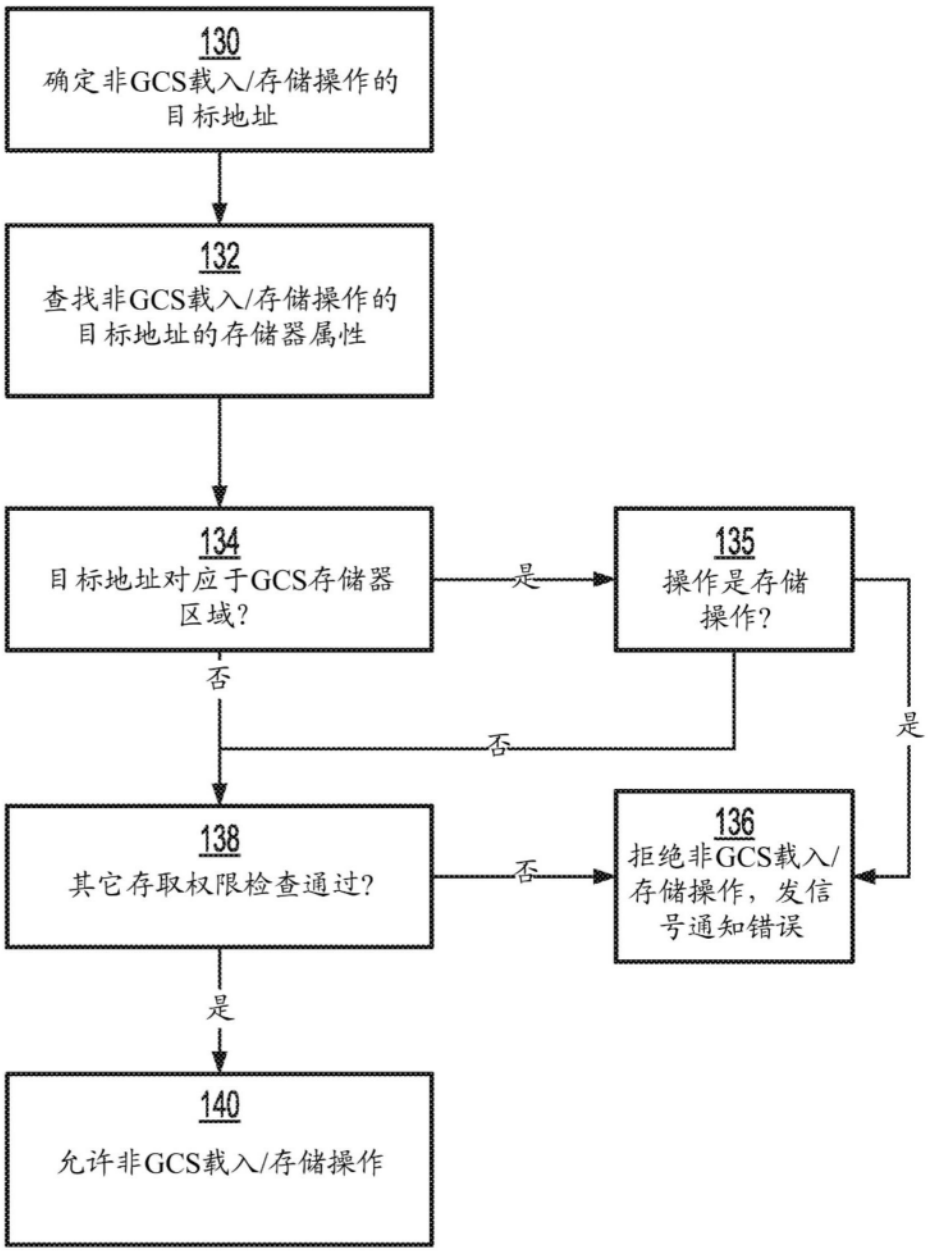


图5

指令提取或分支的
权限检查

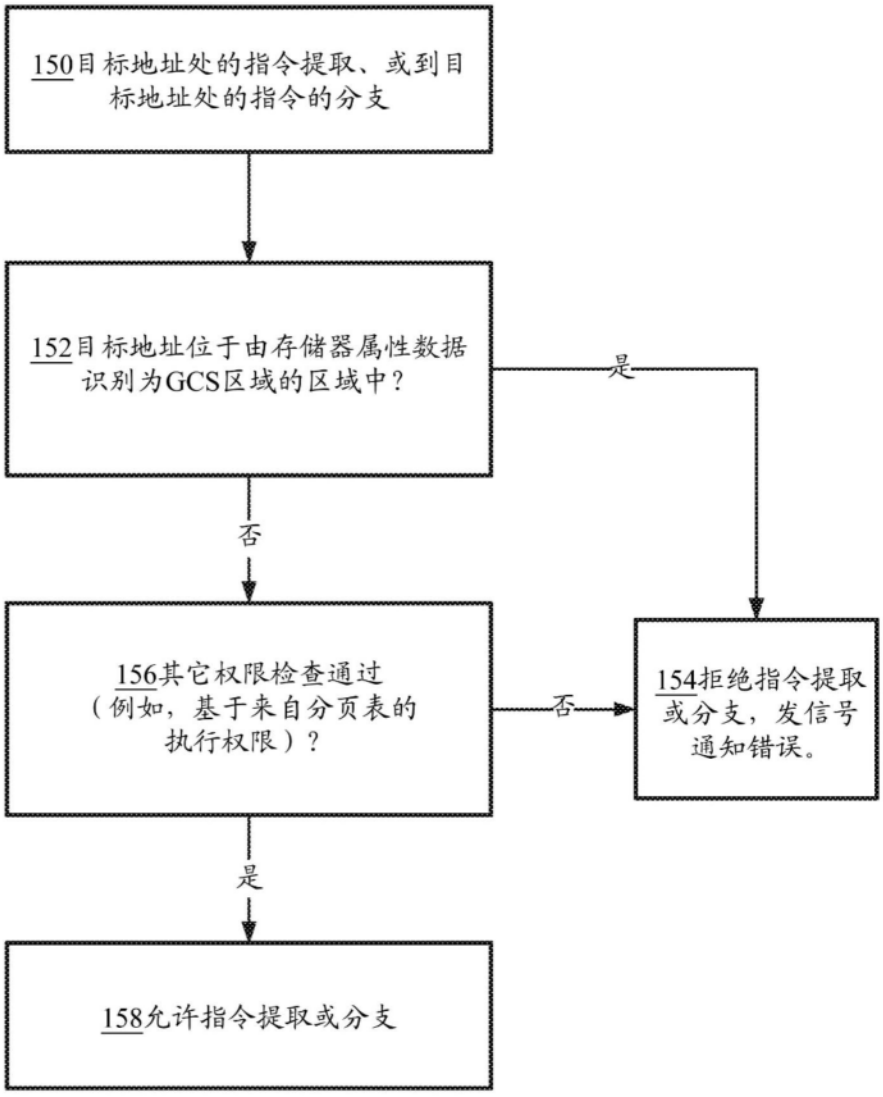


图6

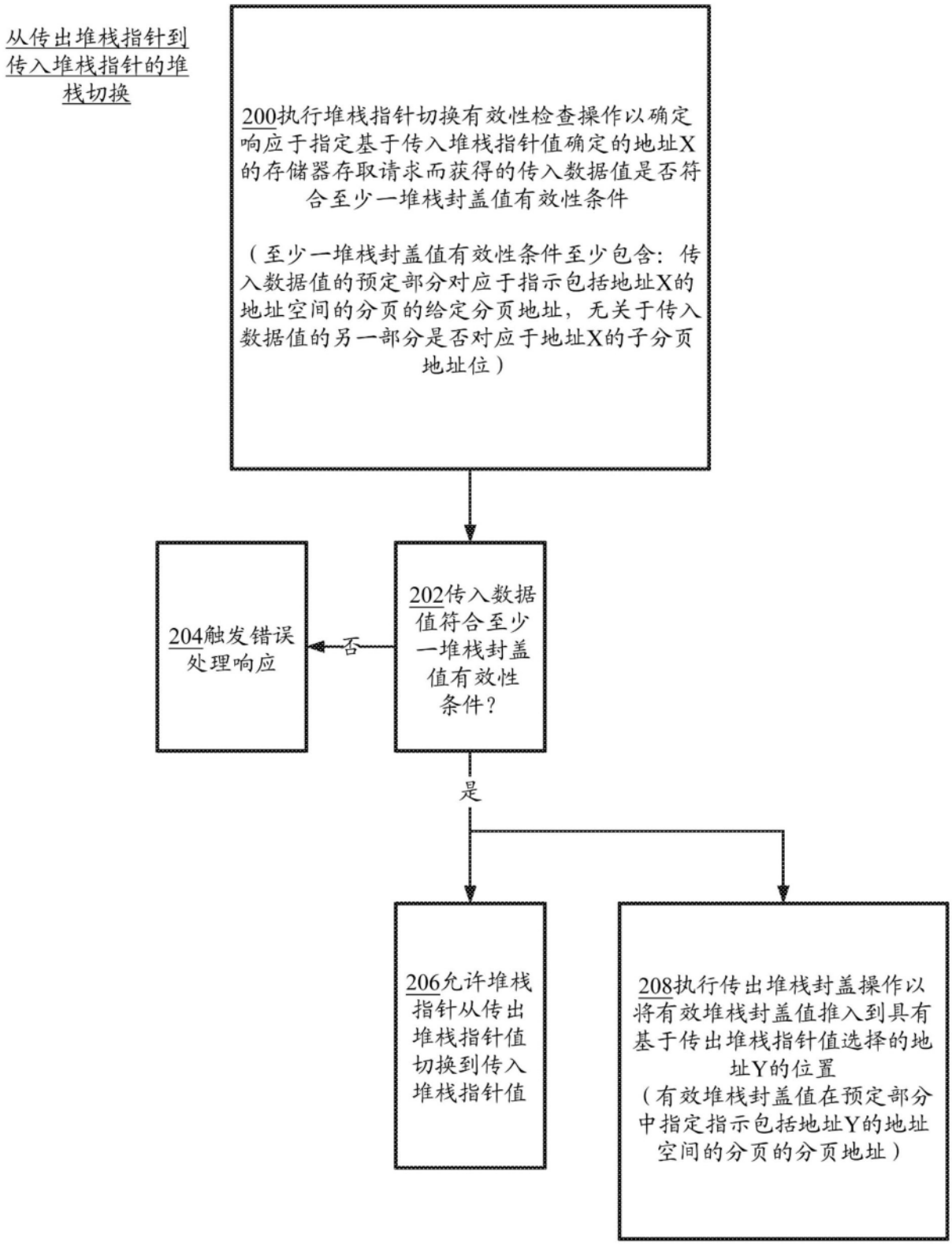

从传出堆栈指针到传入堆栈指针的堆栈切换
200执行堆栈指针切换有效性检查操作以确定响应于指定基于传入堆栈指针值确定的地址X的存储器存取请求而获得的传入数据值是否符合至少一堆栈封盖值有效性条件
（至少一堆栈封盖值有效性条件至少包含：传入数据值的预定部分对应于指示包括地址X的地址空间的分页的给定分页地址，无关于传入数据值的另一部分是否对应于地址X的子分页地址位）
202传入数据值符合至少一堆栈封盖值有效性条件？
否
204触发错误处理响应
是
206允许堆栈指针从传出堆栈指针值切换到传入堆栈指针值
208执行传出堆栈封盖操作以将有效堆栈封盖值推入到具有基于传出堆栈指针值选择的地址Y的位置
（有效堆栈封盖值在预定部分中指定指示包括地址Y的地址空间的分页的分页地址）

