



(12) 发明专利申请

(10) 申请公布号 CN 120181251 A

(43) 申请公布日 2025. 06. 20

(21) 申请号 202510100793.2

(22) 申请日 2025.01.22

(71) 申请人 北京量子信息科学研究院

地址 100193 北京市海淀区中关村软件园
二期国际与区域协同创新中心A座

(72) 发明人 汪景波 黄晨 赵茜 许宏泽
胡孟军

(74) 专利代理机构 北京律和信知识产权代理事
务所(普通合伙) 11446

专利代理师 谢清萍 刘兴

(51) Int. Cl.

G06N 10/20 (2022.01)

G06N 10/40 (2022.01)

G06N 10/80 (2022.01)

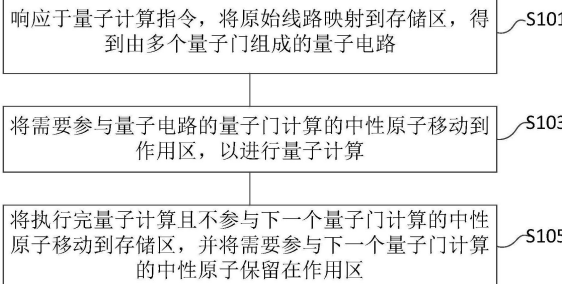
权利要求书2页 说明书8页 附图4页

(54) 发明名称

基于分区架构的量子计算方法和量子计算装置

(57) 摘要

本申请提出一种基于分区架构的量子计算方法和量子计算装置,所述分区架构包括储存区和作用区,所述量子计算方法包括:响应于量子计算指令,将原始线路映射到所述储存区,得到由多个量子门组成的量子电路;将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算;将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述储存区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。根据本申请的实施例,通过采用分区架构,将量子门计算的量子比特限定在作用区,未参与量子门计算的量子比特限定在储存区,从而降低了里德堡全局光束对未参与计算的量子比特影响。



1. 一种基于分区架构的量子计算方法,其特征在于,所述分区架构包括储存区和作用区,所述量子计算方法包括:

响应于量子计算指令,将原始线路映射到所述存储区,得到由多个量子门组成的量子电路;

将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算;

将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。

2. 根据权利要求1所述的量子计算方法,其特征在于,将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算,包括:

将需要参与所述量子门计算的中性原子移动到所述作用区,并通过SLM阵列和AOD阵列进行量子计算。

3. 根据权利要求2所述的量子计算方法,其特征在于,参与所述量子门计算的中性原子包括第一中性原子和第二中性原子,

将需要参与所述量子门计算的中性原子移动到所述作用区,并通过SLM阵列和AOD阵列进行量子计算,包括:

将所述第一中性原子和所述第二中性原子映射到所述SLM阵列中;

将所述第一中性原子从所述SLM阵列中移动到所述AOD阵列中;

利用所述AOD阵列,将所述第一中性原子移动至所述SLM阵列中的第二中性原子相邻位置,以进行量子计算。

4. 根据权利要求2所述的量子计算方法,其特征在于,在将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算之前,还包括:

确定所述量子电路中多个量子门的作用顺序。

5. 根据权利要求4所述的量子计算方法,其特征在于,确定所述量子电路中多个量子门的作用顺序,包括:

采用尽早执行量子门操作方式,确定所述量子电路中多个量子门的作用顺序,以使得总的量子门的执行时间最少。

6. 根据权利要求3所述的量子计算方法,其特征在于,在将所述第一中性原子和所述第二中性原子映射到所述SLM阵列中之前,

将需要参与所述量子门计算的中性原子移动到所述作用区,并通过SLM阵列和AOD阵列进行量子计算,还包括:

利用模拟退火算法确定需要参与所述量子门计算的中性原子在所述作用区的排布。

7. 根据权利要求1所述的量子计算方法,其特征在于,在将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区之前,还包括:

利用模拟退火算法,确定当前参与量子计算的中性原子的排布;

根据确定的排布,将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。

8. 一种基于分区架构的量子计算装置,其特征在于,所述分区架构包括储存区和作用

区,所述量子计算装置包括:

映射单元,用于响应于量子计算指令,将原始线路映射到所述存储区,得到由多个量子门组成的量子电路;

量子计算单元,用于将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算;

移动单元,用于将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。

9.一种电子设备,包括:

