

多逻辑比特表面码结构设计及其逻辑 CNOT 门实现*

权东晓^{1)2)†} 吕晓杰¹⁾ 张雯菲¹⁾

1) (西安电子科技大学通信工程学院, 西安 710071)

2) (西安电子科技大学, 量子信息协同创新中心, 西安 710071)

(2023 年 7 月 14 日收到; 2023 年 11 月 21 日收到修改稿)

量子计算因具有并行处理能力, 相比于经典计算有着指数级的加速, 但量子系统具有脆弱性, 极易受到噪声的影响, 量子纠错码是克服量子噪声的有效手段. 量子表面码是一种拓扑稳定子码, 由于其结构上的最近邻居特点和较高的容错阈值, 表面码在大规模容错量子计算方面具有巨大的潜力. 目前已有的基于边界的表面码均为编码一个逻辑比特的表面码, 本文主要研究基于边界如何实现多逻辑量子比特的编码, 包括设计表面码的结构, 根据结构找出对应的稳定子和逻辑操作, 进一步根据稳定子设计出基于稳定子实现的编码线路; 在研究基于测量和纠正的单量子比特间 CNOT 实现原理和基于融合操作和分割操作的单逻辑量子比特表面码间 CNOT 门实现原理的基础上, 优化了基于融合操作和分割操作的单逻辑量子比特表面码间 CNOT 门实现方案, 将其扩展到所设计的多逻辑量子比特表面码上实现了多逻辑量子比特表面码之间的 CNOT 操作, 并通过仿真验证量子线路的正确性. 本文设计的多逻辑比特表面码克服了单比特表面码不能密铺于量子芯片的缺点且提高了某些逻辑操作的长度, 提高了容错能力. 基于联合测量的思想降低了对辅助比特的要求且减小了实现过程中对量子资源的需求.

关键词: 量子表面码, 多逻辑比特编码, 逻辑 CNOT 门, 融合操作, 分割操作

PACS: 03.67.Lx, 03.67.Hk

DOI: 10.7498/aps.73.20231138

1 引言

1.1 研究背景及意义

1982 年, Feynman^[1] 首次提出了量子计算的概念, 它是运用量子力学的规律, 对量子数据比特进行调控的一种新型的计算方式. 量子计算机在解决一些超复杂的问题时其性能要远优于经典计算机, 例如 NP 难题, 当 N 很大时经典计算机无法解决, 但使用 Shor 量子算法^[2] 可以将其转化为 P 问题而解决. 因此, 对量子计算机的研究和容错量子计算的应用已经成为亟待解决的问题. 2012 年,

Preskill^[3] 提出了“量子霸权”, 他指出如果可以将量子计算机拓展到 50 个量子位上, 那么量子计算机的计算能力将超过世界上所有的计算机. 相比于经典计算的线性计算方式, 量子计算^[4-7] 可以实现并行计算, 它们之间的区别是: 经典 n 比特只能表示 2^n 个状态中的一个状态, 而 n 量子比特可以同时表示 2^n 个状态的叠加态, 因此能够实现并行计算. 但是量子比特具有脆弱性, 在量子计算过程中很容易受到噪声的影响. 量子纠错码^[8-10] 将信息量子比特编入纠缠量子比特中, 这样当量子比特在存储、传输和计算过程中发生错误时, 我们能够通过测量物理比特之间的关系而推断具体的错误, 因此

* 国家自然科学基金 (批准号: 62001351) 和陕西省重点研发计划 (批准号: 2019ZDLGY09-02) 资助的课题.

† 通信作者. E-mail: dxquan@xidian.edu.cn

量子纠错码成为发展量子计算机的必要条件.

由于量子表面码的物理量子比特被放在拉丁格中, 稳定子只关系到附近的几个量子比特, 因此稳定子的测量也只和附近的几个量子比特有关, 有利于物理实现. 因此表面码成为实现容错量子计算的重点研究对象之一, 谷歌、IBM、Intel 和中国科学技术大学等目前公开的量子芯片结构都利于表面码的实现^[11–14]. 表面码主要包括基于边界的表面码和基于缺陷的表面码^[15], 基于缺陷的表面码可以通过删除更多的稳定子构造内部边界而编码多个逻辑量子比特, 而基于边界的矩形结构的表面码只能编码一个逻辑量子比特. 随着可编码逻辑量子比特的增多, 传输和处理的信息也会随之增多. 因此, 设计使用少量的物理量子比特来编码多位逻辑比特的编码方案是一项十分具有意义的研究.

