



(12) 发明专利

(10) 授权公告号 CN 111459551 B

(45) 授权公告日 2022. 08. 16

(21) 申请号 202010289064.3

(22) 申请日 2020.04.14

(65) 同一申请的已公布的文献号
申请公布号 CN 111459551 A

(43) 申请公布日 2020.07.28

(73) 专利权人 上海兆芯集成电路有限公司
地址 201210 上海市张江高科技园区金科
路2537号301室

(72) 发明人 巩凡工 杨梦晨

(74) 专利代理机构 北京市柳沈律师事务所
11105
专利代理师 李芳华

(51) Int. Cl.
G06F 9/38 (2006.01)

(56) 对比文件

- CN 1542625 A, 2004.11.03
- CN 110825442 A, 2020.02.21
- US 2005108508 A1, 2005.05.19
- CN 109388429 A, 2019.02.26
- CN 102508643 A, 2012.06.20
- CN 105426160 A, 2016.03.23
- US 5630157 A, 1997.05.13
- CN 1521635 A, 2004.08.18

审查员 郑攀

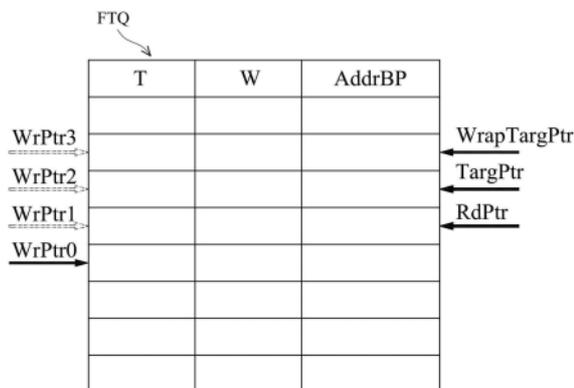
权利要求书3页 说明书13页 附图9页

(54) 发明名称

具高度领先分支预测器的微处理器

(57) 摘要

一种微处理器,其中分支预测器以及指令高速缓存之间以一取指目标队列解耦,且分支预测器系并行对N个指令地址进行分支预测。一当前周期下,经该分支预测器并行完成分支预测的N个地址中,未被跳转避开、且未与前一周期加载该取指目标队列之地址重叠者,被推入该取指目标队列,待后续周期弹出,作为该指令高速缓存的一取指地址。



1. 一种微处理器,包括:

指令高速缓存,依据取指地址进行取指;

分支预测器,并行对N个指令地址进行分支预测,N为大于1的整数;以及

耦接在该分支预测器以及该指令高速缓存之间的取指目标队列,

其中:

当前周期下,经该分支预测器并行完成分支预测的N个指令地址中,未被跳转避开、且未与前一周期推入该取指目标队列的指令地址重叠者,被推入该取指目标队列,待后续周期弹出,作为该指令高速缓存的该取指地址。

2. 如权利要求1所述的微处理器,其中:

该分支预测器是进行多级第一流水线运算;且

该指令高速缓存是进行多级第二流水线运算。

3. 如权利要求1所述的微处理器,其中:

该当前周期下,若该分支预测器并行完成分支预测的所述N个指令地址预测到对应的N个取指单位中存在取跳转的分支指令,且该取跳转的分支指令为跨取指单位,则该取跳转的分支指令所在的取指单位之一相邻取指单位的指令地址亦被推入该取指目标队列,待后续周期弹出,作为该指令高速缓存的该取指地址。

4. 如权利要求1所述的微处理器,其中:

N为3;

各指令地址对应的取指单位为M字节,M为数字;且

该当前周期下,经该分支预测器并行完成分支预测的指令地址包括PC、PC+M、以及PC+2*M。

5. 如权利要求4所述的微处理器,其中:

该当前周期与该前一周期并行完成分支预测的N个指令地址重叠一个指令地址。

6. 如权利要求5所述的微处理器,其中指令地址PC未被跳转避开、且未在该前一周期推入该取指目标队列时:

该取指目标队列提供第一单元格,储存指令地址PC;

该取指目标队列提供第二单元格,在指令地址PC不跳转时,或是指令地址PC取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址PC+M;

该取指目标队列提供第三单元格,在指令地址PC以及PC+M都不跳转时,或是指令地址PC不跳转、但指令地址PC+M取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址PC+2*M;且

该取指目标队列提供第四单元格,在指令地址PC、PC+M都不跳转、但指令地址PC+2*M取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址PC+3*M。

7. 如权利要求5所述的微处理器,其中指令地址PC、PC+M未被跳转避开、但指令地址PC已在该前一周期推入该取指目标队列时:

该取指目标队列提供第一单元格,储存指令地址PC+M;

该取指目标队列提供第二单元格,在指令地址PC+M不跳转时,或是指令地址PC+M取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址PC+2*M;且

该取指目标队列提供第三单元格,在指令地址PC+M不跳转、但指令地址PC+2*M取跳转、

但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+3*M$ 。

8. 如权利要求4所述的微处理器,其中:

该当前周期与该前一周期并行完成分支预测的N个指令地址重叠两个指令地址。

9. 如权利要求8所述的微处理器,其中指令地址PC未被跳转避开、且未在前一周期推入该取指目标队列时:

该取指目标队列提供第一单元格,储存指令地址PC;

该取指目标队列提供第二单元格,在指令地址PC不跳转时,或是指令地址PC取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+M$;

该取指目标队列提供第三单元格,在指令地址PC以及 $PC+M$ 都不跳转时,或是指令地址PC不跳转、但指令地址 $PC+M$ 取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+2*M$;且

该取指目标队列提供第四单元格,在指令地址PC、 $PC+M$ 都不跳转、但指令地址 $PC+2*M$ 取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+3*M$ 。

10. 如权利要求8所述的微处理器,其中指令地址PC、 $PC+M$ 未被跳转避开、但已在前一周期推入该取指目标队列时:

该取指目标队列提供第一单元格,储存指令地址 $PC+2*M$;且

该取指目标队列提供第二单元格,在指令地址 $PC+2*M$ 取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+3*M$ 。

11. 如权利要求4所述的微处理器,其中:

该当前周期与该前一周期并行完成分支预测的N个指令地址中无指令地址重叠。

12. 如权利要求11所述的微处理器,其中指令地址PC未被跳转避开时:

该取指目标队列提供第一单元格,储存指令地址PC;

该取指目标队列提供第二单元格,在指令地址PC不跳转时,或是指令地址PC取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+M$;

该取指目标队列提供第三单元格,在指令地址PC以及 $PC+M$ 都不跳转时,或是指令地址PC不跳转、但指令地址 $PC+M$ 取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+2*M$;且

该取指目标队列提供第四单元格,在指令地址PC、 $PC+M$ 都不跳转、但指令地址 $PC+2*M$ 取跳转、但其中取跳转的分支指令为跨取指单位时,储存指令地址 $PC+3*M$ 。

13. 如权利要求1所述的微处理器,其中该取指目标队列其中之一单元格包括:

第一字段,储存取跳转标志,标志该单元格储存的指令地址所对应的取指单位中,是否预测到取跳转的分支指令;

第二字段,储存跨取指单位标志,标志该取跳转的分支指令是否跨取指单位;以及

第三字段,储存该指令地址。

14. 如权利要求1所述的微处理器,其中该取指目标队列以(N+1)个并行写入指针进行并行推入操作。

15. 如权利要求14所述的微处理器,其中该分支预测器于该当前周期并行推入该取指目标队列的指令地址以上述并行写入指针标示,上述并行写入指针包括指向该当前周期推入该取指目标队列的指令地址所占第一个单元格的起始写入指针、以及自该起始写入指针

递增的指针。

16. 如权利要求1所述的微处理器,其中该取指目标队列以读取指针进行弹出操作。

17. 如权利要求16所述的微处理器,其中如果该读取指针指向的指令地址预测到对应的取指单位包括取跳转的分支指令时,则由目的指针指向该取跳转分支指令的跳转目的地。

18. 如权利要求17所述的微处理器,其中如果该取跳转的分支指令为跨取指单位时,则跨取指单位指针指向该取跳转的分支指令所在的取指单位的相邻取指单位对应的指令地址。

具高度领先分支预测器的微处理器

技术领域

[0001] 本申请涉及微处理器的指令提取。

背景技术

[0002] 计算机架构中,一般会以数字电路实现分支预测器(branch predictor),预测分支指令(例如,'if-then-else'条件分支指令、'call'呼叫指令、'return'返还指令、以及'jump'无条件分支指令)的跳转。分支预测器有效加速微处理器之指令提取,显著提升流水线架构的微处理器(pipelined microprocessor)的效能。

[0003] 在现有的分支预测技术中,分支预测器往往与指令高速缓存(L1i)的取指操作(instruction-fetch operation)同步使用相同的地址,分支预测操作只能和取指操作以相同的速度进行,而且分支预测器和指令高速缓存的延迟都会影响对方。因此分支预测技术的改良,为本技术领域一项重要课题。

发明内容

[0004] 本申请令微处理器中,同采多阶流水线运算的分支预测器以及指令高速缓存之间以一取指目标队列解耦,并适应该分支预测器的多地址并行预测,灵活填写该取指目标队列。

