



(12)发明专利

(10)授权公告号 CN 104106044 B

(45)授权公告日 2017.10.31

(21)申请号 201380008133.7

(22)申请日 2013.02.07

(65)同一申请的已公布的文献号

申请公布号 CN 104106044 A

(43)申请公布日 2014.10.15

(30)优先权数据

61/595,773 2012.02.07 US

13/655,499 2012.10.19 US

(85)PCT国际申请进入国家阶段日

2014.08.06

(86)PCT国际申请的申请数据

PCT/US2013/025187 2013.02.07

(87)PCT国际申请的公布数据

W02013/119842 EN 2013.08.15

(73)专利权人 高通股份有限公司

地址 美国加利福尼亚州

(72)发明人 查尔斯·约瑟夫·塔伯尼

埃里克·詹姆斯·普隆迪克

卢西恩·科德雷斯库

苏雷什·K·文库马汉提

埃万德罗·卡洛斯·梅内塞斯

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 宋献涛

(51)Int.Cl.

G06F 9/30(2006.01)

G06F 9/32(2006.01)

(56)对比文件

US 2005/0262329 A1, 2005.11.24, 说明书第[0025]-[0027]、[0066]-[0080]段,附图1、6.

审查员 白桦

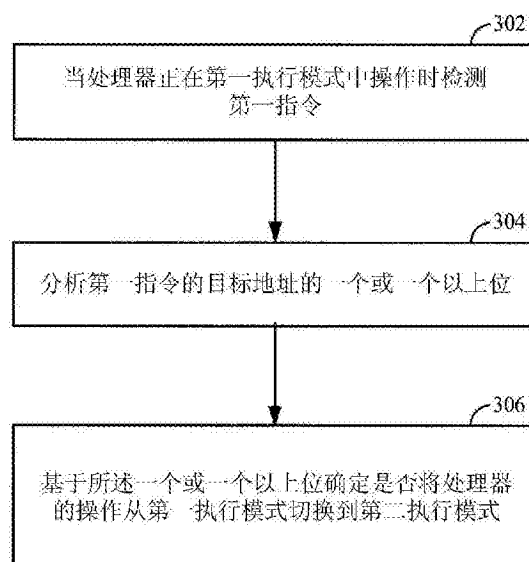
权利要求书2页 说明书8页 附图5页

(54)发明名称

使用经调用函数的地址的最低有效位来切换处理器模式

(57)摘要

本发明揭示用于在处理系统中的执行模式之间进行跟踪和切换的系统和方法。处理系统经配置以在包含选自经典/对准模式和压缩/未对准模式的第一和第二执行模式的至少两个指令执行模式中执行指令。例如调用和返回等选定指令的目标地址在所述压缩模式中被迫未对准,例如所述压缩模式中的所述目标地址的最低有效位(对准位)等此一个或一个以上位不同于所述经典模式中的对应对准位。当在所述第一模式中的执行期间遇到所述选定指令时,将操作切换到所述第二模式的决策是基于分析所述选定指令的所述目标地址的所述对准位。



1. 一种在处理器中的执行模式之间切换的方法,所述方法包括:

当所述处理器正在第一执行模式中操作时检测引起目标地址的流的改变的第一指令;

分析所述目标地址的两个对准位,其中所述对准位是所述目标地址的两个最低有效位,并指示到预定位宽边界的对准;以及

基于两个对准位都具有零值还是至少一个对准位具有非零值来确定是否将所述处理器的操作从所述第一执行模式切换到第二执行模式,

其中所述目标地址的所述两个对准位在所述第一执行模式或所述第二执行模式中的一者中被迫未对准。

2. 根据权利要求1所述的方法,其中:

所述第一执行模式是包括与所述预定位宽边界对准的指令的经典模式;且

所述第二执行模式是包括与所述预定位宽边界未对准的指令的压缩模式。

3. 根据权利要求2所述的方法,其中所述预定位宽边界是32位边界,使得所述经典模式中的所述对准位为“00”,且所述压缩模式中的所述对准位为“01”、“10”或“11”中的一者。

4. 根据权利要求1所述的方法,其中:

所述第一执行模式是包括与所述预定位宽边界未对准的指令的压缩模式;且

所述第二执行模式是包括与所述预定位宽边界对准的指令的经典模式。

5. 根据权利要求4所述的方法,其中所述预定位宽边界是32位边界,使得所述经典模式中的所述对准位为“00”,且所述压缩模式中的所述对准位为“01”、“10”或“11”中的一者。

6. 根据权利要求1所述的方法,其中所述第一指令是调用或返回指令中的一者。

7. 根据权利要求1所述的方法,其中所述第一指令是call或return指令,在其中指定包括所述目标地址的寄存器。

8. 根据权利要求1所述的方法,其中所述第一指令是jump指令,所述方法进一步包括将所述jump指令限于仅跳跃到将不会引起模式切换的指令。

9. 根据权利要求1所述的方法,其中所述第一指令是jump指令,所述方法进一步包括基于所述第一执行模式确定是否将所述处理器的操作从所述第一执行模式切换到第二执行模式。

10. 根据权利要求1所述的方法,其进一步包括基于所述对准位在用户状态寄存器中提供执行模式的指示,所述处理器正在所述执行模式中操作。

11. 一种用于切换处理器模式的设备,其包括:

所述处理器,其经配置以在包含第一执行模式和第二执行模式的至少两个执行模式中执行指令;

经配置以在所述第一执行模式中执行期间检测引起目标地址的流的改变的第一指令的逻辑;

经配置以分析所述目标地址的两个对准位的逻辑,其中所述对准位是所述目标地址的两个最低有效位,并指示到预定位宽边界的对准;以及

经配置以基于两个对准位都具有零值还是至少一个对准位具有非零值来确定是否将所述处理器的操作从所述第一执行模式切换到所述第二执行模式的逻辑,

其中所述目标地址的所述两个对准位在所述第一执行模式或所述第二执行模式中的

一者中被迫未对准。

12. 根据权利要求11所述的设备, 其中:

所述第一执行模式是包括与所述预定位宽边界对准的指令的经典模式; 且

所述第二执行模式是包括与所述预定位宽边界未对准的指令的压缩模式。

13. 根据权利要求12所述的设备, 其中所述预定位宽边界是32位边界, 使得所述经典模式中的所述对准位为“00”, 且所述压缩模式中的所述对准位为“01”、“10”或“11”中的一者。

