



(12) 发明专利

(10) 授权公告号 CN 110795903 B

(45) 授权公告日 2023. 08. 15

(21) 申请号 201910864066.8

CN 110059390 A, 2019.07.26

(22) 申请日 2019.09.12

CN 101369223 A, 2009.02.18

(65) 同一申请的已公布的文献号

US 5713012 A, 1998.01.27

申请公布号 CN 110795903 A

US 2006004996 A1, 2006.01.05

(43) 申请公布日 2020.02.14

审查员 吴峰

(73) 专利权人 中科寒武纪科技股份有限公司

地址 100190 北京市海淀区科学院南路6号

科研综合楼644室

(72) 发明人 请求不公布姓名

(51) Int. Cl.

G06F 30/39 (2020.01)

(56) 对比文件

CN 109740747 A, 2019.05.10

CN 101344843 A, 2009.01.14

US 2006005179 A1, 2006.01.05

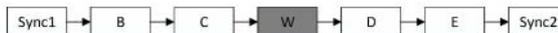
权利要求书2页 说明书13页 附图4页

(54) 发明名称

指令处理方法、装置及相关产品

(57) 摘要

本申请提供了一种指令处理的方法、装置及相关产品,该指令处理方法用于提高指令序列的并行度,本申请实施例通过插入若干指令提高指令序列的并行度,可有效提高指令序列的运算效率。



1. 一种指令处理方法,其特征在于,所述方法包括:

获取当前子序列及所述当前子序列的并行度;其中,所述当前子序列为按照一定规则排列的指令;

判断所述当前子序列的并行度是否符合第一预设并行度;

在所述当前子序列的并行度不符合所述第一预设并行度时,在所述当前子序列中插入额外指令,得到新的子序列,并将所述新的子序列重新确定为所述当前子序列,其中,所述在所述当前子序列中插入额外指令,得到新的子序列,并将所述新的子序列重新确定为当前子序列,包括:

至少移动一次所述额外指令的位置,获得至少一个新的子序列及所述至少一个新的子序列的并行度,确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列,并将所述符合第二预设并行度的新的子序列作为所述当前子序列。

2. 根据权利要求1所述方法,其特征在于,所述额外指令与所述当前子序列中指令之间不存在地址相关性。

3. 根据权利要求1所述方法,其特征在于,所述至少移动一次所述额外指令的位置,获得至少一个新的子序列,包括:

获取所述额外指令的运动区间以及运动步长;

根据所述额外指令的运动步长,每次将所述额外指令从当前位置移动至目标位置,获得至少一个新的子序列,

其中,所述当前位置和所述目标位置均位于所述额外指令的运动区间内。

4. 根据权利要求1所述方法,其特征在于,所述确定新的子序列的并行度,包括:

若所述新的子序列中相邻的两个指令块中第二指令块中指令与第一指令块中至少一个指令在预设时间段内访问同一个部件,则将所述两个指令块划分在不同的指令块中;

若所述新的子序列中相邻的两个指令块中第二指令块中的指令与第一指令块中所有指令在预设时间段内访问不同的部件,则将所述两个指令块划分在相同的指令块中;

根据所述新的子序列中所述指令块,确定所述新的子序列的并行度;

其中,所述每个指令块中包括至少一条指令。

5. 根据权利要求4所述方法,其特征在于,所述方法还包括:

当所述相邻两个指令块中都包括至少两条指令时,在所述相邻指令块之间插入Nope块。

6. 根据权利要求1-5任一项所述方法,其特征在于,所述方法还包括:

将所述当前子序列用指令运行图的形式表示。

7. 根据权利要求1-5任一项所述方法,其特征在于,所述确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列,包括:

若当前新的子序列的并行度小于上一新的子序列的并行度,则根据所述当前新的子序列中插入指令的运动方向,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度;

若当前新的子序列的并行度大于或者等于上一新的子序列的并行度,则根据所述当前新的子序列中所述额外指令的运动方向及预设接受概率,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度。

8. 根据权利要求3所述方法,其特征在于,所述将所述额外指令从当前位置移至目标位置,包括:

若在后指令块中的指令数量大于所述额外指令的运动步长时,则将所述在后指令块拆分为至少一个子指令块,之后执行所述根据所述子序列中额外指令的运动步长,每次将至少一个所述子序列中的额外指令从当前位置移动至目标位置,获得至少一个新的待处理指令序列;

在将所述额外指令从当前位置移动至目标位置后,若在预设时间段,所述当前位置的后一指令与在先指令块中的指令访问不同的部件时,则将所述当前位置的后一指令合并至所述在先指令块;

其中,所述在先指令块是指位于所述额外指令的当前位置之前的指令块,所述在后指令块是指位于所述额外指令的当前位置之后的指令块。

9. 一种计算机设备,其特征在于,包括处理器和存储器,所述存储器中存储有计算机程序,所述处理器执行所述计算机程序时,实现如权利要求1-8任一项所述的方法的步骤。

10. 一种计算机存储介质,其特征在于,所述计算机存储介质中存储有能够被处理器执行的计算机程序,所述处理器执行所述计算机程序时,实现如权利要求1-8任一项所述的方法的步骤。

11. 一种指令处理装置,其特征在于,所述处理装置包括:

指令获取模块,用于获取当前子序列及其并行度信息;其中,所述当前子序列为按照一定规则排列的指令;

判断模块,用于将所述当前子序列的并行度与第一预设并行度进行比较;

插入额外指令模块,用于当所述当前子序列的并行度不符合所述第一预设并行度时,在所述当前子序列中插入额外指令,得到新的子序列,并将所述新的子序列重新确定为所述当前子序列,其中,

所述插入额外指令模块包括指令移动模块,所述指令移动模块用于至少移动一次所述额外指令的位置,获得至少一个新的子序列,确定所述至少一个新的子序列的并行度,确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列为所述当前子序列。

指令处理方法、装置及相关产品

技术领域

[0001] 本公开涉及计算机技术领域，特别是涉及一种指令处理方法及装置以及相关产品。

背景技术

[0002] 随着硅材料半导体工艺接近物理极限，在单位面积的硅材料芯片上集成更多的晶体管变得越来越困难。电子器件中集成电路集成的功能越来越多，处理器需要执行的指令数量也越来越多，这就对指令处理提出了新的挑战。因此，如何优化指令序列，以提高指令序列的处理效率将成为需要考虑的问题。