处理器;以及

存储器,存储有计算机程序,当所述计算机程序被所述处理器执行时,使得所述处理器执行如权利要求1-7中任一项所述的量子计算方法。

10.一种非瞬时性计算机可读存储介质,其上存储有计算机可读指令,当所述指令被处理器执行时,使得所述处理器执行如权利要求1-7中任一项所述的量子计算方法。

基于分区架构的量子计算方法和量子计算装置

技术领域

[0001] 本申请涉及量子计算领域,具体而言,涉及一种基于分区架构的量子计算方法和量子计算装置、电子设备以及非瞬时性计算机可读存储介质。

背景技术

[0002] 近些年来,得益于激光控制技术以及磁光阱束缚技术的发展,中性原子特别是里德堡原子量子计算的发展得到极大进步,越来越受到科研人员以及量子计算产业界的关注。中性原子用被光镊束缚的不带电的原子当做量子比特,量子比特之间通过将原子激发到里德堡态实现相互作用,实现量子比特之间的信息交换,从而在中性原子中也可以实现通用量子计算。将原子当做量子比特,其具有高度全同性、高量子比特寿命、高可操控性,原子里德堡态之间相互作用的特性又使得其具备高度并行性。中性原子一般以二维阵列的方式进行量子比特操控,这使得中性原子相比于诸如超导量子计算、离子阱量子计算平台等其他硬件平台,在扩展性上具有极大的优势,在现有激光功率下,人们已经实现了六千多个里德堡原子量子比特的束缚和操控,这些都显示了里德堡原子作为量子计算的巨大潜力。

[0003] 由于量子算法需要通过编译才能有效地和硬件控制平台结合,真正发挥量子算力的优势,而对于里德堡原子平台,其中一个很重要的特性是量子比特可以在空间上受控的移动,这使得里德堡原子平台中量子比特的连通性、并行性都会随着量子电路线路的执行不断改变。

[0004] 本申请的发明人发现,如何在中性原子量子计算平台实现高效的量子算法的编译和运行工作,降低里德堡全局光束对未参与计算的量子门的影响是当前亟需解决的问题。

发明内容

[0005] 本申请旨在提出一种基于分区架构的量子计算方法和量子计算装置、电子设备以及非瞬时性计算机可读存储介质,以解决里德堡全局光束对未参与计算的量子门的影响问题。

[0006] 根据本申请的一方面,提出一种基于分区架构的量子计算方法,所述分区架构包括储存区和作用区,所述量子计算方法包括:响应于量子计算指令,将原始线路映射到所述储存区,得到由多个量子门组成的量子电路;将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算;将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述储存区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。

[0007] 根据一些实施例,将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算,包括:将需要参与所述量子门计算的中性原子移动到所述作用区,并通过SLM阵列和AOD阵列进行量子计算。

[0008] 根据一些实施例,参与所述量子门计算的中性原子包括第一中性原子和第二中性原子,将需要参与所述量子门计算的中性原子移动到所述作用区,并通过SLM阵列和AOD阵

列进行量子计算,包括:将所述第一中性原子和所述第二中性原子映射到所述SLM阵列中;将所述第一中性原子从所述SLM阵列中移动到所述AOD阵列中;利用所述AOD阵列,将所述第一中性原子移动至所述SLM阵列中的第二中性原子相邻位置,以进行量子计算。

[0009] 根据一些实施例,在将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算之前,还包括:确定所述量子电路中多个量子门的作用顺序。

[0010] 根据一些实施例,确定所述量子电路中多个量子门的作用顺序,包括:采用尽早执行量子门操作方式,确定所述量子电路中多个量子门的作用顺序,以使得总的量子门的执行时间最少。

[0011] 根据一些实施例,在将所述第一中性原子和所述第二中性原子映射到所述SLM阵列中之前,将需要参与所述量子门计算的中性原子移动到所述作用区,并通过SLM阵列和AOD阵列进行量子计算,还包括:利用模拟退火算法确定需要参与所述量子门计算的中性原子在所述作用区的排布。

[0012] 根据一些实施例,在将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区之前,还包括:利用模拟退火算法,确定当前参与量子计算的中性原子的排布;根据确定的排布,将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。

[0013] 根据本申请的一方面,提出一种基于分区架构的量子计算装置,所述分区架构包括储存区和作用区,所述量子计算装置包括:映射单元,用于响应于量子计算指令,将原始线路映射到所述存储区,得到由多个量子门组成的量子电路;量子计算单元,用于将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算;移动单元,用于将执行完量子计算且不参与下一个量子门计算的中性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。