[0005] 根据本申请一种实施方式所实现的一微处理器包括一指令高速缓存、一分支预测器、以及一取指目标队列。该指令高速缓存依据一取指地址进行取指。该分支预测器系并行对N个地址进行分支预测,N为大于1的整数。该取指目标队列耦接在该分支预测器以及该指令高速缓存之间。一当前周期下,经该分支预测器并行完成分支预测的N个指令地址中,未被跳转避开、且未与前一周期推入该取指目标队列之指令地址重叠者,被推入该取指目标队列,待后续周期弹出,作为该指令高速缓存的该取指地址。

[0006] 一种实施方式中,该分支预测器是进行多级第一流水线运算,该指令高速缓存是进行多级第二流水线运算

[0007] 一种实施方式中,该当前周期下,若该分支预测器并行完成分支预测的所述N个指令地址预测到对应的N个取指单位中存在一取跳转的分支指令,且该取跳转的分支指令系跨取指单位,则该取跳转的分支指令所在之取指单位之一相邻取指单位之指令地址亦被推入该取指目标队列,待后续周期弹出,作为该指令高速缓存的该取指地址。

[0008] 一种实施方式中,N为3,各指令地址对应的取指单位为M字节,M为数字。该当前周期下,经该分支预测器并行完成分支预测的指令地址包括PC、PC+M、以及PC+2*M

[0009] 一种实施方式中,该当前周期与该前一周期并行完成分支预测的N个指令地址重叠一个指令地址。

[0010] 指令地址PC未被跳转避开、且未在该前一周期推入该取指目标队列时:该取指目标队列提供一第一单元格,储存指令地址PC;该取指目标队列提供一第二单元格,在指令地址PC不跳转时,或是指令地址PC取跳转、但其中取跳转的分支指令系跨取指单位时,储存指

令地址PC+M;该取指目标队列提供一第三单元格,在指令地址PC以及PC+M都不跳转时,或是指令地址PC不跳转、但指令地址PC+M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+2*M;且该取指目标队列提供一第四单元格,在指令地址PC、PC+M都不跳转、但指令地址PC+2*M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+3*M。

[0011] 指令地址PC、PC+M未被跳转避开、但指令地址PC已在该前一周期推入该取指目标队列时:该取指目标队列提供一第一单元格,储存指令地址PC+M;该取指目标队列提供一第二单元格,在指令地址PC+M不跳转时,或是指令地址PC+M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+2*M;且该取指目标队列提供一第三单元格,在指令地址PC+M不跳转、但指令地址PC+2*M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+3*M。

[0012] 一种实施方式中,该当前周期与该前一周期并行完成分支预测的N个指令地址重叠两个指令地址。

[0013] 指令地址PC未被跳转避开、且未在前一周期推入该取指目标队列时:该取指目标队列提供一第一单元格,储存指令地址PC;该取指目标队列提供一第二单元格,在指令地址PC不跳转时,或是指令地址PC取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+M;该取指目标队列提供一第三单元格,在指令地址PC以及PC+M都不跳转时,或是指令地址PC不跳转、但指令地址PC+M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+2*M;且该取指目标队列提供一第四单元格,在指令地址PC、PC+M都不跳转、但指令地址PC+2*M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+3*M。

[0014] 指令地址PC、PC+M未被跳转避开、但已在前一周期推入该取指目标队列时:该取指目标队列提供一第一单元格,储存指令地址PC+2*M;且该取指目标队列提供一第二单元格,在指令地址PC+2*M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+3*M。

[0015] 一种实施方式中,该当前周期与该前一周期并行完成分支预测的N个指令地址中无指令地址重叠。

[0016] 指令地址PC未被跳转避开时:该取指目标队列提供一第一单元格,储存指令地址PC;该取指目标队列提供一第二单元格,在指令地址PC不跳转时,或是指令地址PC取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+M;该取指目标队列提供一第三单元格,在指令地址PC以及PC+M都不跳转时,或是指令地址PC不跳转、但指令地址PC+M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+2*M;且该取指目标队列提供一第四单元格,在指令地址PC、PC+M都不跳转、但指令地址PC+2*M取跳转、但其中取跳转的分支指令系跨取指单位时,储存指令地址PC+3*M。

[0017] 一种实施方式中,该取指目标队列一单元格包括至少三个字段。一第一字段储存一取跳转标志,标志该单元格储存的一指令地址所对应之取指单位中,是否预测到一取跳转的分支指令。一第二字段储存一跨取指单位标志,标志该取跳转的分支指令是否跨取指单位。一第三字段储存该指令地址。

附图说明

- [0018] 图1根据本申请一种实施方式图解一微处理器100;
- [0019] 图2根据本申请一种实施方式详解该取指目标队列FTQ、以及该多任务器116的设计;
- [0020] 图3根据本申请一种实施方式图解并行分支预测的多条地址如何推入该取指目标队列FTQ,也图解如何将该取指目标队列FTQ单元格内容弹出;
- [0021] 图4A、4B以及4C则根据本申请多种实施方式条列各单元格的地址AddrBP如何填写;
- [0022] 图5A根据本申请一种实施方式图解刷新发生后,同步信号Sync何时再被拉起;
- [0023] 图5B图解图5A周期T5若预测到地址60至地址200跳转(但此时该取指目标队列FTQ并不为空),同步信号Sync将如何变化;以及
- [0024] 图5C根据本申请一种实施方式图解刷新后,又预测出一跳转目标地址时,同步信号Sync何时拉起。
- [0025] 附图标记列表
- [0026] 100:微处理器;
- [0027] 102:指令;
- [0028] 104:指令高速缓存;
- [0029] 106:译码器;
- [0030] 108:执行单元;
- [0031] 110:分支预测器;
- [0032] 114:跳转目标地址;
- [0033] 116:多任务器;
- [0034] 118:(指令高速缓存用)自增地址;
- [0035] 120、122:刷新地址;
- [0036] 124:多任务器;
- [0037] 202:多任务器;
- [0038] 204:取指目标队列FTQ单元格弹出的地址(Addr);
- [0039] 206、208:多任务器;
- [0040] 410、420、430:表格;
- [0041] AddrBP:指令地址;
- [0042] AddrEqual:比较信号;
- [0043] AddrL1i:取指地址;
- [0044] BTACQ:队列;
- [0045] FTQ:取指目标队列;
- [0046] RdPtr:读取指针;
- [0047] PDQ:队列;
- [0048] Sync:同步信号;
- [0049] T:取跳转标志;
- [0050] T0...T7:周期;

- [0051] TargPtr: 指针;
- [0052] W: 跨取指单位标志;
- [0053] WrAddr: 地址;
- [0054] WrapTargPtr: 指针; 以及
- [0055] WrPtr、WrPtr0…WrPtr3: 写入指针, 其中WrPtr0为起始写入指针。

具体实施方式

[0056] 以下叙述列举本发明的多种实施例。以下叙述介绍本发明的基本概念, 且并非意图限制本发明内容。实际发明范围应依照权利要求所保护的范围。

[0057] 图1根据本申请一种实施方式图解一微处理器100。

[0058] 根据取指地址AddrL1i, 指令102自指令高速缓存(instruction cache, 如本技术领域熟知的L1i) 104取出, 由译码器106译码, 最终由执行单元108执行。取指单位可为16字节(16B), 即一次取指操作取回16字节的指令。不同于传统技术将指令高速缓存104的取指地址AddrL1i也同步供分支预测用, 本申请的微处理器100设计使得分支预测器(branch predictor) 110进行分支预测的指令地址AddrBP得以高度领先该指令高速缓存104的取指地址AddrL1i。

[0059] 参阅图1, 微处理器100提供一取指目标队列(fetch target queue) FTQ, 耦接在该分支预测器110以及该指令高速缓存104之间, 储存经该分支预测器110完成分支预测的至少一指令地址AddrBP, 待弹出作为该指令高速缓存116的取指地址AddrL1i, 其中输入该分支预测器110进行分支预测的指令地址AddrBP领先于该取指地址AddrL1i。详言之, 经该分支预测器110完成分支预测的至少一指令地址AddrBP中全部或部分先被推入(push) 该取指目标队列FTQ, 再适时被弹出(pop) 作为该指令高速缓存104的该取指地址AddrL1i。取指目标队列FTQ使得分支预测器110与指令高速缓存104得以解耦(decoupled)。分支预测器110无须同步于该指令高速缓存104对该指令高速缓存104的取指地址AddrL1i进行分支预测, 而是独立地对指令地址AddrBP进行分支预测操作; 依照本申请设计, 分支预测器110可大幅领先该指令高速缓存104运作。

[0060] 分支预测器110与指令高速缓存104之间的解耦设计(取指目标队列FTQ) 可大大增进微处理器分支预测和取指效率。由于分支预测器110早早就预测出分支跳转, 无意义(不在预测分支方向上) 指令地址AddrBP不会被推入取指目标队列FTQ从而可被排除。仅有意义地址(在预测分支方向上) 会被推入该取指目标队列FTQ形成取指轨迹, 指引该指令高速缓存104上的指令提取。

