(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101884025 B
(45)授权公告日 2014.04.09

(21)申请号 200880118806.3
(22)申请日 2008.10.31
(30)优先权数据
11/934,264 2007.11.02 US
(85)PCT国际申请进入国家阶段日
2010.06.01
(86)PCT国际申请的申请数据
PCT/US2008/081947 2008.10.31
(87)PCT国际申请的公布数据
WO2009/059100 EN 2009.05.07
(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 迈克尔·威廉·莫罗
詹姆斯·诺里斯·迪芬德尔费尔
(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 刘国伟
(51)Int.Cl.
G06F 9/32(2006.01)
G06F 9/38(2006.01)

(56)对比文件
US 5812813 A,1998.09.22, 说明书第5栏第1行至第9栏第17行、图2-5.
US 6374350 B1,2002.04.16, 第6栏第35-53行、第9栏51-64行、第10栏6-28行,图1B.
US 6898698 B1,2005.05.24, 全文.

审查员 金霞

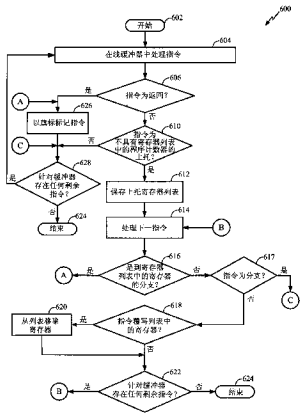
权利要求书2页 说明书10页 附图7页

(54)发明名称

用于使过程返回序列加速的方法和系统

(57)摘要

本发明揭示一种用于当从管线处理器中的过程返回时从链接栈检索返回地址的方法。所述方法识别可操作以从软件栈检索返回地址的检索指令。所述方法进一步识别可操作以分支到所述返回地址的分支指令。所述方法响应于所述指令和所述分支指令两者被识别而从所述链接栈检索所述返回地址,并使用所述返回地址来提取指令。



CN 101884025 B

1. 一种用于当从管线处理器中的过程返回时从链接栈检索返回地址的方法，其包括：

识别可操作以从软件栈检索返回地址的检索指令；

识别可操作以分支到所述返回地址的分支指令；

识别过程返回序列，所述过程返回序列包括所述检索指令和所述分支指令的组合；

响应于所述过程返回序列被识别而从所述链接栈检索所述返回地址；以及

使用所述返回地址来提取后续指令。

2. 根据权利要求 1 所述的方法，其中所述检索指令是上托指令。

3. 根据权利要求 1 所述的方法，其中所述检索指令是加载指令。

4. 根据权利要求 1 所述的方法，其中所述分支指令是 BX 指令。

5. 根据权利要求 1 所述的方法，其中所述分支指令是 MOV 指令。

6. 根据权利要求 1 所述的方法，其中所述识别所述检索指令进一步包括识别含有所述返回地址的寄存器。

7. 根据权利要求 1 所述的方法，其中识别所述检索指令进一步包括维持寄存器列表，其中所述寄存器列表具有多个寄存器，其中所述多个寄存器中的至少一个寄存器含有所述返回地址。

8. 根据权利要求 7 所述的方法，其中维持所述寄存器列表包括在所述多个寄存器中的任一者由后续指令覆写的情况下从所述寄存器列表移除寄存器。

9. 根据权利要求 1 所述的方法，其中由检测逻辑电路执行识别所述分支指令。

10. 根据权利要求 9 所述的方法，其中预解码逻辑电路中包含所述检测逻辑电路。

11. 根据权利要求 9 所述的方法，其中解码逻辑电路中包含所述检测逻辑电路。

12. 根据权利要求 1 所述的方法，其中识别所述分支指令进一步包括在指令高速缓冲存储器中以旗标标记所述分支指令。

13. 一种管线处理器，其包括：

线缓冲器，所述线缓冲器耦合到指令高速缓冲存储器，

提取逻辑电路，其耦合到所述指令高速缓冲存储器，所述提取逻辑电路具有存储预测性返回地址的链接栈，其中指令从所述线缓冲器加载到所述指令高速缓冲存储器中，所述提取逻辑电路从所述指令高速缓冲存储器检索指令，

预解码逻辑电路，其与所述线缓冲器通信，其中所述预解码逻辑电路进一步包括用于识别过程返回序列的检测逻辑电路，所述过程返回序列包括检索指令和分支指令的组合，所述检索指令可操作以从软件栈检索返回地址，所述分支指令分支到所述检索的返回地址，所述管线处理器响应于对所述过程返回序列的所述识别而从所述链接栈检索所述预测的返回地址。

14. 根据权利要求 13 所述的管线处理器，其中当所述分支指令从所述线缓冲器加载到所述指令高速缓冲存储器中时，所述检测逻辑电路以旗标标记所述过程返回序列的所述分支指令。

15. 根据权利要求 14 所述的管线处理器，其中所述提取逻辑电路根据所述以旗标标记的信息识别所述过程返回序列。

16. 根据权利要求 15 所述的管线处理器，其中所述提取逻辑电路内的返回选择器逻辑电路根据所述以旗标标记的信息识别所述返回序列。

17. 根据权利要求 13 所述的管线处理器，其中所述检索指令是上托指令。

18. 根据权利要求 13 所述的管线处理器，其中所述检索指令是加载指令。

19. 根据权利要求 13 所述的管线处理器，其中所述分支指令是 BX 指令。

20. 一种管线处理器，其包括：

提取逻辑电路，其具有存储所预测的返回地址的链接栈，所述提取逻辑电路从指令高速缓冲存储器提取指令，

解码逻辑电路，所述解码逻辑电路耦合到所述提取逻辑电路，其中所述提取的指令由所述解码逻辑电路解码，所述解码逻辑电路进一步包括检测逻辑电路，其中所述检测逻辑电路识别过程返回序列，所述过程返回序列包括检索指令和分支指令的组合，所述检索指令可操作以从软件栈检索地址，所述分支指令可操作以分支到所述检索的地址，所述管线处理器响应于对所述过程返回序列的所述识别而从所述链接栈检索所述预测的返回地址。

21. 根据权利要求 20 所述的管线处理器，其中所述提取逻辑电路使用所述检索的地址来提取指令。

22. 根据权利要求 20 所述的管线处理器，其中所述检索指令是上托指令。

23. 根据权利要求 20 所述的管线处理器，其中所述检索指令是加载指令。

24. 根据权利要求 20 所述的管线处理器，其中所述分支指令分支到所述检索指令所识别的地址。

25. 根据权利要求 20 所述的管线处理器，其中所述分支指令是 MOV 指令。

用于使过程返回序列加速的方法和系统

技术领域

[0001] 本发明大体涉及计算机系统，且更明确地说涉及一种用于通过识别处理器内的上托分支指令序列而使返回序列加速的方法和系统。

背景技术

[0002] 处理器执行的大多数程序包含子例程或过程。过程是通过过程调用序列存取的代码的模块。一旦过程完成，指令执行就通过过程返回序列的执行而返回到调用程序。

[0003] 在一些处理器结构内，过程调用和返回序列可编译为指令的序列。举例来说，过程调用序列可由推送 (PUSH) 指令之后是分支和链接指令组成。推送指令可将过程内的指令使用的参数保存到软件栈上。在推送指令之后，处理器可执行分支和链接指令。分支和链接指令促使在过程的开始地址处开始指令提取和执行，并将分支和链接指令之后的下一顺序指令的地址（称为返回或链接地址）保存在链接寄存器中。链接寄存器可为特殊用途寄存器或处理器使用的通用寄存器 (GPR) 之一。在过程内，通常将链接寄存器内容推送到软件栈上，使得其值在在返回到原始调用程序之前调用另一过程的情况下不被覆写。

[0004] 在过程完成其功能之后，处理器执行过程返回序列以在链接地址（过程调用指令之后的下一顺序指令地址）处继续进行指令执行。因为返回地址经常保存在软件栈上，所以过程返回序列必须首先从软件栈检索返回地址以使用所述地址来确定待提取的下一指令群组。

[0005] 过程返回序列可由一个或一个以上指令组成。在一些处理器结构中，过程返回序列可以是单一指令，例如上托 (POP) 或加载指令，其可从软件栈读取下一返回地址并更新程序计数器 (PC)。或者，处理器可使用上托或加载指令将链接地址从软件栈读取到中间寄存器（例如，GPR）中，之后将所述值移动到程序计数器以完成过程返回序列。在另一说明性实例中，处理器可确定从过程的返回可能是将保存在链接寄存器 (LR) 中的值移动到 PC 中的指令。当处理器在过程调用之后遇到这些过程返回序列中的任一者时，处理器使用从软件栈检索的返回地址值跳回到过程调用指令之后的下一顺序指令。