[0014] 根据本申请的一方面,提出一种电子设备,包括:处理器;以及存储器,存储有计算机程序,当所述计算机程序被所述处理器执行时,使得所述处理器执行如前任一实施例所述的量子计算方法。

[0015] 根据本申请的一方面,提出一种非瞬时性计算机可读存储介质,其上存储有计算机可读指令,当所述指令被处理器执行时,使得所述处理器执行如前任一实施例所述的量子计算方法。

[0016] 根据本申请的实施例,通过采用分区架构,将量子门计算的量子比特限定在作用区,未参与量子门计算的量子比特限定在存储区,从而降低了里德堡全局光束对未参与计算的量子比特影响。

[0017] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性的,并不能限制本申请。

附图说明

[0018] 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。通过参照附图详细描述其示例实施例,本申请的上述和其他目标、特征及优点将变得更加显而易见。

- [0019] 图1示出了根据本申请示例实施例的一种基于分区架构的量子计算方法流程图。
- [0020] 图2示出了根据本申请示例实施例的一种SLM阵列示意图。
- [0021] 图3示出了根据本申请示例实施例的一种AOD阵列中的光镊点位移动方法示意图。
- [0022] 图4示出了根据本申请示例实施例的一种尽早执行量子门的操作方式,确定量子电路中多个量子门的作用顺序的伪代码实现过程示意图。
- [0023] 图5示出了根据本申请示例实施例的一种利用模拟退火算法确定需要参与量子门计算的中性原子在作用区的排布的伪代码实现过程示意图。
- [0024] 图6示出了根据本申请示例实施例的一种基于分区架构的量子计算装置框图。
- [0025] 图7示出根据本申请示例性实施例的一种电子设备。

具体实施方式

[0026] 现在将参考附图更全面地描述示例实施例。然而,示例实施例能够以多种形式实施,且不应被理解为限于在此阐述的实施例;相反,提供这些实施例使得本申请将全面和完整,并将示例实施例的构思全面地传达给本领域的技术人员。在图中相同的附图显示表示相同或类似的部分,因而将省略对它们的重复描述。

[0027] 所描述的特征、结构或特性可以以任何合适的方式结合在一个或更多实施例中。在下面的描述中,提供许多具体细节从而给出对本公开的实施例的充分理解。然而,本领域技术人员将意识到,可以实践本公开的技术方案而没有这些特定细节中的一个或更多,或者可以采用其它的方式、组元、材料、装置或操作等。在这些情况下,将不详细示出或描述公知结构、方法、装置、实现、材料或者操作。

[0028] 附图中所示的流程图仅是示例性说明,不是必须包括所有的内容和操作/步骤,也不是必须按所描述的顺序执行。例如,有的操作/步骤还可以分解,而有的操作/步骤可以合并或部分合并,因此实际执行的顺序有可能根据实际情况改变。

[0029] 本申请的说明书和权利要求书及上述附图中的术语“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。此外,术语“包括”和“具有”以及它们任何变形,意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有限定于已列出的步骤或单元,而是可选地还包括没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其他步骤或单元。

[0030] 在描述本申请的实施例之前,首先对本申请中出现的术语进行解释。

[0031] 量子编译:根据量子算法的逻辑任务需要执行的步骤映射到物理比特上,并进行相应的物理操作以实现相应的逻辑操作。

[0032] 中性原子:不带电的超冷原子。

[0033] 原子阵列:根据一定规则排列整齐的中性原子。

[0034] SLM(Spatial Light Modulator,简称空间光调制器):用来产生光镊阵列的设备。激光通过空间光调制器可以在二维平面上产生多个光镊点位,这些光镊点位的数量可控,位置不可重叠、不可移动,此外不受任何限制。

[0035] AOD(Acousto Optical Deflectors,简称声光偏转器):用来产生光镊阵列的设备。激光通过声光偏转器将在二维平面产生呈网格结构的光镊点位阵列,光镊点位数量为行个数乘以列个数,光镊阵列的每一行(列)可以单独移动,但一行(列)在移动时不能穿过

