



(12) 发明专利申请

(10) 申请公布号 CN 104111818 A

(43) 申请公布日 2014. 10. 22

(21) 申请号 201410163208. 5

(22) 申请日 2014. 04. 22

(30) 优先权数据

10-2013-0044435 2013. 04. 22 KR

(71) 申请人 三星电子株式会社

地址 韩国京畿道水原市

(72) 发明人 郑武垆 柳秀晶 赵渊坤

(74) 专利代理机构 北京铭硕知识产权代理有限公司 11286

代理人 王兆庚 韩明星

(51) Int. Cl.

G06F 9/38 (2006. 01)

G06F 9/30 (2006. 01)

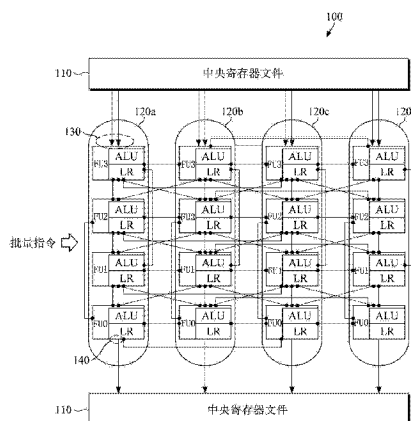
权利要求书3页 说明书9页 附图11页

(54) 发明名称

用于批量线程处理的处理器、处理方法和代码生成设备

(57) 摘要

公开用于批量线程处理的处理器、处理方法和代码生成设备。用于批量线程处理的处理器包括：中央寄存器文件；一个或多个批量功能单元，每个批量功能单元包括两个或多个功能单元以及一个或多个用于访问中央寄存器文件的端口。批量功能单元中的功能单元执行包括一个或多个指令的批量指令，以顺序地执行批量指令中的一个或多个指令。



1. 一种处理器,包括:

中央寄存器文件;

第一批量功能单元,包括第一多个功能单元、第一输入端口和第一输出端口,其中,第一多个功能单元通过第一输入端口访问中央寄存器文件,并且第一多个功能单元通过第一输出端口访问中央寄存器文件;

第二批量功能单元,包括第二多个功能单元、第二输入端口和第二输出端口,其中,第二多个功能单元通过第二输入端口访问中央寄存器文件,并且第二多个功能单元通过第二输出端口访问中央寄存器文件,

其中,第一批量功能单元被配置为接收包括程序的一个或更多个第一指令的第一批量指令,并且顺序地执行所述一个或更多个第一指令,第二批量功能单元被配置为接收包括所述程序的一个或更多个第二指令的第二批量指令,并且顺序地执行所述一个或更多个第二指令。

2. 如权利要求 1 所述的处理器,其中,第一批量功能单元还包括一个或更多个第一本地寄存器文件,第一本地寄存器文件被配置为存储第一多个功能单元的输入/输出数据,

其中,第二批量功能单元还包括一个或更多个第二本地寄存器文件,第二本地寄存器文件被配置为存储第二多个功能单元的输入/输出数据。

3. 如权利要求 2 所述的处理器,其中,第一批量功能单元被配置为通过使用第一多个功能单元、第一多个功能单元之间的连接以及所述一个或更多个第一本地寄存器文件而作为粗粒度可重构阵列 (CGRA) 操作,

其中,第二批量功能单元被配置为通过使用第二多个功能单元、第二多个功能单元之间的连接以及所述一个或更多个第二本地寄存器文件而作为 CGRA 操作。

4. 如权利要求 1 所述的处理器,其中,第一批量功能单元的结构与第二批量功能单元的结构相同。

5. 如权利要求 1 所述的处理器,其中,第一多个功能单元被配置为处理所述一个或更多个第一指令,

其中,第二多个功能单元被配置为处理所述一个或更多个第二指令。

6. 如权利要求 1 所述的处理器,其中,第一批量功能单元被配置为在特定周期期间通过使用偏移指令信息执行所述一个或更多个第二指令中的至少一个指令,

第二批量功能单元被配置为在特定周期期间通过使用偏移指令信息执行所述一个或更多个第一指令中的至少一个指令。

7. 如权利要求 1 所述的处理器,其中,第一批量指令包括第一多个批量指令,第二批量指令包括第二多个批量指令,

其中,第一批量功能单元被配置为当接收到第一多个批量指令时,以包括一个或更多个线程的线程组为单位顺序地执行第一多个批量指令中的每个批量指令,并且第二批量功能单元被配置为当接收到第二多个批量指令时,以线程组为单位顺序地执行第二多个批量指令中的每个批量指令。

8. 如权利要求 7 所述的处理器,其中,第一批量功能单元和第二批量功能单元被配置为:如果在针对批量指令执行线程组期间在特定线程发生阻塞,并且阻塞持续到针对具有对所述批量指令的依赖性的另一批量指令执行所述线程组的时刻,则针对所述另一批量指

令以所述线程组中的最后次序执行所述发生阻塞的特定线程。

9. 如权利要求 7 所述的处理器,其中,第一批量功能单元和第二批量功能单元被配置为:如果在针对批量指令执行线程组期间发生条件分支,则将所述线程组划分为两个或更多个子线程组,并且针对条件分支的多个分支分别执行划分的两个或更多个子线程组。

10. 如权利要求 9 所述的处理器,其中,第一批量功能单元和第二批量功能单元被配置为:如果条件分支的多个分支结束并且合并,则将划分的两个或更多个子线程组合并为所述线程组并执行所述线程组。

11. 一种处理器,包括:

中央寄存器文件;

第一批量功能单元,包括第一多个功能单元、第一输入端口和第一输出端口,其中,第一多个功能单元通过第一输入端口访问中央寄存器文件,并且第一多个功能单元通过第一输出端口访问中央寄存器文件;

第二批量功能单元,包括第二多个功能单元、第二输入端口和第二输出端口,其中,第二多个功能单元通过第二输入端口访问中央寄存器文件,并且第二多个功能单元通过第二输出端口访问中央寄存器文件;

多个偏移寄存器,被分配给第一多个功能单元和第二多个功能单元中的每个功能单元,

其中,使用存储在批量指令存储器中的指令,通过多个偏移寄存器中的一个生成在特定周期期间将被执行的偏移指令,并且将生成的偏移指令发送到分配了多个偏移寄存器中的一个的每个功能单元。

12. 如权利要求 11 所述的处理器,其中,批量指令存储器被配置为被分为两个或更多个单元以对应于第一多个功能单元和第二多个功能单元中的每个功能单元,并存储将被发送到与批量指令存储器对应的功能单元的指令。

13. 如权利要求 11 所述的处理器,还包括:一个或更多个内核队列,存储从批量指令存储器的内核提取至少部分指令,

其中,使用存储在每个内核队列中的指令,通过偏移寄存器生成在特定周期期间将被执行的偏移指令,并且将生成的偏移指令发送到每个分配的功能单元。

14. 一种用于生成代码的设备,所述设备包括:

程序分析单元,被配置为分析将在包括第一批量功能单元和第二批量功能单元的处理器中处理的预定程序,其中,第一批量功能单元包括第一多个功能单元,第二批量功能单元包括第二多个功能单元;

批量指令生成单元,被配置为基于分析结果生成第一批量指令和第二批量指令,第一批量指令和第二批量指令中的每个包括将分别在第一批量功能单元和第二批量功能单元中执行的一个或更多个指令。

15. 如权利要求 14 所述的设备,其中,如果作为分析结果在程序中存在条件分支语句,则批量指令生成单元允许处理条件分支语句的多个分支的指令被包括在不同的批量指令中。

16. 如权利要求 14 所述的设备,其中,批量指令生成单元生成彼此之间具有相似的延迟的第一批量指令和第二批量指令。

17. 如权利要求 14 所述的设备,其中,批量指令生成单元基于将要执行第一批量指令和第二批量指令的第一批量功能单元或第二批量功能单元的读取端口的数量和写入端口的数量生成第一批量指令和第二批量指令。

18. 如权利要求 17 所述的设备,其中,批量指令生成单元生成第一批量指令和第二批量指令,以防止第一批量指令和第二批量指令针对中央寄存器文件的读取请求的数量和写入请求的数量超过将要执行第一批量指令和第二批量指令的第一批量功能单元或第二批量功能单元的读取端口的数量和写入端口的数量。

19. 如权利要求 14 所述的设备,其中,批量指令生成单元生成第一批量指令和第二批量指令,以防止包括在每个批量指令中的指令的数量超过包括在将要执行第一批量指令和第二批量指令的第一批量功能单元或第二批量功能单元中功能单元的数量。

20. 如权利要求 14 所述的设备,其中,批量指令生成单元生成第一批量指令和第二批量指令,以防止特定批量指令中的延迟被用作所述特定批量指令中的源。

21. 一种通过处理器处理批量线程的方法,所述方法包括:

将由代码生成设备生成的第一批量指令和第二批量指令输入到包括第一多个功能单元的第一批量功能单元和包括第二多个功能单元的第二批量功能单元中;

分别通过第一批量功能单元和第二批量功能单元顺序地执行第一批量指令和第二批量指令。

22. 如权利要求 21 所述的方法,其中,在输入一个或更多个批量指令时,以线程组为单位输入第一批量指令和第二批量指令。

23. 如权利要求 22 所述的方法,其中,在执行第一批量指令和第二批量指令时,在以交错方式切换包括在线程组中的每个线程的同时,针对每个批量指令执行线程组。

24. 如权利要求 22 所述的方法,其中,在执行第一批量指令和第二批量指令时,如果在针对批量指令执行线程组期间在特定线程发生阻塞,并且阻塞持续到针对具有对所述批量指令的依赖性的另一批量指令执行所述线程组的时刻,则针对所述另一批量指令以所述线程组中的最后次序执行所述发生阻塞的特定线程。