[0006] 可将额外的逻辑添加到处理器的硬件以改进指令处理的效率。举例来说，可将链接栈添加到处理器的提取逻辑以使指令提取加速。所属领域的技术人员了解，链接栈可含有还可存在于软件栈上的返回地址。然而，链接栈独立于软件栈而操作。与链接栈相关联的硬件逻辑识别过程调用和返回。当在执行之前识别过程调用指令时，将相关联的返回地址加载到链接栈上。相反，当识别过程返回时，从链接栈检索相关联的返回地址并使用其来继续进行指令提取。代替于等待指令执行和从软件栈检索返回地址，处理器可使用存储在链接栈中的地址来推测性地提取指令。

[0007] 随着处理器的发展，过程返回序列继续变化。在一些处理器结构中，过程返回可包括多个指令。如果支持链接栈的硬件逻辑不将这些指令辨认为过程返回序列，那么可能不会从链接栈检索返回地址，且因此链接栈可能会变得与指令序列不同步。当链接栈变得不同步时，链接栈可能会提供可导致多个地址误预测的错误的返回地址信息。

发明内容

[0008] 因此，业内需要有将某些指令序列（更明确地说，上托（或加载）和分支指令序列）辨认为过程返回序列的处理器电路。本发明认识到此需要并揭示一种具有在指令管线中较早识别对应于过程返回的指令的电路的处理器。在识别过程返回之后，处理器通过使用来自链接栈的下一返回地址而提取下一指令群组。通过将上托和分支指令序列辨认为程序返回，处理器可基于从链接栈检索的正确地址来继续提取指令。

[0009] 揭示一种用于当从管线处理器中的过程返回时从链接栈检索返回地址的方法。所述方法识别操作以从链接栈检索返回地址的检索指令。所述方法识别操作以分支到所述返回地址的分支指令。所述方法响应于所述指令和所述分支指令两者被识别而从所述链接栈检索所述返回地址。所述方法使用所述返回地址来提取后续指令。

[0010] 揭示一种管线处理器。所述管线处理器具有线缓冲器。线缓冲器耦合到指令高速缓冲存储器。处理器还具有耦合到指令高速缓冲存储器的提取逻辑电路。提取逻辑电路具有存储预测性返回地址的链接栈，其中指令从线缓冲器加载到指令高速缓冲存储器中。提取逻辑电路从指令高速缓冲存储器检索指令。管线处理器还具有与线缓冲器通信的预解码逻辑电路，其中预解码逻辑电路具有用于识别过程返回序列的检测逻辑电路。过程返回序列识别为操作以从软件栈检索返回地址的检索指令，和分支到所检索的返回地址的分支指令。管线处理器响应于对过程返回序列的识别而从链接栈检索所预测的返回地址。

[0011] 揭示一种管线处理器。所述管线处理器具有提取逻辑电路。提取逻辑电路具有存储所预测的返回地址的链接栈。提取逻辑电路从指令高速缓冲存储器提取指令。管线处理器还具有解码逻辑电路，所述解码逻辑电路耦合到提取逻辑电路，其中所提取的指令由解码逻辑电路解码。解码逻辑电路进一步具有检测逻辑电路，其中所述检测逻辑电路识别过程返回序列。过程返回序列是从软件栈检索地址的检索指令，和操作以分支到所检索的地址的分支指令。管线处理器响应于对过程返回序列的识别而从链接栈检索所预测的返回地址。管线处理器响应于对过程返回的识别而从链接栈检索所预测的返回地址。

[0012] 从以下详细描述和附图将了解对本发明的更完整理解以及本发明的更多特征和优点。

附图说明

[0013] 图 1 展示使用本发明的一实施例的处理器的高级逻辑硬件框图。

[0014] 图 2 展示由图 1 的处理器执行的示范性指令群组。

[0015] 图 3 显示根据本发明的一个实施例并入有检测逻辑电路的图 1 的 CPU 的上部和下部管线的更详细框图。

[0016] 图 4 展示图 3 的提取逻辑电路的更详细视图。

[0017] 图 5 展示利用检测逻辑电路的上部和下部管线的替代实施例。

[0018] 图 6 展示说明由图 1 的处理器执行的指令处理流的流程图，所述处理器辨认程序返回并使用链接栈来提取指令。

[0019] 图 7 展示说明由处理器使用图 4 的上部管线执行的替代指令处理流的流程图。

具体实施方式

[0020] 下文结合附图陈述的详细描述希望作为对本发明的各种实施例的描述，且不希望表示其中可实践本发明的仅有实施例。详细描述包含特定细节以用于提供对本发明的彻底理解。然而，所属领域的技术人员将了解，可在没有这些特定细节的情况下实践本发明。在一些例子中，以框图形式展示众所周知的结构和组件以免混淆本发明的概念。可仅为了方便和清楚而使用首字母缩写词和其它描述性术语，且其不希望限制本发明的范围。

[0021] 图 1 展示利用如下文描述的本发明的一个实施例的超标量处理器 100 的高级视图。处理器 100 具有中央处理单元 (CPU)102，其经由控制信号 104 耦合到指令高速缓冲存储器 106。指令高速缓冲存储器 106 还耦合到线缓冲器 107 且通过通用总线 110 耦合到存储器 108。CPU 102 控制经由线缓冲器 107 将指令从存储器 108 加载到指令高速缓冲存储器 106 中。CPU 102 具有耦合到下部管线 160 和 165 的上部管线 150。在下部管线 160 和 165 内是执行级 220 和 225。在执行级 220 内是执行单元 (EU)130A，且在执行级 225 内是 EU 130B。

[0022] 如所属领域的技术人员了解，指令高速缓冲存储器 106 可以是经设计以弥合存储器 108 与处理器 100 之间的速度差距的专用存储器。将从存储器 108 提取的指令放置在能够以处理器时钟速度读取的较快指令高速缓冲存储器 106 中。如果指令不存在于指令高速缓冲存储器 106 中，那么处理器 100 从存储器 108 检索指令。当从存储器 108 检索到指令时，将其首先加载到线缓冲器 107 中且最终写入到指令高速缓冲存储器 106 中。

[0023] 在指令高速缓冲存储器 106 中加载了指令之后，CPU 102 经由控制信号 104 存取所述指令。将指令从指令高速缓冲存储器 106 加载到上部管线 150 中。在上部管线 150 中处理指令且接着将所述指令发送到下部管线 160 或 165 以供进一步处理。如结合图 3-5 的论述所描述，处理器可具有经设计以检测特定指令序列的逻辑电路。这些特定指令序列可对应于过程返回。在已识别过程返回指令序列之后，处理器 100 可根据本发明的多个实施例基于那些指令执行功能。

[0024] 在上部管线 150 中对指令执行的一些示范性处理功能可包含提取指令、将指令对准、对指令进行解码、将指令发布到下部管线 160 或 165 等。在下部管线 160 和 165 内，可由执行单元 130A 和 130B 执行指令，并记录结果。

[0025] 图 2 中说明具有使用上托和分支指令序列的过程返回的说明性指令群组 200。显示指令 260、指令 270 的操作和执行指令的模块 280。出于清楚的目的，从此指令群组 200 中省略将在软件栈上推送参数以供过程本身使用的任何指令。也省略将组成过程执行的实际功能的任何指令。图 2 中描绘的指令是调用过程、将返回地址保存在链接寄存器（在此实例中为 GPR R_{14}）中、将返回地址存储到软件栈上、从软件栈检索返回地址，并继续处理位于返回地址处的指令的指令。图 2 中以指令群组 200 在追踪指令执行时将呈现的程序次序显示指令群组 200。所属领域的技术人员了解，所追踪的指令是处理器可能已提取的实际代码的子集，且以其将被执行的状态展示。指令群组 200 由三个嵌套过程组成。

[0026] 在指令群组 200 内是三个过程调用及其相关联的返回。第一过程调用是指令 A，其调用过程 PROC1。指令 B 是过程 PROC1 内的预备指令，其将当前返回地址保存到软件栈上。指令 C 是第二过程调用指令，其调用过程 PROC2。指令 D 是过程 PROC2 内的另一预备指令，其将与 PROC2 相关联的返回地址保存到软件栈上。最后过程调用指令是指令 E，其调用过程

PROC3。

[0027] 对应于过程调用指令的是过程返回指令。第一过程返回指令是指令F。在先前处理器结构中，将指令F辨认为过程返回指令。接下来两个指令（指令G与H组合）表示另一过程返回。通常，在先前处理器结构中，上托和分支指令的指令组合可能不会被正确地识别为供硬件链接栈使用的过程返回。因此在这些先前处理器中，当识别指令G和H时，可能未检索链接栈上的下一返回地址。使用一个实施例的处理器可缓解此可能的链接栈讹误。在一个实施例中，在将指令H识别为过程返回指令之后，处理器100可从链接栈检索下一地址并使用所检索的地址来继续提取指令。在此实例中，链接栈上的下一地址指回过程PROC1，且更明确地说其指向指令C之后的下一顺序指令（指令I）。指令H也可称为隐式分支指令。