14. 根据权利要求11所述的设备, 其中:

所述第一执行模式是包括与所述预定位宽边界未对准的指令的压缩模式; 且

所述第二执行模式是包括与所述预定位宽边界对准的指令的经典模式。

15. 根据权利要求14所述的设备, 其中所述预定位宽边界是32位边界, 使得所述经典模式中的所述对准位为“00”, 且所述压缩模式中的所述对准位为“01”、“10”或“11”中的一者。

16. 根据权利要求11所述的设备, 其中所述第一指令是call、return或tailcall指令中的一者。

17. 根据权利要求11所述的设备, 其进一步包括经配置以基于所述对准位在用户状态寄存器中提供执行模式的指示的用户状态寄存器, 所述处理器正在所述执行模式中操作。

18. 根据权利要求11所述的设备, 其集成在至少一个半导体裸片中。

19. 根据权利要求11所述的设备, 其集成到选自由以下各项组成的群组的装置中: 机顶盒、音乐播放器、视频播放器、娱乐单元、导航装置、通信装置、个人数字助理PDA、固定位置数据单元和计算机。

20. 一种处理系统, 其包括:

用于在包含第一执行模式和第二执行模式的至少两个执行模式中执行指令的装置;

用于在所述第一执行模式中执行期间检测到目标地址的流的改变的第一指令的装置;

用于分析所述目标地址的两个对准位的装置, 其中所述对准位是所述目标地址的两个最低有效位, 并指示到预定位宽边界的对准; 以及

用于基于两个对准位都具有零值还是至少一个对准位具有非零值来确定是否将处理器的操作从所述第一执行模式切换到所述第二执行模式的装置,

其中所述目标地址的所述两个对准位在所述第一执行模式或所述第二执行模式中的一者中被迫未对准。

使用经调用函数的地址的最低有效位来切换处理器模式

[0001] 根据35U.S.C.§119主张优先权

[0002] 本专利申请案主张2012年2月7日申请的标题为“使用经调用函数的地址的最低有效位来切换处理器模式 (USING THE LEAST SIGNIFICANT BITS OF A CALLED FUNCTION'S ADDRESS TO SWITCH PROCESSOR MODES)”的第61/595,773号临时申请案的优先权,所述临时申请案指派给本受让人且在此以引用的方式明确地并入本文中。

技术领域

[0003] 所揭示的实施例涉及在处理系统中的执行模式之间切换。更特定来说,示范性实施例针对包括至少两个指令模式的处理系统,以及用于使用经调用函数的一个或一个以上最低有效位在所述至少两个指令模式之间切换的技术。

背景技术

[0004] 处理器通常经配置以支持多个执行模式。举例来说,第一执行模式可包括固定宽度的指令,而第二执行模式可支持可变宽度指令。共同地址空间可在每一模式中的指令之间共享,使得相同地址或程序计数器(PC)值可在处理器处于第一执行模式的情况下指向第一指令,且在处理器处于第二执行模式的情况下指向第二指令。因为处理器可能能够在两个模式之间切换,所以有必要跟踪执行模式以确认两个指令中的哪一者由当前PC值寻址,且因此确保指令正被正确执行。

[0005] 参考已知ARM处理器,ARM模式支持固定宽度(32位)指令,而THUMB模式支持可变宽度(16位和32位)指令。字节可寻址存储器中的共同地址空间可包括ARM和THUMB模式两者中的指令。然而,两个模式中的指令的对准可不同。举例来说,ARM模式指令可在32位边界对准,使得ARM模式指令的两个最低有效地址位将始终为“00”。另一方面,THUMB模式指令可在16位或32位边界对准,使得THUMB模式指令的两个最低有效地址位可为“10”或“00”。因此,单单知道地址不足以最终确定两个模式ARM或THUMB中的哪一者当前正被执行。

[0006] 为跟踪哪一模式正被执行,ARM处理器采用一种方法,其包含将模式位存储在PC的最低有效位(LSB)中。THUMB模式中PC的LSB可为保留位,因为如上所述所述两个最低有效地址位为“10”或“00”且因此LSB(其在两个情况下均为“0”)可用于存储模式位。执行可在函数调用(例如,跳跃指令)时在ARM模式与THUMB模式之间切换。因此,如果跳跃指令将执行转移到包括THUMB模式中的指令的函数,那么ARM处理器通过跳跃到函数+1的地址而实施到THUMB模式的此模式改变。以此方式,THUMB模式中两个最低有效位的值为“11”或“01”,借此将其与ARM模式中两个最低有效位的“00”值区分。

[0007] 虽然ARM处理器采用的以上方法实现跟踪执行模式的目标,但其具有对应局限性。举例来说,如上所述,对于到THUMB模式的每次跳跃需要加“1”。这使软件变复杂且尤其是连接器。连接器必须确定每一函数的模式且接着对于到每一函数的所有调用在常规调用与模式切换调用之间选择。另外,调试器也受到影响,因为调试器将需要跟踪正从调试器内调用的函数以确定这些函数是ARM还是THUMB函数,以便分别确定跳跃到函数的PC值还是跳跃到

函数+1的PC值。

[0008] 与以上方法相关联的又一缺点源自使用PC的LSB来存储模式位。因为LSB用于表示所述模式,所以LSB不再是实际地址的一部分。因此,如果ARM处理器将向THUMB模式包含8位和24位指令,那么THUMB模式中每个函数的开始点将仍需要16位对准以便促进模式跟踪。这将必须在函数开始之前需要所有函数的约一半来引入8位空操作指令(nop)以便将函数中的第一指令与16位边界对准。

[0009] 例如MIPS和PowerPC等其它已知处理器架构面临与在执行模式之间切换相关的类似挑战,且采用类似有缺点的方法来处置这些挑战。相应地,现有架构展现其在学习模式之间跟踪和切换的方法中的缺点。举例来说,一些方法涉及使用存储在相关联存储器管理单元(MMU)中的指令的PC值来确定指令的执行模式,因为MMU中的任何页可属于任一模式。使用页属性来确定模式的增加的硬件成本方面的缺点是,模式位占据MMU的页表条目中的额外空间,且常规页表条目的大小存在限制。使用页属性来确定模式的软件成本方面的缺点包含使加载器变复杂,将必须确定页被推测处于哪一模式且接着填充所述页的页表条目中的适当位。此外,调试器将必须查找特定页的页表中的条目(在其可分解所述页中的指令之前)。其它已知方法同样存在缺陷,其包含用于使用所保留操作代码(OpCode)位确定与地址相关联的模式的特定程序代码。