另一行(列)。

[0036] 路由:在执行量子算法和量子电路任务时,调整量子比特的位置或排列,以满足量子计算机硬件的物理限制,尤其是受限于量子比特之间的直接连接或相互作用。

[0037] 下面结合附图,对根据本申请的具体实施例进行详细说明。

[0038] 图1示出了根据本申请示例实施例的一种基于分区架构的量子计算方法流程图,如图1所示的量子计算方法包括步骤S101、S103和S105。下面以图1为例,对根据本申请示例实施例的一种基于分区架构的量子计算方法进行详细说明。

[0039] 根据本申请的实施例,分区架构包括储存区和作用区。本申请的实施例在中性原子量子计算平台实现。

[0040] 如图1所示,在步骤S101,响应于量子计算指令,将原始线路映射到存储区,得到由多个量子门组成的量子电路。

[0041] 由于任意的单量子比特门可以通过激光单独操作,其他多比特量子门可以转化为单量子门和/或两量子比特门的组合。因此,在本申请的实施例中,将原始线路中的任意两量子比特门转化为中性原子量子计算平台中中性原子可原生执行的CZ门,得到一个仅有两量子比特门的量子电路。

[0042] 在步骤S103,将需要参与量子电路的量子门计算的中性原子移动到作用区,以进行量子计算。

[0043] 根据本申请的实施例,在步骤S103之前,还需要确定量子电路中多个量子门的作用顺序,并按执行顺序将同时操作的量子门分为一组,称为一个阶段。

[0044] 在具体的实施例中,采用尽早执行量子门的操作方式,确定量子电路中多个量子门的作用顺序,以使得总的量子门的执行时间最少。

[0045] 根据本申请的实施例,在步骤S103,将需要参与量子门计算的中性原子移动到作用区,并通过SLM阵列和AOD阵列进行量子计算。

[0046] 在中性原子平台中,量子比特能够在不同光镊阵列之间灵活转移,且一个光镊点位最多只能容纳一个原子。其中,光镊点位可由两种设备产生,一种是空间光调制器,另一种为声光偏转器。

[0047] 空间光调制器产生的光镊点位可以在二维平面上的任意位置,但是这些光镊点位无法移动,在本实施例中,将多个空间光调制器产生的光镊点位称作SLM阵列,如图2所示。

[0048] 声光偏转器产生的光镊点位必须为平面上的晶格阵列结构,在本实施例中,将多个声光偏转器产生的光镊点位称作AOD阵列,AOD光镊阵列可以移动,但需要整行或者整列的移动,如图3所示,为一个3行3列的AOD阵列,AOD必须整行或列一起移动,图3所示为第3行整体的移动。

[0049] 目前技术水平最多仅能同时设置一套SLM原子阵列和一套AOD原子阵列。

[0050] 在本实施例中,原子可以在AOD阵列与SLM阵列之间转换。SLM阵列中的光镊点位在计算过程中可以开启或关闭,AOD阵列中的光镊阵列在计算过程中必须整行或者整列的开启与关闭。原子移动过程中AOD阵列的排序需要保持,移动过程中一行不能越过另一行,一列不能越过另一列。任意的单量子比特门可以通过激光单独操作。两个量子比特只有距离处在里德堡阻塞半径内,并同时被里德堡激光激发时,才能执行两比特CZ门。当两个量子比特执行两比特门时,两个原子的里德堡阻塞半径内不能有其他原子。

[0051] 根据本申请的实施例,参与量子门计算的中性原子包括第一中性原子和第二中性原子。当执行两个SLM行之间的两量子比特门操作时:

[0052] 首先,将第一中性原子和第二中性原子映射到SLM阵列中;

[0053] 然后,将第一中性原子从SLM阵列中移动到AOD阵列中;

[0054] 最后,利用AOD阵列,将第一中性原子移动至SLM阵列中的第二中性原子相邻位置,以进行量子计算。

[0055] 根据一些实施例中,在将第一中性原子和第二中性原子映射到所述SLM阵列中之前,在步骤S103,还需要利用模拟退火算法确定需要参与量子门计算的中性原子在作用区的排布,以降低量子门的操作距离以及移动次数。

[0056] 在步骤S105,将执行完量子计算且不参与下一个量子门计算的中性原子移动到存储区,并将需要参与下一个量子门计算的中性原子保留在作用区。

[0057] 根据本申请的实施例,在步骤S105之前,还需要利用模拟退火算法,确定当前参与量子计算的中性原子的排布;并根据确定的排布,将执行完量子计算且不参与下一个量子门计算的中性原子移动到存储区,并将需要参与下一个量子门计算的中性原子保留在作用区。