25. 如权利要求 22 所述的方法,其中,在执行第一批量指令和第二批量指令时,如果在针对批量指令执行线程组期间发生条件分支,则将所述线程组划分为两个或更多个子线程组,并且针对条件分支的多个分支分别执行划分的两个或更多个子线程组。

26. 如权利要求 25 所述的方法,其中,在执行第一批量指令和第二批量指令时,如果条件分支的多个分支结束并且合并,则将划分的两个或更多个子线程组合并为所述线程组并执行所述线程组。

用于批量线程处理的处理器、处理方法和代码生成设备

[0001] 本申请要求于 2013 年 4 月 22 日提交的第 10-2013-0044435 号韩国专利申请的优先权,其公开通过引用完整地包含于此。

技术领域

[0002] 与示例性实施例一致的方法和设备涉及一种用于批量线程处理的处理器、使用所述处理器的批量线程处理方法和用于支持批量线程处理器的代码生成设备。

背景技术

[0003] 粗粒度可重构阵列 (CGRA) 是指具有阵列形式的多个功能单元 (FU) 的硬件。CGRA 用于加速操作。此外,即使在数据之间存在依赖性,CGRA 也可通过使用软件流水线技术来最大化吞吐量。然而,在编译阶段执行数据处理的所有调度,这增加了编译时间。此外,当执行多线程时,硬件开销很大,使得在执行具有大的、未定义的延迟时间的操作(例如,存储器访问)时操作效率降低。

[0004] 作为另一示例,单指令多线程 (SIMT) 是与 CGRA 相似的包括多个功能单元的架构。SIMT 在执行由每个功能单元处理的单个线程的同时使用由多个功能单元处理的单个指令。根据 SIMT,多个功能单元以相同的指令顺序处理大量数据,从而使需要在同一处理中处理大量数据的大规模并行数据处理应用受益。此外,当操作延迟高时,SIMT 通过执行不同线程的线程切换来提供优越的处理效率。然而,在每个数据之间的依赖性存在于数据处理中的情况下,线程处理非常困难。

发明内容

[0005] 根据示例性实施例的一方面,一种处理器可包括:中央寄存器文件;第一批量功能单元,包括第一多个功能单元、第一输入端口和第一输出端口,其中,第一多个功能单元通过第一输入端口访问中央寄存器文件,并且第一多个功能单元通过第一输出端口访问中央寄存器文件;第二批量功能单元,包括第二多个功能单元、第二输入端口和第二输出端口,其中,第二多个功能单元通过第二输入端口访问中央寄存器文件,并且第二多个功能单元通过第二输出端口访问中央寄存器文件,其中,第一批量功能单元被配置为接收包括程序的一个或更多个第一指令的第一批量指令,并且顺序地执行所述一个或更多个第一指令,第二批量功能单元被配置为接收包括所述程序的一个或更多个第二指令的第二批量指令,并且顺序地执行所述一个或更多个第二指令。

[0006] 第一批量功能单元还可包括一个或更多个第一本地寄存器文件,第一本地寄存器文件被配置为存储第一多个功能单元的输入/输出数据,第二批量功能单元还可包括一个或更多个第二本地寄存器文件,第二本地寄存器文件被配置为存储第二多个功能单元的输入/输出数据。

[0007] 第一批量功能单元可被配置为通过使用第一多个功能单元、第一多个功能单元之间的连接以及所述一个或更多个第一本地寄存器文件而作为粗粒度可重构阵列 (CGRA) 操

作,第二批量功能单元可被配置为通过使用第二多个功能单元、第二多个功能单元之间的连接以及所述一个或更多个第二本地寄存器文件而作为 CGRA 操作。

[0008] 第一批量功能单元的结构可与第二批量功能单元的结构相同。

[0009] 第一多个功能单元可被配置为处理所述一个或更多个第一指令,第二多个功能单元可被配置为处理所述一个或更多个第二指令。

[0010] 第一批量功能单元可被配置为在特定周期期间通过使用偏移指令信息执行所述一个或更多个第二指令中的至少一个指令,第二批量功能单元可被配置为在特定周期期间通过使用偏移指令信息执行所述一个或更多个第一指令中的至少一个指令。

[0011] 第一批量指令可包括第一多个批量指令,第二批量指令可包括第二多个批量指令,第一批量功能单元可被配置为,当接收到第一多个批量指令时,以包括一个或更多个线程的线程组为单位顺序地执行第一多个批量指令中的每个批量指令,并且第二批量功能单元可被配置为,当接收到第二多个批量指令时,以线程组为单位顺序地执行第二多个批量指令中的每个批量指令。

[0012] 第一批量功能单元和第二批量功能单元可被配置为:如果在针对批量指令执行线程组期间在特定线程发生阻塞,并且阻塞持续到针对具有对所述批量指令的依赖性的另一批量指令执行所述线程组的时刻,则针对所述另一批量指令以所述线程组中的最后次序执行所述发生阻塞的特定线程。

[0013] 第一批量功能单元和第二批量功能单元可被配置为:如果在针对批量指令执行线程组期间发生条件分支,则将所述线程组划分为两个或更多个子线程组,并且针对条件分支的多个分支分别执行划分的两个或更多个子线程组。

[0014] 第一批量功能单元和第二批量功能单元可被配置为:如果条件分支的多个分支结束并且合并,则将划分的两个或更多个子线程组合并为所述线程组并执行所述线程组。

[0015] 根据示例性实施例的一方面,一种处理器可包括:第一批量功能单元,包括第一多个功能单元、第一输入端口和第一输出端口,其中,第一多个功能单元通过第一输入端口访问中央寄存器文件,并且第一多个功能单元通过第一输出端口访问中央寄存器文件;第二批量功能单元,包括第二多个功能单元、第二输入端口和第二输出端口,其中,第二多个功能单元通过第二输入端口访问中央寄存器文件,并且第二多个功能单元通过第二输出端口访问中央寄存器文件;多个偏移寄存器,被分配给第一多个功能单元和第二多个功能单元中的每个功能单元,其中,使用存储在批量指令存储器中的指令,通过多个偏移寄存器中的一个生成在特定周期期间将被执行的偏移指令,并且将生成的偏移指令发送到分配了多个偏移寄存器中的一个的每个功能单元。

[0016] 根据示例性实施例的另一方面,提供一种用于生成代码的设备,所述设备包括:程序分析单元,被配置为分析将在包括第一批量功能单元和第二批量功能单元的处理器中处理的预定程序,其中,第一批量功能单元包括第一多个功能单元,第二批量功能单元包括第二多个功能单元;批量指令生成单元,被配置为基于分析结果生成第一批量指令和第二批量指令,第一批量指令和第二批量指令中的每个包括将分别在第一批量功能单元和第二批量功能单元中执行的一个或更多个指令。

[0017] 根据示例性实施例的另一方面,提供一种通过处理器处理批量线程的方法,包括:将由代码生成设备生成的第一批量指令和第二批量指令输入到包括第一多个功能单元的

第一批量功能单元和包括第二多个功能单元的第二批量功能单元中；分别通过第一批量功能单元和第二批量功能单元顺序地执行第一批量指令和第二批量指令。

[0018] 通过以下详细描述、附图和权利要求，以上和其他方面将是清楚的。

附图说明

[0019] 图 1 是示出根据本公开的示例性实施例的处理器的示图。

[0020] 图 2 是根据示例性实施例的程序的流程图。

[0021] 图 3 是示出在一般的 SIMT 架构中执行图 2 的程序的过程的示图。

[0022] 图 4A、图 4B 和图 4C 是示出在一般的 CGRA 中执行图 2 的程序的过程的示图。

[0023] 图 5A 和图 5B 是示出在根据本公开的示例性实施例的处理器中执行图 2 的程序的过程的示图。

[0024] 图 6A 和图 6B 是示出在根据本公开的示例性实施例的处理器批量功能单元中输入的偏移指令 (skewed instruction) 的示图。

[0025] 图 7A 是示出根据本公开的另一示例性实施例的用于偏移指令的输入的处理器的示图。

[0026] 图 7B 是示出根据本公开的另一示例性实施例的用于偏移指令的输入的处理器的示图。

[0027] 图 8 是示出根据本公开的示例性实施例的支持用于批量线程处理的处理器的代码生成设备的框图。

[0028] 图 9 是示出根据本公开的示例性实施例的使用用于批量线程处理的处理器来处理批量线程的方法的流程图。

[0029] 贯穿附图和详细描述，除非另外描述，否则相同的附图标号将被理解为指示相同的元件、特征和结构。为了清楚、说明和方便，可能夸大这些元件的相对大小和描绘。

具体实施方式

[0030] 提供下面的描述以帮助读者获得对这里描述的方法、设备和 / 或系统的全面理解。因此，将向本领域的普通技术人员推荐这里描述的方法、设备和 / 或系统的各种改变、修改和等同物。此外，为了增加清楚和简洁，可省略公知功能和构造的描述。此外，以下描述的术语是考虑到本公开中的功能而定义的术语，并且可根据用户或操作者或常用实践而改变。因此，必须基于整个公开中的内容来进行定义。

[0031] 以下，将参照附图详细描述用于批量线程处理的处理器、使用所述处理器处理批量线程的方法和用于批量线程处理的代码生成设备。

[0032] 图 1 是示出根据本公开的示例性实施例的处理器的示图。

[0033] 参照图 1，根据本公开的示例性实施例的处理器 100 包括中央寄存器文件 110 以及一个或更多个批量功能单元 120a、120b、120c 和 120d。虽然为了便于描述各个批量功能单元 120a、120b、120c 和 120d 中的输入端口 130 和输出端口 140，将中央寄存器文件 110 分开示出在图 1 的上部和下部，但是处理器 100 不包括两个中央寄存器文件 110。

[0034] 批量功能单元 120a、120b、120c 和 120d 中的每个包括两个或更多个功能单元 FU0、FU1、FU2 和 FU3。批量功能单元 120a、120b、120c 和 120d 中的每个包括一个或更多个输入