发明内容

[0003] 基于此，有必要针对上述技术问题，提供一种能够提高指令序列处理效率的指令处理方法、装置以及相关产品。

[0004] 第一方面，提供一种指令处理的方法，所述方法包括：

[0005] 获取当前子序列及所述当前子序列的并行度；

[0006] 判断所述当前子序列的并行度是否符合第一预设并行度；

[0007] 在所述当前子序列的并行度不符合所述第一预设并行度时，在所述当前子序列中插入额外指令，得到新的子序列，并将所述新的子序列重新确定为所述当前子序列。

[0008] 第二方面，本申请实施例提供一种指令处理的装置，所述指令处理装置包括：

[0009] 指令获取模块，用于获取当前子序列及其并行度信息；

[0010] 判断模块，用于将所述当前子序列的并行度与第一预设并行度进行比较；

[0011] 插入额外指令模块，用于当所述当前子序列的并行度不符合所述第一预设并行度时，在所述当前子序列中插入额外指令，得到新的子序列，并将所述新的子序列重新确定为所述当前子序列。

[0012] 第三方面，本申请实施例提供了一种计算机可读存储介质，其存储用于电子数据交换的计算机程序，其中，所述计算机程序使得计算机执行如第一方面所述的方法步骤。

[0013] 第四方面，本申请实施例提供了一种计算机程序产品，所述计算机程序产品包括存储了计算机程序的非瞬时性计算机可读存储介质，所述计算机程序可操作来使计算机执行如第一方面所述的方法步骤。

[0014] 第五方面，本申请实施例提供了一种芯片，该芯片包括上述第二方面所述待测器件验证装置。

[0015] 第六方面，本申请实施例提供了一种芯片封装结构，该芯片封装结构包括上述第五方面所述的芯片。

[0016] 本公开的指令处理方法、装置以及相关产品，通过在当前子序列中插入至少一个额外的指令，所述额外指令可以与当前子序列中至少一个指令并行，使当前子序列中能够并行的指令数增大，直至符合所述第一预设并行度。这样，可以根据该符合第一预设并行度

对应的待处理器指令序列执行该指令序列,从而可以加速指令处理的效率。并且,当需要重复执行该指令序列时,可以直接按照该符合预设条件的并行度信息对应的待处理器指令序列执行该指令序列,无需移动同步指令并确定并行度信息,从而可以提高指令序列的处理效率。

附图说明

[0017] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本申请的一些实施例,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,还可以根据这些附图获得其他的附图。

[0018] 图1示出本申请一实施例提供的指令处理方法的流程图;

[0019] 图2示出本申请一实施例提供的插入额外指令的指令序列图;

[0020] 图3示出本申请一实施例提供的确定并行度的过程图;

[0021] 图4示出本申请一实施例提供的一种确定符合第二预设条件并行度的方法流程图;

[0022] 图5示出本申请一实施例提供的移动额外指令前后的示意图;

[0023] 图6示出本申请一实施例提供的装置图;

[0024] 图7示出本申请一实施例提供的一种电子设备的框图。

具体实施方式

[0025] 下面将结合本披露实施例中的附图,对本披露实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本披露一部分实施例,而不是全部的实施例。基于本披露中的实施例,本领域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,都属于本披露保护的范围。

[0026] 应当理解,本披露的权利要求、说明书及附图中的术语“第一”、“第二”、等是用于区别不同对象,而不是用于描述特定顺序。本披露的说明书和权利要求书中使用的术语“包括”和“包含”指示所描述特征、整体、步骤、操作、元素和/或组件的存在,但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

[0027] 还应当理解,在此本披露说明书中所使用的术语仅仅是出于描述特定实施例的目的,而并不意在限定本披露。如在本披露说明书和权利要求书中所使用的那样,除非上下文清楚地指明其它情况,否则单数形式的“一”、“一个”及“该”意在包括复数形式。还应当进一步理解,在本披露说明书和权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合,并且包括这些组合。

[0028] 如在本说明书和权利要求书中所使用的那样,术语“如果”可以依据上下文被解释为“当...时”或“一旦”或“响应于确定”或“响应于检测到”。类似地,短语“如果确定”或“如果检测到[所描述条件或事件]”可以依据上下文被解释为意指“一旦确定”或“响应于确定”或“一旦检测到[所描述条件或事件]”或“响应于检测到[所描述条件或事件]”。

[0029] 随着电子器件中集成电路集成的功能越来越多,处理器需要执行的指令数量也越来越多,这就对指令处理提出了新的挑战。传统的指令处理方法主要分为两种:一种是直接

对随机生成的指令序列进行处理；另一种是由于指令序列中可能存在竞争冒险的指令，因而要求指令序列中的指令按照一定的顺序运行，基于此，可以在指令序列中随机插入同步指令，以保证指令序列中具有竞争冒险的指令运行时的先后顺序。但是，以上两种指令处理方法均是对随机产生的指令序列进行直接处理，并未对指令序列中各个指令的执行顺序等进行优化，从而导致指令序列的处理效率较低。

[0030] 根据本公开实施例的指令序列处理方法可应用于处理器中，该处理器可以是通用处理器，例如CPU(Central Processing Unit,中央处理器)，也可以是用于执行人工智能运算的人工智能处理器(IPU)。人工智能运算可包括机器学习运算，类脑运算等。其中，机器学习运算包括神经网络运算、k-means运算、支持向量机运算等。该人工智能处理器可例如包括GPU(Graphics Processing Unit,图形处理单元)、NPU(Neural-Network Processing Unit,神经网络处理单元)、DSP(Digital Signal Process,数字信号处理单元)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)芯片中的一种或组合。本公开对处理器的具体类型不作限制。

[0031] 在一种可能的实现方式中，本公开中所提及的处理器可包括多个处理单元，每个处理单元可以独立运行所分配到的各种任务，如：卷积运算任务、池化任务或全连接任务等。本公开对处理单元及处理单元所运行的任务不作限制。

[0032] 图7示出根据本公开实施例的指令序列处理方法的处理器的示意图。如图7所示，处理器100包括多个处理单元101以及存储单元102，多个处理单元101用于执行指令序列，存储单元102用于存储数据，可包括随机存储器(RAM,Random Access Memory)和寄存器堆。处理器100中的多个处理单元101既可共用部分存储空间，例如共用部分RAM存储空间和寄存器堆，又可同时拥有各自的存储空间。该处理器能够执行本申请的指令处理方法，以使得该待处理指令序列的并行度满足预设条件，从而提高待处理指令序列的处理效率和准确性。具体的实现方案详见下文的描述。

