



(12) 发明专利申请

(10) 申请公布号 CN 117971324 A

(43) 申请公布日 2024. 05. 03

(21) 申请号 202410162292.2

(22) 申请日 2024.02.05

(71) 申请人 山东浪潮科学研究院有限公司

地址 250000 山东省济南市高新浪潮路
1036号S02号楼

(72) 发明人 许桂龙 赵鑫鑫 姜凯

(74) 专利代理机构 北京集佳知识产权代理有限公司 11227

专利代理师 王云晓

(51) Int. Cl.

G06F 9/38 (2018.01)

G06F 9/30 (2018.01)

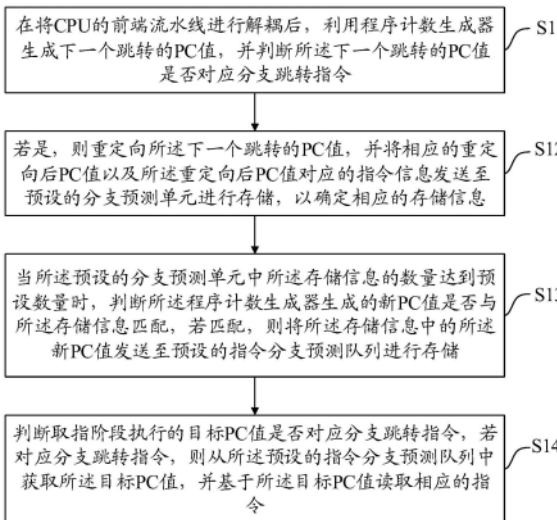
权利要求书2页 说明书8页 附图4页

(54) 发明名称

一种CPU指令预取方法、装置、设备及存储介质

(57) 摘要

本申请公开了一种CPU指令预取方法、装置、设备及存储介质,涉及计算机技术领域,包括:利用程序计数生成器生成下一个跳转的PC值,并判断是否对应分支跳转指令;若是,则重定向下一个跳转的PC值,将重定向后PC值及重定向后PC值对应的指令信息发送至分支预测单元进行存储,以确定存储信息;当分支预测单元中存储信息的数量达到预设数量时,判断程序计数生成器生成的新PC值是否与存储信息匹配,若匹配,则将存储信息中的新PC值发送至预设的指令分支预测队列进行存储;判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应,则从指令分支预测队列中获取目标PC值,基于目标PC值读取指令。由此,本申请提高了预取的准确率。



1. 一种CPU指令预取方法,其特征在于,包括:

在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;

若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;

当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;

判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。

2. 根据权利要求1所述的CPU指令预取方法,其特征在于,所述利用程序计数生成器生成下一个跳转的PC值,包括:

利用所述程序计数生成器根据当前程序执行顺序以及预先获取的分支预测结果生成下一个跳转的PC值。

3. 根据权利要求1所述的CPU指令预取方法,其特征在于,所述判断所述下一个跳转的PC值是否对应分支跳转指令之后,还包括:

若所述下一个跳转的PC值不对应分支跳转指令,则基于所述下一个跳转的PC值对应的下一个PC值读取相应的指令。

4. 根据权利要求1所述的CPU指令预取方法,其特征在于,所述判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配之后,还包括:

若所述新PC值与所述存储信息不匹配,则基于所述新PC值对应的下一个PC值读取相应的指令。

5. 根据权利要求1所述的CPU指令预取方法,其特征在于,所述若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储,包括:

若所述新PC值与所述存储信息匹配,则基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储,并将当前所述待跳转PC值发送至所述程序计数生成器进行重定向处理,以对当前所述新PC值进行更新;

判断当前所述新PC值是否与所述存储信息匹配;

若是,则重新跳转至所述基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储的步骤,直至完成全部所述新PC值的判断。

6. 根据权利要求1所述的CPU指令预取方法,其特征在于,所述若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令,包括:

若所述目标PC值对应分支跳转指令,则根据预设的程序计数值重定向控制模块发出的控制信号从所述预设的指令分支预测队列中获取所述目标PC值,并根据所述目标PC值从预设的指令高速缓存区中读取相应的指令。

7. 根据权利要求1至6任一项所述的CPU指令预取方法,其特征在于,所述判断取指阶段执行的目标PC值是否对应分支跳转指令之后,还包括:

若所述目标PC值不对应分支跳转指令,则基于所述目标PC值对应的下一个PC值读取相应的指令。

8. 一种CPU指令预取装置,其特征在于,包括:

第一判断模块,用于在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;

存储信息获取模块,用于若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;