CNOT 门是通用量子计算中一个非常重要的逻辑门, 目前已有研究表明可以基于表面码的融合和分割操作来实现编码单逻辑量子比特的表面码之间的 CNOT 操作^[16]. 本文首先对已有研究方案提出优化, 然后研究如何将 CNOT 门的实现方法扩展到编码多逻辑量子比特的表面码之间. 利用边界编码多个逻辑比特能够实现量子比特的密铺, 在一定程度上降低对物理比特的需求, 提高编码效率, 从而可以降低量子计算实际实现的复杂性和难度. CNOT 门的实现可为容错量子计算提供基础, 因此本文的研究对量子计算具有重要的理论意义.

1.2 研究现状

1995 年, Shor^[9] 首先提出了量子纠错码的概念, 并提出了用 9 个物理量子比特编码 1 个逻辑量子比特, 可以纠正任意单比特错误的码距为 3 的 [9, 1, 3] Shor 码. 1996 年, Steane^[10] 提出了量子纠错方案的一般表述, 并提出了用 7 个物理量子比特编码 1 个逻辑量子比特, 可以纠正任意单比特错误的码距为 3 的 [7, 1, 3] Steane 码. 表面码是由 Kitaev^[17–19] 提出的复曲面码发展演变而来的, 在复曲面码的基础上, Kitaev 和 Bravyi^[20] 以及 Meyer 和 Freedman^[21] 分别发展出了平面版本的量子码模型. 2003 年, Preskill 等^[22] 发现了表面码优良的容错性, 并通过堆叠多层表面的三维结构实现了逻辑 CNOT 操作. 2006 年, Raussendorf 等^[23] 发现通过编织变换可以在单个表面上实现逻辑 CNOT 操作, 并评估了错误率阈值. 2008 年, 邢莉娟等^[24]

给出了 CSS 量子卷积码的一种新型编译码方法, 该方法描述了编译码网络结构, 将码字的基态转换为信息多项式与生成多项式的乘积, 利用量子态上的多项式乘法操作实现了编译码网络. Flower 等^[25] 于 2009 年提出了基于表面码的高阈值通用量子计算, 其二维近邻耦合量子比特晶格结构, 接近 1% 的高错误率阈值, 自然的非对称和可调强度错误校正, 以及低开销、任意长度逻辑门的特性, 使其成为迄今为止最好和最实用的量子计算方案. 同年, 一种基于超导硬件的表面码架构被 DiVincenzo^[26] 提出. 2012 年, Fowler 等^[15] 使用编织等方法, 在表面码上实现了逻辑通用门集, 包括逻辑 CNOT 门、逻辑 H 门等, 并且全面阐述了表面码量子计算的相关内容. 同年, Horsman 等^[16] 实现了表面码晶格的融合和分割, 进而实现了通用的量子计算.

2014 年, Tomita 和 Svore^[27] 在真实量子噪声下研究了表面码的编码问题, 提出如何减少编码所用门的数量和时隙. 2017 年, Brown 等^[28] 提出了一个统一的框架来描述表面码编码方法, 包括用表面码的角落扭曲缺陷, 基于此提出了一种新的逻辑编码方法, 同时利用码变换, 在不减小码距的情况下, 实现了通用逻辑门集. 2018 年, Litinski 等^[29] 提出了一种基于表面码旋转操作的容错量子计算方案, 该方案消除了单量子比特 Clifford 门的时间开销, 实现了远程多目标 CNOT 门, 其时间开销仅随控制目标的分离而呈对数缩放. 同年, Krylov 和 Lukac^[30] 提出用于量子电路设计的量子进化算法. 2020 年, Beaudrap 和 Horsman^[31] 提出用量子计算图解语言 the ZX calculus 与表面码中的晶格操作建立联系, 使纠错操作更加直观, 并且能在大规模范围下验证, 从而实现更细粒度的晶格操作和纠正; Camps 和 van Beeumen^[32] 提出了使用编码块近似完成了量子线路合成. Shirakawa 等^[33] 于 2021 年提出了一种量子经典混合算法来编码给定的量子态. 同年, Wang 等^[34] 基于机器学习的思想, 构建了一个卷积神经网络译码器来纠正复曲面码中的错误, 译码器的阈值精度达到 10.8%, 更接近于最优阈值. 2022 年, Marques 等^[35] 研究出了一种在表面码中进行错误检测的逻辑量子位运算方法; Kumari 等^[36] 提出了一种双面环形码, 可以用 n 个数据量子比特编码至少 $n/3$ 个逻辑量子比特; Chen 等^[37] 提出了一种仅由 Clifford 门组成的系统高效的量子电路来模拟表面码模型的基态,