端口 130 和一个或多个输出端口 140, 并且通过一个或多个输入端口 130 和一个或多个输出端口 140 访问中央寄存器文件 110。批量功能单元 120a、120b、120c 和 120d 可通过中央寄存器文件 110 彼此执行通信, 诸如数据共享。

[0035] 批量功能单元 120a、120b、120c 和 120d 中的每个可包括一个或多个本地寄存器文件 (LR)。本地寄存器文件 LR 可包括在一个或多个功能单元中, 并且用作用于功能单元的输入 / 输出数据的存储空间, 且以先进先出 (FIFO) 方式操作。

[0036] 根据本公开的示例性实施例的处理器 100 可通过使用包括在批量功能单元中的功能单元、功能单元之间的连接以及功能单元的本地寄存器文件 LR, 而作为粗粒度可重构阵列 (CGRA) 操作。此外, 处理器 100 可通过使用两个或多个批量功能单元 120a、120b、120c 和 120d 而作为单指令多线程 (SIMT) 操作, 其中, 批量功能单元 120a、120b、120c 和 120d 中的每个包括两个或多个功能单元 FU0、FU1、FU2 和 FU3。

[0037] 批量功能单元 120a、120b、120c 和 120d 可具有相同的结构。包括在批量功能单元 120a、120b、120c 和 120d 中的每个批量功能单元中的功能单元 FU0、FU1、FU2 和 FU3 可具有彼此不同的结构。然而, 包括在批量功能单元 120a、120b、120c 和 120d 中的每个批量功能单元中的功能单元 FU0、FU1、FU2 和 FU3 不需要具有彼此不同的结构, 根据需要, 两个或多个功能单元可具有相同的结构。

[0038] 例如, 批量功能单元 120a、120b、120c 和 120d 可包括具有同等的计算能力的功能单元 FU0、FU1、FU2 和 FU3。这里, 计算能力可表示由功能单元执行的操作 (例如, 加法、减法、乘法和除法), 并且批量功能单元 120a、120b、120c 和 120d 中的每个可通过包括执行这种操作的功能单元而具有相同的计算能力。这样, 根据本公开的示例性实施例的处理器 100 通过具有相同的计算能力的批量功能单元 120a、120b、120c 和 120d 而作为 SIMT 操作, 从而支持大规模并行数据线程处理。

[0039] 一般的处理器在每个功能单元的算术逻辑单元 (ALU) 具有一个或多个输入端口和一个或多个输出端口以访问中央寄存器文件, 但是根据本公开的示例性实施例的处理器 100 具有用于批量功能单元 120a、120b、120c 和 120d 中的每个批量处理单元的一个或多个输入端口 130 和一个或多个输出端口 140 以访问中央寄存器文件 110, 从而减少了当访问中央寄存器文件 110 时产生的开销, 并因此提高了处理器 100 的性能。

[0040] 例如, 如果设置有八个功能单元的一般的处理器具有用于每个功能单元的两个输入端口和一个输出端口, 则通过十六个输入端口和八个输出端口实现对中央寄存器文件的访问。相反, 如果根据本公开的示例性实施例的处理器 100 以两个批量功能单元中的每个包括四个功能单元的形式包括八个功能单元, 并且每个批量功能单元具有两个输入端口和一个输出端口, 则通过四个端口和两个输出端口实现对中央寄存器文件的访问, 从而减少了用于输入和输出操作的开销。

[0041] 批量功能单元 120a、120b、120c 和 120d 中的每个可执行一个或多个通过编译生成的批量指令。在这种情况下, 每个批量指令可包括一个或多个指令, 并且各个指令可通过与各个指令对应的功能单元被顺序地执行。

[0042] 同时, 批量功能单元 120a、120b、120c 和 120d 可以以线程组为单位执行一个或多个输入的批量指令, 其中, 每个线程组包括一个或多个线程。

[0043] 在这种情况下, 如果在批量功能单元 120a、120b、120c 和 120d 正针对特定的批量

指令执行预定线程组的同时在特定的线程发生阻塞,并且直到针对具有对发生阻塞的所述批量指令的依赖性的另一批量指令执行所述预定线程组的线程才解除阻塞,则批量功能单元 120a、120b、120c 和 120d 针对所述另一批量指令不执行发生阻塞的线程,从而使发生阻塞的线程在结束所述预定线程组的所有线程的执行的最后时间点才被执行。

[0044] 以这种方式,防止连接到在执行批量指令时被阻塞的线程的所有线程被阻塞,从而提高了处理效率。

[0045] 如果在批量功能单元 120a、120b、120c 和 120d 正针对特定的批量指令执行线程组的同时发生条件分支,则将所述线程组划分为两个或更多个子线程组,并且分别针对条件分支的多个分支执行划分的子线程组。此外,如果针对各个分支的条件分支结束并且分支合并,则划分的子线程组被合并为原始的线程组。

[0046] 图 2 是在根据图 1 的示例性实施例的处理器 100 中执行处理批量线程的过程的程序的流程图。在图 2 中,在彼此具有预定的数据依赖性的同时执行十一个指令 A 至 K,从而在一个指令的执行之后执行具有对所述指令的数据依赖性的另一指令。在这种情况下,延迟表示执行每个指令 A 至 K 所花费的周期 (cycle)。

[0047] 图 3 是示出在一般的 SIMT 架构中执行图 2 的的过程的示图。当通过不同线程分别处理 128 条数据时,需要处理总共 128 个线程。可以假设具有八个 ALU (ALU0 至 ALU7) 的一般的 SIMT 将 128 个线程划分为分别包括 32 个线程的四个线程组,并且允许针对总共 11 个指令 A 至 K 执行所述四个线程组。在这种情况下,如图 3 所示,为了使一般的 SIMT 平稳地操作,每个指令 A 至 K 被设置为具有 4 个周期的延迟,并且 SIMT 针对指令 A 至 K 从指令 A 开始到指令 K 顺序地执行四个线程组。花费总共 180 个周期。

[0048] 图 4A 至图 4C 是示出在一般的 CGRA 中执行图 2 的的过程的示图。图 4A 示出与图 3 的 SIMT 具有相同数量的功能单元的一般的 CGRA 的示例,并且指令从配置存储器或高速缓冲存储器 CMEM 被输入。图 4B 示出被调度以在图 4A 的 CGRA 中执行的图 2 的程序的示例。图 4C 示出执行如图 4B 所示调度的 11 个指令 A 至 K 的示例。

[0049] 在这种情况下,CGRA 的迭代对应于 SIMT 的线程,并且执行 128 个迭代以处理如图 3 所描述的总共 128 个线程。参照图 4B,为了针对 11 个指令 A 至 K 执行一个迭代,需要 16 个周期的延迟。参照图 4C,当在具有启动间隔 (II) 2 的同时执行总共 128 个迭代时,需要总共 272 个周期。

[0050] 图 5A 和图 5B 是示出在根据本公开的示例性实施例的处理器 100 中执行图 2 的的过程的示图。

[0051] 图 5A 示出在编译阶段生成的三个批量指令,以使图 2 的示例性程序在处理器 100 中被执行,其中,批量指令 0 包括四个指令 A、B、D 和 E,批量指令 1 包括四个指令 C、F、G 和 H,并且批量指令 2 包括最后的三个指令 I、J 和 K。

[0052] 图 5B 示出在处理器 100 具有分别包括四个功能单元的两个批量功能单元的情况下由批量功能单元中的一个顺序执行的三个批量指令。批量指令中的每个指令由批量功能单元中的各个功能单元执行。在批量功能单元中,通过与本地寄存器文件的互连来实现批量指令内的数据移动,并且通过中央寄存器文件 100 来实现批量指令之间的数据移动。

[0053] 当与图 3 中示出的内容类似地处理总共 128 个线程时,两个批量功能单元中的每个针对三个批量指令执行 64 个线程,从而需要总共 202 个周期。例如,当以 16 个线程为单

位调度 128 个线程时,批量功能单元中的一个在以交错方案切换 16 个线程的同时顺序地执行三个指令。也就是说,针对 16 个线程执行输入的批量指令,针对所述 16 个线程执行下一个批量指令,并且还针对所述 16 个线程执行最后的批量指令。其后,从第一批量指令开始,针对 16 个新的线程再次执行三个批量指令,以这种方式处理所有线程。在这种情况下,通过两个批量功能单元执行 128 个线程的执行,并且需要总共 202 个周期。

[0054] 图 6A 和图 6B 是示出在批量功能单元中输入的偏移指令 (skewed instruction) 的示图。参照图 6A 和图 6B,利用根据本公开的示例性实施例的处理器 200,当执行一个或更多个输入到每个批量功能单元中的批量指令时,每个批量功能单元作为 CGRA 操作,从而每个批量指令中的指令相对于时间以交叉的方式被输入到功能单元中。这里,如以下所讨论的,因为由一个批量功能单元执行的批量指令随时间改变,所以指令可以是偏移指令。

[0055] 参照图 6A,以如下顺序改变批量指令:A-B-D-E(周期 10)、C-B-D-E(周期 17)、C-F-D-E(周期 21)、C-F-G-E(周期 25)和 C-F-G-H(周期 26)。在 A-B-D-E 和 C-F-G-H 是批量指令的情况下,以在这两个批量指令之间插入偏移指令的形式输入三个偏移指令。因此,在批量功能单元中可(以流水线形式)进行连续的操作。换句话说,偏移指令可以使批量功能单元中的功能单元连续地形成流水线并且进行操作。在偏移指令的特定示例中,在周期 17 的情况下,四个指令 C、B、D 和 E 被输入到批量功能单元中的四个功能单元。然而,如图 5A 所示,指令 C 被包括在批量指令 1 中,而其余的指令 B、D 和 E 属于批量指令 0。这样,当在同一周期输入的指令中的至少一个属于不同的批量指令时,在所述周期输入的指令被称为偏移指令,并且处理器 100 会需要偏移指令信息以精确地将偏移指令输入到各个批量功能单元中。这样,一个批量功能单元可以在特定周期期间通过使用偏移指令信息执行一个或更多个第二指令中的至少一个指令,而另一个批量功能单元可以在特定周期期间通过使用偏移指令信息执行一个或更多个第一指令中的至少一个指令。