[0061] 此段落初步说明分支跳转的信号流。如图所示, 指令地址AddrBP每周期自增送入分支预测器110进行分支预测。预测到跳转时, 跳转目标地址114更新指令地址AddrBP, 使得分支预测器110转而自跳转目标地址逐取指单位地进行分支预测。另外, 如果此时该取指目标队列FTQ恰好为空, 则更新为跳转目标地址114的指令地址AddrBP可不经该取指目标队列FTQ就往后传, 立即作为该指令高速缓存104的取指地址AddrL1i。多任务器116为该跳转目标地址114更新的指令地址AddrBP的直接传递提供通路。此后, 该指令高速缓存104可提供逐周期自增于该跳转目标地址的(指令高速缓存用) 自增地址118, 经该多任务器116更新该取指地址AddrL1i。待该取指目标队列FTQ内容追上该指令高速缓存104需求(例如, 上述指

令高速缓存用自增地址118同样存在于该取指目标队列FTQ,即,指令高速缓存用自增地址118与推入该取指目标队列FTQ的指令地址AddrBP之一相等),该指令高速缓存104可切换回以该取指目标队列FTQ弹出的指令地址为该取指地址AddrL1i。

[0062] 另一个讨论方向是微处理器100运算流水在线发生需要刷新(flush)取指地址AddrL1i的情形,例如分支预测器110可能不准确,载于该取指目标队列FTQ的取指轨迹可能是错的,微处理器100运算流水线较后端的译码器106或执行单元108可在辨识出当初的分支预测错误时发生刷新;又例如译码器106或执行单元108在运行中发生例外(exception)时也会发生刷新,当刷新被发起时,该译码器106或执行单元108会回传刷新地址120或122,经多任务器116刷新(flush)该取指地址AddrL1i,并同时清空该取指目标队列FTQ。之后周期,指令高速缓存104更从刷新地址120/122开始地址自增,逐周期提供指令高速缓存用自增地址118至多任务器116,使取指地址AddrL1i相应更新。另外,刷新地址120/122也耦接多任务器124,输出作为指令地址AddrBP,使分支预测器110亦切换成自上述刷新地址120/122做分支预测,不再浪费在错误方向的分支预测上。

[0063] 微处理器100运算流水在线发生刷新时,指令高速缓存104切换由刷新地址120/122、以及自增于该刷新地址120/122的指令高速缓存用自增地址118更新该取指地址AddrL1i。如此一来,尽管取指目标队列FTQ被清空,指令高速缓存104也不会因为该取指目标队列FTQ没有可弹出内容而停滞。待上述指令高速缓存用自增地址118同样存在于该取指目标队列FTQ(即,指令高速缓存用自增地址118与后续推入该取指目标队列FTQ的指令地址AddrBP之一相等)后,该指令高速缓存104即可切换回以该取指目标队列FTQ弹出的地址为该取指地址AddrL1i。

[0064] 整理之,操作指令高速缓存104的取指地址AddrL1i有两种来源:来自前级(来自该取指目标队列FTQ、或指令地址AddrBP直接传递)以及来自指令高速缓存用自增地址118(包括在预测到分支跳转且同时该取指目标队列FTQ中没有可供弹出的指令地址时自增于该跳转目标地址的指令高速缓存用自增地址118、以及发生刷新时自增于该刷新地址的指令高速缓存用自增地址118);对应的工作模式后文称之为同步模式和异步模式。同步模式下,多任务器116以该取指目标队列FTQ弹出地址、或直接以指令地址AddrBP为取指地址AddrL1i来源。异步模式下,多任务器116以其他输入120、122或118为取指地址AddrL1i来源。基于前述同步模式与异步模式的切换,指令高速缓存104几乎不因分支预测不及延宕,且该取指目标队列FTQ的效用也被极致发挥。

[0065] 另外,指令高速缓存104上可能寻无指令(cache miss)。相应之,指令高速缓存104会先完成指令加载,再接收该取指目标队列FTQ重新弹出的错失地址,完成取指。

[0066] 图1微处理器100还包括队列PDQ。以下讨论之。

[0067] 分支预测器110中通常以内存(如SRAM)记录多种分支预测用信息或表格。例如:分支目的地址高速缓存(Branch Target Address Cache,简称BTAC)、以及分支经验表格(Branch History Table,简称BHT)。分支目的地址高速缓存BTAC可载有取址单位所含的分支指令的分支类型(branch type)、分支目的地址(target address)…等。分支经验表格BHT用于预测分支方向(prediction of branch direction)的查找,判断取跳转taken、或不取跳转not taken。该些信息或表格可随着分支预测器110运作而更新。分支预测器110因大幅领先取指地址AddrL1i运作,累积的更新信息相当可观,即先推入该队列PDQ。该些更新

信息会适时自队列PDQ弹出,送入微处理器100后端流水线,供BTAC、BHT更新时使用。

[0068] 一种实施方式中,指令高速缓存104以及分支预测器110都是在微处理器100的多级(例如,多达20级)流水线运算中的C、I、B、U四级操作。指令地址AddrBP必须完成分支预测(走到最末U级)、且未被领先地址的分支预测避开,方能推入该取指目标队列FTQ。如此一来,推入该取指目标队列FTQ必然有意义(在预测分支方向上)。

[0069] 图2根据本申请一种实施方式详解该取指目标队列FTQ、以及该多任务器116的设计。

[0070] 该取指目标队列FTQ的各单元格(entry)可储存三种信息:取跳转标志(predict taken flag)T;跨取指单位标志(wrap flag)W;以及指令地址AddrBP。一种实施方式中,一单元格有50位(bit)。该指令地址AddrBP占48位。该取跳转标志T可占一位,标志该指令地址AddrBP指示的取指单位中,是否预测到一取跳转(taken)的分支指令。该跨取指单位标志W可占一位,标志该取跳转的分支指令是否跨取指单位(即是否为“wrap”),即对应的取指单位中,是否有预测为取跳转(taken)的分支指令跨越两个取指单位(16B),即该取跳转(taken)的分支指令本身的一部分在第一个16B,另一部分在相邻的第二个16B。值得注意的是,当该指令地址AddrBP并不包含分支指令、或者包含分支指令但并不取跳转(taken)时,即使该分支指令跨取指单位也并不需要标志该跨取指单位标志W,因为如果不跳转则下一个取指单位(16B)不会被跳转避开,其顺序地一定会被取指,因此不必标识。

[0071] 各笔信息(由T、W以及AddrBP组成)可根据指针WrPtr推入(push)该取指目标队列FTQ中相应单元格。指针RdPtr则是用来读取单元格,将其中信息(T、W以及AddrBP)自该取指目标队列FTQ弹出(pop)。

[0072] 图2实施例中图1的多任务器116可由三个多任务器202、206以及208结合实现。其中多任务器202除了接收取指目标队列FTQ单元格弹出的地址(AddrBP)204,也接收不透过取指目标队列FTQ取得、确定会进行而待填入单元格的地址WrAddr。在取指目标队列FTQ为空时,多任务器202将地址WrAddr直接送往多任务器206,避开设取指目标队列FTQ存取的延滞。多任务器206由一同步信号Sync控制,决定取指目标队列FTQ工作于前述同步模式或异步模式:同步信号Sync为真时,取指目标队列FTQ工作于同步模式,在同步模式下取前级(来自多任务器202)提供的地址为指令高速缓存104的取指地址AddrL1i,取指目标队列FTQ弹出的地址、或直接收到的地址WrAddr经由多任务器206、208输出,作为指令高速缓存104的取指地址AddrL1i;同步信号Sync为假时,取指目标队列FTQ工作于异步模式,在异步模式下指令高速缓存104的取指地址AddrL1i可为预测出跳转时自增于自跳转目标地址114的指令高速缓存用自增地址118,或刷新发生时自增于刷新地址120/122的指令高速缓存用自增地址118。

[0073] 指令高速缓存104上寻无指令(cache miss)时,指针RdPtr会重回错失地址所在的单元格。指令高速缓存104完成指令加载后,指针RdPtr所指的错失地址自该取指目标队列FTQ重新弹出。指令高速缓存104完成取指。

[0074] 一种实施方式中,分支预测器110系并行完成多条地址的分支预测。图3根据本申请一种实施方式图解并行分支预测的多条地址如何推入该取指目标队列FTQ,也图解如何将该取指目标队列FTQ单元格内容弹出。以下说明例系并行进行三个指令地址的分支预测。特别是,每周并行完成分支预测的三个指令地址预测到对应的三个取指单位中存在取跳

转 (taken) 的分支指令, 并且该取跳转的分支指令本身为“跨取指单位” (即 wrap, 也即该取跳转的分支指令本身跨越两个 16B), 则该取跳转的分支指令所在的取指单位的相邻取指单位的地址也可同周期推入该取指目标队列 FTQ。因此, 并行推入该取指目标队列 FTQ 的地址可能多达四个 (3+1), 自该取指目标队列 FTQ 所接收的指针 WrPtr0 指向的单元格起始储存。自增自该指针 WrPtr0, 指针 WrPtr1、WrPtr2、以及 WrPtr3, 接续标示三个单元格。指针 WrPtr0 ~ WrPtr3 实现四个单元格的并行储存。该取指目标队列 FTQ 的操作另外还使用到指针 RdPtr、TargPtr 以及 WrapTargPtr。指针 RdPtr 指定弹出内容的单元格, 如果当指针 RdPtr 指向的指令地址 AddrBP 预测到对应的取指单位包括一取跳转的分支 (taken branch) 指令时, 其接续单元格则以指针 TargPtr 标示, 由指针 TargPtr 指向这个取跳转分支指令的跳转目的地址 (target address)。若该取跳转的分支指令本身“跨取指单位” (wrap, 即 W 标志被置位), 则指针 WrapTargPtr 即指向存有相邻取指单位地址的相应单元格。指针 TargPtr 和 WrapTargPtr 是存在取跳转的分支指令或/及该取跳转的分支指令为 wrap 时可以直接从该取指目标队列 FTQ 中得到该取跳转的分支指令的跳转目的地址的信息, 不需要额外的资源去存储。