[0058] 在一些实施例中,步骤S103和S105需要重复执行多次,直至将整个量子线路执行完成。

[0059] 根据图1所示的实施例,通过采用分区架构,将量子门计算的量子比特限定在作用区,未参与量子门计算的量子比特限定在存储区,从而降低了里德堡全局光束对未参与计算的量子比特影响。

[0060] 根据另一些实施例,通过将中性原子从固定位置的光镊点位转移到可移动的光镊点位,实现了非相邻量子比特的耦合。并通过将需要参与下一个量子门计算的中性原子保留在作用区,不仅降低了量子比特的移动次数,而且减少了移动带来的保真度,降低了量子电路执行的路由开销。

[0061] 图4示出了根据本申请示例实施例的一种尽早执行量子门的操作方式,确定量子电路中多个量子门的作用顺序的伪代码实现过程示意图。

[0062] 如图4所示,为了利用中性原子的高度可并行性,采用尽早(As Soon As Possible,简称ASAP)调度策略,将量子电路划分为不同的阶段,每个阶段内的所有两量子比特门都可以并行执行。也即,对于任意给定的量子电路,都能尽可能早地安排量子门操作,使得总的门执行阶段和时间都最小化。

[0063] 如图4所示,首先,初始化一个空的调度列表list_scheduling,用于存储每个阶段需要执行的量子门操作。

[0064] 然后,初始化列表list_qubit_stage,用于记录每个量子比特的当前操作阶段,并设置初始值为0。

[0065] 之后,遍历所有的门操作gate:

[0066] 例如,首先,取出门操作的两个目标量子比特的当前阶段 t_{q0} 和 t_{q1} ;然后,确定当前量子门的操作阶段 t_g 为这两个量子比特阶段的最大值,以确保这两个量子比特之前的操作已完成。如果操作阶段 t_g 尚未包含在调度列表list_scheduling中,则创建新的操作阶段。

[0067] 之后,将当前门的索引添加到对应时间片 t_g ,并更新两个量子比特的操作阶段,表

示它们被安排到了新的操作阶段。

[0068] 最后,返回调度列表`list_scheduling`。

[0069] 根据图4所示的实施例,采用尽早调度策略,将量子电路划分为不同的阶段,每个阶段内的所有两量子比特门都可以并行执行,以使得总的量子门的执行时间最少,并实现了量子比特操作的高并行性编译。

[0070] 图5示出了根据本申请示例实施例的一种利用模拟退火算法确定需要参与量子门计算的中性原子在作用区的排布的伪代码实现过程示意图。

[0071] 在图5所示的实施例中,将消耗函数`cost`为能量`EE`,其值为每一阶段参与双量子比特门的原子之间距离的加权和,其中权重随着阶段增加而递减,具体公式为:对于每一阶段`stage`,权重为 $\max(1-0.1 \times \text{stage}, 0.1)$,距离为量子比特映射位置之间的欧几里得距离,总能量为所有阶段加权距离的累加。

[0072] 本实施例使用模拟退火方法确定需要参与量子门计算的中性原子在作用区的排布,目标是最小化移动时间和退相干效应。

[0073] 如图5所述,首先,设定初始温度并生成量子比特的初始放置方案,计算其成本值`cost`;然后,进入迭代优化过程。

[0074] 在每次迭代中,在满足移动限制条件(例如,只能按行或列顺序移动)的情况下,随机交换两个量子比特的位置,并计算新的放置方案的成本和并行移动数量。如果新的方案比当前方案成本更低且并行移动数量不增加,则直接接受;如果新的方案成本更高,则依据概率决定是否接受,目的是在初始阶段探索更大的解空间以避免陷入局部最优。之后,随着温度逐渐降低,解空间的探索范围收缩,算法趋于收敛。整个过程通过不断降低温度,根据预设的冷却计划逐步收敛到全局最优解,最终输出优化后的量子比特放置方案,以提高量子计算操作的效率和保真度。

[0075] 根据图5所示的实施例,通过模拟物理退火过程中的“降温”操作,以避免陷入局部最优,从而找到全局最优解,从而使原子全局移动距离最短、消耗时间最少,得到了整体保真度相对较高的排列方式,解决了在中性原子量子计算平台高效的量子算法的编译和运行的问题,使得在中性原子量子计算硬件平台上实现大规模量子算法成为现实,为该领域的快速发展奠定了坚实基础。