[0056] 这种偏移指令信息可以在编译阶段由代码生成设备生成。处理器 200 使用偏移指令信息通过每个批量功能单元的程序计数器 (PC) 来访问批量指令存储器 (BIM),以将对应的指令发送到批量功能单元中的对应的功能单元。

[0057] 图 7A 是示出根据本公开的另一示例性实施例的用于偏移指令的输入的处理器的示图。

[0058] 参照图 7A,处理器 300 可包括中央寄存器文件(未示出)、一个或更多个批量功能单元和一个或更多个偏移寄存器 310,其中,每个批量功能单元包括两个或更多个功能单元,一个或更多个偏移寄存器 310 被分配给包括在每个批量功能单元中的功能单元。

[0059] 根据本公开的示例性实施例的处理器 300 包括与功能单元对应的偏移寄存器 310,以更有效地处理上述的偏移指令的输入。处理器 300 可使用存储在批量指令存储器 BIM0、BIM1、BIM2 和 BIM3 中的指令,通过偏移寄存器 310 生成将在一个周期中被执行的偏移指令,并将生成的偏移指令发送到分配的功能单元。每个批量功能单元可使用程序计数器 (PC) 和分配给每个功能单元的偏移寄存器值来访问批量指令存储器。

[0060] 在这种情况下,如图 7A 所示,批量指令存储器 BIM0、BIM1、BIM2 和 BIM3 被配置为被分为两个或更多个,以对应于各个功能单元,并存储将被发送到对应的功能单元的指令。

[0061] 图 7B 是示出根据本公开的另一示例性实施例的用于偏移指令的输入的处理器的示图。除了图 7A 的处理器 300 之外,处理器 400 还可包括一个或更多个内核队列 420。如

图 7B 所示,不具有多个批量指令存储器 BIM0、BIM1、BIM2 和 BIM3,而仅使用一个批量指令存储器 BIM。

[0062] 参照图 7B,处理器 400 可包括与每个批量功能单元的功能单元对应的两个或更多个内核队列 420。处理器 400 可提取批量指令存储器 BIM 的内核中的至少部分指令,并将提取的指令存储在内核队列 420 中。此外,每个批量功能单元通过基于它的 PC 和分配的偏移寄存器的值访问对应的内核队列 420 来读取期望的指令,以生成偏移指令并将生成的偏移指令发送到功能单元。

[0063] 图 8 是示出根据本公开的示例性实施例的支持用于批量线程处理的处理器的代码生成设备的框图。

[0064] 参照图 1 和图 8,代码生成设备 500 包括程序分析单元 510 和批量指令生成单元 520,以生成支持能够处理批量线程的处理器 100 的批量指令。

[0065] 程序分析单元 510 分析将被处理的预定程序,并生成分析结果。例如,程序分析单元 510 可分析程序中的每个数据之间的依赖性以及在程序中是否存在条件分支语句。

[0066] 批量指令生成单元 520 可基于分析结果生成将在处理器 100 的一个或更多个批量功能单元 120a、120b、120c 和 120d 中执行的一个或更多个批量指令。在这种情况下,每个批量指令可包括一个或更多个指令。

[0067] 批量指令生成单元 520 可基于分析结果的依赖性分析信息,生成允许处理器 100 通过使用包括在批量功能单元 120a、120b、120c 和 120d 中的功能单元而作为 CGRA 操作的代码。可选择地,批量指令生成单元 520 可生成用于一个或更多个批量指令的代码,以使功能单元在每个批量功能单元中以 SIMT 架构操作。

[0068] 如果分析出在程序中存在条件分支语句,则批量指令生成单元 520 允许处理条件分支语句的分支的指令被分别包括在不同的批量指令中。例如,当在条件估计为“真”的情况下执行第一路径并且在条件估计为“假”的情况下执行第二路径时,批量指令生成单元 520 可允许处理第一路径的指令和处理第二路径的指令被分别包括在不同的批量指令中。

[0069] 此外,代码生成设备 500 可生成允许批量功能单元中的一个顺序地执行由批量指令生成单元 520 生成的用于处理分支的批量指令或者允许不同的批量功能单元分别执行所述批量指令的指令。以这种方式,可有效地处理一般的 SIMT 或 CGRA 中的与条件分支关联的约束。

[0070] 批量指令生成单元 520 可生成批量指令,以使每个批量指令的总延迟彼此相似。此外,批量指令生成单元 520 可考虑批量功能单元 120a、120b、120c 和 120d 访问中央寄存器文件 110 的输入 / 输出端口的数量而生成批量指令。例如,批量指令生成单元 520 可以以如下方式生成批量指令:防止特定批量指令请求读取中央寄存器文件的数量超过执行所述特定批量指令的批量功能单元的读取端口的数量,同时防止特定批量指令请求写入的数量超过执行所述特定批量指令的批量功能单元的写入端口的数量。

[0071] 此外,批量指令生成单元 520 可以以如下方式生成批量指令:防止包括在每个批量指令中的指令的数量超过包括在每个批量功能单元中的功能单元的数量。参照图 5A,批量指令 0 和 1 中的每个批量指令包括四个指令,批量指令 2 包括三个指令,从而包括在每个批量指令中的指令的数量不超过 4,即,包括在每个批量功能单元 120a、120b、120c 和 120d 中的功能单元的数量。

[0072] 同时,批量指令生成单元 520 可生成批量指令,以使会在特定批量指令中发生延迟的操作的结果(例如,会发生阻塞的操作的结果)在所述特定批量指令中不用作源。例如,在调度期间,批量指令生成单元 520 可允许发生阻塞的操作被布置在批量指令的最先(foremost)的位置,并允许所述操作的线程执行结果以批量指令的最后次序被使用。可选择地,批量指令生成单元 520 可允许发生阻塞的操作被布置在批量指令的最后位置并且在下一批量指令的执行之前被处理。

[0073] 同时,代码生成设备 500 可生成这样的指令,该指令允许生成的批量指令被同等地输入到所有批量功能单元中,或者允许生成的批量指令被划分并且被分别输入到两个或更多个批量功能单元中。

[0074] 代码生成设备 500 可将生成的批量指令信息和各种指令信息存储在配置存储器或高速缓冲存储器中。同时,批量指令生成单元 520 可生成如上参照图 6A 和图 6B 描述的偏移指令信息。

[0075] 以上描述了批量指令生成单元 520。根据示例性实施例,批量指令生成单元 520 不收集可被同时执行的指令,而是通过收集将被顺序地执行的指令来生成批量指令,因此在生成批量指令的过程中不存在困难,并且可实现提高效率。因为可通过多个批量功能单元同时处理多个数据,所以这种生成方式对于实现大规模并行数据处理是有效的。

[0076] 这与超长指令字(VLIW)和超标量架构形成对比。

[0077] VLIW 是这样的架构,其中,编译器创建由可被同时执行的多个指令构成的超长指令字,并且多个功能单元(或执行单元)在单个时钟周期内处理 VLIW。在数字信号处理中广泛使用的 VLIW 架构经常不能找到足够的可被同时执行的指令,这会导致效率降低。此外,因为所有功能单元应该同时访问中央寄存器文件,所以中央寄存器文件的硬件开销不利地增加。

[0078] 超标量是这样一种架构,其中,硬件发现可用于在运行期间并行执行的指令,并且多个执行单元(或功能单元)处理发现的指令。这种架构也在发现可被同时执行的指令方面存在困难,并且会导致非常复杂的硬件。

[0079] 另一方面,示例性实施例可使用多个批量功能单元同时处理多个数据,这对于实现大规模并行数据处理是有效的。

[0080] 图 9 是示出根据本公开的示例性实施例的使用用于批量线程处理的处理器来处理批量线程的方法的流程图。在图 9 中,示出通过使用图 1 中示出的根据本公开的示例性实施例的处理器 100 来处理批量线程的方法。已经通过图 1 至图 8 描述了批量线程处理的细节,以下将简要描述批量线程处理。

[0081] 首先,处理器 100 可将由代码生成设备生成的一个或更多个批量指令输入到一个或更多个批量功能单元 120a、120b、120c 和 120d(610)。在这种情况下,处理器 100 可以以线程为单位分配所有生成的批量指令,并将分配的批量指令输入到批量功能单元 120a、120b、120c 和 120d。即,所有批量指令被同等地输入到批量功能单元 120a、120b、120c 和 120d 中的每个,以在每个批量功能单元处理需要被处理的所有线程组的一部分的同时被顺序地执行,进而使处理器 100 作为 SIMT 操作。

[0082] 可选择地,批量指令可被划分并被输入到批量功能单元 120a、120b、120c 和 120d。例如,当生成的批量指令的数量为 4 时,四个批量指令可被分别输入到批量功能单元 120a、

120b、120c 和 120d,从而以 MIMT 方案处理线程。可选择地,相同的两个批量指令被输入到两个批量功能单元 120a 和 120b,其余的两个批量指令被输入到其余的两个批量功能单元 120c 和 120d,从而 SIMT 和 MIMT 彼此结合地被使用。

[0083] 当批量指令被划分并被输入到批量功能单元 120a、120b、120c 和 120d 时,处理条件分支的批量指令被输入到不同的批量功能单元 120a、120b、120c 和 120d,从而提高了处理条件分支的效率。此外,因为各个批量功能单元 120a、120b、120c 和 120d 独立地操作,所以即使在一个批量功能单元中发生阻塞,其余的批量功能单元也可不管阻塞而执行线程处理。

[0084] 其后,各个批量功能单元 120a、120b、120c 和 120d 可顺序地执行输入的一个或多个批量指令 (620)。在这种情况下,各个批量功能单元 120a、120b、120c 和 120d 可以以上述交错方案,在针对输入的批量指令切换每个线程的同时执行每个批量指令。