能够在任意平面晶格上实现二维复曲面编码的基态; Chen 等^[38]提出了一种面向量子通信系统的低开销容错纠错方案, 该方案简化了电路深度开销和辅助量子态开销, 且适用于通用量子稳定子码. 此外, Xue 等^[39]提出了一种基于周期性表面码强化学习的量子信息保护方案; Ding 等^[40]提出了一种基于量子拓扑稳定子彩色码形态的神经网络译码方法, 该方法将彩色码投射到表面码上, 利用深度 Q 网络对彩色码的解码过程进行迭代训练, 在训练后可以达到较快的预测速度和较好的纠错阈值. 2023 年, Siegel 等^[41]提出了一种用于缺陷晶格表面码量子纠错的新型自适应方法, 结合适当的缺陷检测算法和对已识别区域的隔离, 可以保留有限码大小的量子纠错优势. 从以上介绍能看出, 不管是在理论上还是实验实现上量子表面码都得到了广泛的重视, 取得了快速的发展. 但是, 量子计算理论和实际实现之间的障碍仍然无法跨越, 研究如何提高编码效率以及如何实现通用量子门集对容错量子计算的最终实现具有重要的理论意义.

2 基于边界的多逻辑比特量子表面码结构及编码线路设计

2.1 单逻辑比特表面码结构

表面码由一些具有特定拓扑结构的量子比特构成, 包括最早提出的表面码以及旋转 45° 后形成的更节省物理比特的表面码^[16]. 图 1 为一个码距为 3 的旋转表面码.

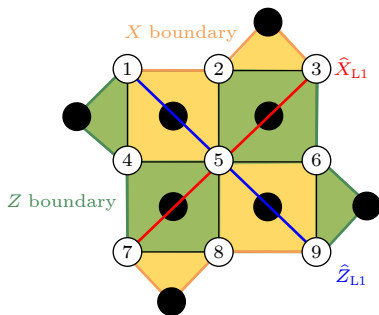


图 1 3×3 表面码的结构图

Fig. 1. Structure diagram of 3×3 surface code.

如图 1 所示, 白色比特为数据比特, 黑色比特为测量比特, 黄色区域为 X 型稳定子, 包括: $X_1X_2X_4X_5$, $X_2X_3X_7X_8$, $X_5X_6X_8X_9$; 绿色区域为 Z 型稳定子, 包括 $Z_1Z_4Z_2Z_3Z_5Z_6$, $Z_4Z_5Z_7Z_8$, Z_6Z_9 .

可以看到不同类型的稳定子间都有 0 个或 2 个共同的数据比特, 而同类型的稳定子本身互相对易, 因此所有稳定子之间满足对易性; 任何一个稳定子都不能由其他稳定子直积得到, 因此也满足独立性. 稳定子码编码的逻辑比特的数目等于物理比特数目和稳定子数目的差, 因此, 图 1 可以编码 1 个逻辑比特.

表面码有两种边界: X 边界是由 X 型稳定子所形成的边界, Z 边界是由 Z 型稳定子所形成的边界. 连接两个 X 边界且与所经过的每个 Z 型稳定子有偶数个共同数据比特的连线上的所有数据比特 X 操作的直积就是 X 逻辑操作 X_L , 信息比特要选择 X_L 所涉及的量子比特; 类似地连接两个 Z 边界且与所经过的每个 X 型稳定子有偶数个共同数据比特的连线上所有数据比特 Z 操作的直积就是 Z 逻辑操作 Z_L . 据此, 可以写出图 1 中表面码的 X 逻辑操作: $X_{L1} = X_3X_5X_7$, $X_{L2} = X_2X_5X_7$, $X_{L3} = X_3X_5X_8$, $X_{L4} = X_2X_5X_8$; Z 逻辑操作: $Z_{L1} = Z_1Z_5Z_9$, $Z_{L2} = Z_4Z_5Z_9$, $Z_{L3} = Z_1Z_5Z_6$ 和 $Z_{L4} = Z_4Z_5Z_6$. 容易验证, X 逻辑操作和 Z 逻辑操作与所有的稳定子对易, 两者反对易.