[0076] 上面主要从方法的角度对本申请实施例进行了介绍。本领域技术人员应该很容易意识到,结合本文中所公开的实施例所描述的各示例的操作或步骤,本申请能够以硬件或硬件和计算机软件的结合形式来实现。本领域技术人员可以对每个特定的操作或方法使用不同方式来实现所描述的功能,这种实现不应认为超出本申请的范围。

[0077] 下面描述本申请的装置实施例。对于本申请装置实施例中未说明的细节,可参照本申请方法实施例。

[0078] 图6示出了根据本申请示例实施例的一种基于分区架构的量子计算装置框图,如图6所示的量子计算装置包括映射单元601、量子计算单元603和移动单元605。

[0079] 根据本申请的实施例,分区架构包括储存区和作用区。其中,映射单元601用于响应于量子计算指令,将原始线路映射到所述存储区,得到由多个量子门组成的量子电路;量子计算单元603用于将需要参与所述量子电路的量子门计算的中性原子移动到所述作用区,以进行量子计算;移动单元605用于将执行完量子计算且不参与下一个量子门计算的中

性原子移动到所述存储区,并将需要参与所述下一个量子门计算的中性原子保留在所述作用区。

[0080] 图7示出根据本申请示例性实施例的一种电子设备。下面参照图7来描述根据本申请的这种实施方式的电子设备200。图7显示的电子设备200仅仅是一个示例,不应对本申请实施例的功能和使用范围带来任何限制。

[0081] 如图7所示,电子设备200以通用计算设备的形式表现。电子设备200的组件可以包括但不限于:至少一个处理单元210、至少一个存储单元220、连接不同系统组件(包括存储单元220和处理单元210)的总线230、显示单元240等。

[0082] 其中,存储单元存储有程序代码,程序代码可以被处理单元210执行,使得处理单元210执行本说明书描述的根据本申请各种示例性实施方式的方法。例如,处理单元210可以执行如图1中所示的方法。

[0083] 存储单元220可以包括易失性存储单元形式的可读介质,例如随机存取存储单元(RAM) 2201和/或高速缓存存储单元2202,还可以进一步包括只读存储单元(ROM) 2203。

[0084] 存储单元220还可以包括具有一组(至少一个)程序模块2205的程序/实用工具2204,这样的程序模块2205包括但不限于:操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0085] 总线230可以为表示几类总线结构中的一种或多种,包括存储单元总线或者存储单元控制器、外围总线、图形加速端口、处理单元或者使用多种总线结构中的任意总线结构的局域总线。

[0086] 电子设备200也可以与一个或多个外部设备300(例如键盘、指向设备、蓝牙设备等)通信,还可与一个或者多个使得用户能与该电子设备200交互的设备通信,和/或与使得该电子设备200能与一个或多个其它计算设备进行通信的任何设备(例如路由器、调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口250进行。并且,电子设备200还可以通过网络适配器260与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。网络适配器260可以通过总线230与电子设备200的其它模块通信。应当明白,尽管图中未示出,可以结合电子设备200使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0087] 通过以上的实施方式的描述,本领域的技术人员易于理解,这里描述的示例实施方式可以通过软件实现,也可以通过软件结合必要的硬件的方式来实现。根据本申请实施方式的技术方案可以以软件产品的形式体现出来,该软件产品可以存储在一个非易失性存储介质(可以是CD-ROM, U盘, 移动硬盘等)中或网络上,包括若干指令以使得一台计算设备(可以是个人计算机、服务器、或者网络设备等)执行根据本申请实施方式的上述方法。

[0088] 软件产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以为但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0089] 计算机可读存储介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了可读程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。可读存储介质还可以是可读存储介质以外的任何可读介质,该可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。可读存储介质上包含的程序代码可以用任何适当的介质传输,包括但不限于无线、有线、光缆、RF等等,或者上述的任意合适的组合。

[0090] 可以以一种或多种程序设计语言的任意组合来编写用于执行本申请操作的程序代码,程序设计语言包括面向对象的程序设计语言—诸如Java、C++等,还包括常规的过程式程序设计语言—诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。在涉及远程计算设备的情形中,远程计算设备可以通过任意种类的网络,包括局域网(LAN)或广域网(WAN),连接到用户计算设备,或者,可以连接到外部计算设备(例如利用因特网服务提供商来通过因特网连接)。