[0085] 同时,如果在批量功能单元 120a、120b、120c 和 120d 正在针对特定批量指令执行预定线程组的同时在特定线程发生阻塞,并且直到针对具有对发生阻塞的所述批量指令的依赖性的另一批量指令执行所述预定线程组才解除阻塞,则批量功能单元 120a、120b、120c 和 120d 针对所述另一批量指令不执行发生阻塞的线程,从而使发生阻塞的线程在结束所述预定线程组的所有线程的执行的最后时间点才被执行。

[0086] 此外,如果当批量功能单元 120a、120b、120c 和 120d 正在针对特定批量指令执行预定线程组时发生条件分支,则线程组被划分为两个或多个子线程组,并且划分的子线程组可分别针对条件分支的多个分支被执行。此外,当多个分支的条件分支结束并且多个分支合并时,划分的子线程组被合并为原始的线程组。

[0087] 示例性实施例的功能可被实现为计算机可读记录介质中的计算机可读代码。计算机可读记录介质包括存储计算机可读数据的所有类型的记录介质。计算机可读记录介质的示例包括 ROM、RAM、CD-ROM、磁带、软盘和光学数据存储介质。此外,记录介质可以以载波 (诸如在互联网传输中使用的载波) 的形式被实现。此外,计算机可读记录介质可被分布到网络上的计算机系统,在网络中,计算机可读代码可以以分布方式被存储和执行。

[0088] 本领域技术人员还将理解,可以以执行特定任务的软件和 / 或硬件组件 (诸如现场可编程门阵列 (FPGA) 或专用集成电路 (ASIC)) 的任意组合来实现示例性实施例。单元或模块可被有利地配置为寄存在可寻址的存储介质中,并被配置为在一个或多个处理器或微处理器上运行。因此,作为示例,模块可包括诸如软件组件、面向对象的软件组件、类组件和任务组件的组件、进程、函数、属性、过程、子程序、程序代码段、驱动程序、固件、微码、电路、数据、数据库、数据结构、表、数组和变量的组件。所述组件和单元中提供的功能可被组合为更少的组件和单元或模块,或者可被进一步分为另外的组件和单元或模块。

[0089] 以上已经描述了多个示例性实施例。然而,将理解的是,可进行各种修改。例如,如果描述的技术以不同的顺序被执行和 / 或如果描述的系统、架构、装置或电路中的组件以不同的方式被组合和 / 或被其他组件或它们的等同物替换或补充,则可实现适当的结果。因此,其他实施方式包括在权利要求的范围内。