[0075] 图 4A、4B 以及 4C 则根据本申请多种实施方式条列各单元格的地址 AddrBP 如何填写。分支预测器 110 并行三个地址的分支预测, 此外, 分支预测器 110 是进行多级第一流水线运算 (C/I/B/U 级); 指令高速缓存 104 是进行多级第二流水线运算 (例如也是 C/I/B/U 级)。以下讨论当前周期下完成分支预测的地址标为 PC、PC+16、以及 PC+32, 例如如后面第 5A、5B 及 5C 图所示, 分支预测流水线包括 C/I/B/U 四级, 并且前述每个取指单位对应取指 16 字节 (16B) 的指令, 因此以下讨论当前周期完成分支预测 U 级流水线运算的地址标为 PC、PC+16、以及 PC+32。单元格的写入条件包括考虑一旗标 afterbr。由于这里讨论的是分支预测器 110 前后两个周期并行完成分支预测的三个指令地址 AddrBP 中有一个或两个重叠的情形, 因此旗标 afterbr 为真时, 代表分支预测完毕 (完成 U 级流水线运算) 的地址 PC 没出现在领先一周期的分支预测中。例如, 地址 PC 为分支跳转后的头一笔地址, 或是刷新后的头一笔地址。旗标 afterbr 为否 (~afterbr) 时, 代表分支预测完毕 (完成 U 级流水线运算) 的地址 PC 已出现在领先一周期的分支预测中。

[0076] 图 4A 令相邻周期的并行分支运算只有一个地址重叠。例如, 第一周期将指令地址 A、B、C 送入分支预测的第一级流水线, 即第 5 图所示的 C 级流水线运算, 第二周期将地址 C、D、E 送入 C 级流水线运算。分支预测用自增地址 C、D、E 中, 有一个地址 C 与前一周期重叠。当前周期下完成分支预测最后一级流水线, 即第 5 图所示的 U 级流水线运算的地址 PC、PC+16、以及 PC+32 中, 若地址 PC 确定未被跳转避开, 则依照表 410 将推入该取指目标队列 FTQ 的指令地址 AddrBP 填入相关单元格。

[0077] 地址 PC 未在前一周期推入该取指目标队列时 (afterbr), 则在本实例中指针 WrPtr0...WrPtr3 所指的四个单元格都可能会用到。指针 WrPtr0 所指单元格无额外条件即储存地址 PC。在地址 PC 不跳转时, 或是地址 PC 取跳转、但取跳转的分支指令本身系跨取指单位 (即 wrap) 时, 指针 WrPtr1 所指单元格储存地址 PC+16。在地址 PC 以及 PC+16 都不跳转时, 或是地址 PC 不跳转、但 PC+16 取跳转且取跳转的分支指令本身系跨取指单位 (即 wrap) 时, 指针 WrPtr2 所指单元格储存地址 PC+32。在地址 PC、PC+16 都不跳转、但 PC+32 系取跳转且取跳转的分支指令本身跨取指单位 (即 wrap) 时, 指针 WrPtr3 所指单元格储存地址 PC+48。

[0078] 地址PC已在前一周期推入该取指目标队列时(\sim afterbr),若PC+16未被跳转避开,推入该取指目标队列FTQ的指令地址AddrBP填写如下。指针WrPtr0所指单元格无额外条件即储存地址PC+16。在地址PC+16不跳转时,或是地址PC+16取跳转、但取跳转的分支指令本身系跨取指单位(即wrap)时,指针WrPtr1所指单元格储存地址PC+32。在地址PC+16不跳转、但PC+32取跳转且取跳转的分支指令本身系跨取指单位(即wrap)时,指针WrPtr2所指单元格储存地址PC+48。

[0079] 图4B令相邻周期的并行分支运算有两个地址重叠。例如,第一周期将指令地址A、B、C送入分支预测的第一级流水线,即第5图所示的C级流水线运算,第二周期将地址B、C、D送入C级流水线运算。分支预测用自增地址B、C、D中,有两个地址B、C与前一周期重叠。当前周期下完成分支预测最后一级流水线,即第5图所示的U级流水线运算的地址PC、PC+16、以及PC+32中,若地址PC确定未被跳转避开,则依照表420将推入该取指目标队列FTQ的指令地址AddrBP填入相关单元格。相较表格410,差异在 \sim afterbr相关字段,因此与表格410相同的afterbr相关字段以下不再赘述。

[0080] 地址PC已在前一周期推入该取指目标队列时(\sim afterbr),若PC+16未被跳转避开,指令地址AddrBP填写如下。指针WrPtr0所指单元格无额外条件即储存地址PC+32。在地址PC+32取跳转、但取跳转的分支指令本身系跨取指单位(即wrap)时,指针WrPtr1所指单元格储存地址PC+48。

[0081] 图4C令相邻周期的并行分支运算完全不重叠。例如,第一周期将指令地址A、B、C送入分支预测的第一级流水线,即第5图所示的C级流水线运算,第二周期将地址E、F、G送入C级流水线运算。分支预测用自增地址E、F、G中,无一与前一周期重叠。当前周期下完成分支预测最后一级流水线,即第5图所示的U级流水线运算的地址PC、PC+16、以及PC+32中,若地址PC确定未被跳转避开,此例无需考虑旗标afterbr。表格430显示,其指令地址AddrBP填写状况。

[0082] 在其他实施方式中,分支预测器110并行处理的指令地址数量可为其他数目N。各周期填写的单元格数量可达N+1个。取指单位也不限定为16字节,可为其他数字M。

[0083] 整理之,该取指目标队列FTQ的单元格的填写,包括考虑该指令地址AddrBP是否有意义(未被跳转避开),并检查该指令地址AddrBP在前一周期是否已经填入该取指目标队列FTQ(例如,检查afterbr或者是否与前一周重叠)。

[0084] 以下段落讨论指令高速缓存104所接收的取指地址AddrL1i来源,说明其如何随该取指目标队列FTQ工作于同步模式或异步模式,切换取指地址AddrL1i来源。以图3为例,该取指目标队列FTQ是根据指针WrPtr0(以下称之为起始写入指针)以及RdPtr(以下称之为读取指针)操作。在一实施例中是藉由比较该起始写入指针WrPtr0以及该读取指针RdPtr,判断指令高速缓存104工作于同步模式或异步模式,进而切换图2的同步信号Sync,设定指令高速缓存104的取指地址AddrL1i来源。

[0085] 一种实施方式中,该读取指针RdPtr每周期递增;该分支预测器110并行分支预测完毕的指令地址AddrBP系根据该起始写入指针WrPtr0推入该取指目标队列FTQ,由并行写入指针(包括WrPtr0、WrPtr1...)标示,即是说,该起始写入指针WrPtr0每周期订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格。如前所述,当微处理器100的流水线发生刷新的情形时,该取指目标队列FTQ清空,相应刷新地址120/122的回传,该起

始写入指针WrPtr0以及该读取指针RdPtr初始化指向该取指目标队列FTQ的一起始单元格。如前所述,当该分支预测器110预测出一跳转目标地址(即进行分支预测操作的指令地址AddrBP存在取跳转的分支指令),同时该取指目标队列FTQ中没有可供弹出的指令地址时,该读取指针RdPtr与该起始写入指针WrPtr0均订正为指向该取指目标队列FTQ中已存入指令地址的单元格后接续的空白单元格。相应以上刷新或跳转事件,该指令高速缓存104的取指地址AddrL1i可以是刷新地址120/122、或跳转目标地址(绕过该取指目标队列FTQ、直接经AddrBP传来的114)、或后续逐周期自增于前述两者的指令高速缓存用自增地址118。同步信号Sync为放下(de-asserted)。待该读取指针RdPtr与确实有储存内容的上述并行写入指针(包括WrPtr0、WrPtr1…)中任一相等后,该指令高速缓存104切换回以该取指目标队列FTQ弹出的指令地址AddrBP为该取指地址AddrL1i,在一实施例中,该读取指针RdPtr是在第一周期与上述并行写入指针(包括WrPtr0、WrPtr1…)中任一相等,该指令高速缓存104可在该第一周期的下一周期切换回以该取指目标队列FTQ弹出的指令地址AddrBP为该取指地址AddrL1i。同步信号Sync拉起(asserted)。