[0091] 上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被一个该设备执行时,使得该计算机可读介质实现前述功能。

[0092] 本领域技术人员可以理解上述各模块可以按照实施例的描述分布于装置中,也可以进行相应变化唯一不同于本实施例的一个或多个装置中。上述实施例的模块可以合并为一个模块,也可以进一步拆分成多个子模块。

[0093] 根据本申请的实施例,提出一种计算机程序,包括计算机程序或指令,该计算机程序或指令被处理器执行时,可以执行以上描述的方法。

[0094] 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,以上实施例的说明仅用于帮助理解本申请的方法及其核心思想。同时,本领域技术人员依据本申请的思想,基于本申请的具体实施方式及应用范围上做出的改变或变形之处,都属于本申请保护的范围。综上所述,本说明书内容不应理解为对本申请的限制。

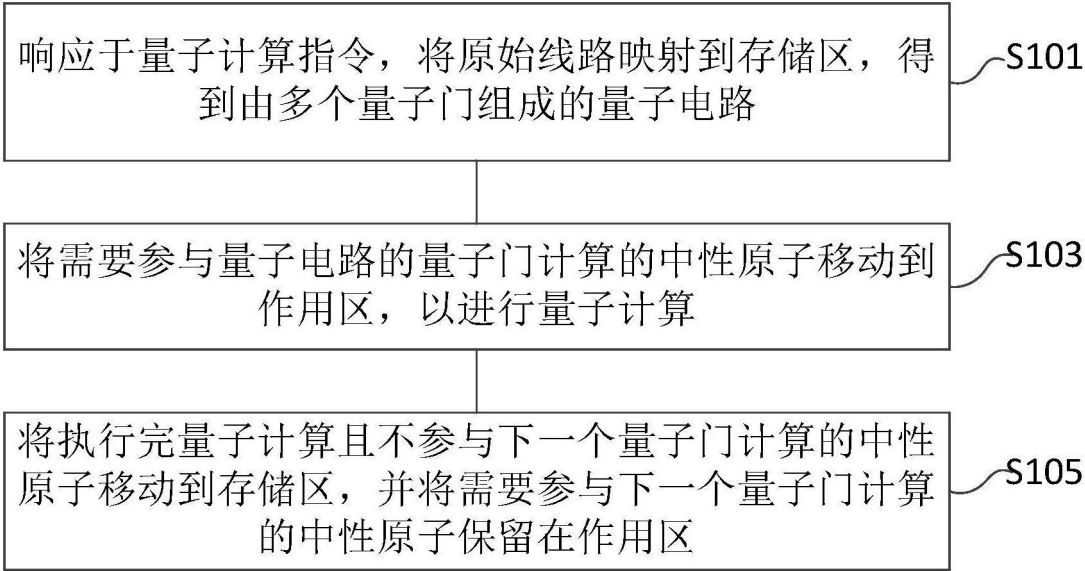


图1



图2

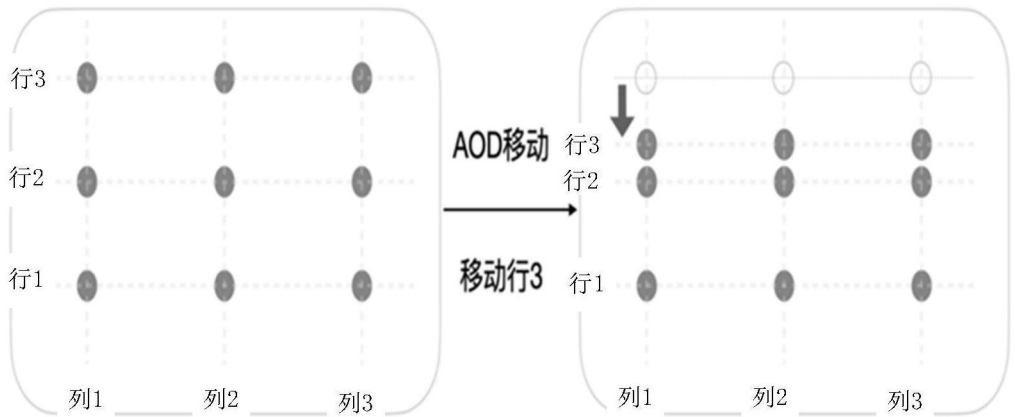


图3

Algorithm 1 Scheduling Procedure

```

1: procedure SCHEDULING
2:   list_scheduling  $\leftarrow$  []
3:   list_qubit_stage  $\leftarrow$  [0 for each qubit in n_q]
4:   for each gate  $g$  in gate_list do
5:      $t_{q0} \leftarrow$  list_qubit_stage[gate[0]]
6:      $t_{q1} \leftarrow$  list_qubit_stage[gate[1]]
7:      $t_g \leftarrow \max(t_{q0}, t_{q1})$ 
8:     if  $t_g \geq \text{len}(\text{list\_scheduling})$  then
9:       Append empty list to list_scheduling
10:    end if
11:    Append index of  $g$  to list_scheduling[ $t_g$ ]
12:     $t_g \leftarrow t_g + 1$ 
13:    list_qubit_stage[gate[0]]  $\leftarrow t_g$ 
14:    list_qubit_stage[gate[1]]  $\leftarrow t_g$ 
15:  end for
16:  return list_scheduling
17: end procedure

```

图4