[0010] 因此,此项技术中对应地需要克服用于在执行模式之间跟踪和切换的上文提及的方法的缺点。

发明内容

[0011] 本发明的示范性实施例针对用于在处理系统中的执行模式之间切换的系统和方法。更特定来说,示范性实施例针对包括至少两个指令模式(经典/对准模式和压缩/未对准模式)的处理系统,以及使用经调用函数的地址的一个或一个以上最低有效位在所述至少两个指令模式之间切换。

[0012] 举例来说,一示范性实施例针对一种在处理器中的执行模式之间切换的方法,所述方法包括:当处理器正在第一执行模式中操作时检测第一指令;分析第一指令的目标地址的一个或一个以上位;以及基于所述一个或一个以上位确定是否将处理器的操作从第一执行模式切换到第二执行模式。

[0013] 另一示范性实施例针对一种设备,其包括:处理器,所述处理器经配置以在包含第一执行模式和第二执行模式的至少两个执行模式中执行指令;经配置以在第一执行模式中执行期间检测第一指令的逻辑;经配置以分析第一指令的目标地址的一个或一个以上位的逻辑;以及经配置以基于所述一个或一个以上位确定是否将处理器的操作从第一执行模式切换到第二执行模式的逻辑。

[0014] 又一示范性实施例针对一种处理系统,其包括:用于在包含第一执行模式和第二执行模式的至少两个执行模式中执行指令的装置;用于在第一执行模式中执行期间检测第一指令的装置;用于分析第一指令的目标地址的一个或一个以上位的装置;以及用于基于所述一个或一个以上位确定是否将处理器的操作从第一执行模式切换到第二执行模式的装置。

[0015] 另一示范性实施例针对一种非暂时性计算机可读存储媒体,其包括代码,所述代

码当由处理器执行时致使所述处理器执行用于在处理器的执行模式之间切换的操作,所述非暂时性计算机可读存储媒体包括:用于当处理器正在第一执行模式中操作时检测第一指令的代码;用于分析第一指令的目标地址的一个或一个以上位的代码;以及用于基于所述一个或一个以上位确定是否将处理器的操作从第一执行模式切换到第二执行模式的代码。

附图说明

[0016] 呈现附图来辅助对本发明的实施例的描述,且仅出于说明而非限制所述实施例的目的来提供附图。

[0017] 图1是根据示范性实施例配置的处理系统的简化示意图。

[0018] 图2说明根据示范性实施例由经配置以在执行模式之间切换的处理器执行的示范性代码序列。

[0019] 图3说明根据示范性实施例用于在执行模式之间切换的方法的操作流程。

[0020] 图4A-B说明根据示范性实施例用于在执行模式之间跟踪和切换的方案。

[0021] 图5说明其中可有利地采用本发明的实施例的示范性无线通信系统500。

具体实施方式

[0022] 在以下针对本发明特定实施例的描述和有关图式中揭示本发明的若干方面。可在不脱离本发明的范围的情况下设计替代实施例。此外,将不会详细描述本发明的众所周知的元件,或将省略所述元件以免混淆本发明的相关细节。

[0023] 本文使用词语“示范性”来表示“充当实例、例子或说明”。本文中被描述为“示范性”的任何实施例不必被理解为比其它实施例优选或有利。同样,术语“本发明的实施例”并非要求本发明的所有实施例均包含所论述的特征、优点或操作模式。

[0024] 本文中所使用的术语仅出于描述特定实施例的目的且并不希望限制本发明的实施例。如在本文中所使用,除非上下文另外清楚地指示,否则希望单数形式“一”和“所述”也包含复数形式。将进一步了解,术语“包括”和/或“包含”在用于本文中时指定所陈述的特征、整数、步骤、操作、元件和/或组件的存在,但不排除一个或一个以上其它特征、整数、步骤、操作、元件、组件和/或其群组的存在或添加。

[0025] 此外,依据将由(例如)计算装置的元件执行的动作序列来描述许多实施例。将认识到,可由特定电路(例如,专用集成电路(ASIC))、由正由一个或一个以上处理器执行的程序指令或由两者的组合来执行本文中描述的各种动作。此外,可认为本文中描述的这些动作序列完全实施于任何形式的计算机可读存储媒体内,所述计算机可读存储媒体中已存储了对应计算机指令集,所述指令在被执行时将致使相关联处理器执行本文中描述的功能性。因此,本发明的各个方面可以许多不同形式来实施,所有所述形式均预期在所主张的标的物的范围内。此外,对于本文中描述的实施例的每一者来说,任何此类实施例的对应形式可在本文中被描述为(例如)“经配置以(执行所描述的动作)的逻辑”。

[0026] 示范性实施例包含用于在执行模式之间跟踪和切换同时避免与现有方法相关联的缺点的技术。一个实施例可包含至少两个执行模式,其中关于模式内指令的相应对准来界定执行模式。对指令的对准的参考通常相对于存储器地址空间的32位(或4字节)进行,但此不应被解释为是限制。所揭示的技术可容易扩展到其它存储器地址边界,而不脱离本发

明的范围。

[0027] 因此,在第一执行模式中,可允许指令未对准/不对准或无限制地任意对准。此第一执行模式将在本文中称为“压缩”模式。在第二执行模式中,指令的对准可限于对应存储器地址空间中的指定边界,例如32位边界。此第二执行模式将在本文中称为“经典”模式。

[0028] 现参看图1,展示示范性处理系统100的简化示意图。处理系统100被展示为包括耦合到存储器104的处理器102。虽然未说明,但处理系统100可包括各种其它组件,例如一个或一个以上指令和/或数据高速缓冲存储器、I/O装置、协处理器等,如此项技术中众所周知。存储器104可为字节可寻址的且包括可未对准或在32位边界对准的指令(即,两个指令分别处于压缩模式和经典模式中)。处理器102可经配置以在经典模式中以及压缩模式中执行指令。