2.2 一种可以编码 3 位逻辑比特的表面码

2.2.1 表面码结构设计

在表面码中, 数据比特个数与稳定子个数之差等于可以编码逻辑比特的个数, 所以想要同时编码多位逻辑比特, 可以固定数据比特的个数后减少稳定子的个数. 减少稳定子个数的方法有两种: 一种是在表面码内部采用“挖洞”的方式来减少稳定子个数, 实际上是构造内部边界; 另一种是构造更多的互相交替的外边界, 能够形成更多连接同类型边界的逻辑操作, 从而可以编码多个逻辑比特. 经典的表面码结构包含 2 个 X 边界和 2 个 Z 边界, X 稳定子数目与 Z 稳定子数目相等, 可以编码一位逻辑比特.

本文提出了一种基于边界可以同时编码三位逻辑比特的表面码结构, 该表面码结构如图 2 所示, 由 12 个数据比特构成, 包含 5 个 X 型稳定子 $X_1X_2X_4$, $X_3X_6X_8$, $X_4X_6X_7X_9$, $X_5X_7X_{10}$, $X_9X_{11}X_{12}$ 以及 4 个 Z 型稳定子 $Z_1Z_3Z_4Z_6$, $Z_2Z_4Z_5Z_7$, $Z_6Z_8Z_9Z_{11}$, $Z_7Z_9Z_{10}Z_{12}$, 4 个 X 边界和 4 个 Z 边界, 因此可以编码 3 位逻辑比特.

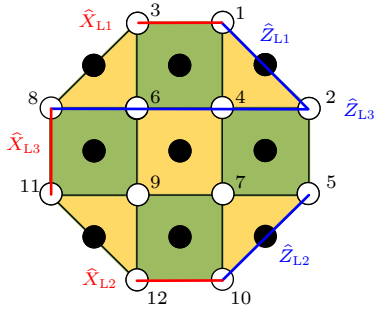


图 2 基于边界编码三位逻辑比特的表面码结构

Fig. 2. Structure of surface code based on boundary encoding three logical qubits.

2.2.2 逻辑操作的选择

逻辑操作的选择需要满足以下原则: 第一, 逻辑操作要与所有的稳定子对易; 第二, Z 逻辑操作 Z_{Li} 要与相应的 X 逻辑操作 X_{Li} 反对易, 同时与编码其他逻辑比特的 X 逻辑操作 $X_{Lj} (j \neq i)$ 对易; X 逻辑操作 X_{Li} 要与相应的 Z 逻辑操作 Z_{Li} 反对易, 同时与编码其他逻辑比特的 Z 逻辑操作 $Z_{Lj} (j \neq i)$ 对易. 在该表面码中, 边界是互相交替的 4 条 X 边界和 4 条 Z 边界, 但它们是不独立的. 理论上可以选择其中的 3 个 Z 边界作为逻辑 X 操作, 3 个 X 边界作为逻辑 Z 操作, 但是当固定了 X 逻辑操作后, 会出现 Z 逻辑操作同时与两个 X 逻辑操作反对易的情况, 因此逻辑操作不能全部选择最短的. 这里第 3 个逻辑 Z 操作选择为: $Z_{L3} = Z_8 Z_6 Z_4 Z_2$, 或将其等价到边界上, 写成 $Z'_{L3} = Z_8 Z_3 Z_1 Z_2$. 最终选择的逻辑操作见表 1.

表 1 图 2 所示表面码的逻辑操作

Table 1. Logical operation of the surface code shown in Fig. 2.

X_L	Z_L
$X_{L1} = X_1 X_3$	$Z_{L1} = Z_1 Z_2$
$X_{L2} = X_{10} X_{12}$	$Z_{L2} = Z_5 Z_{10}$
$X_{L3} = X_8 X_{11}$	$Z_{L3} = Z_8 Z_6 Z_4 Z_2$

2.2.3 编码线路的设计与仿真

编码线路主要包括基于稳定子测量和纠正的编码线路^[15]以及基于稳定子实现的编码线路^[42]. 因为基于稳定子测量的编码线路总是随机地塌缩, 需要根据测量结果进行纠正操作才能正确编码, 消耗的资源较多, 本文基于稳定子的实现设计编码线路. 选定物理比特 1, 10, 8 作为信息比特, 编码线路的第 1 部分是逻辑操作的实现, 即利用信息比特

控制相应的逻辑操作的其他比特; 接下来利用 H 门和 $CNOT$ 门逐个实现 X 型稳定子, 得到的编码线路如图 3 所示.