第二判断模块,用于当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;

指令读取模块,用于判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。

9. 一种电子设备,其特征在于,包括:

存储器,用于保存计算机程序;

处理器,用于执行所述计算机程序,以实现如权利要求1至7任一项所述的CPU指令预取方法的步骤。

10. 一种计算机可读存储介质,其特征在于,用于存储计算机程序;其中,所述计算机程序被处理器执行时实现如权利要求1至7任一项所述的CPU指令预取方法的步骤。

一种CPU指令预取方法、装置、设备及存储介质

技术领域

[0001] 本发明涉及计算机技术领域,特别涉及一种CPU指令预取方法、装置、设备及存储介质。

背景技术

[0002] 在现代超高性能处理器设计当中,指令的预取以及预测十分重要,现今,预取技术可以按照预取的实现形式不同分为两种,分别是软件预取、硬件预取。软件预取是在程序的编译阶段或者编写程序时手动插入预取指令,提前将处理器会用到的数据从下级存储器提取到当前的高速缓存中,这样便可以达到处理器只需要多执行一条预取指令接下来的指令数据便不会在一级缓存产生较多缺失。硬件预取是处理器根据内部电路实时运行的状态,由硬件电路自身的控制模块进行预测处理器可能会发生访存缺失,进而在合适的时间向下级存储器发出访问数据请求,将接下来处理器可能会用到的数据返回一级Cache(高速缓冲存储器)。

[0003] 虽然传统的硬件预取对处理器有不错的性能提升,但是不能有效判断分支指令,使得预取准确率较低,造成流水效率降低。因此,如何提高预取的准确率,增加流水线取指令的效率是目前亟待解决的问题。

发明内容

[0004] 有鉴于此,本发明的目的在于提供一种CPU指令预取方法、装置、设备及存储介质,能够提高预取的准确率,增加流水线取指令的效率。其具体方案如下:

[0005] 第一方面,本申请公开了一种CPU指令预取方法,包括:

[0006] 在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;

[0007] 若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;

[0008] 当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;

[0009] 判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。

[0010] 可选的,所述利用程序计数生成器生成下一个跳转的PC值,包括:

[0011] 利用所述程序计数生成器根据当前程序执行顺序以及预先获取的分支预测结果生成下一个跳转的PC值。

[0012] 可选的,所述判断所述下一个跳转的PC值是否对应分支跳转指令之后,还包括:

[0013] 若所述下一个跳转的PC值不对应分支跳转指令,则基于所述下一个跳转的PC值对

应的下一个PC值读取相应的指令。

[0014] 可选的,所述判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配之后,还包括:

[0015] 若所述新PC值与所述存储信息不匹配,则基于所述新PC值对应的下一个PC值读取相应的指令。

[0016] 可选的,所述若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储,包括:

[0017] 若所述新PC值与所述存储信息匹配,则基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储,并将当前所述待跳转PC值发送至所述程序计数生成器进行重定向处理,以对当前所述新PC值进行更新;

[0018] 判断当前所述新PC值是否与所述存储信息匹配;

[0019] 若是,则重新跳转至所述基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储的步骤,直至完成全部所述新PC值的判断。

[0020] 可选的,所述若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令,包括:

[0021] 若所述目标PC值对应分支跳转指令,则根据预设的程序计数值重定向控制模块发出的控制信号从所述预设的指令分支预测队列中获取所述目标PC值,并根据所述目标PC值从预设的指令高速缓存区中读取相应的指令。

[0022] 可选的,所述判断取指阶段执行的目标PC值是否对应分支跳转指令之后,还包括:

[0023] 若所述目标PC值不对应分支跳转指令,则基于所述目标PC值对应的下一个PC值读取相应的指令。

[0024] 第二方面,本申请公开了一种CPU指令预取装置,包括:

[0025] 第一判断模块,用于在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;

[0026] 存储信息获取模块,用于若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;

[0027] 第二判断模块,用于当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;

[0028] 指令读取模块,用于判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。

[0029] 第三方面,本申请公开了一种电子设备,包括:

[0030] 存储器,用于保存计算机程序;

[0031] 处理器,用于执行所述计算机程序以实现前述的CPU指令预取方法。

[0032] 第四方面,本申请公开了一种计算机可读存储介质,用于保存计算机程序,所述计算机程序被处理器执行时实现前述的CPU指令预取方法。

[0033] 本申请在预取CPU指令时,首先在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。可见,本申请通过解耦前端设计,将超前的分支预测PC指导指令存储起来,在用到指令时,直接从指令分支预测队列中取。这样一来,不仅消除了取指令单元当周期取指令不足而无法满足后续流水线的需要,降低了成本,而且提高了预取的准确率。同时指令分支预测队列与取指流水线结合的设计,增加了流水线取指令的效率,避免了流水线空泡。