[0029] 在非限制性示范性实施例中,经典模式中的指令可对准到存储器104中的32位边界,因此强制经典模式中每个指令的地址的两个最低有效位具有值“00”。此外,在此实施例中,压缩模式中的选定指令可被强迫未对准使得选定指令的地址的两个最低有效位具有非零值,即除“00”外的任何值(例如,选定指令的地址的两个最低有效位可被迫为值“01”、“10”或“11”)。因此,地址的这些最低有效位可指示对准且可在本文中称为“对准位”。

[0030] 压缩模式中被迫未对准的选定指令可对应于进入点。举例来说,进入点可标记包括指令的函数在压缩模式中的开始;进入点可为遵循来自压缩模式的调用的指令;或进入点可标记异常/中断处置程序的开始。不管进入点的精确性质如何,可触发从在经典模式中执行切换到在压缩模式中执行的选定指令可被迫未对准。例如遵循到压缩模式的函数调用的压缩模式指令等所有剩余指令的对准可保持未受干扰(即,除选定指令外的所有指令的地址的对准位可具有任何值,包含“00”)。利用此配置,示范性实施例可从在经典模式中执行切换到在压缩模式中执行,同时避免常规方法中在执行模式之间跟踪和切换的缺点。

[0031] 现参看图2,说明处理器102执行的示范性代码序列。代码序列210和214包括经典(对准)模式中的指令,且代码序列212包括压缩(未对准或不对准)模式中的指令。如图所示,代码序列210包括函数1,其在指令a1处开始且包含指令a1-a7。函数1和对应指令a1-a7以及调用(调用1)的地址的对准位已说明为“00”。一些实施例可包含任选用户状态寄存器(或“USR”,未明确展示),其中USR可经配置以提供对准位的状态的指示。因此,虽然在代码序列210的执行期间对准位为“00”,但USR可任选地更新以指示执行处于经典模式。当到函数2的调用(调用1)在指令a4之后遇到时,执行可遵循路径202到函数2。

[0032] 相对于函数2,此函数可在指令b1处开始且包含指令b1-b6,处于压缩模式。处理器102有必要辨识由于采取路径202引起的执行模式的切换,以便避免在错误执行模式中的错误执行。因此,在示范性实施例中,函数2可辨识为到压缩模式的进入点。辨识此切换可通过迫使函数2的地址“AB”的两个最低有效位(或对准位)为除“00”外的任何值来实现。举例来说,函数2的地址可在存储器104中被迫未对准使得“AB”为“01”、“10”或“11”。此被迫未对准可例如在处理系统100(未明确展示)的编译器、软件或专门逻辑中实现。在一个实例中,使函数2被迫未对准可通过配置汇编器以沿着路径202在函数2之前添加空操作来实现。空操作的大小可经调整以确保对准位“AB”具有除“00”以外的值。举例来说,如果对准位“AB”已具有除“00”以外的值,那么汇编器可不插入空操作。另一方面,如果“AB”的值等于“00”,那么汇编器可插入单一一个字节空操作,这将调整函数2的开始地址使得“AB”等于“01”。

[0033] 处理器102可经配置以调查指令执行期间遇到的每个进入点的对准位。因此,一旦处理器102辨识进入点函数2具有作为非零值的对准位“AB”,就可任选地更新USR以指示执行现处于压缩模式中。指令b1-b3的对准可保持不受干扰,且其对应地址的对准位可具有任何值,包含“00”。

[0034] 随后,一旦到函数3的调用(调用2)在指令b3之后遇到,执行就可遵循路径206到函数3。再次,处理器102可经配置以调查函数3的对准地址位,因为其是进入点。然而,此时,可辨识进入点函数3的对准位为“00”,因为函数3在对准经典模式中。因此,处理器102可辨识对于函数3执行回到经典模式,函数3在指令c1处开始且包含指令c1-c5。处理器102还可任选地更新USR以指示返回到经典模式。

[0035] 现来到沿着路径204从压缩模式中的代码序列212到经典模式中的代码序列210的返回(返回1),处理器102可调查进入点的对准位。在此情况下,进入点为指令a5,且因此可辨识指令a5的对准位为“00”。因此,可得出结论,执行已返回到经典模式。对应地,USR可经更新以指示执行在经典模式中。

[0036] 对于沿着路径208到指令b4的返回(返回2),进入点将为指令b4。为确保正确操作,处理器102必须能够辨识指令b4在压缩模式中。为实现此目的,指令b4可被迫未对准,使得指令b4的对准位不为“00”。在一个实例中,使指令b4被迫未对准可通过配置汇编器以沿着路径206在到函数3的调用(调用2)之前添加空操作来实现。空操作的大小可经调整以确保最低有效位“CD”具有除“00”以外的值。在另一实例中,调用函数(沿着路径206到函数3的调用2)的大小可增加使得指令b4的最低有效位“CD”被迫为值“01”、“10”或“11”。指令b5和b6的对准不受限制且不受干扰。因此,指令b5和b6的地址的对准位可为任何值,包含“00”。

[0037] 现参看图4A,说明用于针对调用/返回在经典模式(对准)与压缩模式(未对准)之间跟踪和切换的一般化方案。如图所示,到对准地址的调用/返回可保持在经典模式402中,如回路406展示。从经典模式402到未对准地址的调用/返回可沿着路径408转变到压缩模式404。到未对准地址的调用/返回可保持在压缩模式404中,如回路412展示。最后,从压缩模式404到对准地址的调用/返回可沿着路径410转变到经典模式402。

[0038] 虽然已相对于调用/返回描述以上实施例,但示范性技术可扩展到其它此类指令。举例来说,一些实施例可经扩展以监视例如跳跃(jump)、jumpr、callr、dealloc_return等指令的模式切换,其将参看图4B简要描述。这些指令可被监视以查看模式切换,且因此可通过遵循如上文针对调用/返回描述的类似技术确保执行在正确模式中。

[0039] 相对于跳跃指令,其可相对于示范性实施例以两种不同方式的一者利用。在一个实例中,跳跃指令当在函数内遇到时可用于跳跃到所述函数内的一指令。在另一实例中,跳跃指令可充当尾部调用(或“tailcall”),其如所属领域的技术人员将了解,可用于执行调用型行为。虽然到函数的调用可需要返回指令以返回到调用的位置(其可保存在调用堆栈上),但尾部调用可不需要记住调用的位置或将其保存在堆栈上。另一方面,尾部调用可直接返回到原始调用器。例如跳跃等指令可需要额外考虑,如下文将进一步阐释。