[0028] 接下来两个指令（指令I和J）也解译为过程返回序列。当指令J由处理器100识别为过程返回指令时，检索链接栈上的下一地址并使用所述地址来继续指令提取。指令J是显式分支指令。在此实例中，链接栈外的下一地址将程序执行指回主程序。在先前处理器结构中，指令I与J的组合可能未被正确地识别为供硬件链接栈使用的过程返回序列。如图3-7的论述中更详细描述，本发明的各种实施例将上托和分支指令的组合识别为过程返回序列。

[0029] 图3显示利用本发明实施例的CPU 102的更详细框图。在CPU 102内，上部管线150具有提取级203，其含有通过控制信号104耦合到指令高速缓冲存储器106的提取逻辑电路202。同样在CPU 102中的是具有检测逻辑电路250的预解码逻辑电路201。预解码逻辑电路201耦合到线缓冲器107，所述线缓冲器107耦合到指令高速缓冲存储器106。提取级203耦合到解码级205，解码级205又耦合到发布级207。耦合到解码级205的是解码逻辑电路（为了便于说明未图示），其对关于指令的特定信息进行解码。在发布级207内的可为若干指令队列（为了便于说明未图示），其在指令发布到下部管线160和165之前保持指令。

[0030] 如所属领域的技术人员可了解，管线级可具有经设计以保持指令的寄存器或寄存器群组。当指令进入特定级时，处理器100将指令加载到链接到所述级的寄存器或寄存器群组中。当指令保持在每一级内的寄存器或寄存器群组中时，逻辑电路可依据指令而执行某些操作。在逻辑电路已执行既定操作之后，指令接着传递到下一顺序级。另外，当指令在上部管线150中时，其由各种逻辑电路“处理”。处理指令可包含提取指令、对指令进行解码、将指令对准、发布指令等。

[0031] 指令进入上部管线150并从提取级203移动穿过发布级207。指令在提取级203期间由提取逻辑电路202提取。在提取指令之后，其在解码级205期间由解码逻辑电路解码。在解码级205之后，在发布级207中处理指令。在指令离开发布级207之后，在下部管线160或下部管线165中执行指令。如先前论述，在下部管线160内的是执行级220和EU 130A。在下部管线165内的是执行级225和EU 130B。下部管线160和165分别存取寄存器堆230或235。

[0032] 预解码逻辑电路201可由处理器100使用以在指令保存在指令高速缓冲存储器106中之前对关于指令的信息进行部分解码和识别。经预解码的信息可在指令存储在指令高速缓冲存储器106中时连同指令一起保存。在预解码逻辑电路201内，检测逻辑电路250

可识别指令之间的相依性。举例来说，检测逻辑电路 250 可经设计以识别何时上托指令和分支指令利用相同寄存器。如图 4 的论述中所阐释，在检测逻辑电路 250 将由上托和分支指令组成的指令序列识别为来自过程调用的返回之后，提取逻辑电路 202 在从指令高速缓冲存储器 106 提取分支指令时解译此信息。

[0033] 当指令加载到指令高速缓冲存储器 106 中时，可通过设定与指令相关联的信息字段内的特定位置中的位来实现将经预解码的信息与指令相关联。将经预解码的信息保存在指令高速缓冲存储器 106 中也可称为以旗标标记指令。举例来说，在确定指令是过程返回指令之后，可在指令标头中的一个位置中设定一位，其识别所述指令是过程返回指令。或者，处理器 100 可针对经识别的指令将经预解码的信息编码到指令标头中。以此方式，处理器 100 可针对不同指令基于选定或预定标准使用多个位来编码不同信息。可在正从指令高速缓冲存储器 106 提取指令时检索经预解码的信息。处理器 100 可接着基于经识别的信息执行某些功能。

[0034] 图 4 显示根据本发明一个实施例的提取逻辑电路 202。提取逻辑电路 202 包含控制地址选择多路复用器 302 的地址选择器逻辑电路 320。地址选择器逻辑电路 320 包含返回选择器逻辑电路 350。耦合到地址选择多路复用器 302 的输入的是来自链接栈 304 的链接栈输出 316。链接栈逻辑电路 310 与地址选择器逻辑电路 320 通信并控制链接栈 304 的输入和输出两者。当识别过程调用时，链接栈 304 从地址总线接收返回地址。

[0035] 在链接栈 304 内，可保存预测性返回地址。链接栈 304 可以是存储器的存储对应于与过程返回相关联的返回地址的指令地址的后进先出 (LIFO) 部分。链接栈 304 独立于软件栈而操作。当指令在指令管线中较早地被识别为过程返回指令时，处理器 100 可使用存储在链接栈上的返回地址抢先提取指令，而不是等待过程返回在下部管线 160 或 165 中执行。

[0036] 如图 4 中显示，地址选择多路复用器 302 可接收下一顺序程序地址。下一顺序程序地址可以是当前程序计数器递增 8 个地址位置 (PC+8)。在此实施例中，从指令高速缓冲存储器 106 提取指令，一次提取两个指令，其中每一指令为四个字节长。在其它处理器实施例中，下一顺序程序地址可以是递增不同量的程序计数器。如先前所提及，地址选择多路复用器 302 还可从链接栈 304 接收预测性地址信息。当处理器 100 确定已发生过程返回时，检索链接栈 304 中的下一地址并将其用作开始位置以提取下一指令群组。

[0037] 地址选择多路复用器 302 可从其它来源接收地址信息。举例来说，分支目标地址高速缓冲存储器 (BTAC) 可提供用于提取指令的地址。或者，中断地址可用于提取指令。为了便于说明，未展示地址的这些其它来源。

[0038] 地址选择器逻辑电路 320 确定其输入中的哪一者将通过地址选择多路复用器 302 并用于提取下一指令群组。如果地址选择器逻辑电路 320 确定待提取的下一地址群组是下一顺序地址 (PC+8)，那么选择 PC+8 输入，或者，如果地址选择器逻辑电路 320 内的返回选择器逻辑电路 350 确定链接栈 304 含有下一提取地址，那么选择链接栈输出 316。

[0039] 为了利用链接栈 304，处理器 100 需要确定何时在上部管线 150 内的指令处理序列期间识别过程调用和对应的返回。由于链接栈 304 用于预测性地提取指令，所以处理器 100 不会等到指令执行再提取后续指令。替代地，在处理器 100 已识别为上部管线 150 中的过程调用指令之后，处理器 100 经由地址总线将与过程调用相关联的返回地址加载到链接栈

304 上。接着处理器 100 提取过程的指令。

[0040] 在过程结束时，处理器 100 遇到过程返回序列。由于过程返回序列的缘故，处理器将“上托”链接栈 304 以检索对应的返回地址并分支到所述返回地址以继续进行指令提取。处理器 100 识别过程返回指令并检索链接栈外的下一返回地址。过程返回指令可以是上托指令或加载指令，其读取软件栈并写入 PC。如果返回选择器逻辑电路 350 识别特定上托指令是过程返回，那么返回选择器逻辑电路 350 接着促使地址选择器逻辑电路 320 促使引导链接栈输出 316 穿过地址选择多路复用器 302。接着用取自链接栈 304 的返回地址来提取下一指令集。

[0041] 如先前所描述，过程返回序列可由一个或一个以上指令组成。举例来说，在一些 ARM 实施方案中，到存储在链接寄存器 (R_{14}) 中的值的分支指令可解译为过程返回。或者，将链接寄存器 (R_{14}) 的值移动到程序计数器 (R_{15}) 中的移动指令也可解译为过程返回。重要的是，处理器 100 准确地识别过程返回。如果处理器 100 未准确地识别过程返回，那么链接栈 304 将变得相对于过程返回指令不同步。如果链接栈 304 变得不同步，那么处理器 100 可能必须进入分支校正序列，且可影响执行性能。

[0042] 由于处理器指令集已发展，可将替代指令序列识别为过程返回序列。在一个示范性实施例中，将返回地址上托到特定寄存器的上托或加载指令 (其不更新 PC) 之后是到特定寄存器中存储的值的分支指令可解译为过程返回序列。分支指令可能是或可能不是上托指令之后的下一顺序指令。

[0043] 为了促进识别由上托和分支指令组成的过程返回序列，搜集关于所述两个指令的信息。过程返回的上托指令可涉及一个或一个以上寄存器。当识别上托指令时，可保存上托指令的寄存器列表并将其与任何后续指令的寄存器目标进行比较。寄存器列表的保存和比较也可称为认定已识别上托指令。如果不分支 (non-branching) 指令在遇到到与上托指令相关联的寄存器列表中识别的寄存器的分支之前利用所述寄存器，那么将所述寄存器从所保存的寄存器列表中扣除。如果在确实使用所保存的寄存器列表中的寄存器的分支指令之前遇到不使用所保存的寄存器列表中的寄存器的分支指令，那么对先前上托的上托 - 分支返回序列的搜索终止。当遇到使用寄存器列表中的寄存器的分支指令时，处理器 100 可接着确定过程返回正被处理。因此，链接栈 304 的顶部处的地址可接着被检索并用于提取下一指令群组。