图7

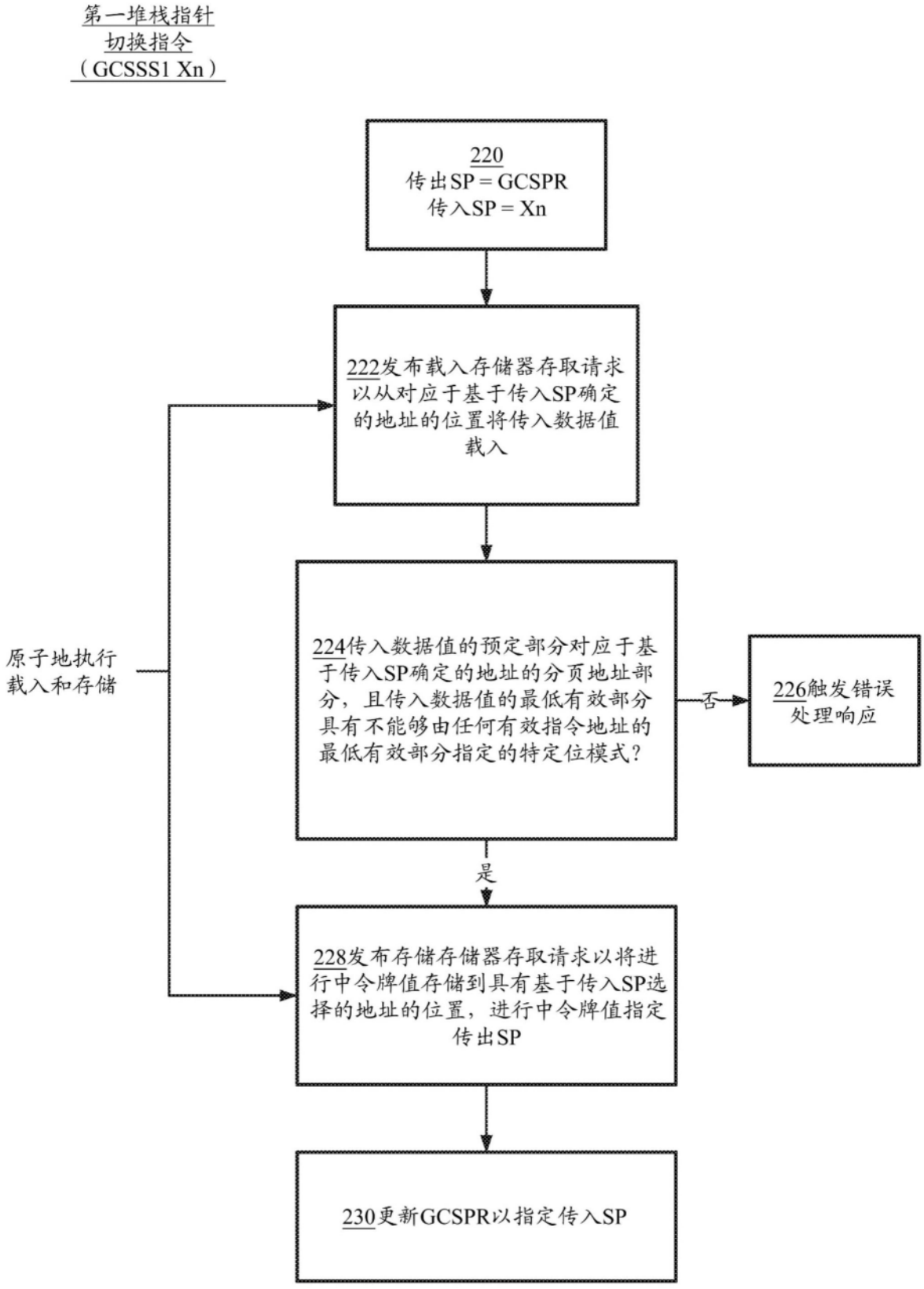

第一堆栈指针
切换指令
（GCSSS1 Xn）
220
传出SP = GCSPR
传入SP = Xn
222发布载入存储器存取请求以从对应于基于传入SP确定的地址的位置将传入数据值载入
224传入数据值的预定部分对应于基于传入SP确定的地址的分页地址部分，且传入数据值的最低有效部分具有不能够由任何有效指令地址的最低有效部分指定的特定位模式？
否
226触发错误处理响应
是
228发布存储存储器存取请求以将进行中令牌值存储到具有基于传入SP选择的地址的位置，进行中令牌值指定传出SP
230更新GCSPR以指定传入SP
原子地执行载入和存储