[0040] 如果在经典模式中的函数内遇到跳跃指令,且跳跃指令的行为是跳跃到函数内的一指令,那么一定不存在模式改变。然而,如果跳跃指令具有例如尾部调用等类型的行为,那么执行跳跃指令可致使控制流程移动到可在函数外部的进入点。所述进入点可位于经典模式中或压缩模式中。因此,通过调查进入点的最低有效位,可作出关于是否需要模式切换

的确定。因此,经典模式中遇到的跳跃指令的行为可类似于上文描述的调用指令的行为。

[0041] 然而,如果在处于压缩模式中的函数内遇到跳跃指令,那么可引发特殊考虑。如先前描述,在压缩模式中,除了被迫未对准的特定进入点外,剩余指令可或可不对准。换句话说,函数内剩余指令的地址的最低有效位可为任何值,包含“00”。因此,当跳跃指令致使跳跃到指令内的目标指令时,目标指令的地址可尚未被迫未对准,且因此可为任何值。因此,调查目标指令的地址的最低有效位可能不足以确定是否已发生模式切换,因为如果目标指令的地址的最低有效位为“00”,那么这不指示已发生到经典模式的切换。为处理此类情形,实施例可包含特殊指令或具有特殊行为的指令。

[0042] 一个此类特殊行为可包含在压缩模式中限制跳跃指令的行为仅跳跃到将不会引起模式切换的指令,例如跳跃到与跳跃指令相同的函数内的目标指令;以及针对可引起模式改变的情形界定单独尾部调用指令,例如跳跃到可在相同函数内部且可能在不同模式中的目标指令。因此,调查尾部调用函数的最低有效位可揭示是否已发生模式改变,且处理器可相应地处理此模式改变。

[0043] 因此,对于例如跳跃指令等指令,关于指令的执行是否将引起模式改变的确定可基于在其内遇到所述指令的模式。如果在经典模式内遇到跳跃指令,那么目标地址的最低有效位可用于确定是否已发生模式改变。另一方面,如果在压缩模式内遇到跳跃指令,那么将不发生模式改变或模式改变可被抑制,而不管目标地址的最低有效位所提供的指示如何。而是,特殊尾部调用指令可在压缩模式中使用以实现跳跃型行为,其可能需要模式改变,且此模式改变可基于尾部调用指令的最低有效位来确定。类似地,在一些实施例中,返回指令也可限于压缩模式,且不在经典模式中使用。

[0044] 其它指令可包含 `jumpr` 和 `tailcallr`。虽然跳跃和尾部调用可指定目标指令的地址,但 `jumpr` 和 `tailcallr` 指令可指定可固持目标指令的地址的寄存器。类似地, `callr` 指令可指定可固持待调用的函数的地址的寄存器。可用于解除分配寄存器的 `dealloc_return` 指令也可经配置使得与 `dealloc_return` 指令的执行相关的模式改变可基于调查 `dealloc_return` 指令的目标地址的最低有效位。

[0045] 现参看图4B,其说明模式切换行为以及对于包含以上特殊指令的示范性指令的跟踪。基于经调用函数的目标指令/进入点的地址的最低有效位,例如调用 (`call`)、`callr`、跳跃 (`jump`)、`jumpr` 和 `dealloc_return` 等指令可保持在经典模式452中而不引起模式切换(按照回路456)。或者,可发生模式切换,且这些指令 `call`、`callr`、`jump`、`jumpr` 和 `dealloc_return` 可沿着路径458转变到压缩模式454。从压缩模式454,将不针对模式切换而跟踪例如 `jump` 和 `jumpr` 等指令,如先前描述。而是,可跟踪尾部调用 (`tailcall`)、`tailcallr` 和返回 (`return`)。因此,基于经调用函数的目标指令/进入点的地址的最低有效位,例如 `call`、`callr`、`tailcall`、`tailcallr`、`return` 和 `dealloc_return` 等指令可保持在压缩模式454中(按照回路462)。或者,可发生模式切换,且这些指令 `call`、`callr`、`tailcall`、`tailcallr`、`return` 和 `dealloc_return` 可沿着路径460转变到经典模式452。

[0046] 在一些示范性实施例中,仅例如到压缩模式中的函数的进入点等选定指令可被迫未对准,且压缩模式中的剩余指令的地址可保持不受干扰。因此,与现有方法(例如,THUMB模式中的ARM处理器)相比,不需要使用LSB来保持模式的指示或跳跃到经调用函数+1的PC值。如上文论述,在示范性实施例中,USR可经配置以有效提供执行模式的指示。此外,仅沿

着路径202到函数2的调用1和沿着路径208到指令b4的返回2的类型的指令可需要插入空操作,而例如沿着路径204到指令a5的返回1和沿着路径206到函数3的调用2等指令将不需要此类空操作。因此,四种类型的调用/返回指令中的仅两者可需要通过引入空操作实现未对准,这意味着由于引入空操作引起的代码空间的浪费减少。

[0047] 此外,在示范性实施例中,指令的地址可本身便利地用于辨识函数调用和返回的执行模式,而不需要任何复杂的代码或昂贵的跟踪机制。对于所有其它指令,USR可经配置以有效地提供执行模式的指示。因为示范性实施例避免需要保留位来指示执行模式,所以实施例对应地避免与限制遵循到函数的进入点的指令的地址相关联的缺点。此外,使用地址的值确定执行模式需要调查仅调用或返回上的地址位。因此,不需要事先知道与目标指令相关联的模式是什么。另外,在示范性实施例中,MMU页表条目不需要保持关于指令的执行模式的信息。因此,处理系统100中的MMU(未图示)不需要当执行在压缩模式与经典模式之间切换或反之亦然时切换操作或执行特殊函数。

[0048] 将了解,实施例包含用于执行本文揭示的过程、功能和/或算法的各种方法。举例来说,如图3中说明,实施例可包含在处理器(例如,处理器102)中的执行模式之间切换的方法,其包括:当处理器正在第一执行模式(例如,经典模式)中操作时检测第一指令(例如,分别具有图2中例如函数2/指令a5等目标指令/进入点的调用/返回指令)-框302;分析第一指令的目标地址的一个或一个以上位(例如,分别图2中函数2/指令a5的两个最低有效位或对准位“AB”/“00”)-框304;以及基于所述一个或一个以上位确定是否将处理器的操作从第一执行模式切换到第二执行模式(例如,压缩模式)-框306。