[0033] 如图1所示，图1为本申请一实施例提供的指令处理方法的流程图。该方法能够应用于上述的计算机设备中，所述方法包括：

[0034] 11:处理器获取当前子序列及所述当前子序列的并行度。

[0035] 其中，处理器可以从存储器中读取上述当前子序列，该当前子序列可以是按照一定规则(如先进先出)排列的指令。所述子序列的起始指令或终止指令中至少一个为同步指令，所述同步指令(sync指令)用于保证所述同步指令之前的所有指令执行完毕后才能执行所述同步指令之后的指令，在某种程度上具有保证指令运行顺序的作用，本方案中起始指令和或终止指令中的同步指令用于保证两个子序列之间的执行顺序。具体地，将相邻两条同步指令之间的指令序列作为当前子序列，该当前子序列的起始指令和终止指令均为同步指令。

[0036] 可选地，处理器可以从外部直接获得所述当前子序列并行度信息，也可以在获取当前子序列后，计算获得该当前子序列的并行度信息，具体的当前子序列的并行度信息的方法详见下文。应当清楚的是，在计算机体系结构中，并行设计可以包括数据并行和指令并行，其中，在指令流水中，处理器同时执行多条指令称为指令并行，该并行执行的指令之间无相互依赖关系。该并行度信息可以用于衡量待处理指令序列中并行执行的指令数量。可选地，该当前子序列中并行执行的指令数量越多，则该当前子序列的并行度越高，进

而该当前子序列的处理时间越短,处理效率越高。该当前子序列中并行指令的指令数量越少,则当前子序列的并行度越低,进而处理器对当前子序列的处理时间越长,处理效率越低。

[0037] 12:判断所述当前子序列的并行度是否符合第一预设并行度。

[0038] 其中,所述当前子序列中的指令数量并不是确定的,为了提高处理器执行当前子序列的效率,我们应尽可能提高所述当前子序列的并行度。为了方便判断,我们可以预设一个并行度值来作为衡量并行度高低的的标准,所以我们用第一预设并行度来判断所述当前子序列的并行度是否足够大。

[0039] 例如,在当前子序列的指令数量较少时,则认为该当前子序列的并行度较低,其不符合第一预设并行度。在当前子序列的指令数量较多时,则认为该当前子序列的并行度较高,其符合第一预设并行度。

[0040] 13:在所述当前子序列的并行度不符合所述第一预设并行度时,在所述当前子序列中插入额外指令,得到新的子序列,并将所述新的子序列确定为当前子序列。

[0041] 其中,所述额外指令来自于指令集合其他的指令。可选地,所述额外指令与所述当前子序列中指令之间不存在数据竞争关系。即所述额外指令是与所述当前子序列中任意指令都没有地址相关性的指令,无地址相关是为了防止插入新的指令而引起新的数据冒险。如果说我们为了丰富两条sync指令之间的指令种类和数目,插入了一堆指令,但是这些指令之间有地址相关(大概率会引起数据冒险,比如IO指令),那么就会需要再次插入sync指令对数据冒险进行消除,这样就又回到了起点(因为又多了sync指令)。可选地,该额外指令可以插入在当前子序列的任意位置。在当前子序列的并行度不符合第一预设并行度时,处理器可以将额外指令随机插入当前子序列中,并将该位置作为额外指令的初始位置。

[0042] 参阅图2,图2示出本申请一实施例提供的插入额外指令的指令序列图。假设所述指令序列为一条当前子序列,所述当前子序列的起始指令和终止指令为两个同步指令sync1和sync2,所述当前子序列还包括指令B、指令C、指令D和指令E,在所述当前子序列的并行度不符合第一预设条件时,在所述当前子序列中插入额外指令W,所述额外指令的插入位置是任意的,可以位于以下任意一个位置:

[0043] Sync1指令与指令B之间、指令B和指令D之间(也就是图中所述位置)、指令D和指令E之间、指令E和sync2之间。

[0044] 可选地,所述每次插入的额外指令为一个或多个。

[0045] 可选地,在所述当前子序列中插入额外指令,得到新的子序列,并将所述新的子序列重新确定为当前子序列还包括:至少移动一次所述额外指令的位置,获得至少一个新的子序列及所述至少一个新的子序列的并行度,确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列为所述当前子序列。

[0046] 其中,由于上述当前子序列中的额外指令是随机插入的,即所述额外指令在所述当前子序列中的初始位置是随机的,因此,该包含额外指令的当前子序列的并行度可能并不符合第二预设条件,因此处理器可以移动所述额外指令的位置,得到新的子序列并确定新的子序列中符合第二预设条件的额外指令的位置,以获得新的子序列的并行度信息,并根据各个子序列的并行度信息确定符合第二预设条件的并行度信息。其中,所述第二预设条件是使新的子序列并行度达到最大的额外指令的插入位置。

[0047] 可选地,在所述当前子序列的并行度符合所述第一预设并行度时,无需在当前子序列中插入额外指令。例如,一条当前子序列的并行度不小于第一预设并行度时,处理器直接对所述当前子序列进行处理。

[0048] 在一个可选地的实施例中,上述至少移动一次所述额外指令的位置,获得至少一个新的子序列,包括:

[0049] 获取所述额外指令的运动区间以及运动步长;

[0050] 根据所述额外指令的运动步长,每次将所述额外指令从当前位置移动至目标位置,获得至少一个新的子序列,

[0051] 其中,所述当前位置和所述目标位置均位于所述额外指令的运动区间内。

[0052] 所述额外指令的运动步长是指所述额外指令每移动一次位置,所述额外指令的起始位置与终止位置两个位置点之间的差值。可选地,该额外指令的运动步长也可以是预设值,例如,该额外指令的运动步长可以是用户自定义输入的。根据具体情况具体分析,例如,若所述指令序列的长度比较短,所述运动步长可以设为1个单位,这种情况我们也称之为遍历,也是使结果最准确的设置;若所述指令序列的长度很长,遍历的情况可能会花费很长时间,此时所述运动步长可以设置大一些,比如3个单位、5个单位甚至更大,本公开对此不作限制。

[0053] 可选地,所述运动步长可以以指令数为单位,也可以根据并行度以指令块确定,本公开对此不作限制。其中,所述指令块在下文中会详细介绍。

[0054] 额外指令的运动区间是指额外指令在当前子序列中所有可能位置的集合,额外指令可以移动至运动区间中的任意位置。可选地,所述额外指令的运动区间可以是根据该额外指令所在的当前子序列中的指令确定的。具体地,该额外指令的运动区间可以根据两个同步指令之间的指令确定。可选地,该额外指令的运动区间可以采用坐标的形式进行表示,其中,该额外指令的初始位置可以作为坐标的原点,也可以根据确定其他位置为坐标原点,所述运动区间不限于坐标的表示方式,只要能正确表达出额外指令的移动范围即可,本公开对所述运动区间的表示方法不做限制。