[0044] 如先前所描述，预解码逻辑电路 201(图 3) 可能已识别利用相同寄存器的上托和分支指令序列，且因此分支指令被识别为过程返回指令。处理器 100 可能已在分支指令被存储到指令高速缓冲存储器 106 中时将此信息保存到指令标头中。当提取逻辑电路 202 检索随分支指令保存的经预解码信息时，处理器 100 使用返回选择器逻辑电路 350 来识别分支指令是过程返回。在返回选择器逻辑电路 350 已确定分支指令是过程返回之后，返回选择器逻辑电路 350 促使地址选择逻辑电路 320 引导链接栈输出 316 穿过地址选择多路复用器 302。返回选择器逻辑电路 350 还与链接栈逻辑电路 310 通信，从而促使返回链接栈中的下一值。因此，用链接栈地址来提取下一指令集。

[0045] 图 5 显示根据替代实施例的具有上部管线 151 的 CPU 102，所述 CPU 102 具有能够检测由上托 / 分支指令序列组成的过程返回的解码逻辑电路。更明确地说，CPU 102 含有具有检测逻辑电路 450 的解码逻辑电路 406。当解码逻辑电路 406 对指令进行解码时，识别

关于指令的信息。检测逻辑电路 450 可监视经解码指令以确定何时识别过程返回。如先前所论述，过程返回序列可由一个或一个以上指令组成。当上托指令和后续分支指令被解码时，检测逻辑电路 450 可确定发生过程返回序列。

[0046] 当检测逻辑电路 450 确定已经识别过程返回时，检测逻辑电路 450 将此信息传送到返回选择器逻辑电路 350，返回选择器逻辑电路 350 又将此信息传送到链接栈逻辑电路 310（图 4）。返回选择器逻辑电路 350 接着促使地址选择器逻辑电路 320 引导链接栈输出 316 穿过地址选择多路复用器 302。接着使用取自链接栈 304 的返回地址来提取下一指令集。

[0047] 可通过返回参考图 2 中的指令群组 200 来进一步阐释与实施例相关联的发明概念。指令 A 是过程 PROC1 的调用。当指令 A 分支到 PROC1 时，处理器 100 将下一顺序地址存储到链接寄存器（R_{14}）中。下一顺序地址是与返回到主程序相关联的返回地址。当指令 A 被识别为过程调用时，链接栈逻辑电路 310 促使将与指令 A 相关联的返回地址加载到链接栈 304 上。如图 2 中显示，指令 A 是主程序的一部分。指令 A 分支到 PROC1 且下一经处理指令为指令 B。

[0048] 指令 B 是 PROC1 内的第一指令且是过程 PROC2 的调用的预备指令。指令 B 通过将 R_{14} 的值推送到软件栈上来保存当前返回地址。接下来，处理指令 C。指令 C 是过程 PROC2 的调用。当指令 C 被识别为过程调用时，链接栈逻辑电路 310 将与指令 C 相关联的返回地址保存到链接栈 304 上。指令 C 分支到过程 PROC2 且经处理的下一指令为指令 D。

[0049] 指令 D 是过程 PROC2 内的第一指令，且通过将 R_{14} 的值推送到软件栈上来保存当前返回地址。指令 D 是另一预备指令，其为下一过程调用指令（指令 E）作准备。当指令 E 被识别为过程调用时，链接栈逻辑电路 310 促使将与指令 E 相关联的返回地址加载到链接栈 304 上。指令 E 是过程 PROC2 内的第二指令且调用过程 PROC3。指令 E 分支到与指令 F 相关联的地址，指令 F 是过程 PROC3 内的第一指令。指令 F 是过程 PROC3 内仅有的指令且是返回。明确地说，指令 F 分支到当前在链接寄存器（R_{14}）中的值。通常，在现有处理器结构中，将指令 F 辨认为指令返回。当处理指令 F 时，检测逻辑电路 450 确定指令 F 是过程返回且促使检索链接栈 304 上的下一返回地址。处理器使用返回地址而返回到过程 PROC2。

[0050] 在过程 PROC2 内，待处理的下一指令是指令 G，其将当前值从软件栈“上托”出来并将其保存到寄存器 R_{12} 中。为了便于说明，指令 G “上托”单一寄存器。然而，在替代实施例中，上托指令可返回多个寄存器的多个值。在此替代实施例中，处理器 100 可保持“经上托”寄存器的列表以便使用寄存器列表中的那些寄存器中的一者作为分支目标地址而将寄存器列表与后续分支指令进行比较。在一个实施例中，检测逻辑电路 450 可存储“经上托”寄存器的列表。

[0051] 指令 H 分支到现在在 R_{12} 中的所检索的地址。尽管指令 H 不是显式分支指令（BX），但其是等效分支指令。如所属领域的技术人员了解，MOV、PC、RN 也可解译为隐式分支指令。如图 6 和 7 的指令流程图 600 和 700 中所阐释，检测逻辑电路 250、450 确定上托指令（指令 G）连同到“经上托”寄存器（指令 H 的 R_{12}）的分支指令一起组成过程返回序列。因此，处理器 100 使用链接栈 304 来提供下一提取地址，且指令提取返回到过程 PROC1。

[0052] 在处理指令 H 之后，指令提取返回到过程 PROC1 并识别指令 I。指令 I 将下一值从软件栈上托出来到 R_2 中。仍在过程 PROC1 内，指令 J 分支到存储在 R_2 中的地址。类似于

指令 H，指令 J 分支到存储在先前“经上托”寄存器中的地址。因此，检测逻辑电路 250、450 确定指令 J 是过程返回指令，且使用来自链接栈 304 的下一值来提取下一指令群组。在此实例中，在处理指令 J 之后，提取指令 K。指令 K 可以是如图 3 中显示的主程序内的任何指令。

[0053] 在一个实施例中，处理器 100 使用检测逻辑电路 250 来识别指令 F 和指令 G 与 H 以及 I 与 J 的序列将解译为过程返回。因此，当检测逻辑电路 250 在线缓冲器 107 中遇到指令集 200 时，将指令 F、H 和 J 预解码为过程返回指令，其中经预解码信息保存在指令高速缓冲存储器 106 中。因此，当提取逻辑电路 202 从指令高速缓冲存储器 106 提取指令 F、H 和 J 时，返回选择逻辑电路 350 促使从链接 304 检索返回地址，所述返回地址用于提取下一指令群组。

[0054] 在替代实施例中，检测逻辑电路 450 也可经设计以识别指令 F 和指令 G 与 H 以及 I 与 J 的序列将解译为过程返回。在此情况下，当在解码级 205 中对指令群组 200 进行解码时，检测逻辑电路 450 识别指令 F、H 和 J 为过程返回指令并将此传送到返回选择器逻辑电路 350。返回选择器逻辑电路 350 接着促使使用链接栈 304 内的下一返回地址来确定下一提取地址。

[0055] 图 6 显示说明由处理器 100 执行的步骤的指令流 600，所述处理器 100 具有图 3 的 CPU 102 内的检测逻辑电路 250。为了便于说明，流程图 600 假定 CPU 102 内的线缓冲器 107 的宽度仅为单一指令，且指令从高速缓冲存储器线地址的开始依次返回。所属领域的技术人员了解，一些处理器可具有能够不以顺序次序处理多个指令的线缓冲器。本文描述的发明概念可应用于任一类型的处理器。

[0056] 指令流 600 在开始框 602 处开始。指令流从框 602 进行到框 604，其中线缓冲器 107 中的第一指令由检测逻辑电路 250 处理。指令流 600 接着进行到决策框 606。在决策框 606 中，检测逻辑电路 250 确定指令是否为已知的过程返回。如先前论述，已知的过程返回可以是先前识别的过程返回中的任一者，上托 / 分支序列除外。如果在决策框 606 处检测逻辑电路 250 确定指令是先前已知的过程返回，那么指令流 600 进行到框 626，其中将指令识别或以旗标标记为过程返回。如果在决策框 606 处检测逻辑电路 250 确定指令不是先前已知的过程返回，那么指令流进行到决策框 610。

[0057] 在决策框 610 处，检测逻辑电路 250 确定指令是否为不具有经上托寄存器列表中的程序计数器 (PC) 的上托指令。如果指令不是没有寄存器列表中的 PC 的上托指令，那么指令流 600 进行到决策框 628。否则，如果指令是不含有寄存器列表中的 PC 的上托指令，那么指令流 600 进行到框 612。在框 612 处，检测逻辑电路 250 保存上托指令的寄存器列表以供用于分析线缓冲器 107 中的任何后续指令。