附图说明

[0034] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据提供的附图获得其他的附图。

[0035] 图1为本申请公开的一种CPU指令预取方法流程图;

[0036] 图2为本申请公开的一种解耦流水线示意图;

[0037] 图3为本申请公开的一种解耦的前端取指架构图;

[0038] 图4为本申请公开的一种IBFQ设计示意图;

[0039] 图5为本申请公开的一种CPU指令预取装置结构示意图;

[0040] 图6为本申请公开的一种电子设备结构图。

具体实施方式

[0041] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0042] 目前,传统的硬件预取对处理器有不错的性能提升,但是不能有效判断分支指令,使得预取准确率较低,造成流水效率降低。为了解决上述技术问题,本申请公开了一种CPU指令预取方法,能够提高预取的准确率,增加流水线取指令的效率。

[0043] 参见图1所示,本发明实施例公开了一种CPU指令预取方法,包括:

[0044] 步骤S11、在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令。

[0045] 本实施例中,如图2所示,首先将原本CPU(Central Processing Unit,中央处理器)前端流水进行解耦重新设计,在指令译码之前会经过4级流水,分别是指令分支预测两个阶段(IB0,IB1),指令分支预测队列(IBFQ)阶段,取指(IFETCH)阶段,译码(DECODE)阶段,在译码阶段之前共4个阶段,因此为4级流水。虽然解耦设计增加了一级流水来缓存了指令队列,增大了分支预测的代价,但是现代高性能处理器分支预测器的准确率能达到95%以上,因此该设计方法对CPU的性能理论上来说是正向提升的。

[0046] 在具体实施过程中,如图3所示,流水分为几个步骤,在CPU一开始取指的时候,利用所述程序计数生成器根据当前程序执行顺序以及预先获取的分支预测结果生成下一个跳转的PC值。之后判断所述下一个跳转的PC值是否对应分支跳转指令。需要说明的是,这里的PC值是指当前正在执行指令的指令存储地址,指令地址即指令所在内存单元的地址,它被储存在指令地址寄存器(instruction address register)中,而控制器会根据指令计数器中存放的指令地址从内存存储器中将指令取出到指令寄存器中,然后对该指令进行分析并执行,同时在指令计数器中自动形成下一条指令的地址。

[0047] 步骤S12、若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息。

[0048] 本实施例中,如果后端流水判断该指令为分支跳转指令将会重定向PC,并将该分支跳转的PC和指令信息送给分支预测单元(BTB,BHT)进行训练存储。其中,分支目标缓冲器(BTB,Branch Target Buffer)会将跳转目标(target)缓存在缓冲区中。分支历史表(BHT,Branch History Table),会对每条分支跳转指令是否跳转进行判断。若所述下一个跳转的PC值不对应分支跳转指令,则基于所述下一个跳转的PC值对应的下一个PC值读取相应的指令。

[0049] 步骤S13、当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储。

[0050] 本实施例中,当分支预测器完全工作起来时,也即预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若所述新PC值与所述存储信息不匹配,则基于所述新PC值对应的下一个PC值读取相应的指令。若所述新PC值与所述存储信息匹配,则基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储,并将当前所述待跳转PC值发送至所述程序计数生成器进行重定向处理,以对当前所述新PC值进行更新;判断当前所述新PC值是否与所述存储信息匹配;若是,则重新跳转至所述基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储的步骤,直至完成全部所述新PC值的判断。通俗来说,每一次pcgen(程序计数生成器)产生的PC如果命中BTB和BHT后,IB_CTRL(程序计数值重定向控制模块)都会将要跳转的PC值发送给IBFQ(指令分支预测队列)单元存储起来,同时会将要跳转的PC发送给pcgen进行重定向。以此类推,当pcgen产生的重定向的PC又命中BTB和BHT的话,也即重定向的PC又是分支跳转指令的PC,IB_CTRL又会将要跳转的PC值发送IBFQ和pcgen模块往复循环。