[0086] 上述实施例是通过“指针比对”的方式来决定该指令高速缓存104是否从前述异步模式切换回同步模式,即,是否从以指令高速缓存用自增地址118切换回以该取指目标队列弹出的指令地址AddrBP为该取指地址AddrL1i。在其它实施例中,也可以通过直接比较指令高速缓存用自增地址118是否与推入该取指目标队列FTQ的指令地址AddrBP之一相等来决定该指令高速缓存104是否从前述异步模式切换回同步模式,即,是否从以指令高速缓存用自增地址118切换回以该取指目标队列FTQ弹出的指令地址AddrBP为该取指地址AddrL1i。值得注意的是,第一种实施例中的指针比对方式——比较并行写入指针(包括WrPtr0、WrPtr1…)以及读取指针RdPtr——损耗的资源有限。相比于第二种实施例中直接比较48位的地址的方式,指针RdPtr、WrPtr0、WrPtr1…长度通常很短,例如,只有三位。微处理器100不用花太多资源就能进行指针比对,快速决定拉起、或放下同步信号Sync。

[0087] 图5A根据本申请一种实施方式图解刷新发生后,同步信号Sync何时再被拉起。此说明例中,分支预测器110是并行对三个地址做分支预测,且相邻两个周期所输入处理的地址有一个重叠(参考图4A),此外,该分支预测器110是进行多级第一流水线运算(C/I/B/U级);指令高速缓存104是进行多级第二流水线运算(例如也是C/I/B/U级)。值得注意的是,为了图面简洁,分支预测各周期只标出起始地址,另外两个接续地址未标。取指目标队列FTQ操作的起始写入指针WrPtr0在左侧以空心箭头标示,至于读取指针RdPtr则在右侧以实心箭头标示。为了图面简洁,其他写入指针(WrPtr1…WrPtr3)并未标示。

[0088] 周期T0,相应地址刷新事件,同步信号Sync为放下(de-asserted),取指目标队列FTQ进入异步模式:取指目标队列FTQ清空,起始写入指针WrPtr0以及读取RdPtr初始化至起始单元格。刷新地址10及其相邻的地址20、30输入分支预测器110,并行进行C级流水线运算。即分支预测器110切换为对刷新地址10进行分支预测。同时,刷新地址10也输入该指令高速缓存104做C级流水线运算,即刷新地址10亦刷新该指令高速缓存104的取指地址AddrL1i。

[0089] 在异步模式下,周期T1,读取指针RdPtr移到下一个单元格。分支预测用自增地址30、40、50(图中只标出起始地址30)输入分支预测器110,并行进行C级流水线运算。指令高速缓存用的自增地址20也输入该指令高速缓存104做C级流水线运算。

[0090] 周期T2,读取指针RdPtr移到下一个单元格。分支预测用自增地址50、60、70(图中只标出起始地址50)输入分支预测器110,并行进行C级流水线运算。指令高速缓存用的自增地址30也输入该指令高速缓存104做C级流水线运算。

[0091] 周期T3,读取指针RdPtr移到下一个单元格。地址10、20、30(图中只标出起始地址10)的分支预测进行到U级流水线运算(即完成分支预测),且没有预测到跳转。根据图4A,当前周期完成分支预测的地址10、20、30填入该取指目标队列FTQ,指针WrPtr0、WrPtr1、以及WrPtr2订正为指向这三个单元格,其中起始写入指针WrPtr0指向当前周期(T3)推入该取指目标队列FTQ的指令地址所占第一个单元格,即起始写入指针WrPtr0指向地址10所占的单元格。此外在T3,分支预测用自增地址70、80、90(图中只标出起始地址70)输入分支预测器110,并行进行C级流水线运算。同时在T3,指令高速缓存用的自增地址40也输入该指令高速缓存104做C级流水线运算。

[0092] 周期T4,读取指针RdPtr移到下一个单元格。地址30、40、50(图中只标出起始地址30)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,当前周期完成分支预测的地址30、40、50中与前一周期不重叠的地址40、50填入该取指目标队列FTQ,指针WrPtr0以及WrPtr1订正为指向这两个单元格,其中起始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址40所占的单元格)。此外在T4,分支预测用自增地址90、A0、B0(图中只标出起始地址90)输入分支预测器110,并行进行C级流水线运算。同时在T4,指令高速缓存用的自增地址50也输入该指令高速缓存104做C级流水线运算。值得注意的是,在当前周期T4,读取指针RdPtr与并行写入指针之一WrPtr1相等。比较信号AddrEqual拉起。所指单元格确实载有地址(50),符合同步信号Sync拉起的条件。

[0093] 周期T5,同步信号Sync相应于比较信号AddrEqual拉起而拉起,读取指针RdPtr移到下一个单元格,地址50、60、70(图中只标出起始地址50)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,当前周期完成分支预测的地址50、60、70中与前一周期不重叠的地址60、70填入该取指目标队列FTQ,指针WrPtr0以及WrPtr1订正为指向这两个单元格,其中起始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址60所占的单元格)。此外在T5,分支预测用自增地址B0、C0、D0(图中只标出起始地址B0)输入分支预测器110,并行进行C级流水线运算。由于同步信号Sync拉起,该取指目标队列FTQ由异步模式切换回同步模式,即指令高速缓存104切换回自该取指目标队列FTQ取得取指地址AddrL1i。根据读取指针RdPtr,该取指目标队列FTQ弹出地址60为取指地址AddrL1i,输入该指令高速缓存104做C级流水线运算。因为已经切换回同步模式,指针WrPtr以及RdPtr的比较无须再做。

[0094] 周期T6,读取指针RdPtr移到下一个单元格。地址70、80、90(图中只标出起始地址70)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,当前周期完成分支预测的地址70、80、90中与前一周期不重叠的地址80、90填入,指针WrPtr0以及WrPtr1订正为指向这两个单元格,其中起始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址80所占的单元格)。特别是,此时指针WrPtr0已经指向该取指目标队列FTQ最后一个单元格,则指针WrPtr1是指向该取指目标队列FTQ中被之前所取的无用地地址占用的单元格(图5A所示的实施例是地址90覆盖无

用地址所占的单元格中第一个单元格,其中读取指针RdPtr所指向单元格之前的单元格中的地址均系无用地址)。此外在T6,分支预测用自增地址D0、E0、F0(图中只标出起始地址D0)输入分支预测器110,并行进行C级流水线运算。由于T6同步信号Sync维持拉起,表明为同步模式,则根据读取指针RdPtr,该取指目标队列FTQ弹出地址70为取指地址AddrL1i,输入该指令高速缓存104做C级流水线运算。

[0095] 周期T7,读取指针RdPtr移到下一个单元格,地址90、A0、B0(图中只标出起始地址90)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,当前周期完成分支预测的地址90、A0、B0中与前一周期不重叠的地址A0、B0填入,指针WrPtr0以及WrPtr1订正为指向这两个单元格,其中起始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址A0所占的单元格)。分支预测用自增地址F0、G0、H0(图中只标出起始地址F0)输入分支预测器110,并行进行C级流水线运算。由于T7同步信号Sync维持拉起,表明为同步模式,则根据读取指针RdPtr,该取指目标队列FTQ弹出地址80为取指地址AddrL1i,输入该指令高速缓存104做C级流水线运算。

[0096] 图5A中,刷新事件使得该取指目标队列FTQ在周期T0...T4准备不及,来不及在指令高速缓存104使用该取指地址AddrL1i前备妥相应地址。但是,这个状况完全不延误该指令高速缓存104运作。刷新后的地址10、20、30、40、50还是由其他管道(120/122、118)逐周期供应至该指令高速缓存104作为该取指地址AddrL1i。

[0097] 图5B图解图5A周期T5若预测到地址60至地址200跳转(但此时该取指目标队列FTQ并不为空),同步信号Sync将如何变化。

[0098] 相较于图5A,由于地址70将会被地址60处的跳转避开,图5B周期T5不会将地址70推入该取指目标队列FTQ。跳转目标地址200、210、220(图中只标出起始地址200)输入分支预测器110,并行进行C级流水线运算。

[0099] 周期T6,由于在前一周T5预测到跳转且同时该取指目标队列FTQ中没有可供弹出的指令地址(因为地址60及其之前的地址都在读取指针RdPtr之后,说明已经被弹出过),则同步信号Sync拉下,读取指针RdPtr与起始写入指针WrPtr0均订正为指向该取指目标队列FTQ中已存入指令地址AddrBP的单元格后接续的空白单元格(这里是指向60所占单元格后接续的空白单元格)。地址70、80、90的分支预测进行到U级流水线运算(即完成分支预测),但无意义,因已跳转避开,不会推入该取指目标队列FTQ。此外在T6,分支预测用自增地址220、230、240(图中只标出起始地址220)输入分支预测器110,并行进行C级流水线运算。同时在T6,由于同步信号Sync拉下进入异步模式,跳转目标地址200不经该取指目标队列FTQ直接输入该指令高速缓存104做C级流水线运算。

[0100] 周期T7,读取指针RdPtr移到下一个单元格。地址90、A0、B0的分支预测进行到U级流水线运算(即完成分支预测),但无意义,因已跳转避开,不会推入该取指目标队列FTQ。分支预测用自增地址240、250、260(图中只标出起始地址240)输入分支预测器110,并行进行C级流水线运算。由于处于异步模式,以指令高速缓存用的自增地址210输入该指令高速缓存104做C级流水线运算。

[0101] 周期T8,读取指针RdPtr移到下一个单元格。地址200、210、220(图中只标出起始地址200)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,地址200、210、220填入,指针WrPtr0、WrPtr1以及WrPtr2订正为指向这三个单元格,其中起