[0058] 指令流从框 612 进行到框 614。在框 614 处，检测逻辑电路 250 从线缓冲器 107 检索下一指令。处理流从框 614 继续到决策框 616。在决策框 616 处，检测逻辑电路 250 确定线缓冲器 107 中的下一指令是否为到寄存器列表中所保存的寄存器中的任一者的分支指令。如果指令是到寄存器列表中的寄存器的分支，那么指令流进行到框 626，其中将指令以旗标标记为过程返回指令。如果在决策框 616 处检测逻辑电路 250 确定指令不是到所保存的寄存器列表中的寄存器的分支指令，那么指令流 600 继续到决策框 617。

[0059] 在决策框 617 处，检测逻辑电路 250 确定指令是否为分支指令。如果指令为分支

指令，那么指令流进行到决策框 628。如果在决策框 617 处检测逻辑电路 250 确定指令不是分支指令，那么指令流进行到决策框 618。在决策框 618 处，检测逻辑电路 250 确定指令是否覆写所保存的寄存器列表中的寄存器中的任一者。如果指令覆写所保存的寄存器列表中的寄存器中的任一者，那么指令流 600 继续到框 620，其中从所保存的寄存器列表移除被覆写的寄存器。指令流 600 从框 620 继续到决策框 622。

[0060] 如果在决策框 618 处检测逻辑电路 250 确定指令未覆写所保存的寄存器列表中的任何寄存器，那么指令流 600 进行到决策框 622。在决策框 622 处，检测逻辑电路 250 确定针对线缓冲器 107 是否存在任何剩余指令。如果针对线缓冲器不存在剩余指令，那么指令流 600 在框 624 处结束。如果线缓冲器 107 中存在剩余指令，那么指令流 600 回到框 614，其中处理线缓冲器 107 中的下一指令。

[0061] 在框 626 处，检测逻辑电路将指令以旗标标记为返回指令。如先前所提及，以旗标标记返回指令允许提取逻辑电路 202 在从指令高速缓冲存储器 106 提取返回指令时识别所述返回指令。指令流 600 从框 626 进行到决策框 628。在决策框 628 处，检测逻辑电路 250 确定是否存在待在线缓冲器 107 中处理的任何剩余指令。如果不存在待在线缓冲器 107 中处理的剩余指令，那么指令流 600 在框 624 处结束。如果存在待处理的剩余额外指令，那么指令流 600 进行到框 604，其中检测逻辑电路 250 处理下一指令。

[0062] 图 7 显示说明由 CPU 102 执行的步骤的指令流 700，所述 CPU 102 内具有耦合到图 4 的上部管线 151 的解码逻辑电路 406 中的检测逻辑电路 450。为了便于说明，对指令流 700 中概述的指令的处理假定解码逻辑电路 406 每处理器循环处理单一指令。所属领域的技术人员了解，一些处理器可具有能够每处理器循环处理多个指令的解码逻辑电路。本文描述的发明概念可应用于任一类型的处理器。

[0063] 指令流 700 在开始框 702 处开始。指令流从框 702 进行到框 704，其中解码逻辑电路 406 在解码级 205 中处理指令。指令流从框 704 继续到决策框 706。在决策框 706 处，检测逻辑电路 450 确定指令是否为过程返回。在此实例中，如果指令是除上托 / 分支序列外的先前已知的过程返回中的任一者，那么检测逻辑电路 450 确定指令为过程返回。如果检测逻辑电路 450 确定指令为过程返回，那么指令流 700 继续到框 708。如果检测逻辑电路 450 确定指令不是过程返回，那么指令流继续到决策框 710。

[0064] 在决策框 710 处，检测逻辑电路 450 确定指令是否为不具有寄存器列表中的程序计数器 (PC) 的上托指令。如果指令不是没有其寄存器列表中的 PC 的上托指令，那么处理流返回到框 704。如果在决策框 710 处检测逻辑电路 450 确定经解码指令是不含有其寄存器列表中的 PC 的上托指令，那么指令流 700 继续到框 712。由于处理器 100 可能能够从软件栈上托多个寄存器，所以在框 712 处检测逻辑电路 450 保存经上托寄存器列表。指令流 700 从框 712 进行到框 714。

[0065] 在框 714 处，处理器 100 将下一指令加载到解码级 205 中，且解码逻辑电路 406 处理所述指令。在在框 714 处加载指令之后，指令流 700 进行到决策框 716。在决策框 716 处，检测逻辑电路 450 确定指令是否为到所保存寄存器列表中的寄存器的分支。如果检测逻辑电路 450 确定指令是到所保存寄存器列表中的寄存器的分支，那么处理流继续到框 708。如果检测逻辑电路 450 确定指令不是到所保存的寄存器列表中的寄存器的分支指令，那么指令流 700 继续到决策框 718。

[0066] 在决策框 718 处，检测逻辑电路 450 确定指令是否为分支指令。如果指令为分支指令，那么指令流返回到框 704，其中将下一指令加载到解码级 205 中。如果在决策框 718 处指令不是分支指令，那么指令流 700 进行到决策框 720。在决策框 720 处，检测逻辑电路 450 确定指令是否覆写所保存的寄存器列表中的寄存器。

[0067] 如果指令未覆写所保存的寄存器列表中的寄存器，那么指令流 700 返回到框 714，其中将下一指令加载到解码级 205 且由解码逻辑电路 406 处理所述指令。如果在决策框 720 处指令覆写所保存的寄存器列表中的寄存器，那么指令流 700 继续到框 722，其中从所保存的寄存器列表移除经覆写的寄存器。指令流 700 从框 722 返回到框 714，其中将下一指令加载到解码级 205 中且由解码逻辑电路 406 处理所述指令。

[0068] 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、元件和 / 或组件可用通用处理器、数字信号处理器 (DSP)、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA) 或其它可编程逻辑组件、离散门或晶体管逻辑、离散硬件组件或其经设计以执行本文所描述的功能的任何组合来实施或执行。通用处理器可以是微处理器，但在替代实施例中，处理器可以是任何常规处理器、控制器、微控制器或状态机。处理器也可实施为计算组件的组合，例如 DSP 与微处理器的组合、多个微处理器、结合 DSP 核心的一个或一个以上微处理器，或任何其它此配置。

[0069] 尽管本文已说明和描述特定实施例，但所属领域的一般技术人员了解，预计实现相同目的的任何布置可替代所展示的特定实施例，且本发明在其它环境中具有其它应用。本申请案希望涵盖本发明的任何修改或变化。所附权利要求书决不希望将本发明的范围限于本文描述的特定实施例。

