



(12) 发明专利申请

(10) 申请公布号 CN 103631566 A

(43) 申请公布日 2014. 03. 12

(21) 申请号 201310619745. 1

(22) 申请日 2013. 11. 29

(71) 申请人 深圳中微电科技有限公司

地址 518057 广东省深圳市南山区高新南区  
科技南 12 路 18 号长虹科技大厦 706-8  
室

(72) 发明人 梅思行 劳咏仪

(74) 专利代理机构 深圳市科吉华烽知识产权事  
务所（普通合伙） 44248

代理人 刘显扬

(51) Int. Cl.

G06F 9/38 (2006. 01)

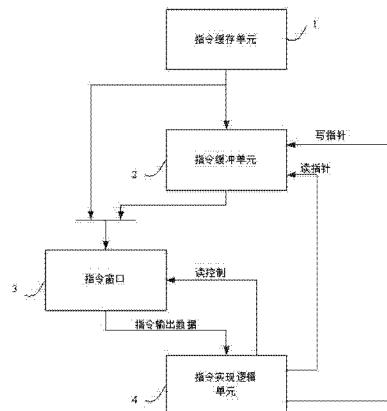
权利要求书1页 说明书4页 附图1页

(54) 发明名称

一种用于处理器中的指令取得装置及其处理  
器

(57) 摘要

本发明涉及一种用于处理器中的指令取得装  
置，包括指令缓存单元、指令缓冲单元、指令窗口  
和指令实现逻辑单元；所述指令缓存单元用于存  
储指令并将其存储的指令按顺序传输到所述指  
令缓冲单元；所述指令缓冲单元将当前需要执行  
的、所述指令实现逻辑单元一次取指取得数量的  
指令传输到所述指令窗口中存储；所述指令实现  
逻辑单元由所述指令窗口中取得指令并处理，使  
其得到输出和执行；所述指令实现逻辑单元还将  
当前执行指令的读指针和写指针返回到所述指  
令缓冲单元。本发明还涉及一种使用上述指令取得  
装置的处理器。实施本发明的用于处理器中的指  
令取得装置及其处理器，具有以下有益效果：从  
总体上减少了指令输出的时间。



1. 一种用于处理器中的指令取得装置,其特征在于,包括指令缓存单元、指令缓冲单元、指令窗口和指令实现逻辑单元;所述指令缓存单元用于存储指令并将其存储的指令按顺序传输到所述指令缓冲单元;所述指令缓冲单元将当前需要执行的、所述指令实现逻辑单元一次取指取得数量的指令传输到所述指令窗口中存储;所述指令实现逻辑单元由所述指令窗口中取得指令并处理,使其得到输出和执行;所述指令实现逻辑单元还将当前执行指令的读指针和写指针返回到所述指令缓冲单元。

2. 根据权利要求 1 所述的用于处理器中的指令取得装置,其特征在于,所述指令缓存单元还与所述指令窗口连接,所述指令缓冲单元未存储指令时,所述指令缓存单元直接输出指令到所述指令窗口。

3. 根据权利要求 2 所述的用于处理器中的指令取得装置,其特征在于,所述指令缓存单元通过读取所述指令实现逻辑单元输出到所述指令缓冲单元的读指针的值来判断所述指令缓冲单元是否存储有指令。

4. 根据权利要求 3 所述的用于处理器中的指令取得装置,其特征在于,所述指令窗口为缓冲存储单元,所述缓冲存储单元的存储量为两条指令的长度。

5. 根据权利要求 4 所述的用于处理器中的指令取得装置,其特征在于,所述指令实现逻辑单元每次由所述指令窗口按照指令排列的顺序依次读取两条指令。

6. 根据权利要求 5 所述的用于处理器中的指令取得装置,其特征在于,所述指令缓冲单元依据所述指令实现逻辑单元在上一个取指周期产生的读指针指向位置开始存储的两条指令输出到所述指令窗口,等待所述指令实现逻辑单元读取。