始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址200所占的单元格)。此外在T8,分支预测用自增地址260、270、280(图中只标出起始地址260)输入分支预测器110,并行进行C级流水线运算。同时在T8,由于处于异步模式,指令高速缓存用的自增地址220也输入该指令高速缓存104做C级流水线运算。值得注意的是,在当前周期T8,读取指针RdPtr与并行指针之一WrPtr2相等。比较信号AddrEqual拉起。所指单元格确实载有地址(220),符合同步信号Sync拉起的条件。

[0102] 周期T9,同步信号Sync相应于比较信号AddrEqual拉起而拉起,读取指针RdPtr移到下一个单元格,地址220、230、240(图中只标出起始地址220)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,当前周期完成分支预测的地址220、230、240中与前一周周期不重叠的地址230、240填入,指针WrPtr0以及WrPtr1订正为指向这两个单元格,其中起始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址230所占的单元格)。此外在T9,分支预测用自增地址280、290、300(图中只标出起始地址280)输入分支预测器110,并行进行C级流水线运算。由于同步信号Sync拉起,该取指目标队列FTQ由异步模式切换回同步模式,即指令高速缓存104切换回自该取指目标队列FTQ取得取指地址AddrL1i。根据读取指针RdPtr,该取指目标队列FTQ弹出地址230为取指地址AddrL1i,输入该指令高速缓存104做C级流水线运算。因为已经切换回同步模式,指针WrPtr以及RdPtr的比较无须再做。

[0103] 图5B中,周期T5预测到的跳转事件使得该取指目标队列FTQ在周期T6...T8准备不及,来不及在指令高速缓存104使用该取指地址AddrL1i前备妥相应地址。但是,这个状况完全不延误该指令高速缓存104运作。跳转后的地址200、210、220还是由其他管道(绕过该取指目标队列FTQ直接经AddrBP传来的114、或指令高速缓存104提供的118)逐周期供应至该指令高速缓存104作为该取指地址AddrL1i。

[0104] 图5C根据本申请一种实施方式图解刷新后,又预测出一跳转目标地址时,同步信号Sync何时拉起。

[0105] 相较图5A,图5C在周期T3预测到地址10分支跳转到地址200。分支预测器110在T3预测完毕的地址包括10、20、30(图中只标出起始地址10)。因应地址10跳转到地址200的预测,地址20、30会被跳转避开而不会推入该取指目标队列FTQ而只有地址10填入。跳转目标地址200、210、220(图中只标出起始地址200)则是输入分支预测器110并行进行C级流水线运算。

[0106] 周期T4,读取指针RdPtr与起始写入指针WrPtr0均订正为指向该取指目标队列FTQ中已存入指令地址AddrBP的单元格后接续的空白单元格(这里是指向10所占单元格后接续的空白单元格)。地址30、40、50的分支预测进行到U级流水线运算(即完成分支预测),但无意义,因已跳转避开,不会推入该取指目标队列FTQ。此外在T4,分支预测用自增地址220、230、240(图中只标出起始地址220)输入分支预测器110,并行进行C级流水线运算。同时在T4,由于处于异步模式,跳转目标地址200不经该取指目标队列FTQ直接输入该指令高速缓存104做C级流水线运算。比较信号AddrEqual拉起。但是,起始写入指针WrPtr0所指单元格是空的,不当被视为并行写入指针,不符合切换至同步模式的条件。

[0107] 周期T5,读取指针RdPtr移到下一个单元格。地址50、60、70的分支预测进行到U级流水线运算(即完成分支预测),但无意义,因已跳转避开,亦不会推入该取指目标队列FTQ。

分支预测用自增地址240、250、260(图中只标出起始地址240)输入分支预测器110,并行进行C级流水线运算。由于处于异步模式,以指令高速缓存用的自增地址210输入该指令高速缓存104做C级流水线运算。

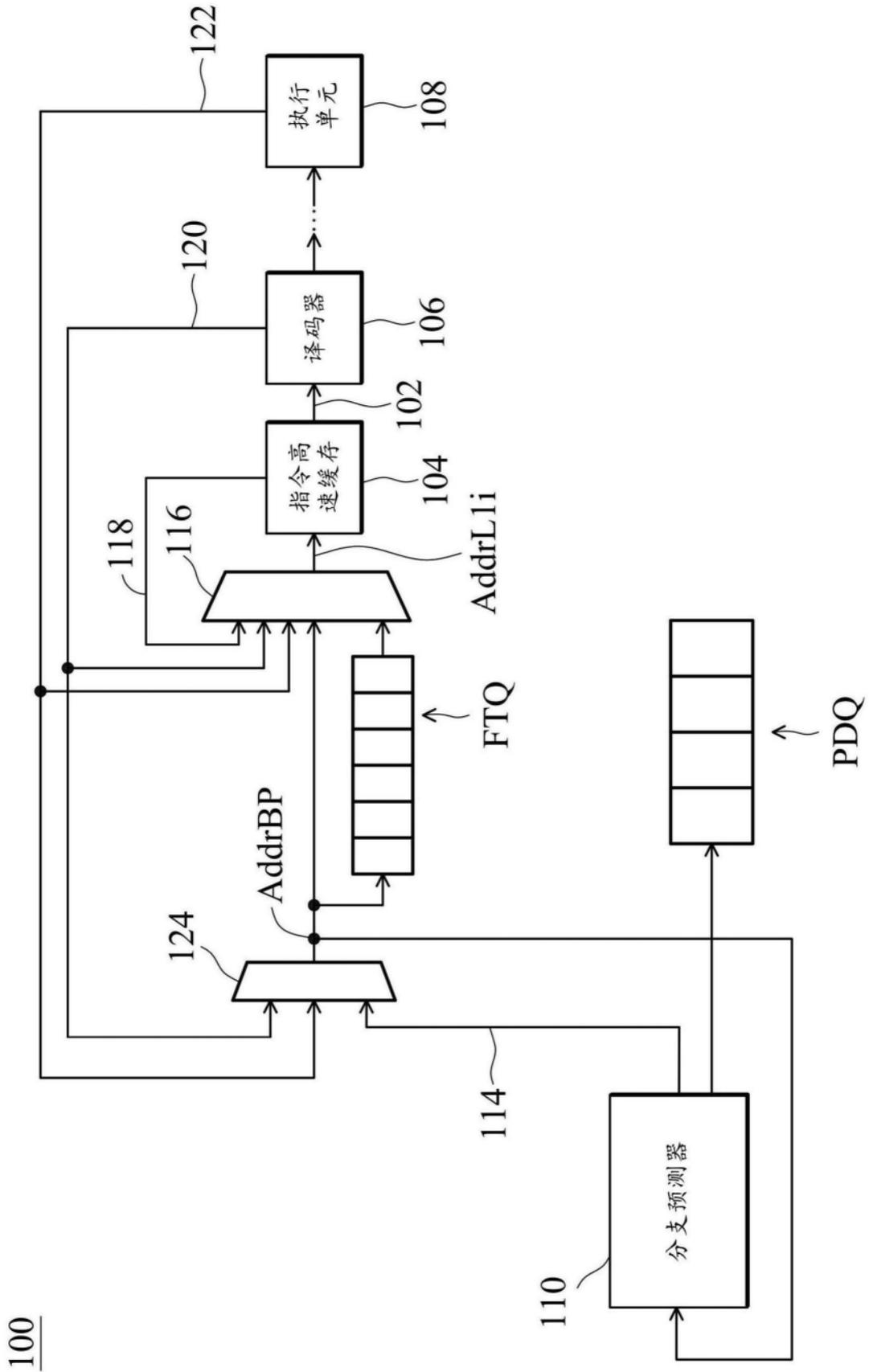
[0108] 周期T6,读取指针RdPtr移到下一个单元格。地址200、210、220(图中只标出起始地址200)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,地址200、210、220填入,指针WrPtr0、WrPtr1以及WrPtr2订正为指向这三个单元格,其中起始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址200所占的单元格)。此外在T6,分支预测用自增地址260、270、280(图中只标出起始地址260)输入分支预测器110,并行进行C级流水线运算。同时在T6,由于处于异步模式,指令高速缓存用的自增地址220也输入该指令高速缓存104做C级流水线运算。值得注意的是,在当前周期T6,读取指针RdPtr与并行写入指针之一WrPtr2相等。比较信号AddrEqual拉起。所指单元格确实载有地址(220),符合同步信号Sync拉起的条件。

[0109] 周期T7,同步信号Sync相应于比较信号AddrEqual拉起而拉起,读取指针RdPtr移到下一个单元格,地址220、230、240(图中只标出起始地址220)的分支预测进行到U级流水线运算(即完成分支预测),且没有跳转发生。根据图4A,当前周期完成分支预测的地址220、230、240中与前一周期不重叠的地址230、240填入,指针WrPtr0以及WrPtr1订正为指向这两个单元格,其中起始写入指针WrPtr0订正为指向当前周期推入该取指目标队列FTQ的指令地址所占第一个单元格(这里即地址230所占的单元格)。此外在T7,分支预测用自增地址280、290、300(图中只标出起始地址280)输入分支预测器110,并行进行C级流水线运算。由于同步信号Sync拉起,该取指目标队列FTQ由异步模式切换回同步模式,即指令高速缓存104切换回自该取指目标队列FTQ取得取指地址AddrL1i。根据读取指针RdPtr,该取指目标队列FTQ弹出地址230为取指地址AddrL1i,输入该指令高速缓存104做C级流水线运算。因为已经切换回同步模式,指针WrPtr以及RdPtr的比较无须再做。