图8

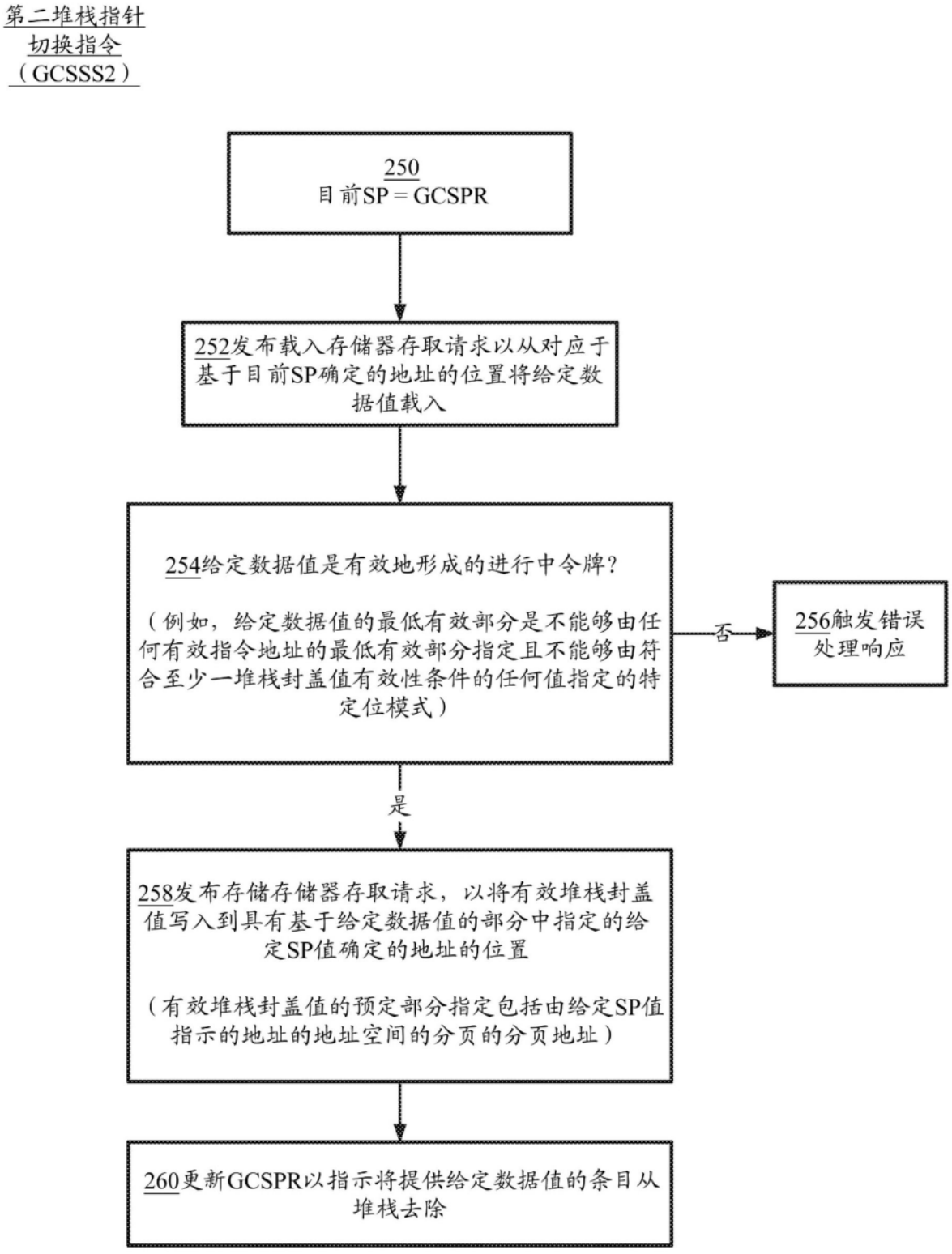

第二堆栈指针
切换指令
（GCSSS2）
250
目前SP = GCSPR
252发布载入存储器存取请求以从对应于基于目前SP确定的地址的位置将给定数据值载入
254给定数据值是有效地形成的进行中令牌？
（例如，给定数据值的最低有效部分是不能够由任何有效指令地址的最低有效部分指定且不能够由符合至少一堆栈封盖值有效性条件的任何值指定的特定位模式）
否
256触发错误处理响应
是
258发布存储存储器存取请求，以将有效堆栈封盖值写入到具有基于给定数据值的部分中指定的给定SP值确定的地址的位置
（有效堆栈封盖值的预定部分指定包括由给定SP值指示的地址的地址空间的分页的分页地址）
260更新GCSPR以指示将提供给定数据值的条目从堆栈去除