[0051] 步骤S14、判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。

[0052] 本实施例中,此后在取指阶段,判断执行的目标PC值是否对应分支跳转指令,若所述目标PC值对应分支跳转指令,则根据预设的程序计数值重定向控制模块发出的控制信号从所述预设的指令分支预测队列中获取所述目标PC值,并根据所述目标PC值从预设的指令高速缓存区中读取相应的指令。若所述目标PC值不对应分支跳转指令,则基于所述目标PC值对应的下一个PC值读取相应的指令。换句话说,如果在后端流水时判断执行的为分支跳转指令的话,取指单元会根据IB_CTRL发出的控制信号,立刻从IBFQ单元取得分支跳转之后PC值,并根据PC值去指令Cache(ICache (Instruction Cache,指令高速缓存))中读取指令,而不需要重新冲刷前面的流水重新从pcgen产生重定向的PC,因为在后端判断执行的指令为分支跳转指令并反馈到取指单元时,分支跳转的下一条PC值已经准备好了,这样就增加了CPU执行流水线的效率。同时分支预测为非阻塞的,并且pcgen产生的PC肯定会超前于取指令单元,就能进行提前预测并将提前预测的要跳转的PC值存储在IBFQ中,这些超前的PC就可以用来指导ICache进行指令预取。

[0053] 由上可知,本申请在预取CPU指令时,首先在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。可见,本申请通过解耦前端设计,将超前的分支预测PC指导指令存储起来,在用到指令时,直接从指令分支预测队列中取。这样一来,不仅消除了取指令单元当周期取指令不足而无法后续流水线的需要,降低了成本,而且提高了预取的准确率。同时指令分支预测队列与取指流水线结合的设计,增加了流水线取指令的效率,避免了流水线空泡。

[0054] 基于上一实施例可知,本申请IBFQ中保存了当前的PC以及PC对应的分支预测信息。这样一来,当指令Cache未命中、访问TLB(Translation lookaside buffer,旁路转换缓冲)未命中或者执行单元暂停流水线阻塞取指时,分支预测器可以继续运行产生新的预测结果存入IBFQ,也就是上面提到的超前的PC值。接下来,将针对IBFQ工作流程进行具体的说明。

[0055] 当PC存入IBFQ队列时,取指单元就会顺序的从IBFQ队尾取出PC,而IB_CTRL单元会根据分支预测结果源源不断的将PC值压入IBFQ队首,这样只要IBFQ表项未使用完,分支预测器可以一直工作。如果后端要执行刷新流水线,那么该队列中所有的数据都会被清空掉。

[0056] 如图4所示,设置的队列深度为10,该深度可以根据情况增加。

[0057] IBFQ的空满判断逻辑如下所示,当IBFQ的出队指针deqpnr等于入队指针enqpnr时,可能表示当前IBFQ队列为空,也可能为满。所以本申请将IBFQ的出入队列指针都扩宽一

位,用来追踪IBFQ的空满情况,此扩展位并不表现IBFQ的深度,当分支预测器送入一条PC进入队列会使得IBFQ加一,而如果送入PC数量超过IBFQ深度则会将最高位翻转,表示IBFQ进队已经超过一圈,同样的进行出队操作也会加一,如果出队数量超过IBFQ深度也会翻转最高位标志位,当最高位不相等为相反值,最低位都相等时表示为满,表示写指针追上了读指针,就算是满了。当最高位和最低位都完全相等时,表示为空。这样就可以通过出入队指针是否完全相等判断IBFQ是否处于满状态。当IBFQ由于取指令单元的阻塞而导致队列被塞满时会停止IB_CTRL单元继续更新IBFQ,只要IBFQ未产生满信号,IB_CTRL单元就会一直压入新预测的PC。

[0058] IBFQ还定义了取指令指针(fetchptr)进行指导取指单元访问指令Cache,当IBFQ不为空时,每周取出对应PC同时将fetchptr加一,但不会立即进行IBFQ的出队,确认取指单元能够从指令Cache得到指令才会真正执行出队,如果未取得指令,会一直将fetchptr指向的PC重播取指。同时增加了一个prefetchptr指针,用来执导Cache提前从内存中取指。接着是进行数据冲刷指针ifuptr和完成指针commptr,当后续流水线发生预测错误冲刷流水线时,会通过ifuptr指针恢复出入队列指针,并将IBFQ内的信息清空。完成指针commptr会在所有有效的IBFQ指令提交完成后加一,用来指示预测的PC有效执行,同时送给分支预测器训练。

[0059] 这样一来,指令分支预测队列与取指流水线结合的设计,增加了流水线取指令的效率,避免了流水线空泡。

[0060] 参见图5所示,本发明实施例公开了一种CPU指令预取装置,包括:

[0061] 第一判断模块11,用于在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;

[0062] 存储信息获取模块12,用于若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;