[0110] 图5C中,该取指目标队列FTQ到周期T7才为指令高速缓存104备妥地址作为该取指地址AddrL1i。但是,这个状况完全不延误该指令高速缓存104运作。刷新地址10、以及其跳转目标地址200、和其后地址210、220都还是由其他管道快速供应至该指令高速缓存104作为该取指地址AddrL1i。指令高速缓存104虽然还是耗费资源在无意义地址20、30、40的取指上,仍然算是高效能设计。

[0111] 无论是刷新事件,或是预测出的分支跳转事件(即预测出跳转目标地址),解耦该指令高速缓存104以及该分支预测器110的该取指目标队列FTQ都展现相当功效。凡是使用所述取指目标队列FTQ的微处理器架构,都属于本申请范畴。

[0112] 虽然本发明已以较佳实施例公开如上,然其并非用以限定本发明,任何本领域技术人员,在不脱离本发明的精神和范围内,当可做些许更动与润饰,因此本发明的保护范围应以权利要求所保护的范围为准。



100

图1

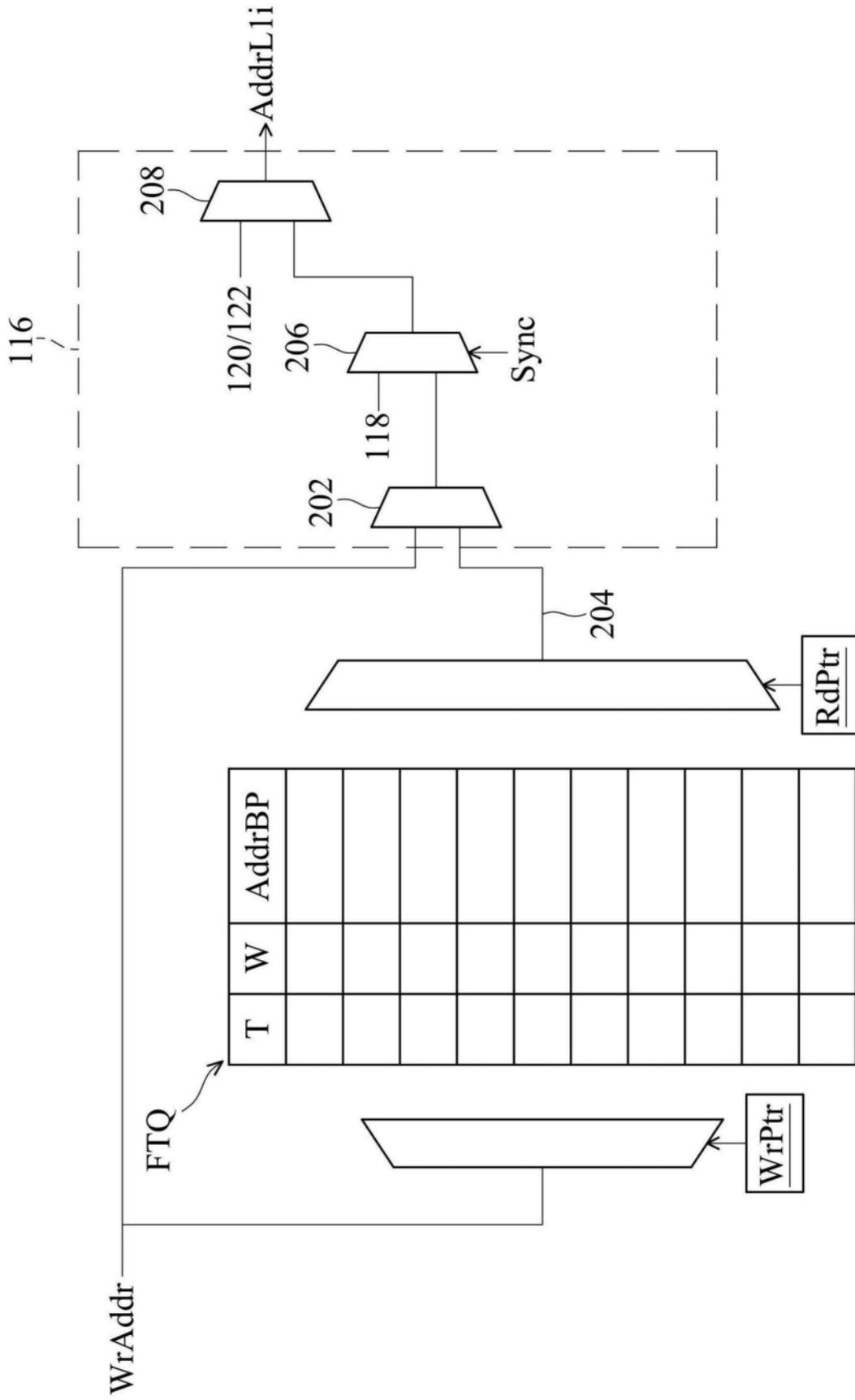


图2

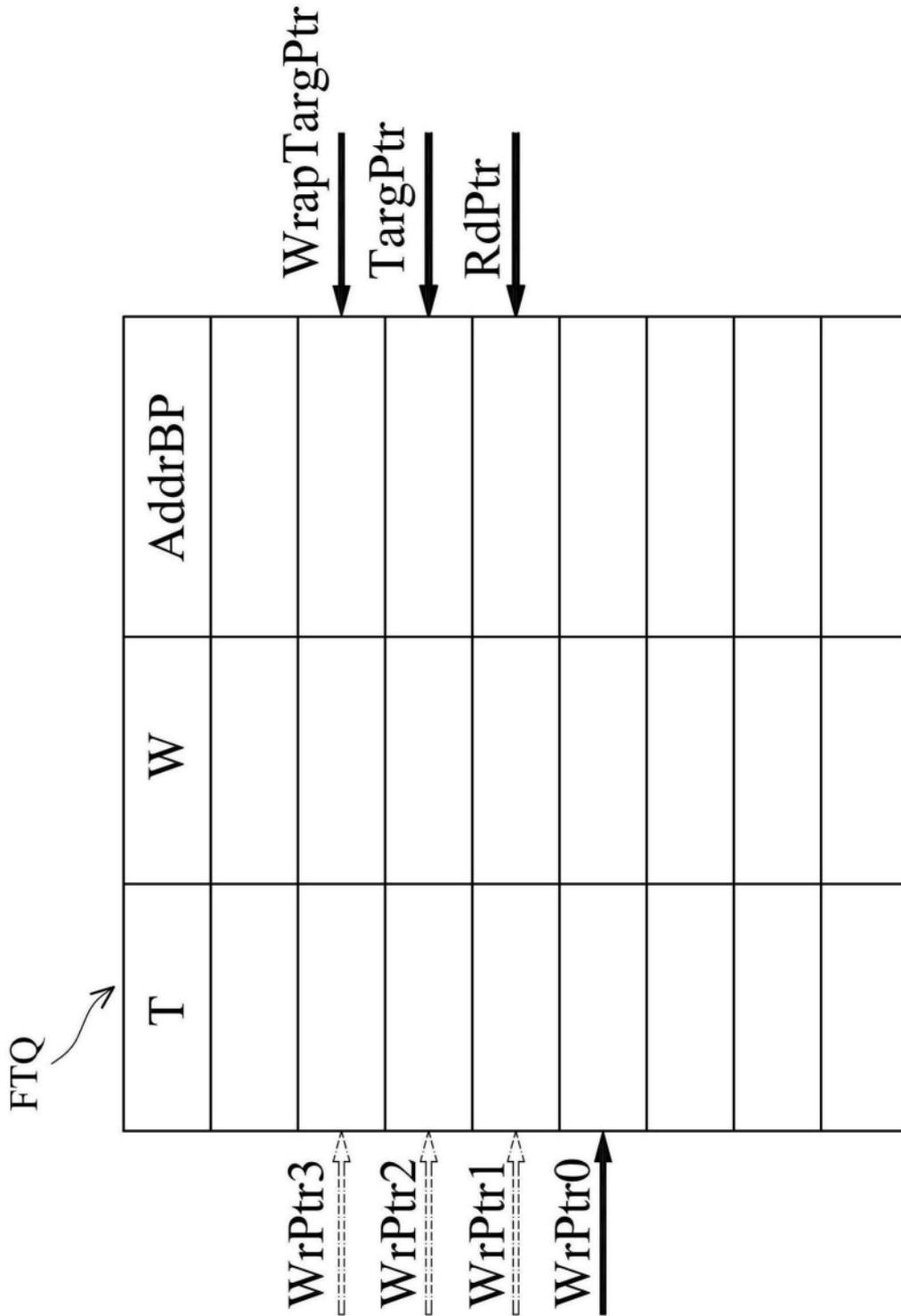


图3

410

		写入条件	AddrBP
WrPtr0	afterbr	无额外条件	PC
	~afterbr		PC+16
WrPtr1	afterbr	PC不跳转	PC+16
	~afterbr	PC取跳转、且其中取跳转的分支指令跨取指单位	
			PC+16不跳转
WrPtr2	afterbr	PC+16取跳转、且其中取跳转的分支指令跨取指单位	PC+32
	~afterbr	PC不跳转、PC+16不跳转	
			PC不跳转、但PC+16取跳转、且其中取跳转的分支指令跨取指单位
WrPtr3	afterbr	PC+16不跳转、但PC+32取跳转、且其中取跳转的分支指令跨取指单位	PC+48
	~afterbr	PC与PC+16不跳转、但PC+32取跳转、且其中取跳转的分支指令跨取指单位	PC+48