7. 根据权利要求 6 所述的用于处理器中的指令取得装置,其特征在于,所述读指针为零时,所述指令缓存单元将排列在最前面的两条指令直接输送到所述指令窗口中。

8. 一种处理器,包括取得指令装置,其特征在于,所述取得指令装置为权利要 1-7 任意一项所述的指令取得装置。

9. 根据权利要求 8 所述的处理器,其特征在于,所述处理器包括 CPU 或 GPU。

10. 根据权利要求 8 所述的处理器,其特征在于,所述处理器包括具有多个内核的并行处理器或具有多个内核的、并行的流处理器。

## 一种用于处理器中的指令取得装置及其处理器

### 技术领域

[0001] 本发明涉及处理器,更具体地说,涉及一种用于处理器中的指令取得装置及其处理器。

### 背景技术

[0002] 一个典型的指令执行流水线包括一个指令流出阶段(instruction issue stage),在该阶段中,需要在得知一个指令是否可以输出之前检查其是否存在数据冲突(data hazard)。这种检查通常使时序变得复杂并可能带来时间上的临界性。在超标量(super-scalar)设计中,一个能够取得多条指令并存储的指令缓冲单元可以在一定程度上减少上述问题。但是,在这种情况下,需要再次由该指令缓冲单元中读出其不久之前得到并存储的指令,这需要花费一定的时间,特别是在寻址上。因此,这种加入一个指令缓冲单元的结构虽然使得时序较为简单且不会带来时间上的临界性,但是其实现指令的时间较长。

### 发明内容

[0003] 本发明要解决的技术问题在于,针对现有技术的上述实现指令的时间较长的缺陷,提供一种实现指令的时间较短的用于处理器中的指令取得装置及其处理器。

[0004] 本发明解决其技术问题所采用的技术方案是:构造一种用于处理器中的指令取得装置,包括指令缓存单元、指令缓冲单元、指令窗口和指令实现逻辑单元;所述指令缓存单元用于存储指令并将其存储的指令按顺序传输到所述指令缓冲单元;所述指令缓冲单元将当前需要执行的、所述指令实现逻辑单元一次取指取得数量的指令传输到所述指令窗口中存储;所述指令实现逻辑单元由所述指令窗口中取得指令并处理,使其得到输出和执行;所述指令实现逻辑单元还将当前执行指令的读指针和写指针返回到所述指令缓冲单元。

[0005] 更进一步地,所述指令缓存单元还与所述指令窗口连接,所述指令缓冲单元未存储指令时,所述指令缓存单元直接输出指令到所述指令窗口。

[0006] 更进一步地,所述指令缓存单元通过读取所述指令实现逻辑单元输出到所述指令缓冲单元的读指针的值来判断所述指令缓冲单元是否存储有指令。

[0007] 更进一步地,所述指令窗口为缓冲存储单元,所述缓冲存储单元的存储量为两条指令的长度。

[0008] 更进一步地,所述指令实现逻辑单元每次由所述指令窗口按照指令排列的顺序依次读取两条指令。

[0009] 更进一步地,所述指令缓冲单元依据所述指令实现逻辑单元在上一个取指周期产生的读指针指向位置开始存储的两条指令输出到所述指令窗口,等待所述指令实现逻辑单元读取。

[0010] 更进一步地,所述读指针为零时,所述指令缓存单元将排列在最前面的两条指令直接输送到所述指令窗口中。

[0011] 本发明还涉及一种处理器,包括取得指令装置,所述取得指令装置为上述任意一

项所述的指令取得装置。

[0012] 更进一步地，所述处理器包括 CPU 或 GPU。

[0013] 更进一步地，所述处理器包括具有多个内核的并行处理器或具有多个内核的、并行的流处理器。

[0014] 实施本发明的用于处理器中的指令取得装置及其处理器，具有以下有益效果：由于在指令缓冲单元和指令实现逻辑单元之间加入了指令窗口，由指令缓冲单元或指令缓存单元将一次取指数量的指令输送到指令窗口，指令实现逻辑单元直接由上述指令窗口取得指令，不需要再由上述指令缓冲单元取得指令；由于该指令窗口中的指令就是一次取指所需要的指令，因此，上述指令实现逻辑单元由该指令窗口取得指令时，不需要进行寻址；所以节省了在指令缓冲单元中寻址的时间或者将该时间与数据传输或指令处理的时间重叠了，从总体上减少了指令输出的时间。