[0063] 第二判断模块13,用于当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;

[0064] 指令读取模块14,用于判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中获取所述目标PC值,并基于所述目标PC值读取相应的指令。

[0065] 由上可知,本申请在预取CPU指令时,首先在将CPU的前端流水线进行解耦后,利用程序计数生成器生成下一个跳转的PC值,并判断所述下一个跳转的PC值是否对应分支跳转指令;若是,则重定向所述下一个跳转的PC值,并将相应的重定向后PC值以及所述重定向后PC值对应的指令信息发送至预设的分支预测单元进行存储,以确定相应的存储信息;当所述预设的分支预测单元中所述存储信息的数量达到预设数量时,判断所述程序计数生成器生成的新PC值是否与所述存储信息匹配,若匹配,则将所述存储信息中的所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储;判断取指阶段执行的目标PC值是否对应分支跳转指令,若对应分支跳转指令,则从所述预设的指令分支预测队列中

获取所述目标PC值,并基于所述目标PC值读取相应的指令。可见,本申请通过解耦前端设计,将超前的分支预测PC指导指令存储起来,在用到指令时,直接从指令分支预测队列中取。这样一来,不仅消除了取指令单元当周期取指令不足而无法后续流水线的需要,降低了成本,而且提高了预取的准确率。同时指令分支预测队列与取指流水线结合的设计,增加了流水线取指令的效率,避免了流水线空泡。

[0066] 在一些具体的实施例中,所述第一判断模块11,具体可以包括:

[0067] 下一个跳转的PC值生成单元,用于利用所述程序计数生成器根据当前程序执行顺序以及预先获取的分支预测结果生成下一个跳转的PC值。

[0068] 在一些具体的实施例中,所述装置,还可以包括:

[0069] 第一指令读取模块,用于若所述下一个跳转的PC值不对应分支跳转指令,则基于所述下一个跳转的PC值对应的下一个PC值读取相应的指令。

[0070] 在一些具体的实施例中,所述装置,还可以包括:

[0071] 第二指令读取模块,用于若所述新PC值与所述存储信息不匹配,则基于所述新PC值对应的下一个PC值读取相应的指令。

[0072] 在一些具体的实施例中,所述第二判断模块13,具体可以包括:

[0073] PC值更新单元,用于若所述新PC值与所述存储信息匹配,则基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储,并将当前所述待跳转PC值发送至所述程序计数生成器进行重定向处理,以对当前所述新PC值进行更新;

[0074] PC值单元,用于判断当前所述新PC值是否与所述存储信息匹配;

[0075] 跳转单元,用于若是,则重新跳转至所述基于预设的程序计数值重定向控制模块将当前所述新PC值的指令对应的待跳转PC值发送至预设的指令分支预测队列进行存储的步骤,直至完成全部所述新PC值的判断。

[0076] 在一些具体的实施例中,所述指令读取模块14,具体可以包括:

[0077] 指令读取单元,用于若所述目标PC值对应分支跳转指令,则根据预设的程序计数值重定向控制模块发出的控制信号从所述预设的指令分支预测队列中获取所述目标PC值,并根据所述目标PC值从预设的指令高速缓存区中读取相应的指令。

[0078] 在一些具体的实施例中,所述装置,还可以包括:

[0079] 第三指令读取模块,用于若所述目标PC值不对应分支跳转指令,则基于所述目标PC值对应的下一个PC值读取相应的指令。

[0080] 进一步的,本申请实施例还公开了一种电子设备,图6是根据一示例性实施例示出的电子设备20结构图,图中的内容不能认为是对本申请的使用范围的任何限制。

[0081] 图6为本申请实施例提供的一种电子设备20的结构示意图。该电子设备20,具体可以包括:至少一个处理器21、至少一个存储器22、电源23、通信接口24、输入输出接口25和通信总线26。其中,所述存储器22用于存储计算机程序,所述计算机程序由所述处理器21加载并执行,以实现前述任一实施例公开的CPU指令预取方法中的相关步骤。另外,本实施例中的电子设备20具体可以为电子计算机。

[0082] 本实施例中,电源23用于为电子设备20上的各硬件设备提供工作电压;通信接口24能够为电子设备20创建与外界设备之间的数据传输通道,其所遵循的通信协议是能够适

用于本申请技术方案的任意通信协议,在此不对其进行具体限定;输入输出接口25,用于获取外界输入数据或向外界输出数据,其具体的接口类型可以根据具体应用需要进行选取,在此不进行具体限定。