[0055] 仍以图2为例,假设所述额外指令W的位置如图2所示,额外指令W位于指令C和指令D之间,用坐标的方式表示所述额外指令的运动区间,当前额外指令W为坐标原点,所述额外指令W的运动区间为 $[-2, 2]$ 。

[0056] 可选地,上述确定子序列的并行度也可以用指令块表示,所述确定子序列的并行度,包括:

[0057] 根据所述新的子序列中所述指令块,确定所述新的子序列的并行度;其中,所述子序列每个指令块中包括至少一条指令。可选地,子序列中的指令可以是按照一定的规则划分为不同的指令块,例如按照指令间的并行关系划分为不同的指令块。

[0058] 进一步可选地,在根据指令块确定子系列的并行度的操作之前,处理器可以首先将子序列划分为至少一个指令块。具体地,上述方法还包括:

[0059] 若所述子序列中相邻的两个指令块中第二指令块中指令与第一指令块中至少一个指令在预设时间段内访问同一个部件,则将所述两个指令块划分在不同的指令块中;

[0060] 若所述子序列中相邻的两个指令块中第二指令块中的指令与第一指令块中所有指令在预设时间段内访问不同的部件,则将所述两个指令块划分在相同的指令块中。

[0061] 例如,参阅图3,图3示出本申请一实施例提供的确定并行度的过程图。其中图3中的A为一条子序列,所述子序列包括指令B~J和两个同步指令sync1和sync2,对于一条未处理过的指令序列,我们可以认为每条指令为一个指令块,也就是指令块B~指令块J和指令块sync1、指令块sync2。在确定该子序列的并行度时,因为同步指令与其他指令在预设时间段内访问不同的部件,与其他指令都不能并行,因此sync1和sync2都单独成一个指令块。所述子序列中相邻的两个指令块中第二指令块中的指令C与第一指令块B在预设时间段内访问不同的部件,因此将所述两个指令块划分在相同的指令块中,此时,所述子序列中相邻的两个指令块中第二指令块中的指令D与第一指令块所有指令包括指令BC指令在预设时间段内访问不同的部件,因此将所述两个指令块划分在相同的指令块中,此时,所述子序列中相邻的两个指令块中第二指令块中的指令E与第一指令块至少一个指令包括指令BCD指令在预设时间段内访问相同的部件,将所述两个指令块划分在不同的指令块中,以此类推,对该子序列的其他指令执行相同的操作,得到整个子序列的指令块。

[0062] 例如如图3所示,图3中的B示出本申请提供的一个可能的指令运行图,根据所述子序列中所述指令块,确定所述新的子序列的并行度。我们可以用当前子序列地指令块数量衡量所述当前子序列的并行度,同一条当前子序列中,所述指令块的数量越少,并行度越高。

[0063] 可选地,所述子序列可以用指令运行图的形式表示,将所述指令块依次排列连接起来,当所述相邻两个指令块中都多于一条指令时,在所述两个指令块之间插入Nope块,其中,所述Nope指令块不在所述并行度的计算范围,所述Nope块只是为了所述两个指令块中指令的连接清楚。得到所述当前子序列的指令运行图。

[0064] 例如,如图3所示,其中,B是A示出的当前子序列划分指令块后得到的指令运行图。当前子序列中指令块FG和指令块HI中均包含至少两个指令,则在所述两个指令块中间加入Nope块。进一步地,所述指令运行图中指令块为5(同步指令和Nope块不计算在内),我们可以认为所述当前子序列的并行度为5。

[0065] 可选地,当同一当前子序列中,可能存在指令块数相同的情况,所述情况可能是由于插入额外指令的位置不同,导致指令块划分的结果不同,但是所述至少两种情况下得到的指令块数相同,此时所述指令块中指令数多的情况下所述当前子序列的并行度越高。例如假如图3中指令运行图B存在一种情况,指令D不能与指令B和C划分在一个指令块中,而与指令E划分为一个指令块中,那么所述当前子序列得到的指令块数也是5,与图3中指令运行图B所述划分方式比,图3中指令运行图B中指令块中包含3个指令BCD,而上述情况下该指令块只包括两个指令BC,此时我们可认为图3中示出的指令运行图B所示情况下的并行度更高。

[0066] 可选地,当子序列中存在指令块中指令数大于2个指令时,在插入额外指令时可能存在指令拆分与合并的情况。例如在图3中B所示指令序列的基础上,插入额外指令w,假设插入的额外指令的位置在指令F和指令G之间,我们需要先把指令块FG拆分为指令块F和指令块G,然后根据上文所述方法判断指令W和指令F或指令G在预设时间段内是否访问同一部件,然后确定所述指令FGW中至少两个指令能否合并。

[0067] 可选地,所述插入的额外指令可以以划分的指令块为单位进行插入或移动,可以减少额外指令的运动步长。

[0068] 可选地,在确定额外指令的运动步长和运动区间时,有以下几种情况:

[0069] 1、当所述运动步长以指令数为单位时

[0070] (1) 运动步长小于在后指令块的指令数时,将所述在后指令块拆分为至少一个子指令块,同时判断所述额外指令在当前位置时前后两个指令块中的指令是否可以合并,并进行相应操作,然后移动所述额外指令的位置至目标位置,其中,所述在后指令块指所述额外指令后面的指令块。

[0071] 例如,如图5所示,A为一个插入额外指令的指令序列,运动步长为2时,小于所述在后指令块的指令数3,将所述在后指令块BCD拆分为指令块BC和指令块D两个指令块,得到拆分后的指令序列B。所述额外指令W在当前位置时前后两个指令块分别为sync1和指令块BC,不可以进行合并,之后将所述子序列中的额外指令从当前位置移动至目标位置指令块BC后,获得新的指令序列。

[0072] (2) 运动步长等于在后指令块的指令数时,直接将所述额外指令移至所述在后指令块的后面,同时判断所述额外指令在当前位置时前后两个指令块中的指令是否可以合并,并进行相应操作,然后移动所述额外指令的位置至目标位置,得到新的指令序列。

[0073] 例如,运动步长为3,等于在后指令块的指令数,直接将所述额外指令移至指令块BCD的后面,同上sync1与在后指令块不能合并,得到新的指令序列。

[0074] 当运动步长大于在后指令块中的指令数时,继续向后面的指令块移动,直至所述指令的总数量等于所述运动步长的大小。

[0075] 例如,运动步长大于3时,假设运动步长为5,将指令块FG进行拆分,同上额外指令的前后指令块sync1与指令块BCD不能合并,将额外指令移动至目标指令位置指令块F之后。