## 附图说明

[0015] 图 1 是本发明用于处理器中的指令取得装置及其处理器实施例中指令取得装置的结构示意图。

## 具体实施方式

[0016] 下面将结合附图对本发明实施例作进一步说明。

[0017] 如图 1 所示，在本发明的一种用于处理器中的指令取得装置及其处理器实施例中，该用于处理器中的指令取得装置包括指令缓存单元 1、指令缓冲单元 2、指令窗口 3 和指令实现逻辑单元 4；基本上而言，上述指令缓存单元 1 和指令缓冲单元 2 构成一个二级的 cache（高速缓冲存储器）；这个二级的 cache 并不是直接连接在指令实现逻辑单元 4 上，而是通过指令窗口 3 与指令实现逻辑单元 4 连接；也就是说，虽然其间存在其他的连接或信号交互等，但是，指令实现逻辑单元 4 并不会直接访问上述 cache 并得到指令内容；相反，上述指令实现逻辑单元 4 只能直接地由上述指令窗口得到指令，并将得到的指令处理后执行或输出执行。具体来讲，指令缓存单元 1 用于存储指令并将其存储的指令按执行或排列的顺序传输到指令缓冲单元 2；指令缓冲单元 2 将当前需要执行的、指令实现逻辑单元一次取指取得数量的指令传输到指令窗口 3 中存储；指令实现逻辑单元 4 由指令窗口 3 中取得指令并处理，使其得到输出和执行；在本实施例中，上述指令实现逻辑单元 4 还将其产生的读指针信号（Read-pointer）和写指针信号（Write-pointer）返回到指令缓冲单元 2 的相应端口，以便于该指令缓冲单元 2 产生正确的读指针和写指针。

[0018] 在本实施例中，指令缓冲单元 2 依据指令实现逻辑单元 4 在上一个取指周期产生的读指针信号使得其读指针指向位置开始存储的一次取指所能够得到的指令输出到指令窗口 3，等待指令实现逻辑单元 4 由上述指令窗口 3 读取这些指令或者说指令窗口 3 将这些指令输送到上述指令实现逻辑单元 4。普遍而言，上述指令窗口 3 中的指令都是按照上述方法，经过指令缓存单元 1、指令缓冲单元 2 而得到的。但是，在一些特殊的情况下，上述方法可能带来时间上的滞后，例如，在上述指令窗口 3 和指令缓冲单元 2 均为空时（即在指令窗口 3 和指令缓冲单元 2 中均没有存储指令，例如，在系统上电时），如果仍然按照上面的方法，则指令实现逻辑单元 4 需要等待较长的时间才能得到指令。为了减少这些情况下指令

实现逻辑单元 4 的等待时间,进一步减少整个系统指令输出及实现的时间,在本实施例中,当出现上述情况时,并不按照上述流程进行指令传输,而是直接将指令缓存单元 1 中本来要输出到指令缓冲单元 2 的指令(即本来在当前取指周期中输出到指令缓冲单元 2 的、一次取指能够得到的最大指令数量的指令),跳过上述指令缓冲单元 2,直接输出到上述指令窗口中。为此,在本实施例中,指令缓存单元 1 的指令输出端口还与指令窗口 3 的指令输入端口连接,请参见图 1,该指令窗口 3 的指令输入端口还与上述指令缓冲单元 2 的指令输出端口连接;也就是说,在本实施例中,上述指令窗口 3 的指令输入端口同时连接在上述指令缓冲单元 2 的指令输出端和指令缓存单元 1 的指令输出端。这样,在指令缓冲单元 3 未存储指令时,指令缓存单元 1 直接输出指令到指令窗口 3 或者说指令窗口 3 直接由上述指令缓存单元 1 取得指令。在本实施例中,通过读取指令实现逻辑单元 4 输出到指令缓冲单元 2 的读指针信号所确定的读指针的值来判断指令缓冲单元 2 是否存储有指令。一般来讲,当上述读指针的值为零或为一个事先设定的表示其存储的指令为空的值的时候,指令缓存单元 1 将排列在最前面的、一次取指所能得到的最大数量指令直接输送到指令窗口 3 中。

[0019] 在本实施例中,一个具体的例子是,由于每次取指能够取得两条指令,所以,上述指令实现逻辑单元 4 每次由指令窗口 3 读取两条指令,而指令窗口 3 在每个取指周期中由上述指令缓冲单元 2 或指令缓存单元 1 中得到的指令也是两条;而在本实施例中,指令窗口 3 为缓冲存储单元,该缓冲存储单元的存储量为两条指令的长度;也就是说,该指令窗口能够且只能容纳两条指令。故当指令实现逻辑单元 4 由指令窗口读取指令时并不需要寻址,直接读取即可。在其他的例子中,上述每次取指的指令条数可能不同,在这种情况下,上述指令窗口 3 的长度也就不同。例如,另一个例子中如果每次只能取得一条指令,则其对应的指令窗口 3 的长度就只能是一条指令的长度。总之,该指令窗口 3 的长度与该系统每次取指的最大指令数的长度相同。