图9

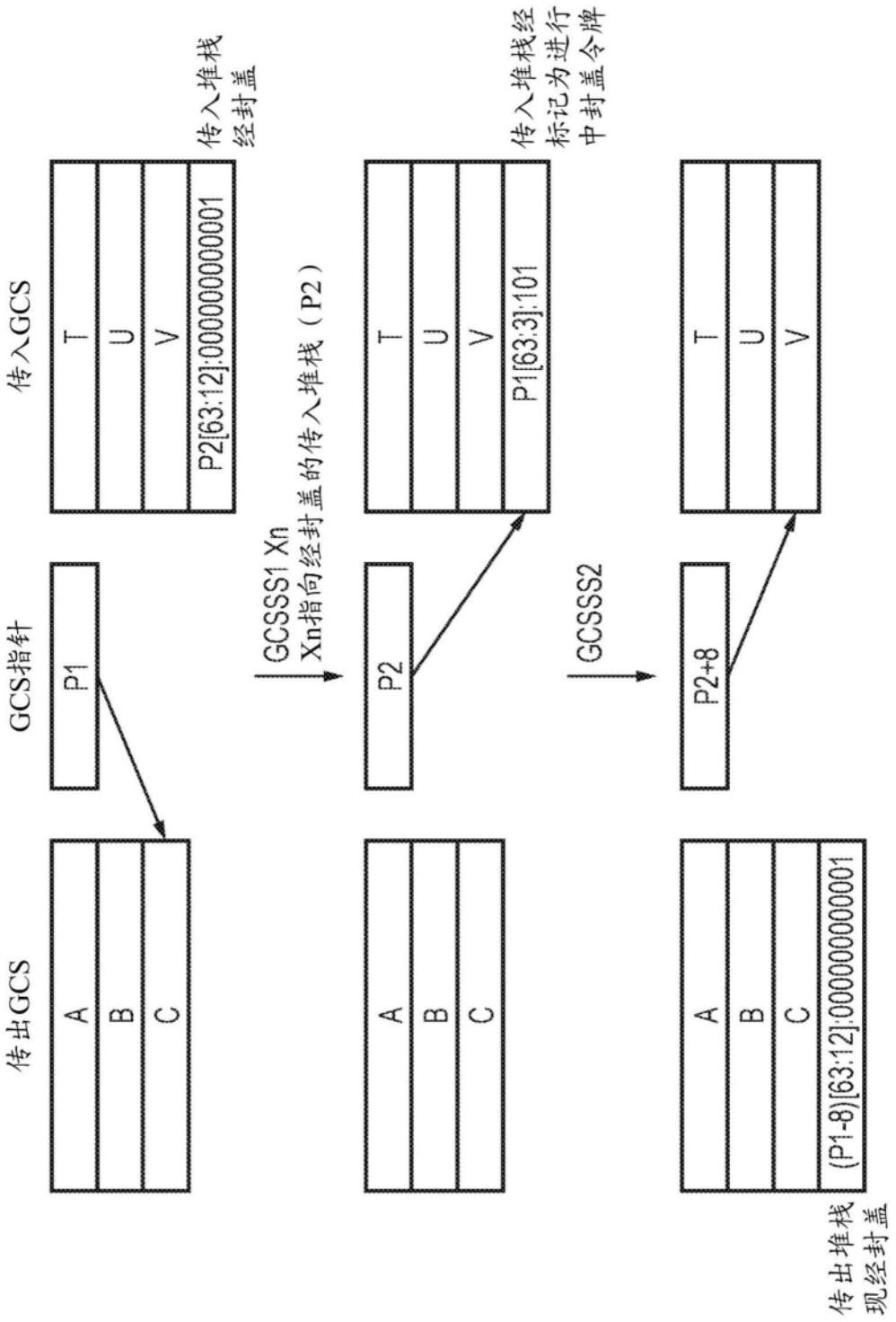

传入GCS
T
U
V
P2[63:12]:000000000001
传入堆栈经封盖
GCS指针
P1
传出GCS
A
B
C
GCSSS1 Xn
Xn指向经封盖的传入堆栈（P2）
P2
P1[63:3]:101
传入堆栈经标记为进行中封盖令牌
GCSSS2
P2+8
(P1-8)[63:12]:000000000001
传出堆栈现经封盖