[0076] 2、当所述运动步长以指令块为单位时

[0077] 当所述运动步长以指令块为单位时,不存在对指令块进行拆分的过程,移动到的目标位置均为当前位置加运动步长所得到的位置,在一个可能的实施例中,在第一次移动所述额外指令的位置时,需要判断所述额外指令前后两个指令块中的指令是否可以合并。例如,如图5所示指令运行图A,假设运动步长以指令块为单位,运动步长为1、2、3等等,相应的所述额外指令从当前位置移动至指令块BCD后面、指令块E后面或者指令块FG后面等等,与此同时,需判断指令块sync1与指令块BCD是否能合并,此处不可以合并,可选地,假如在前指令块与在后指令块内指令在预设时间段内,不访问同一个部件,那么所述两个指令块可以并行,将所述两个指令块合并为同一个指令块。

[0078] 可选地,在所述当前子序列中插入额外指令,至少移动一次所述额外指令的位置,获得至少一个新的子序列及所述至少一个新的子序列的并行度,确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列为所述当前子序列,包括:

[0079] 若当前新的子序列的并行度小于上一新的子序列的并行度,则根据所述当前新的子序列中插入指令的运动方向,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度;

[0080] 若当前新的子序列的并行度大于或者等于上一新的子序列的并行度,则根据所述当前新的子序列中所述额外指令的运动方向及预设接受概率,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度。

[0081] 参阅图4,图4示出本申请一实施例提供的一种确定符合第二预设条件并行度的方法流程图。开始随机产生一个初始解x,所述解可以是衡量子序列并行度的指令块数,然后

按照某种方法产生新解 x_1 ,所述某种方法可以通过移动指令的位置或通过插入额外的指令获得新的指令块数 x_1 ,然后判断所述新的指令块数与初始指令块数的大小,在所述获得的新的指令块数 x_1 小于初始解 x 时,将所述新解 x_1 代替初始解 x ,判断此时子序列并行度是否达到要求,假若仍不符合要求,继续缩减并行度,重复执行上述操作,直至得到的子序列的并行度达到要求,符合所述第二预设条件,得到最优解 x 。

[0082] 当插入的额外指令数量为两个以上时,本申请实施例可以采用上述方法同时移动两条以上的额外指令,以快速确定各个额外指令的插入位置,从而使得新的子序列能够符合第二并行度,从而提高该子序列的处理效率。应当清楚的是,本申请实施例仅以上述方法举例说明确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列的过程,并不限于上述方法,上述确定新的子序列的方法还可以是其他优化算法,本公开对此不作限制。

[0083] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本公开并不受所描述的动作顺序的限制,因为依据本公开,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于可选实施例,所涉及的动作和模块并不一定是本公开所必须的。

[0084] 进一步需要说明的是,虽然图1的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,这些步骤可以以其它的顺序执行。而且,图1中的至少一部分步骤可以包括多个子步骤或者多个阶段,这些子步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,这些子步骤或者阶段的执行顺序也不必然是依次进行,而是可以与其它步骤或者其它步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

[0085] 图6示出本申请一实施例提供的装置图,装置610包括:

[0086] 指令获取模块601,用于获取当前子序列及其并行度信息;

[0087] 判断模块602,用于将所述当前子序列的并行度与第一预设并行度进行比较;

[0088] 插入额外指令模块603,用于当所述当前子序列的并行度不符合所述第一预设并行度时,在所述当前子序列中插入额外指令,得到新的子序列,并确定所述新的子序列为当前子序列。

[0089] 在一个可能地实施例中,所述插入额外指令模块603还包括指令移动模块,所述指令移动模块可以用于至少移动一次所述额外指令的位置,获得至少一个新的子序列,确定所述至少一个新的子序列的并行度,确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列为所述当前子序列。

[0090] 在一个可选地实施例中,插入的额外指令与所述当前子序列中指令之间不存在数据竞争关系。

[0091] 在一种可能的实现方式中,所述指令移动模块还包括区间步长确定模块,所述区间步长确定模块还可以用于:

[0092] 获取所述额外指令的运动区间以及运动步长;

[0093] 根据所述额外指令的运动步长,每次将所述额外指令从当前位置移动至目标位

置,获得至少一个新的子序列,

[0094] 其中,所述当前位置和所述目标位置均位于所述额外指令的运动区间内。

[0095] 在一个可能地实施例中,所述指令移动模块还包括并行度确定模块,所述并行度确定模块,用于确定指令序列的并行度,所述确定子序列的并行度,包括:

[0096] 若所述子序列中相邻的两个指令块中第二指令块中指令与第一指令块中至少一个指令在预设时间段内访问同一个部件,则将所述两个指令块划分在不同的指令块中;

[0097] 若所述子序列中相邻的两个指令块中第二指令块中的指令与第一指令块中所有指令在预设时间段内访问不同的部件,则将所述两个指令块划分在相同的指令块中;

[0098] 其中,所述子序列每个指令块中包括至少一条指令,根据所述新的子序列中所述指令块,确定所述新的子序列的并行度。

[0099] 可选地,所述装置还用于将所述指令块用指令运行图的形式表示。

[0100] 可选地,所述装置还用于当所述相邻两个指令块中都包括至少两条指令时,在所述相邻指令块之间插入Nope块,所述Nope块不在所述并行度的计算范围。

[0101] 在一种可能的实现方式中,所述指令移动模块还包括确定子序列模块,所述确定子序列模块还用于:

[0102] 若当前新的子序列的并行度小于上一新的子序列的并行度,则根据所述当前新的子序列中插入指令的运动方向,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度;

[0103] 若当前新的子序列的并行度大于或者等于上一新的子序列的并行度,则根据所述当前新的子序列中所述额外指令的运动方向及预设接受概率,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度。

[0104] 在一种可能的实现方式中,所述区间步长确定模块还包括指令拆分合并模块,所述指令拆分合并模块,用于在若在后指令块中的指令数量大于所述额外指令的运动步长时,则将所述在后指令块拆分为至少一个子指令块,之后执行所述根据所述子序列中额外指令的运动步长,每次将至少一个所述子序列中的额外指令从当前位置移动至目标位置,获得至少一个新的待处理指令序列;

[0105] 在将所述额外指令从当前位置移动至目标位置后,若在预设时间段,所述当前位置的后一指令与在先指令块中的指令访问不同的部件时,则将所述当前位置的后一指令合并至所述在先指令块;

[0106] 其中,所述在先指令块是指位于所述额外指令的当前位置之前的指令块,所述在后指令块是指位于所述额外指令的当前位置之后的指令块。

[0107] 本公开提供的指令序列处理装置610能够实现图1-5所示方法实施例中的各个步骤,并实现相同的技术效果,为避免重复,这里不再赘述。

[0108] 应该理解,上述的装置实施例仅是示意性的,本披露的装置还可通过其它的方式实现。例如,上述实施例中所述单元/模块的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式。例如,多个单元、模块或组件可以结合,或者可以集成到另一个系统,或一些特征可以忽略或不执行。