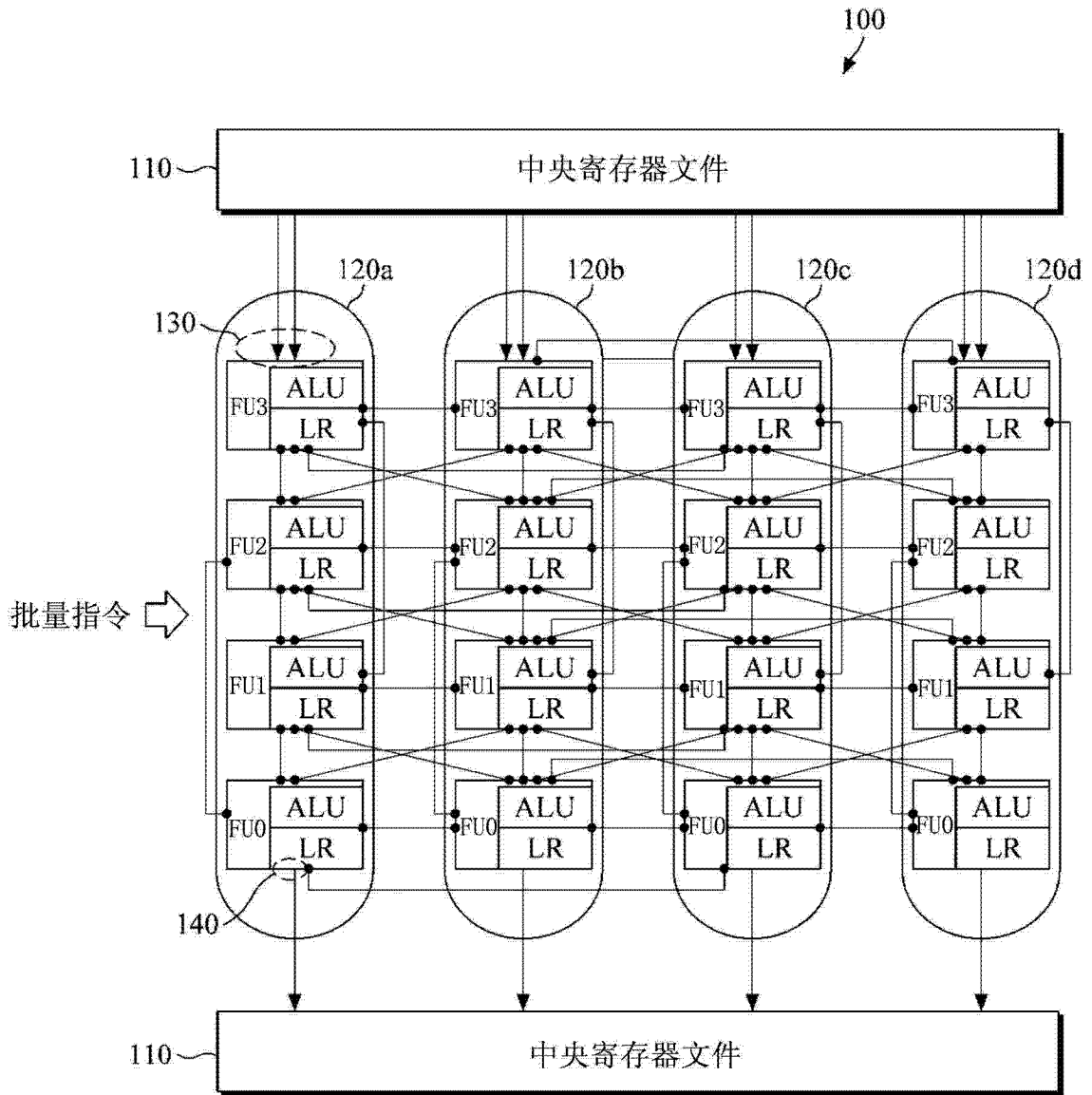


图 1

控制流程图

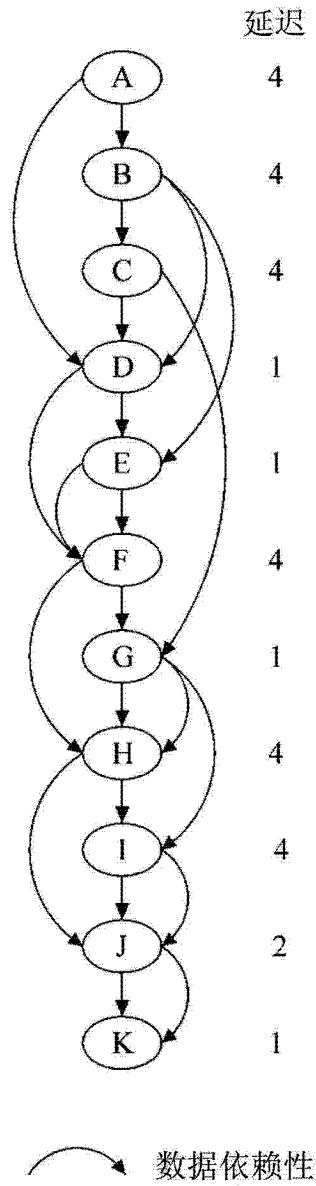


图 2

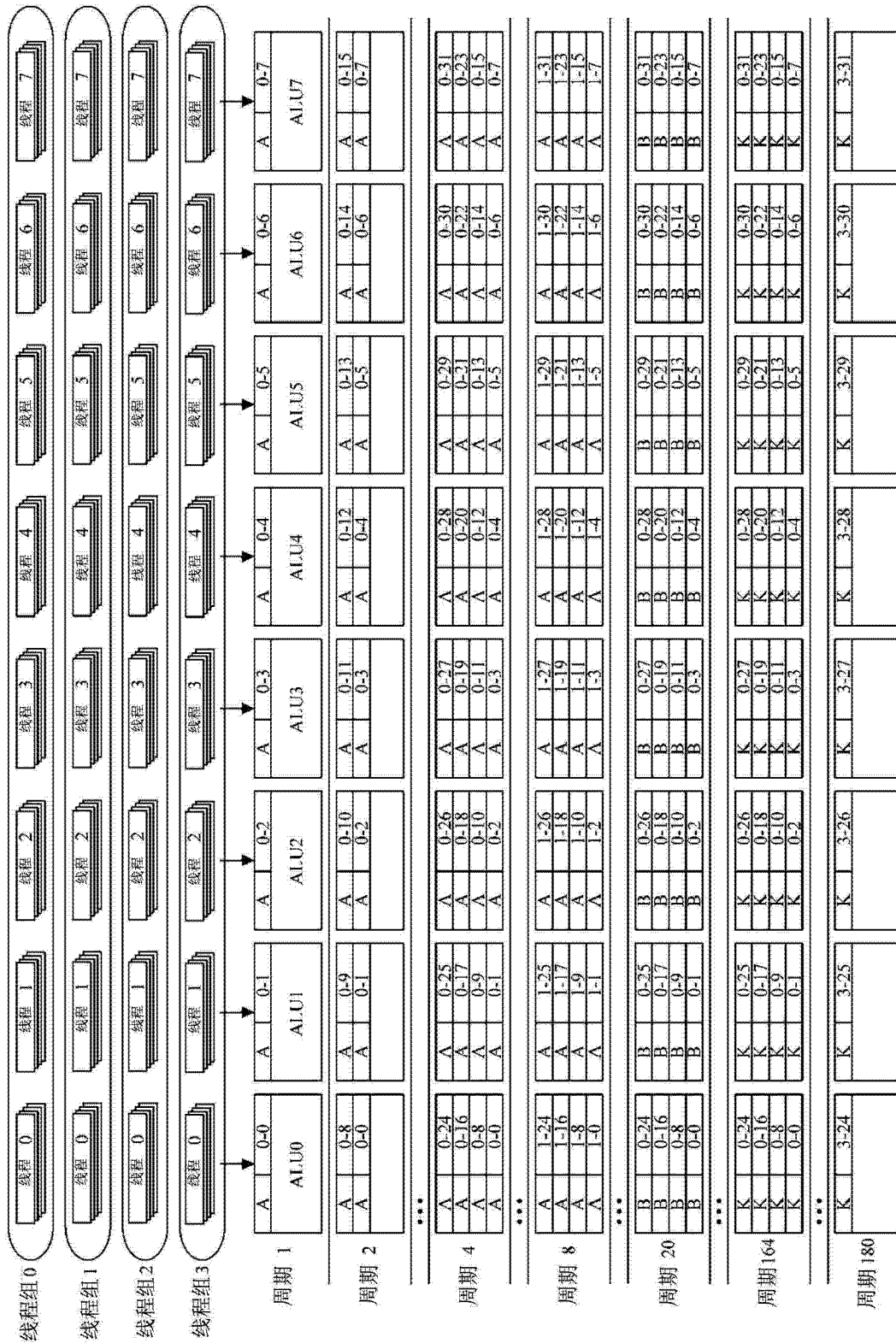


图 3

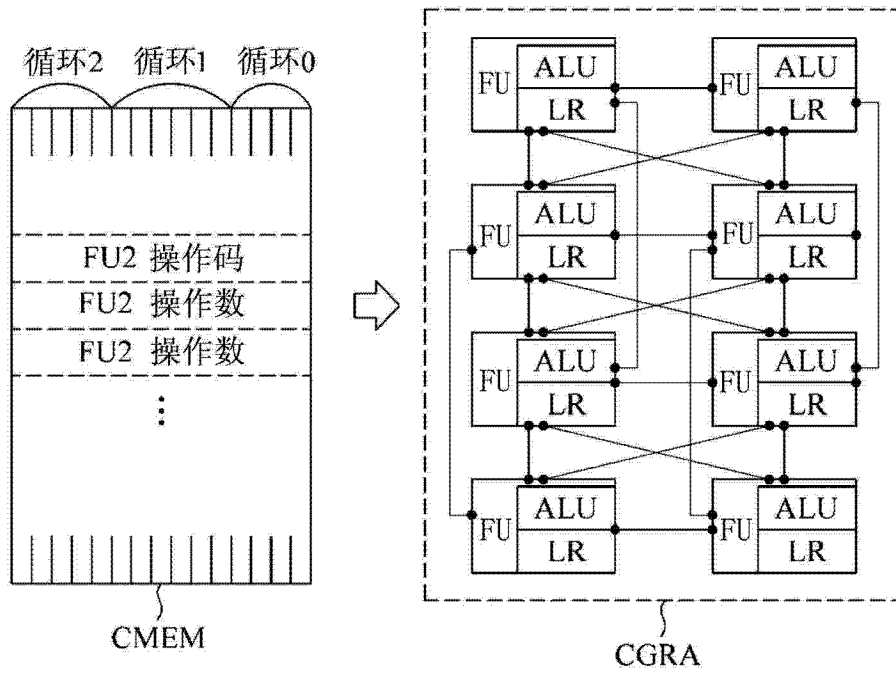


图 4A

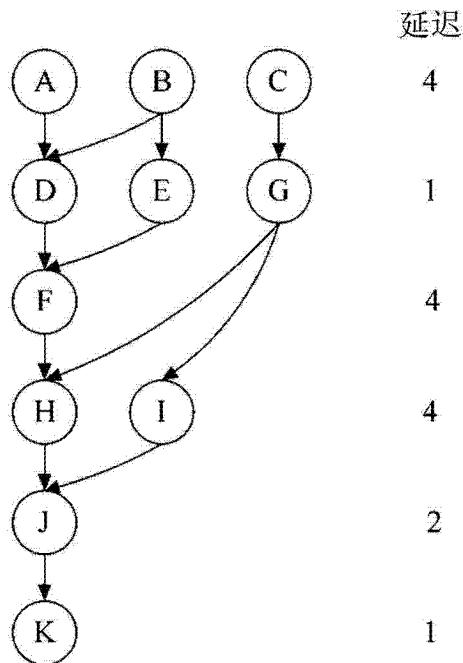