[0083] 另外,存储器22作为资源存储的载体,可以是只读存储器、随机存储器、磁盘或者光盘等,其上所存储的资源可以包括操作系统221、计算机程序222等,存储方式可以是短暂存储或者永久存储。

[0084] 其中,操作系统221用于管理与控制电子设备20上的各硬件设备以及计算机程序222,其可以是Windows Server、Netware、Unix、Linux等。计算机程序222除了包括能够用于完成前述任一实施例公开的由电子设备20执行的CPU指令预取方法的计算机程序之外,还可以进一步包括能够用于完成其他特定工作的计算机程序。

[0085] 进一步的,本申请还公开了一种计算机可读存储介质,用于存储计算机程序;其中,所述计算机程序被处理器执行时实现前述公开的CPU指令预取方法。关于该方法的具体步骤可以参考前述实施例中公开的相应内容,在此不再进行赘述。

[0086] 本说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其它实施例的不同之处,各个实施例之间相同或相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。

[0087] 专业人员还可以进一步意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0088] 结合本文中所公开的实施例描述的方法或算法的步骤可以直接用硬件、处理器执行的软件模块,或者二者的结合来实施。软件模块可以置于随机存储器(RAM)、内存、只读存储器(ROM)、电可编程ROM、电可擦除可编程ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0089] 最后,还需要说明的是,在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0090] 以上对本申请所提供的技术方案进行了详细介绍,本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本申请的方法及其核心思想;同时,对于本领域的一般技术人员,依据本申请的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本申请的限制。