图10

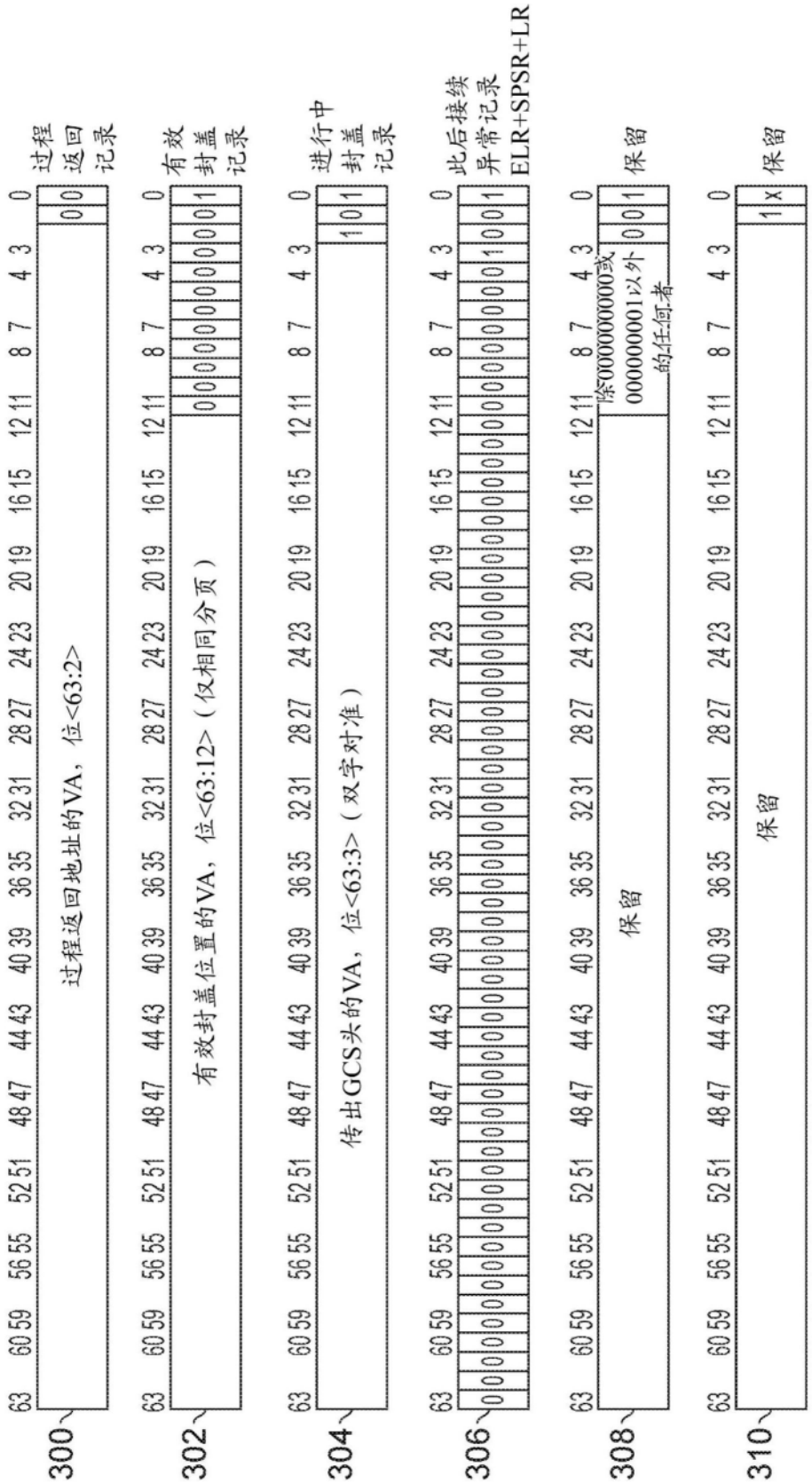

300
过程返回记录
过程返回地址的VA，位<63:2>
302
有效封盖记录
有效封盖位置的VA，位<63:12>（仅相同分页）
304
进行中封盖记录
传出GCS头的VA，位<63:3>（双字对准）
306
此后接续异常记录
ELR+SPSR+LR
308
保留
除0000000001以外的任何值
310
保留
保留