[0109] 所述作为分离部件说明的单元或模块可以是物理上分开的,也可以不是物理上分开的。作为单元或模块说明的部件可以是物理单元,也可以不是物理单元,即可以位于一个

装置中,或者也可以分布到多个装置上。本披露中实施例的方案可以根据实际的需要选择其中的部分或者全部单元来实现。

[0110] 另外,若无特别说明,在本披露各个实施例中的各功能单元/模块可以集成在一个单元/模块中,也可以是各个单元/模块单独物理存在,也可以两个或两个以上单元/模块集成在一起。上述集成的单元/模块既可以采用硬件的形式实现,也可以采用软件程序模块的形式实现。

[0111] 所述集成的单元/模块如果以硬件的形式实现时,该硬件可以是数字电路,模拟电路等等。硬件结构的物理实现包括但不限于晶体管,忆阻器等等。若无特别说明,所述人工智能处理器可以是任何适当的硬件处理器,比如CPU、GPU、FPGA、DSP和ASIC等等。若无特别说明,所述存储单元可以是任何适当的磁存储介质或者磁光存储介质,比如,阻变式存储器RRAM(Resistive Random Access Memory)、动态随机存取存储器DRAM(Dynamic Random Access Memory)、静态随机存取存储器SRAM(Static Random-Access Memory)、增强动态随机存取存储器EDRAM(Enhanced Dynamic Random Access Memory)、高带宽内存HBM(High-BandwidthMemory)、混合存储立方HMC(Hybrid Memory Cube)等等。

[0112] 所述集成的单元/模块如果以软件程序模块的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储器中。基于这样的理解,本披露的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储器中,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备等)执行本披露各个实施例所述方法的全部或部分步骤。而前述的存储器包括:U盘、只读存储器(ROM,Read-OnlyMemory)、随机存取存储器(RAM,Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0113] 本申请还揭露了一种计算机存储介质,其存储计算机程序,其中,所述计算机程序是计算机执行如图1示的指令处理方法的步骤。

[0114] 本申请还揭露了一种存储介质,其存储计算机程序,其中,所述计算机程序被计算机设备执行时,实现上述任一实施例的指令处理方法的步骤。具体地,计算机设备执行上述的计算机程序时,实现如下步骤:

[0115] 获取当前子序列及所述当前子序列的并行度;

[0116] 判断所述当前子序列的并行度是否符合第一预设并行度;

[0117] 在所述当前子序列的并行度不符合所述第一预设并行度时,在所述当前子序列中插入额外指令,得到新的子序列,并将所述新的子序列重新确定为所述当前子序列。

[0118] 应当清楚的是,该计算机程序被计算机设备执行时,其实现的步骤与上述指令处理方法中的各个步骤相对应,具体可参见上文的描述,此处不再一一赘述。

[0119] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中未详述的部分,可以参见其他实施例的相关描述。上述实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0120] 根据以下条款可更好的理解前述内容:

[0121] 条款1:一种指令处理方法,所述方法包括:

- [0122] 获取当前子序列及所述当前子序列的并行度；
- [0123] 判断所述当前子序列的并行度是否符合第一预设并行度；
- [0124] 在所述当前子序列的并行度不符合所述第一预设并行度时，在所述当前子序列中插入额外指令，得到新的子序列，并将所述新的子序列重新确定为所述当前子序列。
- [0125] 条款2:根据条款1所述的方法，所述在所述当前子序列中插入额外指令，得到新的子序列，并将所述新的子序列重新确定为当前子序列还包括：
- [0126] 至少移动一次所述额外指令的位置，获得至少一个新的子序列及所述至少一个新的子序列的并行度，确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列为所述当前子序列。
- [0127] 条款3:根据条款1所述方法，所述额外指令与所述当前子序列中指令之间不存在数据竞争关系。
- [0128] 条款4:根据条款2所述方法，所述至少移动一次所述额外指令的位置，获得至少一个新的子序列，包括：
- [0129] 获取所述额外指令的运动区间以及运动步长；
- [0130] 根据所述额外指令的运动步长，每次将所述额外指令从当前位置移动至目标位置，获得至少一个新的子序列，
- [0131] 其中，所述当前位置和所述目标位置均位于所述额外指令的运动区间内。
- [0132] 条款5:根据条款2所述方法，所述确定子序列的并行度，包括：
- [0133] 若所述子序列中相邻的两个指令块中第二指令块中指令与第一指令块中至少一个指令在预设时间段内访问同一个部件，则将所述两个指令块划分在不同的指令块中；
- [0134] 若所述子序列中相邻的两个指令块中第二指令块中的指令与第一指令块中所有指令在预设时间段内访问不同的部件，则将所述两个指令块划分在相同的指令块中；
- [0135] 根据所述新的子序列中所述指令块，确定所述新的子序列的并行度；
- [0136] 其中，所述每个指令块中包括至少一条指令。
- [0137] 条款6:根据条款5所述方法，所述方法还包括：
- [0138] 当所述相邻两个指令块中都包括至少两条指令时，在所述相邻指令块之间插入Nope块。
- [0139] 条款7:根据条款1-6任一项所述方法，述方法还包括：
- [0140] 将所述当前子序列用指令运行图的形式表示。
- [0141] 条款8:根据条款2-6任一项所述方法，所述确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列，包括：
- [0142] 若当前新的子序列的并行度小于上一新的子序列的并行度，则根据所述当前新的子序列中插入指令的运动方向，继续获取下一新的子序列，直至所述新的子序列的并行度符合所述第二预设并行度；
- [0143] 若当前新的子序列的并行度大于或者等于上一新的子序列的并行度，则根据所述当前新的子序列中所述额外指令的运动方向及预设接受概率，继续获取下一新的子序列，直至所述新的子序列的并行度符合所述第二预设并行度。
- [0144] 条款9:根据条款4所述方法，所述将所述额外指令从当前位置移至目标位置，包括：

[0145] 若在后指令块中的指令数量大于所述额外指令的运动步长时,则将所述在后指令块拆分为至少一个子指令块,之后执行所述根据所述子序列中额外指令的运动步长,每次将至少一个所述子序列中的额外指令从当前位置移动至目标位置,获得至少一个新的待处理指令序列;

[0146] 在将所述额外指令从当前位置移动至目标位置后,若在预设时间段,所述当前位置的后一指令与在先指令块中的指令访问不同的部件时,则将所述当前位置的后一指令合并至所述在先指令块;