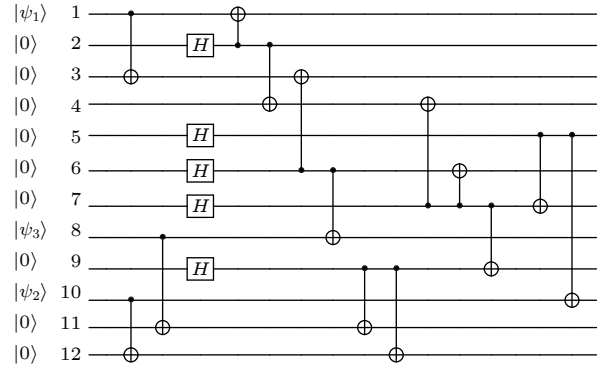


图 3 图 2 所示编码三位逻辑比特表面码的编码线路图

Fig. 3. Quantum encoding circuit for the surface code shown in Fig. 2.

采用 IBM 的 Qiskit 对所设计的编码线路进行仿真. 令 $|\bar{c}\bar{b}\bar{a}\rangle, a, b, c \in (0, 1)$ 表示信息比特 1 处于 $|a\rangle$, 信息比特 10 处于 $|b\rangle$, 信息比特 8 处于 $|c\rangle$ 时对应的逻辑码字, 分别设置 3 个信息比特 1, 10, 8 的值为 $|0\rangle$ 和 $|1\rangle$, 得到 $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle$ 和 $|111\rangle$ 的码字见附录 A. 容易验证码字满足所有的稳定子, 且所有的逻辑操作能够按照理论在码字之间转换: $X_{L1}|\bar{c}\bar{b}\bar{a}\rangle = |\bar{c}\bar{b}\bar{X}a\rangle$; $X_{L2}|\bar{c}\bar{b}\bar{a}\rangle = |\bar{c}\bar{X}b\bar{a}\rangle$; $X_{L3}|\bar{c}\bar{b}\bar{a}\rangle = |\bar{X}c\bar{b}\bar{a}\rangle$; 当 $i = 1, a = 0$ 或 $i = 2, b = 0$ 或 $i = 3, c = 0$ 时, $Z_{Li}|\bar{c}\bar{b}\bar{a}\rangle = |\bar{c}\bar{b}\bar{a}\rangle$; 当 $i = 1, a = 1$ 或 $i = 2, b = 1$ 或 $i = 3, c = 1$ 时, $Z_{Li}|\bar{c}\bar{b}\bar{a}\rangle = -|\bar{c}\bar{b}\bar{a}\rangle$. 验证了表面码结构设计以及编码线路设计的正确性.

3 基于融合与分割的单逻辑量子比特表面码间 CNOT 门实现的优化

3.1 基于联合测量的单量子比特间 CNOT 门实现方法

图 4 是利用联合测量在单量子比特之间实现 CNOT 门的原理图, $|C\rangle$ 是控制位, $|T\rangle$ 是受控位, 实现过程中需要一个额外的初始化为 $|+\rangle$ 的辅助比特 M . 实现过程分为 4 步: 第 1 步, 对控制位 $|C\rangle$ 和辅助比特 M 进行 Z 基下的联合测量, 结果为 M_1 ; 第 2 步, 对辅助比特 M 和受控位 $|T\rangle$ 进行 X 基下的联合测量, 结果记为 M_2 ; 第 3 步, 对辅助比特 M 在 Z 基下进行测量, 结果为 M_3 ; 第 4 步, 根据

前 3 步的测量结果, 在控制位 $|C\rangle$ 添加逻辑门 Z^{M_2} , 在受控位 $|T\rangle$ 添加逻辑门 $X^{M_1+M_3}$ 作为纠正操作, 即当 $M_2 = 1$ 时, 添加 Z 门, 当 $M_1 + M_3 = 1$ 时, 添加 X 门.

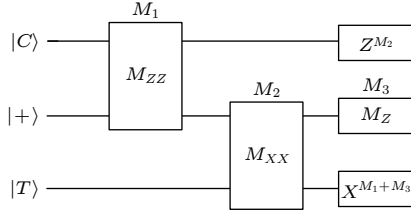


图 4 基于联合测量的单量子比特间 CNOT 门的实现原理
Fig. 4. CNOT gate implementation for single qubit based on joint measurement.