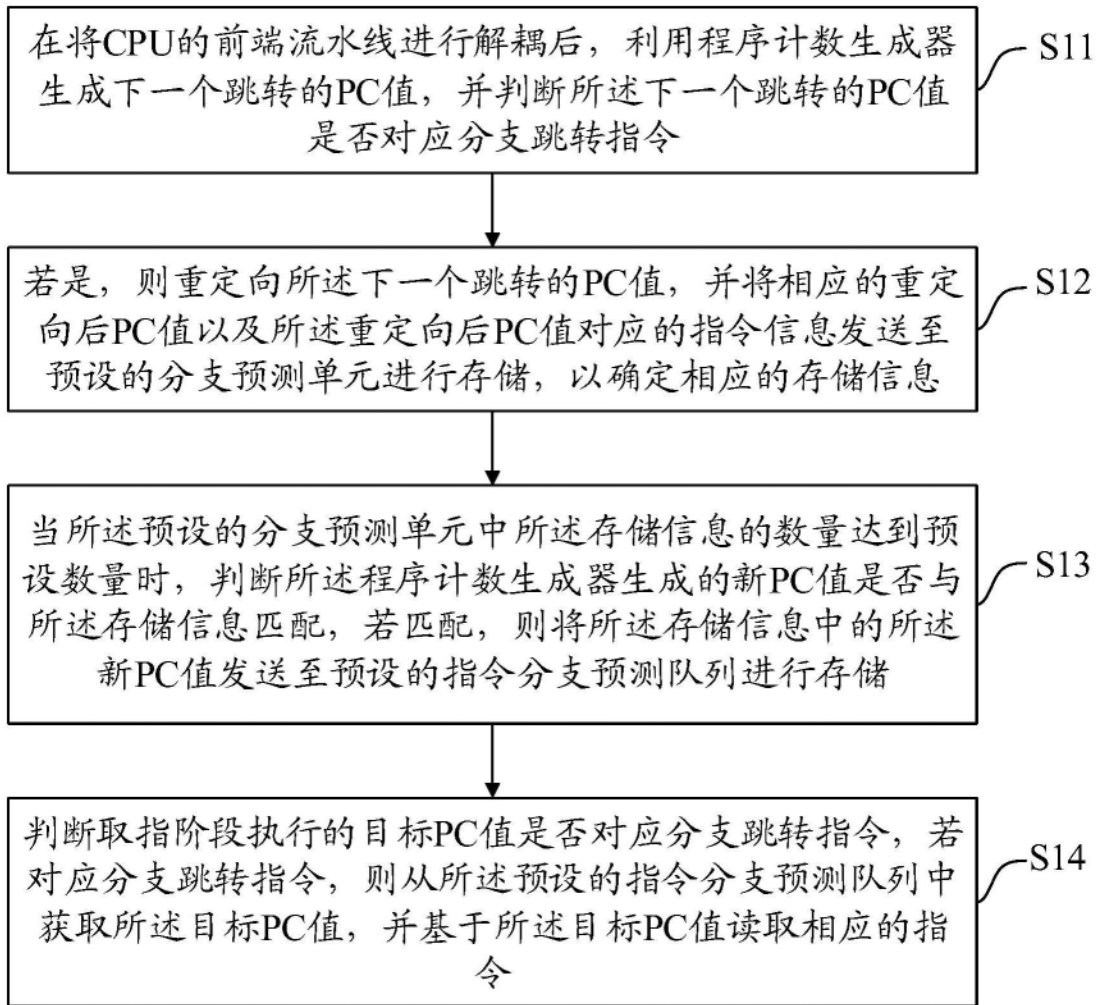


图1



图2

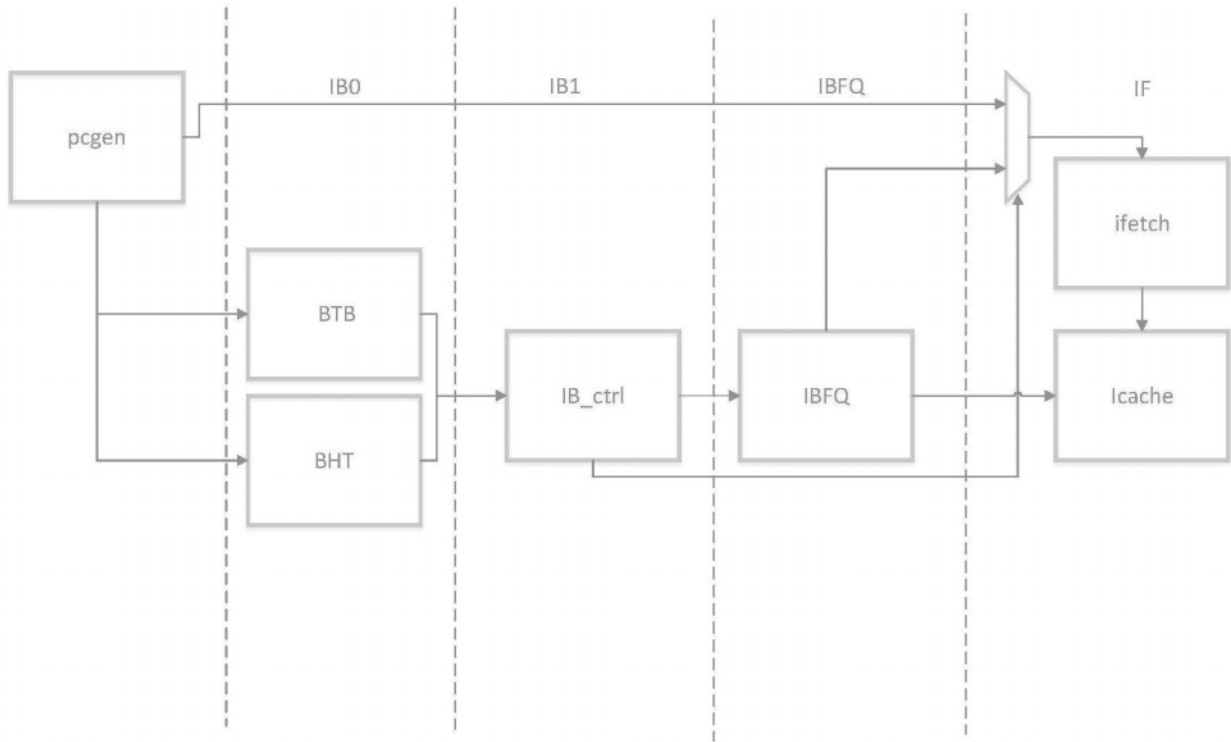


图3