[0147] 其中,所述在先指令块是指位于所述额外指令的当前位置之前的指令块,所述在后指令块是指位于所述额外指令的当前位置之后的指令块。

[0148] 条款10:一种计算机设备,包括处理器和存储器,所述存储器中存储有计算机程序,所述处理器执行所述计算机程序时,实现如条款1-9任一项所述的方法的步骤。

[0149] 条款11:一种计算机存储介质,所述计算机存储介质中存储有能够被处理器执行的计算机程序,所述处理器执行所述计算机程序时,实现如条款1-9任一项所述的方法的步骤。

[0150] 条款12:一种指令处理装置,所述处理装置包括:

[0151] 指令获取模块,用于获取当前子序列及其并行度信息;

[0152] 判断模块,用于将所述当前子序列的并行度与第一预设并行度进行比较;

[0153] 插入额外指令模块,用于当所述当前子序列的并行度不符合所述第一预设并行度时,在所述当前子序列中插入额外指令,得到新的子序列,并将所述新的子序列重新确定为所述当前子序列。

[0154] 条款13:根据条款12所述的装置,所述插入额外指令模块还用于:

[0155] 至少移动一次所述额外指令的位置,获得至少一个新的子序列及所述至少一个新的子序列的并行度,确定所述至少一个新的子序列的并行度中符合第二预设并行度所对应的新的子序列,并将所述符合第二预设并行度的新的子序列作为所述当前子序列。

[0156] 条款14:根据条款12所述装置,所述额外指令与所述当前子序列中指令之间不存在数据竞争关系。

[0157] 条款15:根据条款13所述装置,所述插入额外指令模块还用于:

[0158] 获取所述额外指令的运动区间以及运动步长;

[0159] 根据所述额外指令的运动步长,每次将所述额外指令从当前位置移动至目标位置,获得至少一个新的子序列,

[0160] 其中,所述当前位置和所述目标位置均位于所述额外指令的运动区间内。

[0161] 条款16:根据条款13所述装置,所述插入额外指令模块还用于:

[0162] 若所述子序列中相邻的两个指令块中第二指令块中指令与第一指令块中至少一个指令在预设时间段内访问同一个部件,则将所述两个指令块划分在不同的指令块中;

[0163] 若所述子序列中相邻的两个指令块中第二指令块中的指令与第一指令块中所有指令在预设时间段内访问不同的部件,则将所述两个指令块划分在相同的指令块中;

[0164] 根据所述新的子序列中所述指令块,确定所述新的子序列的并行度;

[0165] 其中,所述每个指令块中包括至少一条指令。

[0166] 条款17:根据条款16所述装置,所述装置还用于:

[0167] 当所述相邻两个指令块中都包括至少两条指令时,在所述相邻指令块之间插入Nope块。

[0168] 条款18:根据条款12-17任一项所述装置,所属装置还用于:

[0169] 将所述当前子序列用指令运行图的形式表示。

[0170] 条款19:根据条款13-17任一项所述装置,所述插入额外指令模块还用于:

[0171] 若当前新的子序列的并行度小于上一新的子序列的并行度,则根据所述当前新的子序列中插入指令的运动方向,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度;

[0172] 若当前新的子序列的并行度大于或者等于上一新的子序列的并行度,则根据所述当前新的子序列中所述额外指令的运动方向及预设接受概率,继续获取下一新的子序列,直至所述新的子序列的并行度符合所述第二预设并行度。

[0173] 条款20:根据条款15所述装置,所述插入额外指令模块还用于:

[0174] 若在后指令块中的指令数量大于所述额外指令的运动步长时,则将所述在后指令块拆分为至少一个子指令块,之后执行所述根据所述子序列中额外指令的运动步长,每次将至少一个所述子序列中的额外指令从当前位置移动至目标位置,获得至少一个新的待处理指令序列;

[0175] 在将所述额外指令从当前位置移动至目标位置后,若在预设时间段,所述当前位置的后一指令与在先指令块中的指令访问不同的部件时,则将所述当前位置的后一指令合并至所述在先指令块;

[0176] 其中,所述在先指令块是指位于所述额外指令的当前位置之前的指令块,所述在后指令块是指位于所述额外指令的当前位置之后的指令块。

[0177] 以上对本申请实施例进行了详细介绍,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。

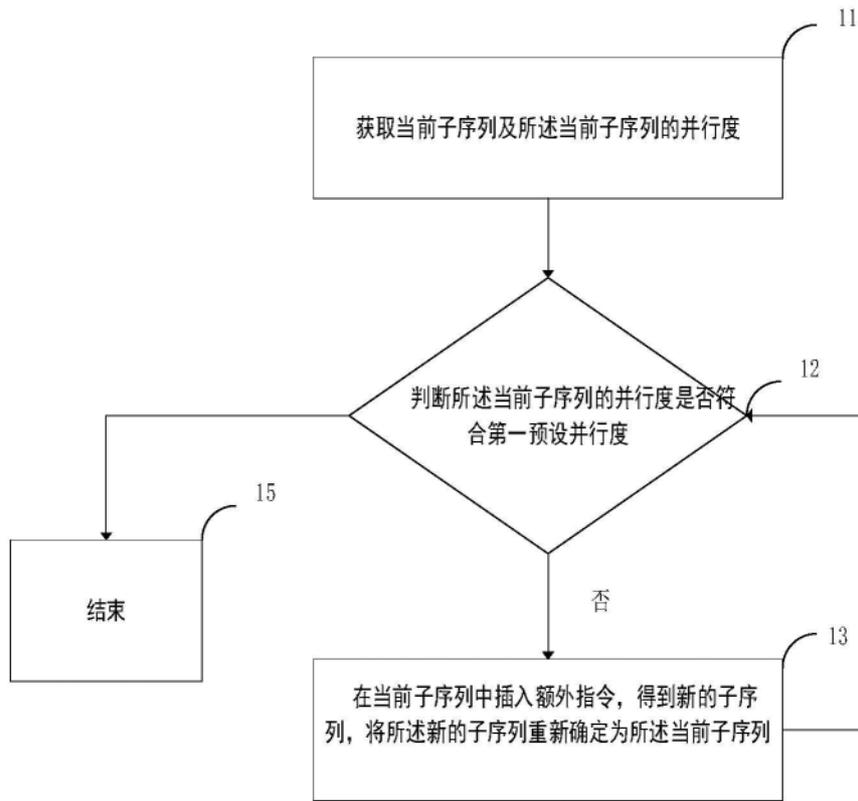
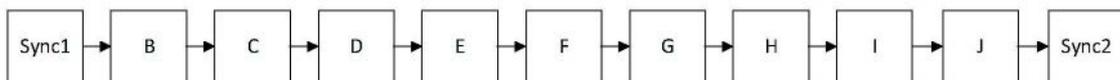