为了和融合分割操作的理论分析一致, 这里将联合测量理解为稳定子测量. 假设两个量子态分别为 $|\phi\rangle$ 和 $|\psi\rangle$, 进行联合 Z 测量, 相当于进行稳定子 $Z_1 Z_2$ 测量, 如果结果为 0, 表示满足这个稳定子, 这时候得到的态为

$$\frac{1}{\sqrt{2}} (I + Z_1 Z_2) |\phi\rangle |\psi\rangle. \quad (1)$$

如果结果为 1, 表示不满足这个稳定子, 或者理解为满足稳定子 $-Z_1 Z_2$, 这时候得到的态为

$$\frac{1}{\sqrt{2}} (I - Z_1 Z_2) |\phi\rangle |\psi\rangle. \quad (2)$$

上述两种情况可以合并为

$$\frac{1}{\sqrt{2}} (I + (-1)^M Z_1 Z_2) |\phi\rangle |\psi\rangle. \quad (3)$$

在 CNOT 的实现过程中, 基于上述理论进行分析, 具体分析过程如下:

$$|C\rangle = \alpha |0\rangle + \beta |1\rangle, \quad (4)$$

$$|T\rangle = m |0\rangle + n |1\rangle, \quad (5)$$

表 2 对辅助比特 M 在 Z 基测量后的输出结果

Table 2. Output states after the measurements of ancilla qubit M in the Z basis.

测量结果	输出态
$M_1=0, M_2=0, M_3=0$	$\alpha 00\rangle (m 0\rangle + n 1\rangle) + \beta 10\rangle (m 1\rangle + n 0\rangle)$
$M_1=0, M_2=0, M_3=1$	$\alpha 01\rangle (m 1\rangle + n 0\rangle) + \beta 11\rangle (m 0\rangle + n 1\rangle)$
$M_1=0, M_2=1, M_3=0$	$\alpha 00\rangle (m 0\rangle + n 1\rangle) - \beta 10\rangle (m 1\rangle + n 0\rangle)$
$M_1=0, M_2=1, M_3=1$	$-\alpha 01\rangle (m 1\rangle + n 0\rangle) + \beta 11\rangle (m 0\rangle + n 1\rangle)$
$M_1=1, M_2=0, M_3=0$	$\alpha 00\rangle (m 1\rangle + n 0\rangle) + \beta 10\rangle (m 0\rangle + n 1\rangle)$
$M_1=1, M_2=0, M_3=1$	$\alpha 01\rangle (m 0\rangle + n 1\rangle) + \beta 11\rangle (m 1\rangle + n 0\rangle)$
$M_1=1, M_2=1, M_3=0$	$-\alpha 00\rangle (m 1\rangle + n 0\rangle) + \beta 10\rangle (m 0\rangle + n 1\rangle)$
$M_1=1, M_2=1, M_3=1$	$\alpha 01\rangle (m 0\rangle + n 1\rangle) - \beta 11\rangle (m 1\rangle + n 0\rangle)$

$$M = |+\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle. \quad (6)$$

1) 对控制位 $|C\rangle$ 和辅助比特 M 进行 Z 基下的联合测量, 可以得到输出态为

$$\frac{1}{\sqrt{2}} ((\alpha |0\rangle + \beta |1\rangle) |+\rangle + (-1)^{M_1} (\alpha |0\rangle - \beta |1\rangle) |-\rangle) |T\rangle. \quad (7)$$

2) 对辅助比特 M 和受控位 $|T\rangle$ 进行 X 基下的联合测量, 即进行稳定子 $X_1 X_2$ 测量, 可以得到输出态为

$$\begin{aligned} & \frac{1}{2} ((\alpha |0\rangle + \beta |1\rangle) |+\rangle \\ & + (-1)^{M_1} (\alpha |0\rangle - \beta |1\rangle) |-\rangle) (m |0\rangle + n |1\rangle) \\ & + \frac{1}{2} (-1)^{M_2} ((\alpha |0\rangle + \beta |1\rangle) |+\rangle \\ & - (-1)^{M_1} (\alpha |0\rangle - \beta |1\rangle) |-\rangle) (m |1\rangle + n |0\rangle). \end{aligned} \quad (8)$$

3) 对辅助比特 M 进行 Z 基下的测量, 可以得到输出态, 见表 2.

4) 根据测量结果添加纠正操作, 无论哪一种情况, 通过纠正操作后可以得到最后的输出结果 (忽略系数) 为

$$\alpha |0\rangle (m |0\rangle + n |1\rangle) + \beta |1\rangle (n |0\rangle + m |1\rangle). \quad (9)$$

根据输出结果, 可以看到, 当 $|C\rangle$ 为 $|0\rangle$ 时, 输出 $|T\rangle$; 当 $|C\rangle$ 为 $|1\rangle$ 时, 输出 $X|T\rangle$, 符合 CNOT 门的逻辑功能, 实现了单量子比特间的 CNOT 门.