Algorithm 2: Qubit Placement via Simulated Annealing

Input: Quantum circuit structure, Initial placement of qubits in the storage zone, AOD constraints (rows/columns can only move in order), Parameters: initial temperature T , cost function, max iterations

Output: Optimized qubit placement that minimizes movement time and decoherence

Initialize:

- Set temperature T to initial temperature
- Generate initial mapping of qubits
- Compute initial cost based on current mapping
- Set iteration count to 0

while $T > \text{frozen_temperature}$ **do**

- for** *each iteration* **do**
 - Randomly swap positions of selected qubits within allowed regions
 - Compute new cost based on the current mapping
 - Calculate the number of parallel movements for the new configuration
 - if** $\text{new_cost} < \text{current_cost}$ **and** $\text{new_parallel_movements} \leq \text{current_parallel_movements}$ **then**
 - | Accept new placement
 - else**
 - | Calculate probability of acceptance for worse solutions **if** $\text{random}() < \text{probability}$ **then**
 - | Accept new placement with penalty
 - | Revert to previous placement
- Reduce temperature T according to cooling schedule

Output optimized placement with minimized total movement time and enhanced fidelity

图5



图6

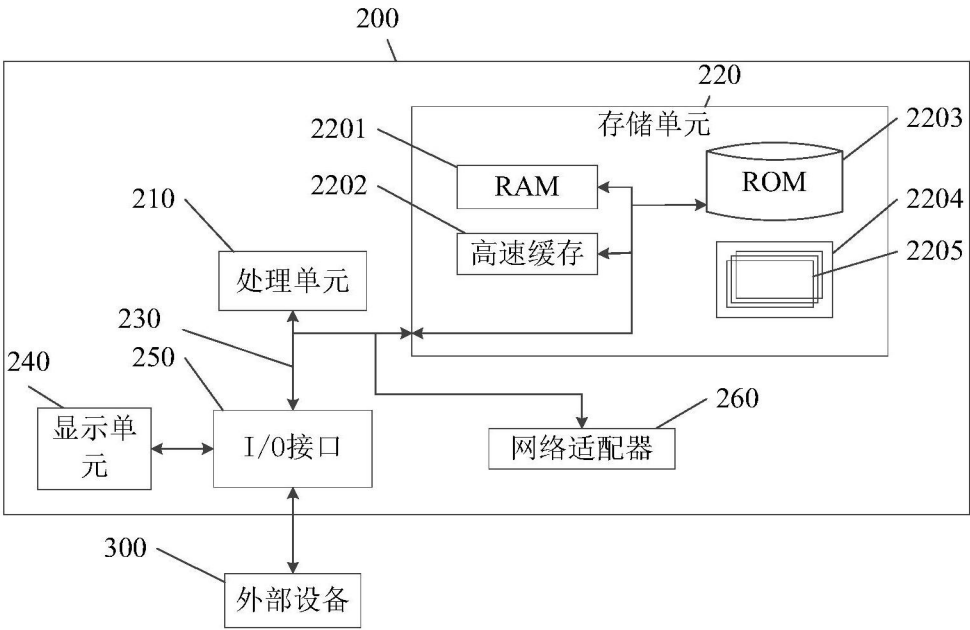


图7