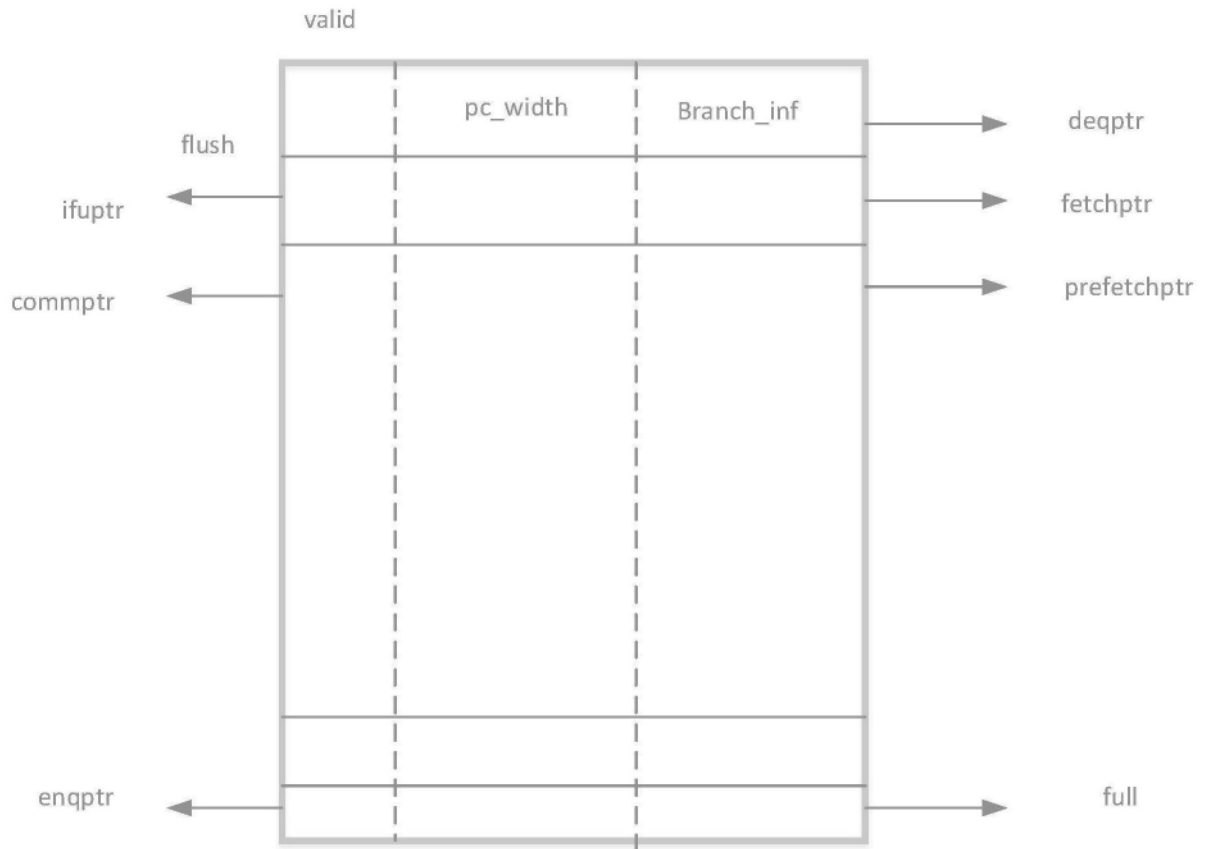


图4



图5

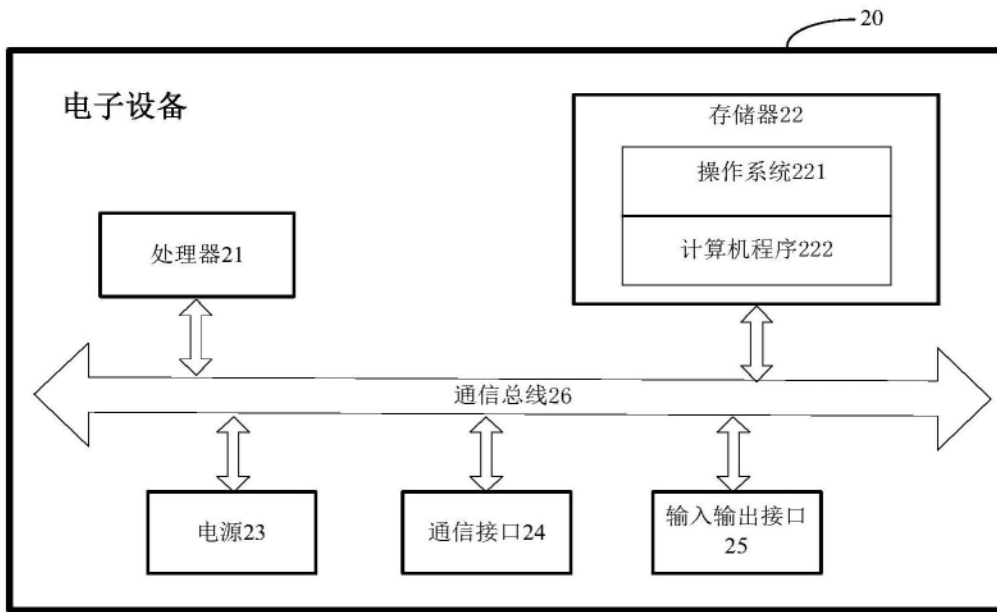


图6