[0049] 所属领域的技术人员将了解,可使用多种不同技术和技法中的任一者来表示信息和信号。举例来说,可由电压、电流、电磁波、磁场或磁粒子、光场或光粒子或其任何组合来表示贯穿以上描述中可参考的数据、指令、命令、信息、信号、位、符号和码片。

[0050] 此外,所属领域的技术人员将了解,结合本文所揭示的实施例而描述的各种说明性逻辑块、模块、电路和算法步骤可实施为电子硬件、计算机软件或两者的组合。为清楚地说明硬件与软件的此互换性,上文已大致关于其功能性而描述了各种说明性组件、块、模块、电路和步骤。所述功能性实施为硬件还是软件取决于特定应用和强加于整个系统的设计约束。熟练的技术人员可针对每一特定应用以不同方式来实施所描述的功能性,但此类实施决策不应被解释为会导致脱离本发明的范围。

[0051] 结合本文所揭示的实施例而描述的方法、序列和/或算法可直接以硬件、以由处理器执行的软件模块或以两者的组合体现。软件模块可驻留在RAM存储器、闪存存储器、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、可装卸盘、CD-ROM,或此项技术中已知的任一其它形式的存储媒体中。示范性存储媒体耦合到处理器,使得处理器可从存储媒体读取信息并将信息写入到存储媒体。在替代方案中,存储媒体可与处理器成一体式。

[0052] 参看图5,描绘包含根据示范性实施例配置的多核处理器的无线装置的特定说明性实施例的框图且其大体表示为500。装置500包含数字信号处理器(DSP)564,其可包含耦合到如图所示的存储器532的图1的处理器102。图5还展示耦合到DSP 564且耦合到显示器528的显示器控制器526。译码器/解码器(CODEC)534(例如,音频和/或语音CODEC)可耦合到DSP 564。还说明例如无线控制器540(其可包含调制解调器)等其它组件。扬声器536和麦克风538可耦合到CODEC 534。图5还指示无线控制器540可耦合到无线天线542。在特定实施例

中,DSP 564、显示器控制器526、存储器532、CODEC534和无线控制器540包含在系统级封装或芯片上系统装置522中。

[0053] 在特定实施例中,输入装置530和电源544耦合到芯片上系统装置522。此外,在特定实施例中,如图5中说明,显示器528、输入装置530、扬声器536、麦克风538、无线天线542和电源544在芯片上系统装置522外部。然而,显示器528、输入装置530、扬声器536、麦克风538、无线天线542和电源544的每一者可耦合到芯片上系统装置522的组件,例如接口或控制器。

[0054] 应注意,尽管图5描绘无线通信装置,但DSP 564和存储器532也可集成到机顶盒、音乐播放器、视频播放器、娱乐单元、导航装置、个人数字助理(PDA)、固定位置数据单元或计算机中。处理器(例如,DSP 564)也可集成到此装置中。

[0055] 因此,本发明的实施例可包含体现用于在处理器中的执行模式之间切换的方法的计算机可读媒体。因此,本发明不限于所说明的实例,且用于执行本文描述的功能性的任何装置包含在本发明的实施例中。

[0056] 虽然以上揭示内容展示本发明的说明性实施例,但应注意,可在不脱离如所附权利要求书界定的本发明的范围的情况下在其中作出各种改变和修改。无需以任何特定次序来执行根据本文中描述的本发明的实施例的方法权利要求项的功能、步骤和/或动作。此外,尽管可以单数形式描述或主张本发明的元件,但除非明确陈述限于单数形式,否则也预期复数形式。