[0020] 从一个方面来讲,在本实施例中,为了消除由于数据冲突检测逻辑导致时间临界而带来的在指令缓冲单元 2 中的时间延迟,设置了一个可以容纳两条指令的指令窗口 3 在上述指令缓冲单元的指令输出端,并使其与指令实现逻辑单元 4 连接,指令逻辑单元 4 由该指令窗口 3 取得指令。数据冲突检测逻辑总是考虑这两条在数据窗口 3 中的指令的实现或输出条件,以此来替代由指令缓冲单元本身中取得指令。在每个取值周期中,1 条或两条指令被输出并实现后,更多的指令被由指令缓冲单元 2 中读入上述指令窗口 3。这消除了指令缓冲单元指令读取加上数据冲突检测路径都带来的、耗时的访问时间,由于通过复杂的数据冲突检测逻辑被由指令直接由指令缓冲单元 2 中进入指令窗口 3 代替,当上述指令窗口 3 和指令缓冲单元 2 均为空(未存储指令)时,指令直接由指令缓存单元 1 跳过指令缓冲单元 2 而直接传输到指令窗口 3。这避免了由于指令缓冲单元 2 的空闲而导致流水线空闲的情况,进一步节省了时间。

[0021] 在本实施例中,还涉及一种处理器,该处理器包括取得指令装置,该取得指令装置为上述的指令取得装置。在本实施例中,这种处理器包括了所有具有这种取指装置的处理器。例如,这些处理器可以是 CPU 或 GPU,也可以是具有多个内核的并行处理器或具有多个内核的、并行的流处理器。

[0022] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对本发明专利范围的限制。应当指出的是,对于本领域的普通技术人员

来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,本发明专利的保护范围应以所附权利要求为准。

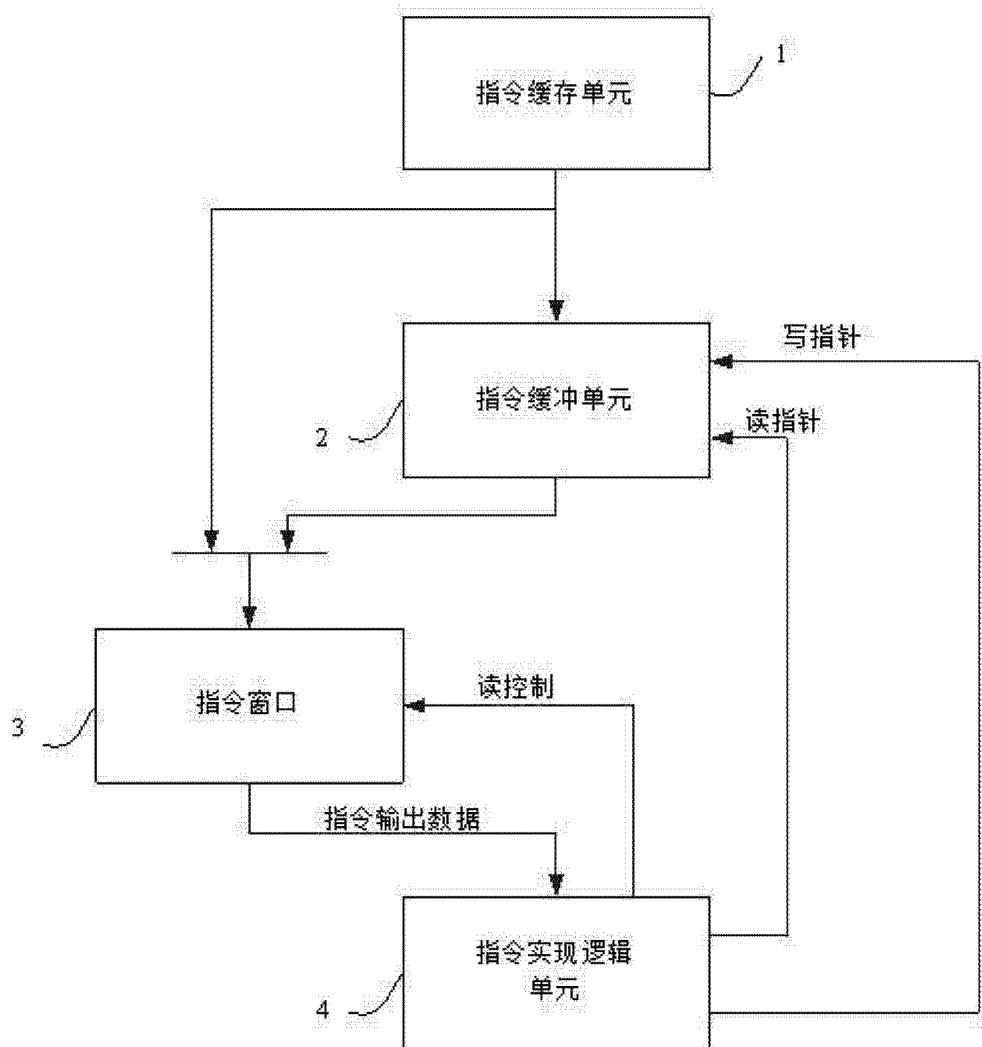


图 1