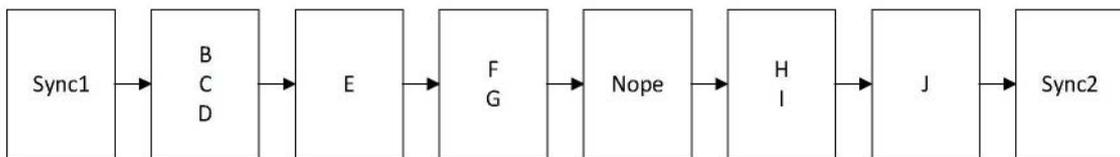
图1



图2



A



B

图3

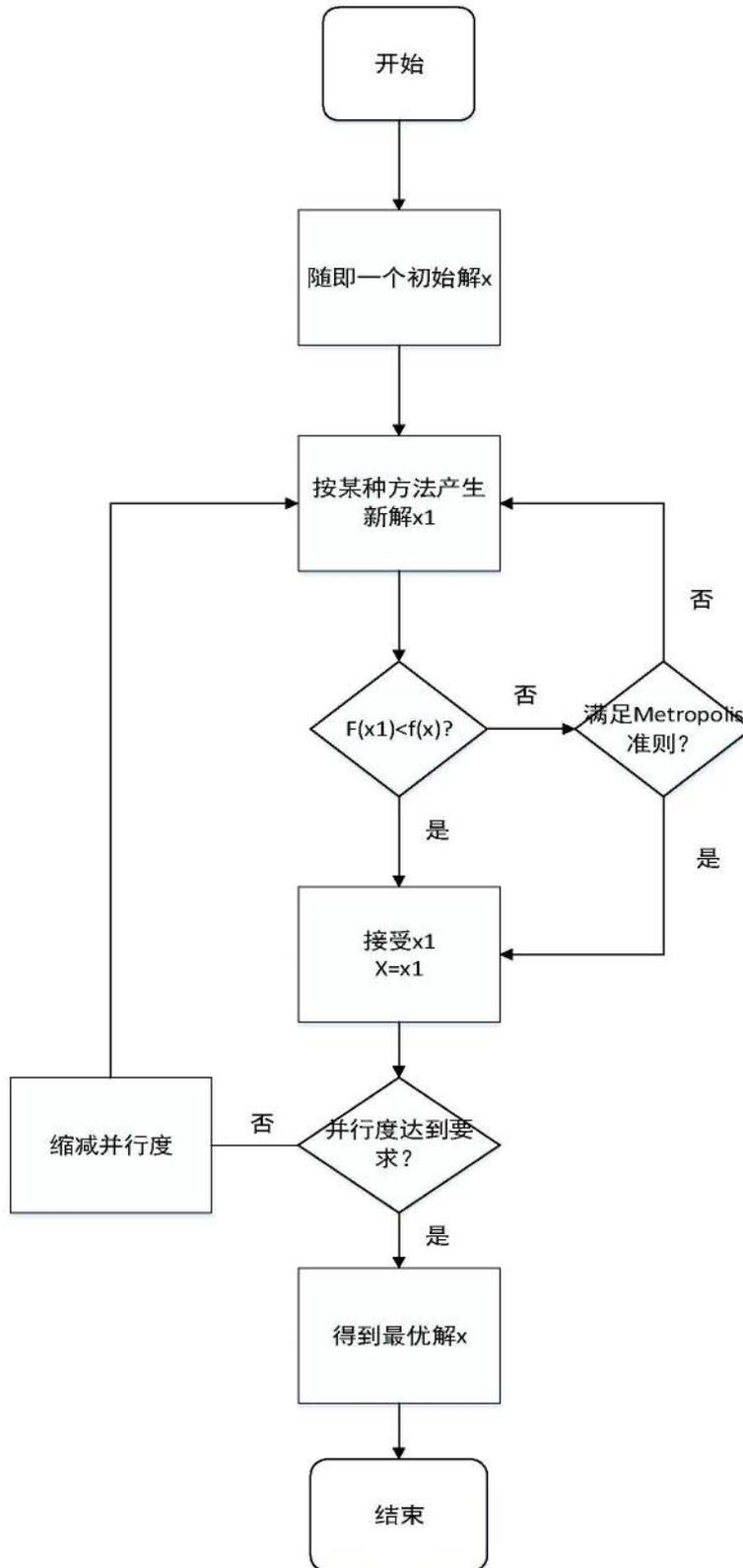
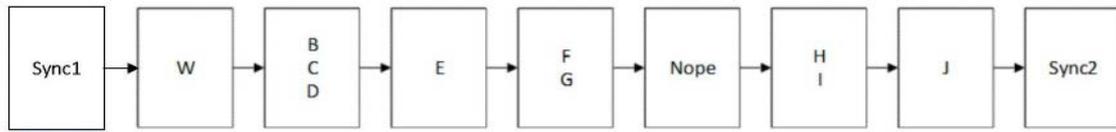
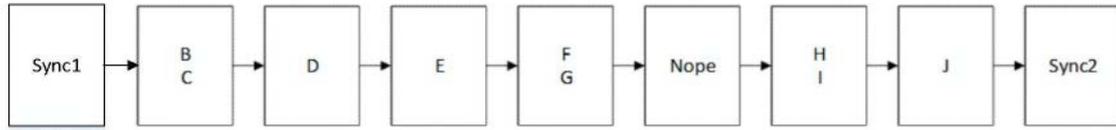


图4



A



B

图5

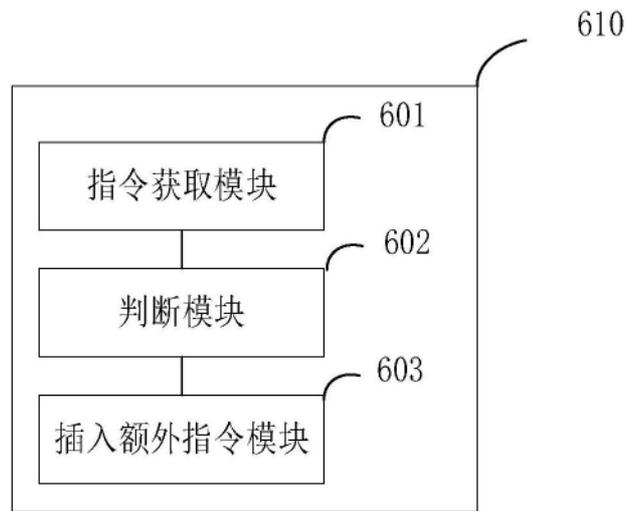


图6



图7