图4A

		写入条件	AddrBP
WrPtr0	afterbr	无额外条件	PC
	~afterbr		PC+32
WrPtr1	afterbr	PC不跳转	PC+16
	~afterbr	PC取跳转、且其中取跳转的分支指令跨取指令单位	PC+48
WrPtr2	afterbr	PC不跳转、PC+16不跳转	PC+32
	~afterbr	PC不跳转、但PC+16取跳转、且其中取跳转的分支指令跨取指令单位	PC+48
WrPtr3	afterbr	PC与PC+16不跳转、但PC+32取跳转、且其中取跳转的分支指令跨取指令单位	PC+48

420

图4B

	写入条件	AddrBP
WrPtr0	无额外条件	PC
WrPtr1	PC不跳转	PC+16
	PC取跳转、且其中取跳转的分支指令跨取指单位	
WrPtr2	PC不跳转、PC+16不跳转	PC+32
	PC不跳转、但PC+16取跳转、且其中取跳转的分支指令跨取指单位	
WrPtr3	PC与PC+16不跳转、但PC+32取跳转、且其中取跳转的分支指令跨取指单位	PC+48

430

图4C

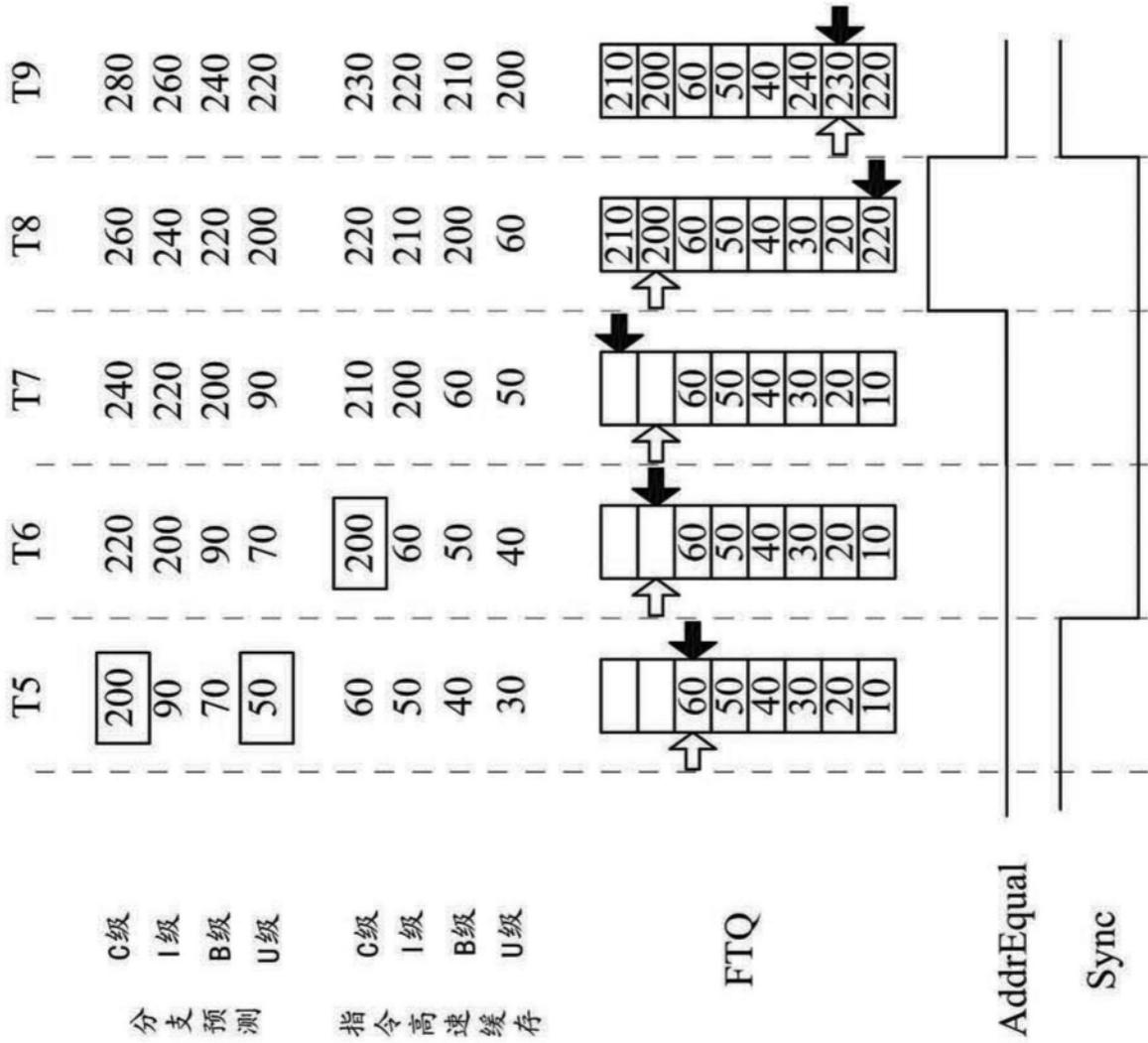


图5B

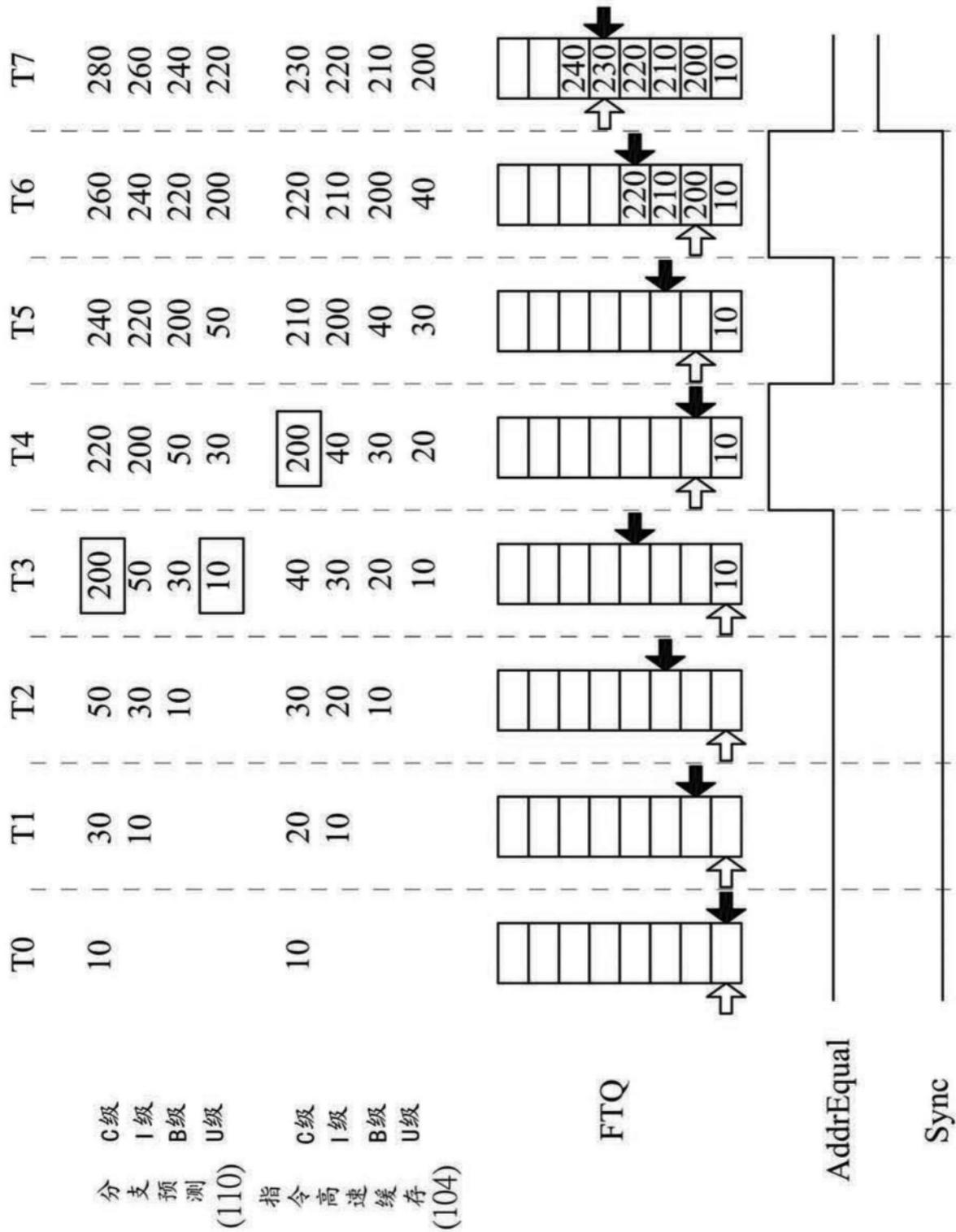


图5C