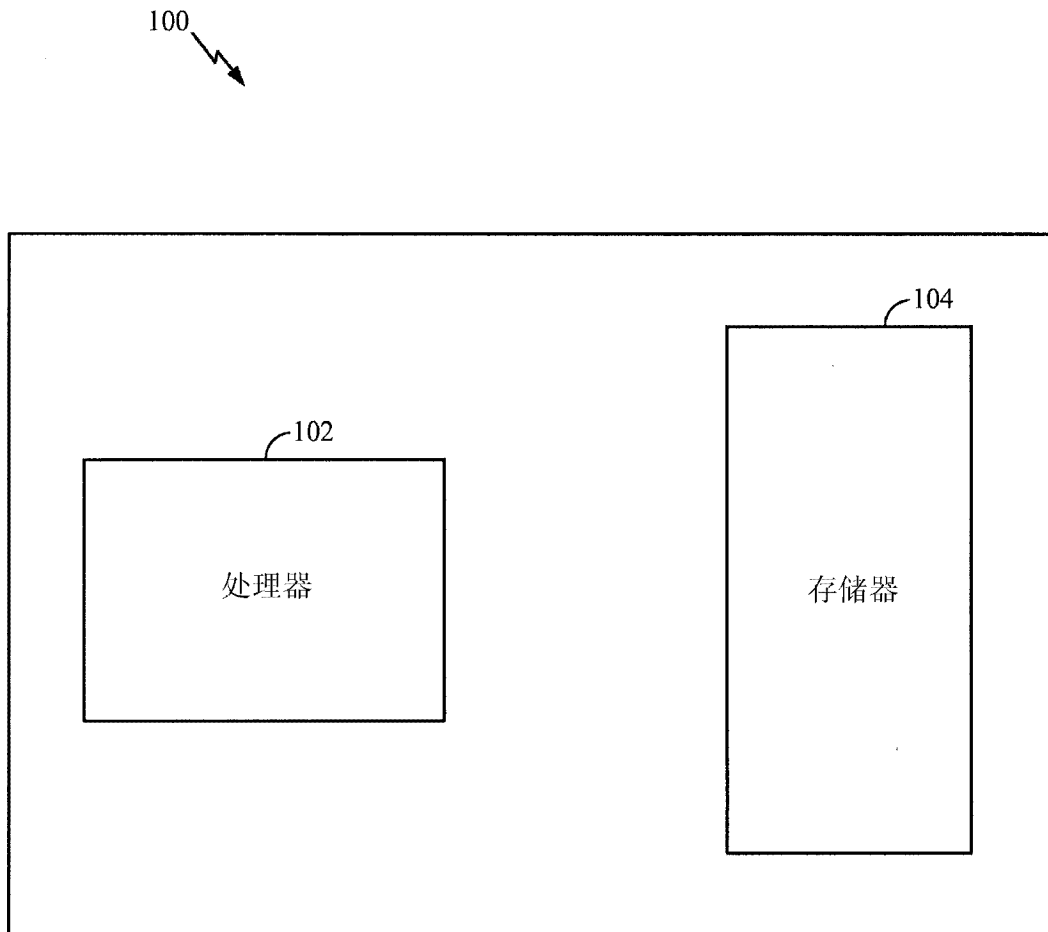


图1

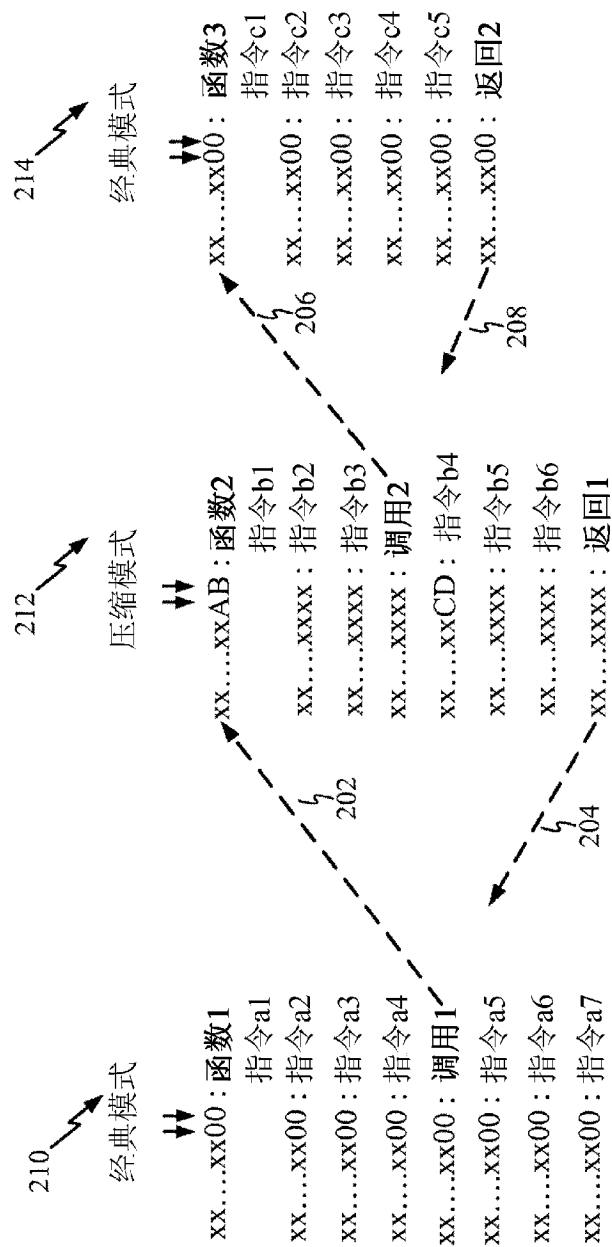


图2

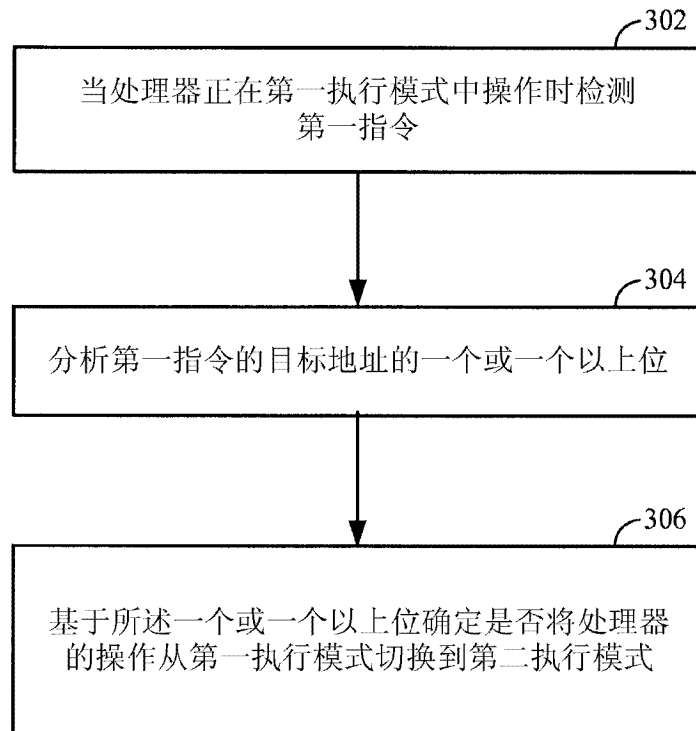


图3