图11

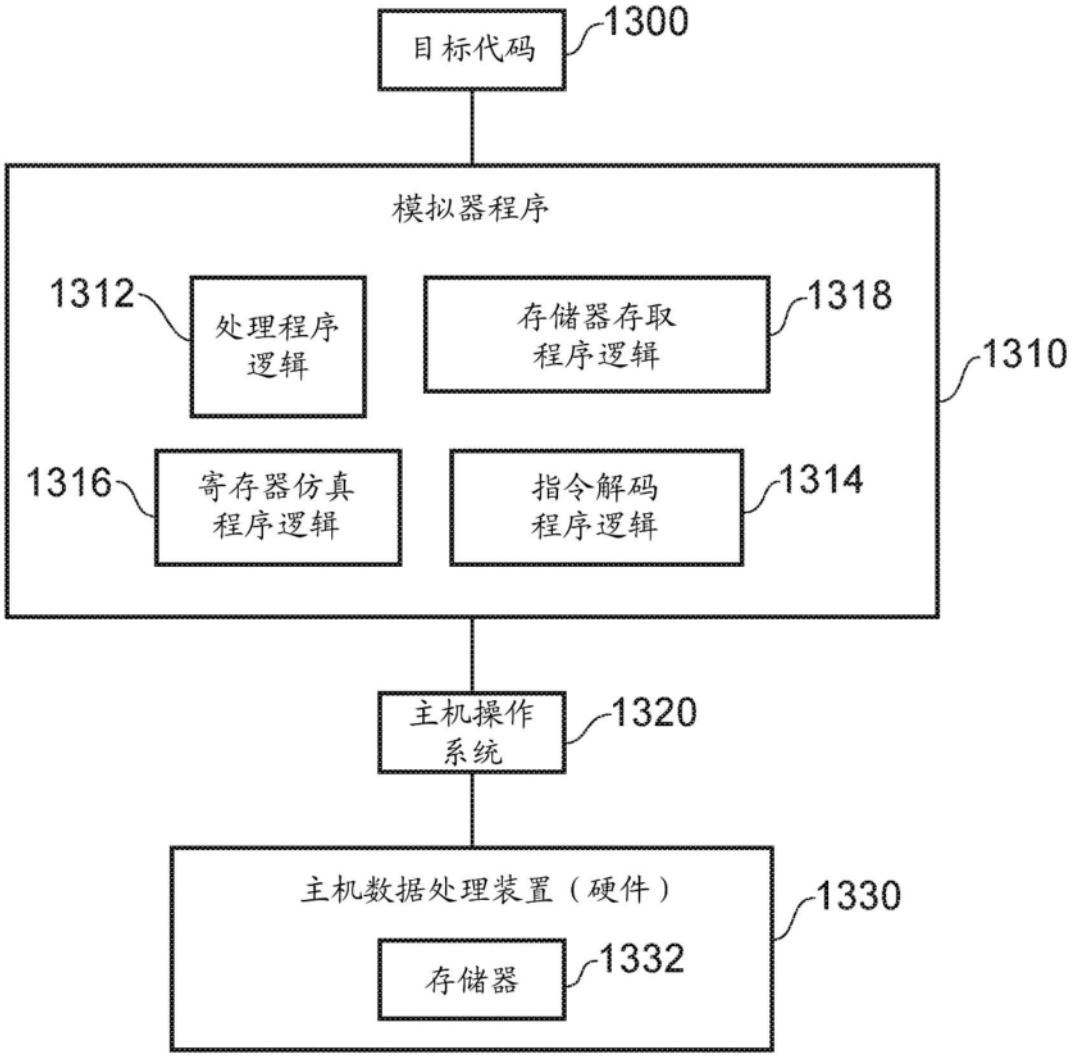

目标代码
1300
模拟器程序
1312
处理程序逻辑
存储器存取程序逻辑
1318
1310
1316
寄存器仿真程序逻辑
指令解码程序逻辑
1314
主机操作系统
1320
主机数据处理装置（硬件）
1330
存储器
1332

图12