图 4B

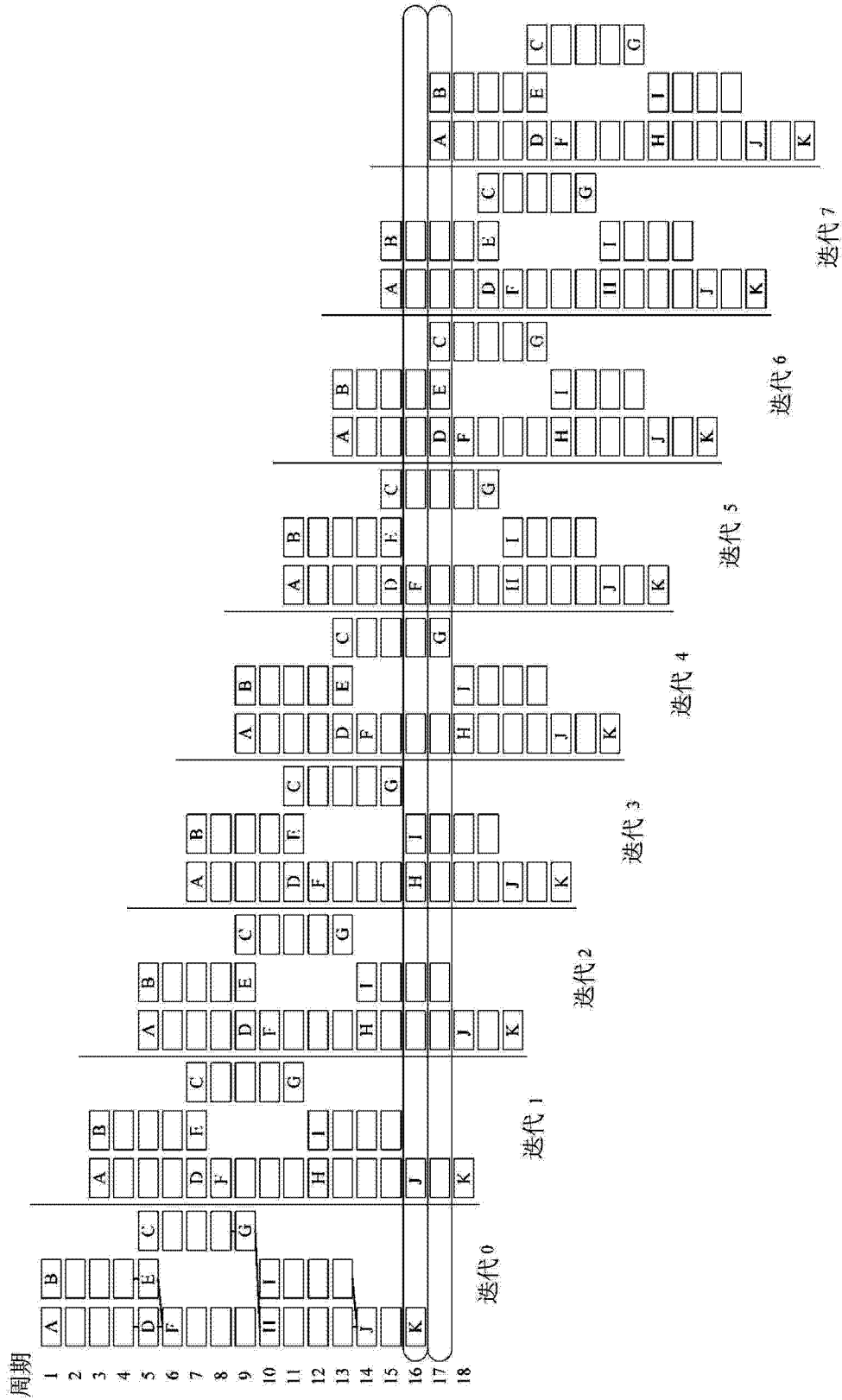


图 4C

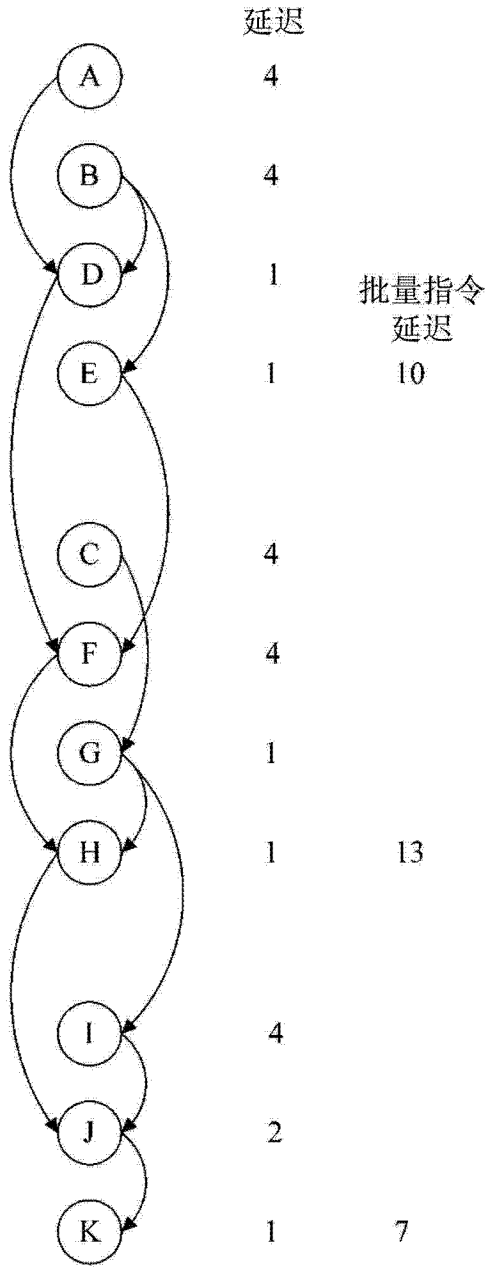


图 5A

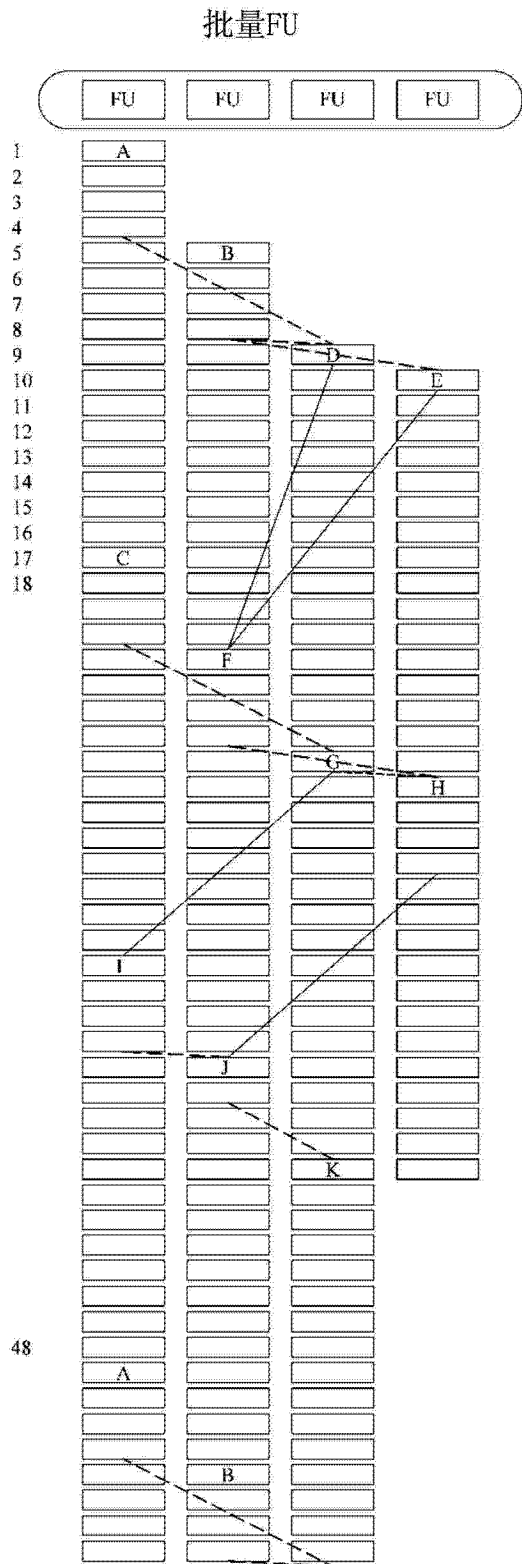


图 5B

批量FU

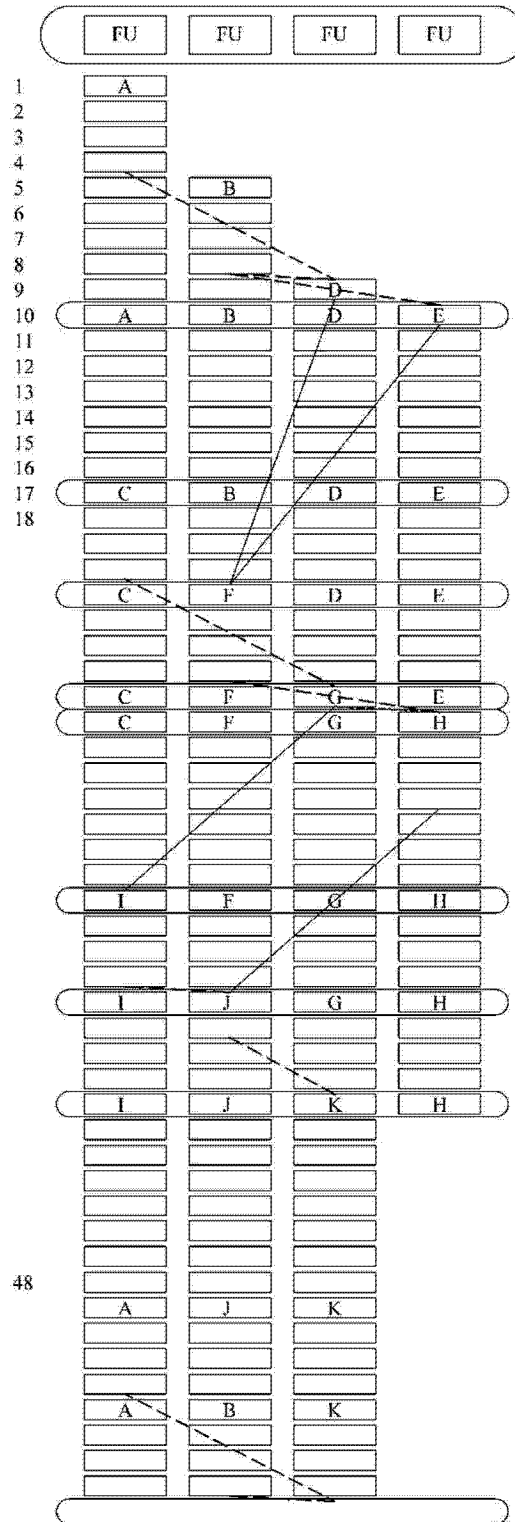


图 6A