3.2 融合与分割操作

2012 年, Horsman 等^[16] 提出了基于晶格融合与分割的方式实现逻辑 CNOT 门. 这里首先介绍粗糙融合、粗糙分割、光滑融合和光滑分割的原理, 然后对 Horsman 等^[16] 提出的 CNOT 门的实现方法进行改进.

3.2.1 粗糙融合和粗糙分割

在基于边界的表面码中, X 边界称为光滑边界, Z 边界称为粗糙边界, 因此这里粗糙融合指的是对两个表面码的 Z 边界进行融合, 如图 5 所示. 对于两个分别编码一个逻辑比特码距为 d 的表面码, 左边记为 $|\bar{\phi}\rangle$, 右边记为 $|\bar{\psi}\rangle$, 在两个表面码的 Z 边界中间有 3 个辅助数据比特 Q_1, Q_2, Q_3 和 4 个测量比特充当媒介. $|\bar{\phi}\rangle$ 的逻辑操作为 $X_{L1} = X_1 X_2 X_3 X_4$, $Z_{L1} = Z_4 Z_5 Z_6 Z_7$, $|\bar{\psi}\rangle$ 的逻辑操作为 $X_{L2} = X_a X_b X_c X_d$, $Z_{L2} = Z_d Z_e Z_f Z_g$. 表面码粗糙融合的步骤为: 将 Q_1, Q_2, Q_3 都初始化为 $|0\rangle$, 则边界处的三端 Z 型稳定子变成了自动满足的四端 Z 型稳定子; 然后对两个表面码之间的 X 稳定子 $S_1^X = X_1 X_a X_{Q_1}$, $S_2^X = X_2 X_b X_{Q_2}$, $S_3^X = X_3 X_c X_{Q_3}$, $S_4^X = X_4 X_d X_{Q_3}$ 进行测量, 通过稳定子的测量过程就完成了融合操作. X 型稳定子如果满足, 则测量结果记为 0, 否则记为 1, 将 4 个 X 型稳定子的测量结果记为: M_1, M_2, M_3, M_4 . 因为 $S_1^X \otimes S_2^X \otimes S_3^X \otimes S_4^X = X_{L1} \otimes X_{L2}$, 所以 $M = M_1 \oplus M_2 \oplus M_3 \oplus M_4$ 表示联合逻辑 $X_{L1} X_{L2}$ 测量的结果. 粗糙融合的过程可以表示为

$$|\bar{\phi}\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\bar{\psi}\rangle = \alpha'|0\rangle + \beta'|1\rangle. \quad (10)$$

融合之后的状态为

$$\alpha|\psi\rangle_{\text{new}} + (-1)^M \beta|\bar{\psi}\rangle_{\text{new}} = \alpha'|\phi\rangle_{\text{new}} + \beta'|\bar{\phi}\rangle_{\text{new}}. \quad (11)$$

其中,

$$\begin{aligned} |\psi\rangle_{\text{new}} &= \alpha'|0\rangle_{\text{new}} + \beta'|1\rangle_{\text{new}}, \\ |\bar{\psi}\rangle_{\text{new}} &= \alpha'|1\rangle_{\text{new}} + \beta'|0\rangle_{\text{new}}, \\ |\phi\rangle_{\text{new}} &= \alpha|0\rangle_{\text{new}} + (-1)^M \beta|1\rangle_{\text{new}}, \\ |\bar{\phi}\rangle_{\text{new}} &= \alpha|1\rangle_{\text{new}} + (-1)^M \beta|0\rangle_{\text{new}}, \\ |0\rangle_{\text{new}} &= \frac{1}{\sqrt{2}}(|00\rangle + (-1)^M |11\rangle), \\ |1\rangle_{\text{new}} &= \frac{1}{\sqrt{2}}(|01\rangle + (-1)^M |10\rangle). \end{aligned}$$

上述公式不严谨, 融合以后中间的 3 个数据比特 Q_1, Q_2 和 Q_3 将和左右两边的表面码是纠缠在一起的, 因此 (11) 式表述的不够严格, 因为它并没有考虑这 3 个数据比特的状态.