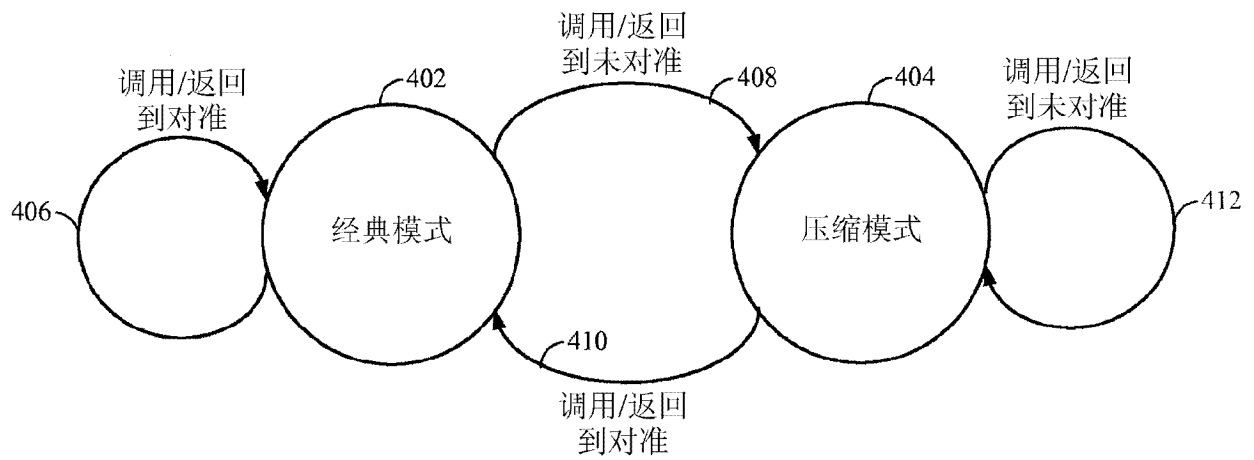


图4A

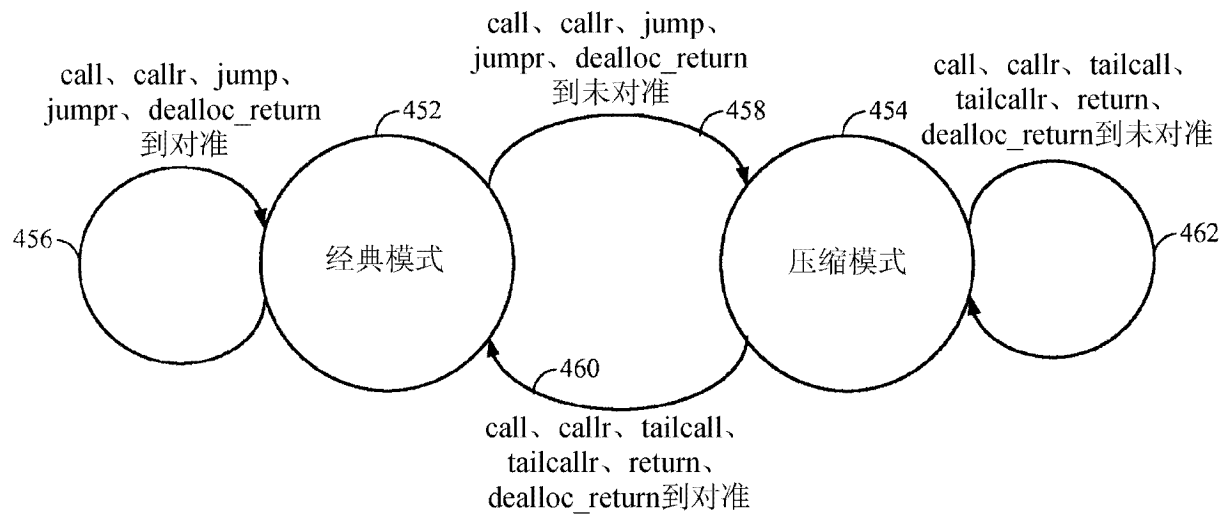


图4B

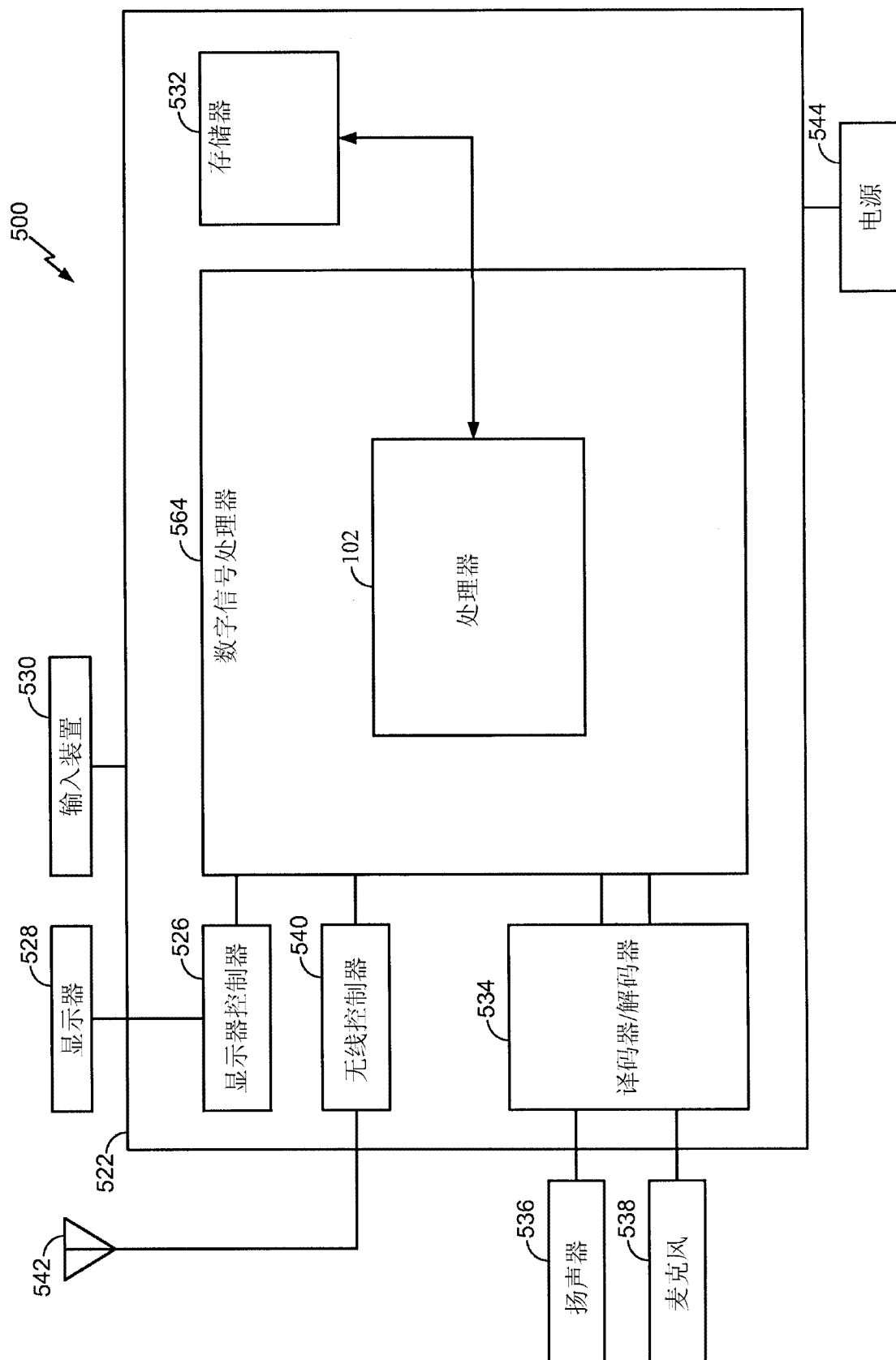


图5