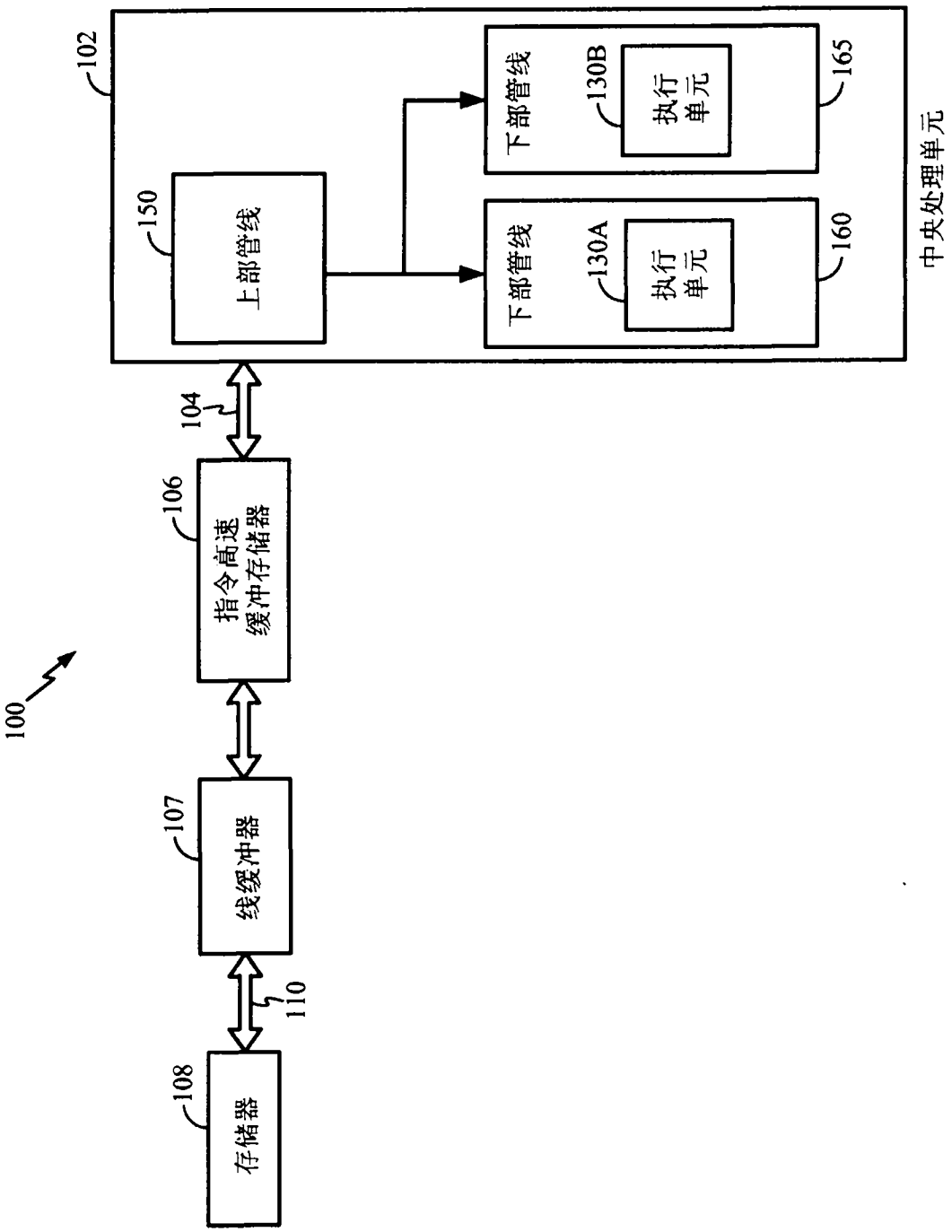

100
102
中央处理单元
150
上部管线
165
下部管线
130B
执行单元
160
下部管线
130A
执行单元
104
106
指令高速缓冲存储器
107
线缓冲器
110
108
存储器

图 1

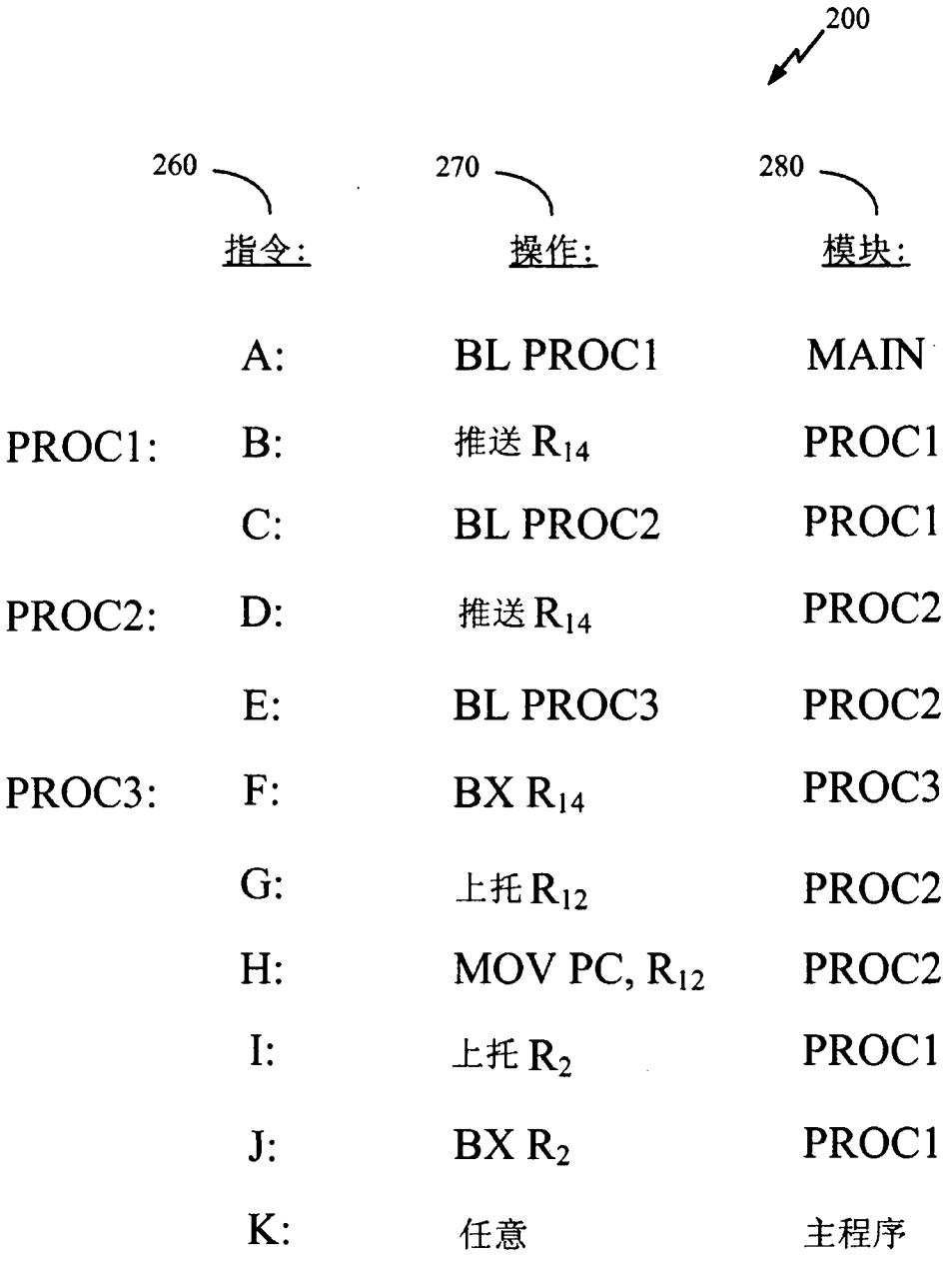

200
260
270
280
指令:
操作:
模块:
A: BL PROC1 MAIN
PROC1: B: 推送 R_{14} PROC1
C: BL PROC2 PROC1
PROC2: D: 推送 R_{14} PROC2
E: BL PROC3 PROC2
PROC3: F: BX R_{14} PROC3
G: 上托 R_{12} PROC2
H: MOV PC, R_{12} PROC2
I: 上托 R_2 PROC1
J: BX R_2 PROC1
K: 任意 主程序

图 2

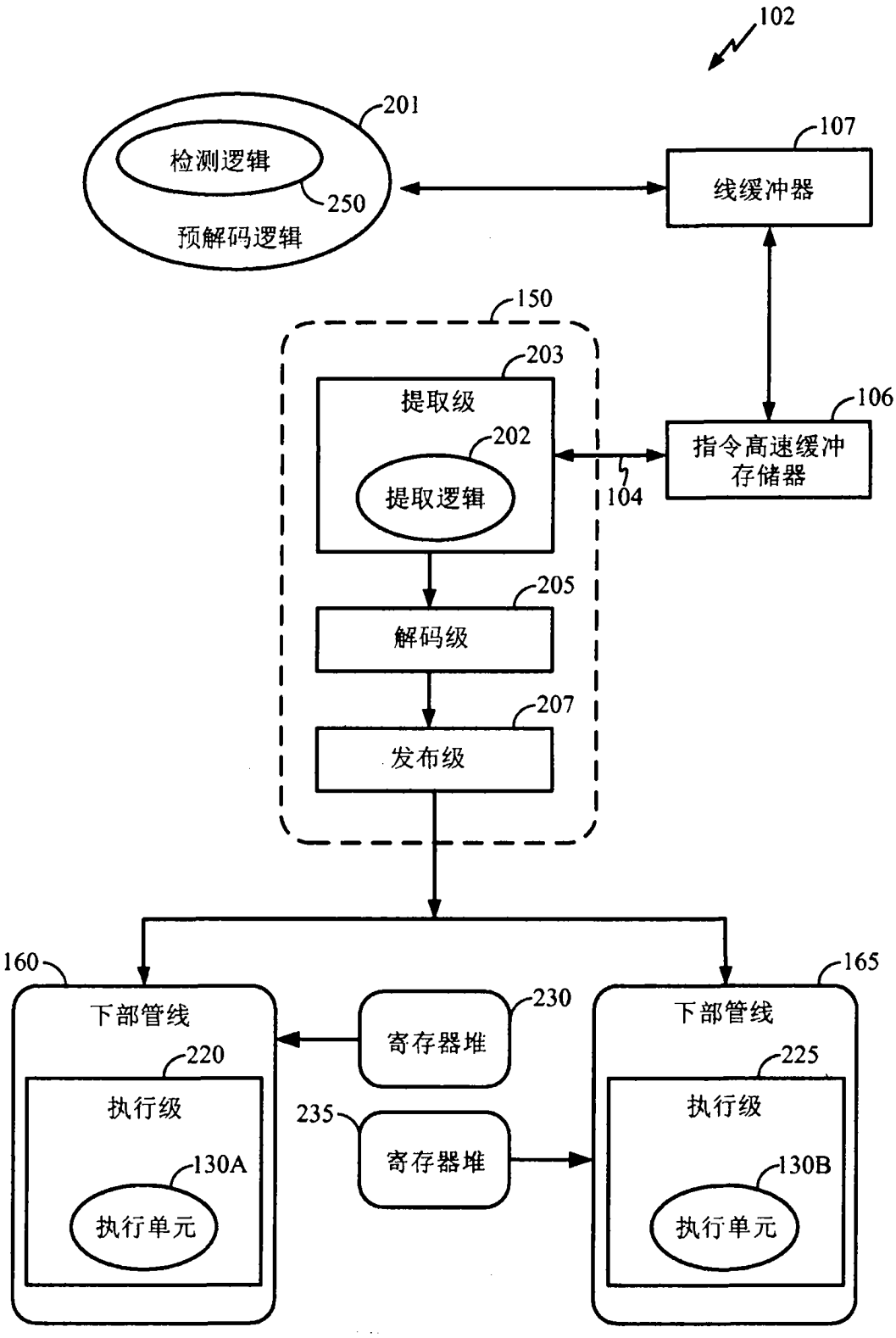

102
201
检测逻辑
250
预解码逻辑
107
线缓冲器
150
203
提取级
202
提取逻辑
106
指令高速缓冲存储器
104
205
解码级
207
发布级
160
下部管线
220
执行级
130A
执行单元
230
寄存器堆
235
寄存器堆
165
下部管线
225
执行级
130B
执行单元