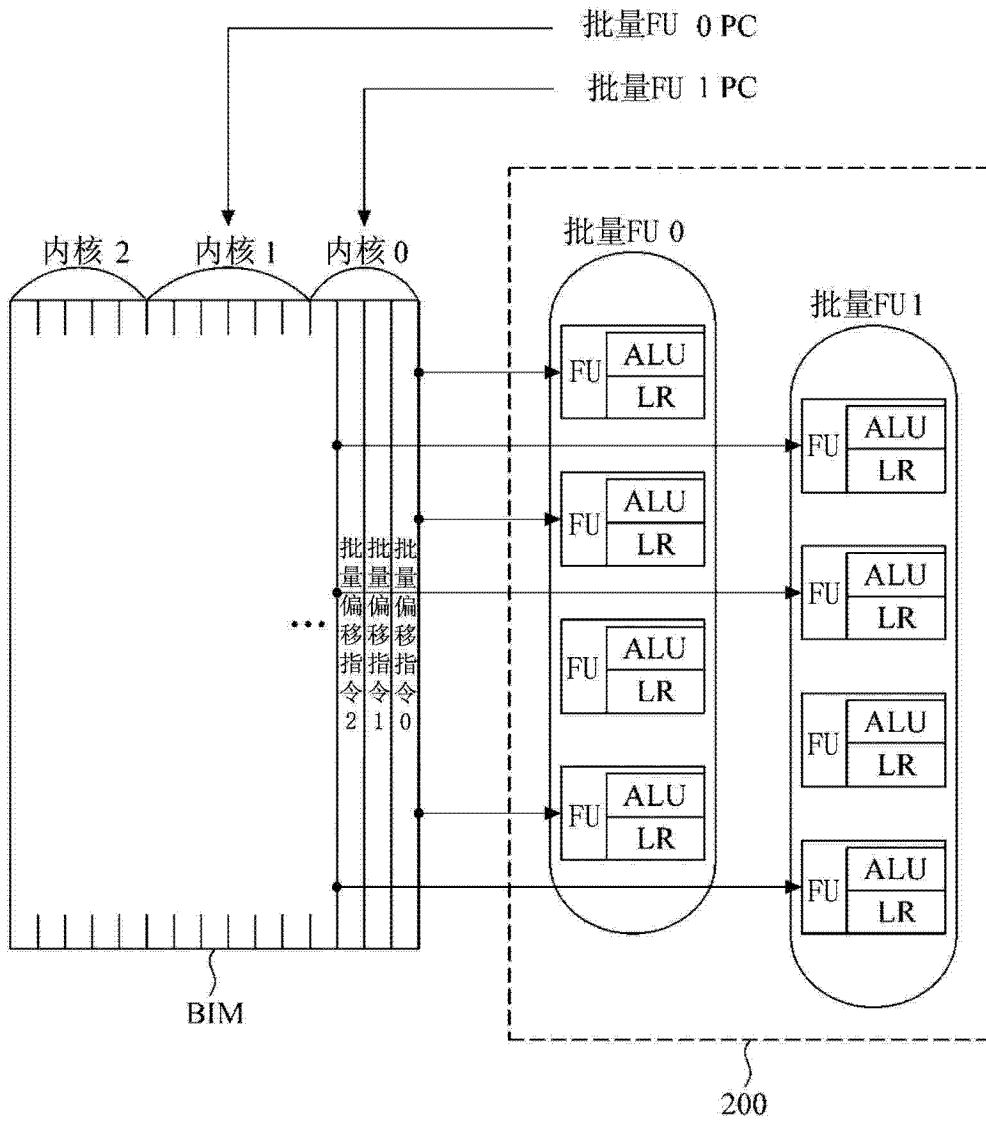


图 6B

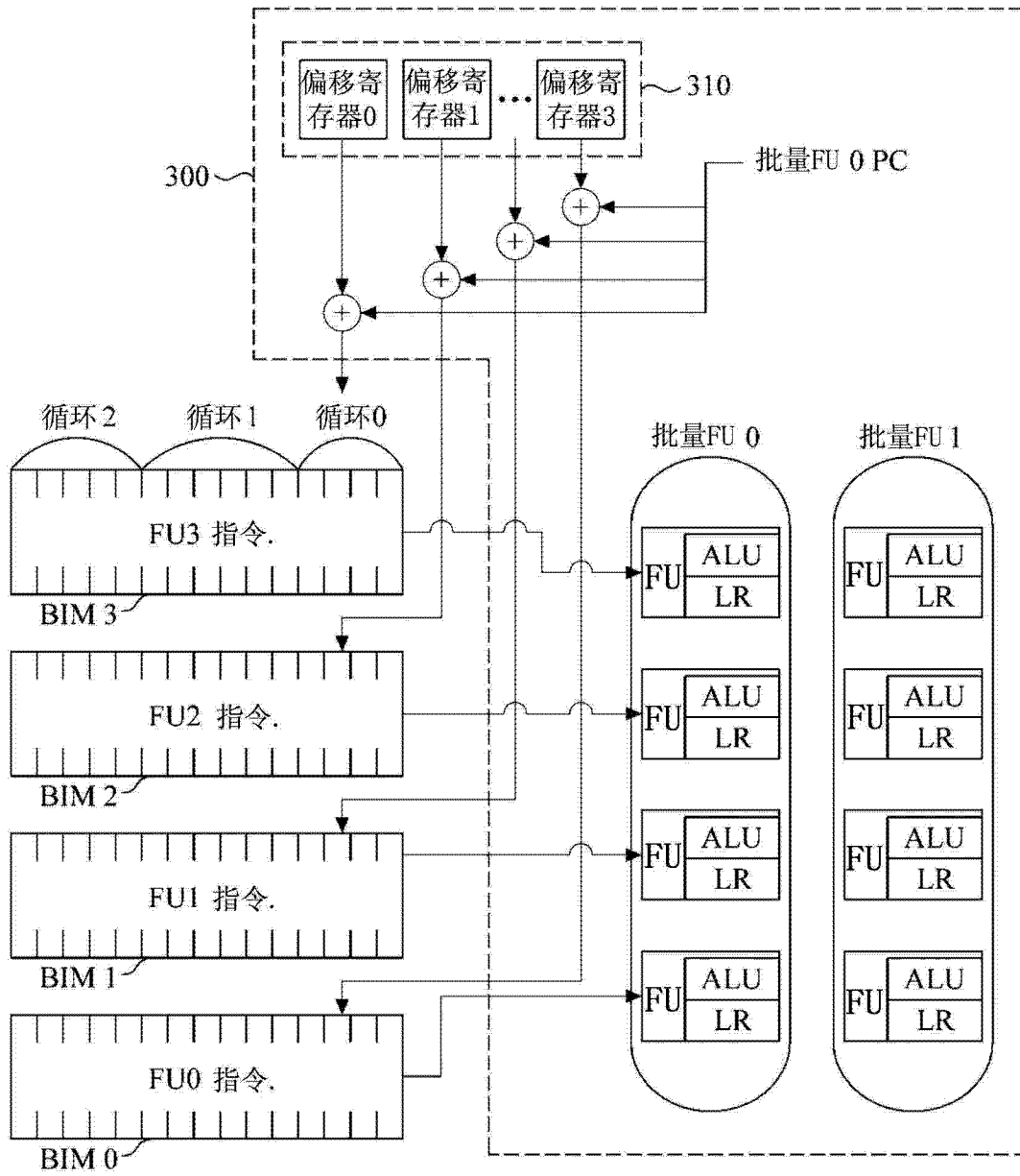


图 7A

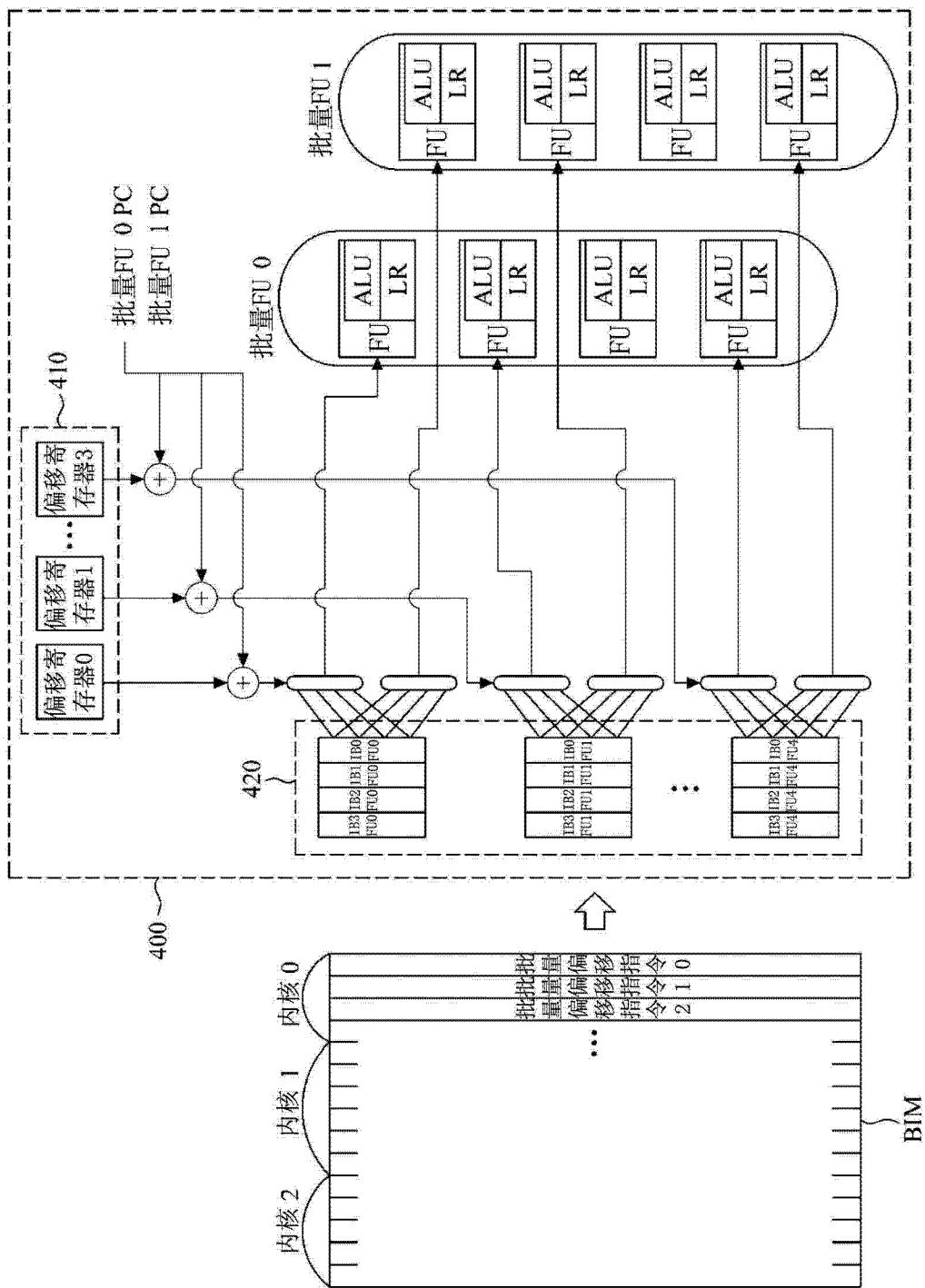


图 7B

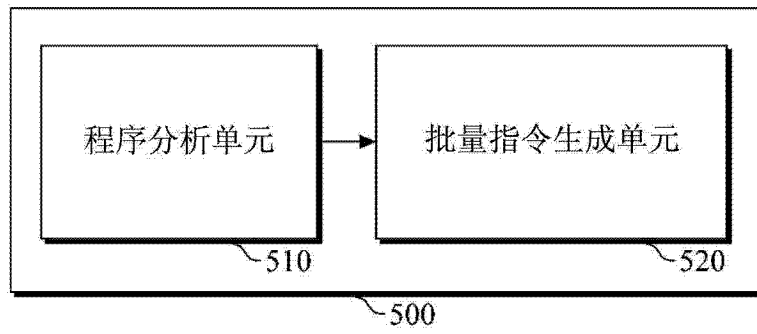


图 8

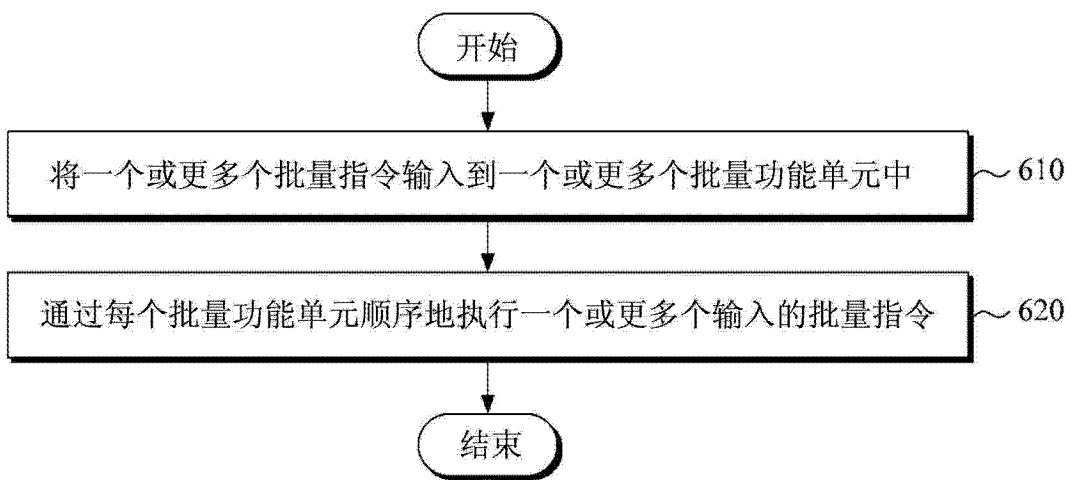


图 9