图 3

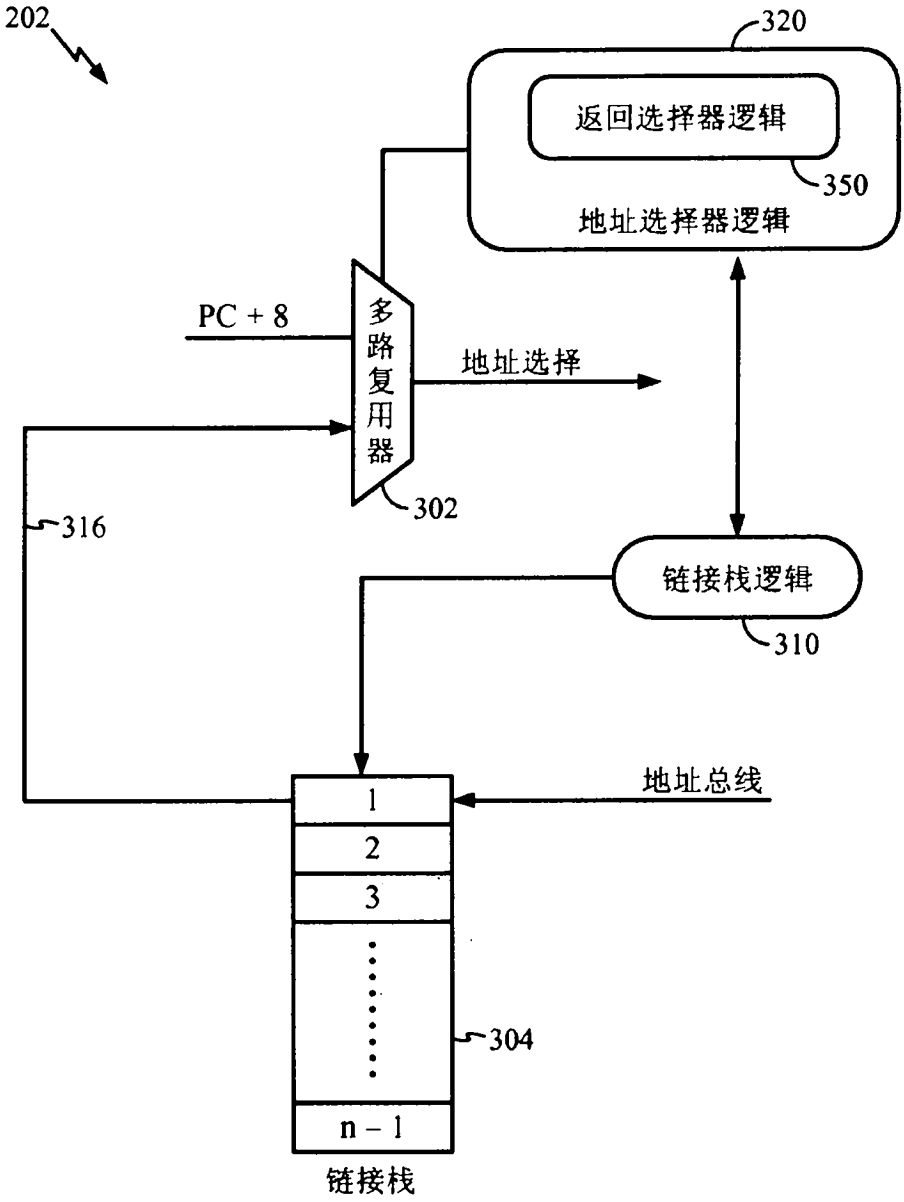

202
320
返回选择器逻辑
350
地址选择器逻辑
PC + 8
多路复用器
地址选择
316
302
链接栈逻辑
310
地址总线
1
2
3
304
n – 1
链接栈

图 4

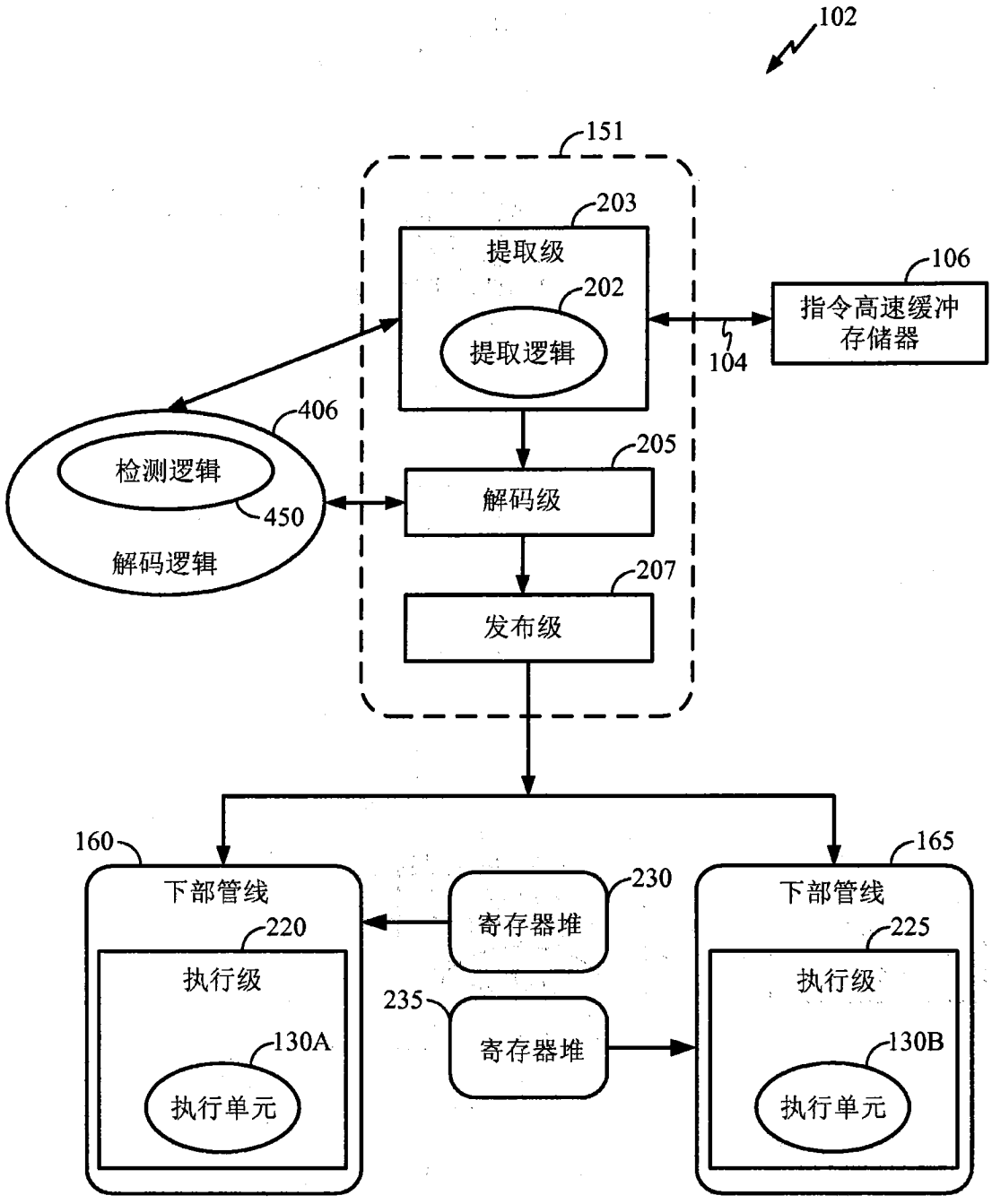

102
151
203
提取级
202
提取逻辑
106
指令高速缓冲存储器
104
406
检测逻辑
450
解码逻辑
205
解码级
207
发布级
160
下部管线
220
执行级
130A
执行单元
230
寄存器堆
235
寄存器堆
165
下部管线
225
执行级
130B
执行单元

图 5

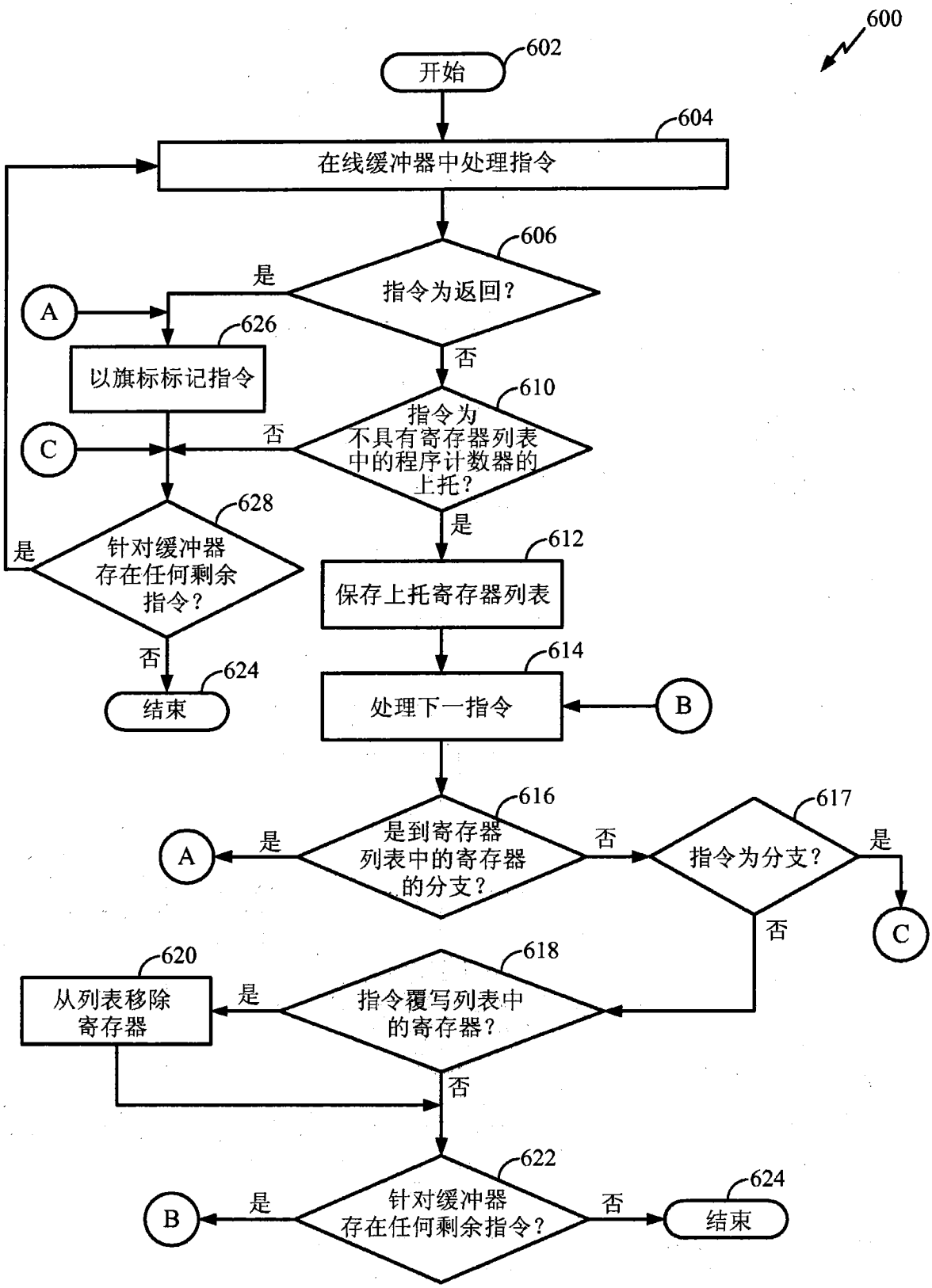

600
开始
602
在线缓冲器中处理指令
604
指令为返回？
606
是
否
A
以旗标标记指令
626
C
否
指令为不具有寄存器列表中的程序计数器的上托？
610
是
针对缓冲器存在任何剩余指令？
628
是
否
结束
624
保存上托寄存器列表
612
处理下一指令
614
B
是到寄存器列表中的寄存器的分支？
616
是
否
A
指令为分支？
617
是
C
否
从列表移除寄存器
620
是
指令覆写列表中的寄存器？
618
否
针对缓冲器存在任何剩余指令？
622
是
否
B
结束
624

图 6

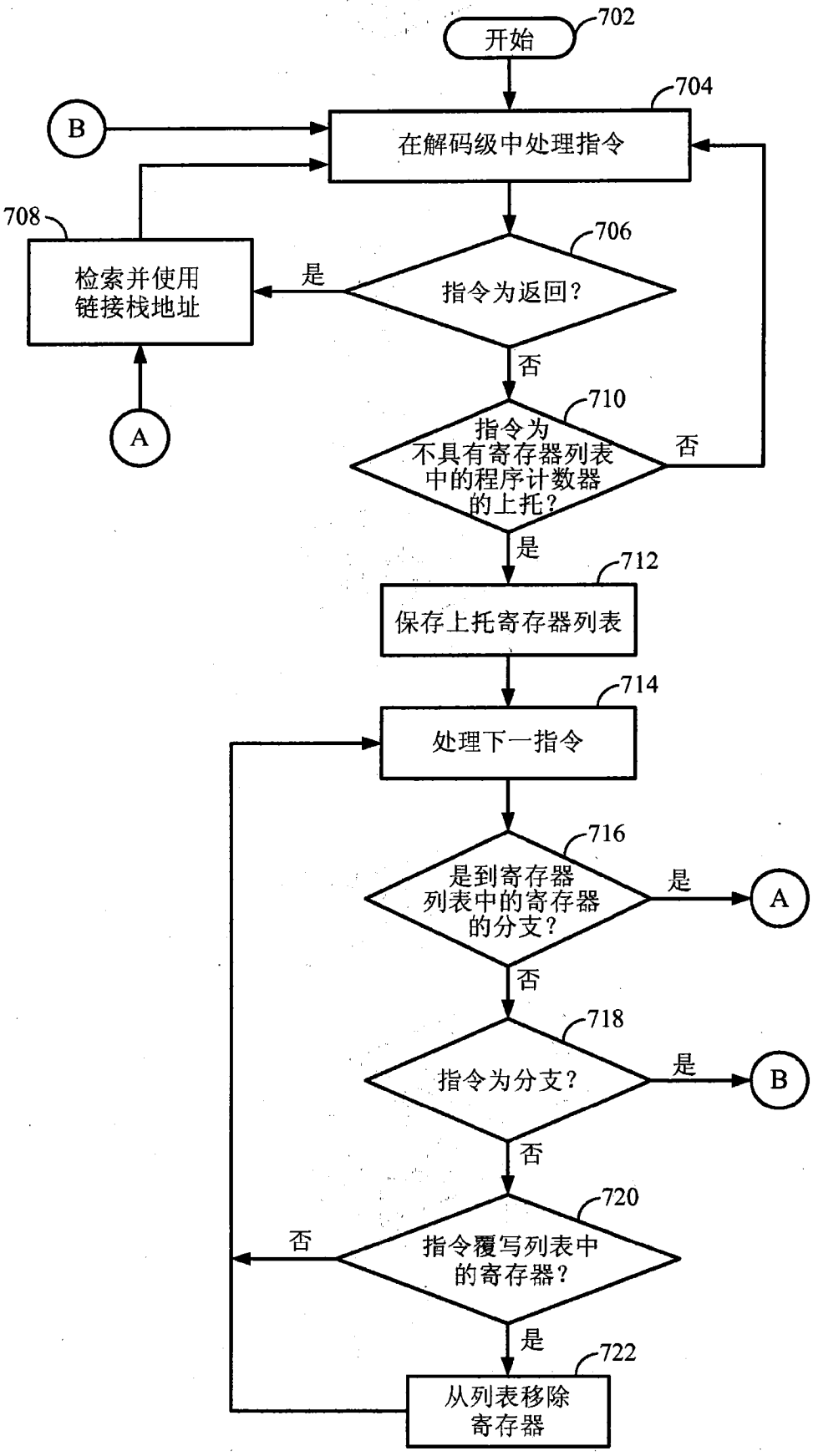

开始
702
在解码级中处理指令
704
B
708
检索并使用
链接栈地址
是
指令为返回?
706
否
A
指令为
不具有寄存器列表
中的程序计数器
的上托?
710
否
是
保存上托寄存器列表
712
处理下一指令
714
是到寄存器
列表中的寄存器
的分支?
716
是
A
否
指令为分支?
718
是
B
否
指令覆写列表中
的寄存器?
720
否
是
从列表移除
寄存器
722

图 7