为了便于分析, 对融合过程增加的第 4 个稳定子进行变形:

$$\begin{aligned} S_4^{X'} &= S_4^X \otimes S_1^X \otimes S_2^X \otimes S_3^X \\ &= X_1 X_2 X_3 X_4 X_a X_b X_c X_d = X_{L1} \otimes X_{L2}. \end{aligned} \quad (12)$$

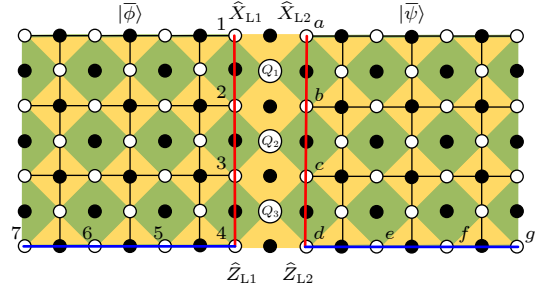


图 5 表面码的粗糙融合

Fig. 5. Rough fusion of surface codes.

对量子比特 Q_3, Q_2 和 Q_1 在 Z 基下进行测量, 可以完成粗糙分割操作. 在对 Q_3 进行测量之后, 与 Q_3 相关的 X 型稳定子 $S_3^X = X_3 X_c X_{Q_2} X_{Q_3}$ 因测量而消失; 与 Q_3 相关的两个 Z 型稳定子由四端变为三端, 当测量结果为 0 时, 三端稳定子自动满足; 当测量结果为 1 时, 需要添加 X 操作使这两个三端稳定子满足. 同理在对 Q_2 和 Q_1 进行测量之后, 稳定子 $S_2^X = X_2 X_b X_{Q_1} X_{Q_2}$ 和稳定子 $S_1^X = X_1 X_a X_{Q_1}$ 会因为测量而消失. 这时仅保留了变型过的 $S_4^{X'}$, 如图 6 所示.

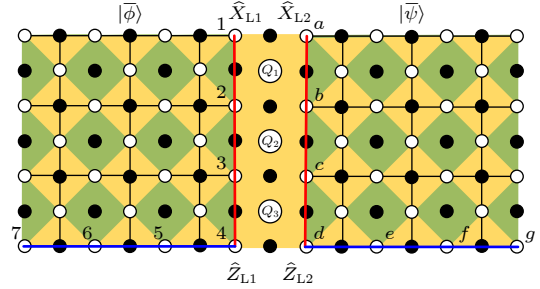


图 6 表面码的粗糙分割

Fig. 6. Rough segmentation of surface codes.

通过粗糙融合和分割, 实际上是对两个逻辑态进行联合逻辑测量 $M_{X_{L1} X_{L2}}$. 结合单量子比特间 CNOT 门的实现原理 (图 4) 可知, 为了实现逻辑量子比特间的 CNOT 门, 只需要联合测量和纠正操作就可以, 因此粗糙融合和分割过程可以简化为对图 6 所示的中间条形黄色稳定子的联合测量, 这样就能够简化 CNOT 的实现过程. 将稳定子测量的结果记为 M , 稳定子测量后的量子态表示为

$$\begin{aligned} &(\alpha|0\rangle + \beta|1\rangle)(\alpha'|0\rangle + \beta'|1\rangle) \\ &+ (-1)^M (\alpha|1\rangle + \beta|0\rangle)(\alpha'|1\rangle + \beta'|0\rangle) \\ &= (\alpha\alpha' + (-1)^M \beta\beta')(|00\rangle + (-1)^M |11\rangle) \\ &+ (\alpha\beta' + (-1)^M \beta\alpha')(|01\rangle + (-1)^M |10\rangle). \end{aligned} \quad (13)$$

3.2.2 光滑融合和光滑分割

光滑融合 (见图 7) 与粗糙融合原理相似, 它的具体过程为: 将 Q_1 , Q_2 和 Q_3 都初始化为 $|+\rangle$, 则边界处的三端 X 型稳定子变成了自动满足的四端 X 型稳定子; 然后对两个表面码之间的 Z 稳定子 $S_1^Z = Z_1 Z_a Z_{Q_1}$, $S_2^Z = Z_2 Z_b Z_{Q_1} Z_{Q_2}$, $S_3^Z = Z_3 Z_c Z_{Q_2} Z_{Q_3}$, $S_4^Z = Z_4 Z_d Z_{Q_3}$ 进行测量, 通过稳定子的测量过程就完成了融合操作. Z 型稳定子如果满足, 则测量结果记为 0, 否则记为 1, 将 4 个 Z 型稳定子的测量结果记为 M_1, M_2, M_3, M_4 , 因为 $S_1^Z \otimes S_2^Z \otimes S_3^Z \otimes S_4^Z = Z_{L1} \otimes Z_{L2}$, 所以 $M = M_1 \oplus M_2 \oplus M_3 \oplus M_4$ 表示联合逻辑 $Z_{L1} Z_{L2}$ 测量的结果.

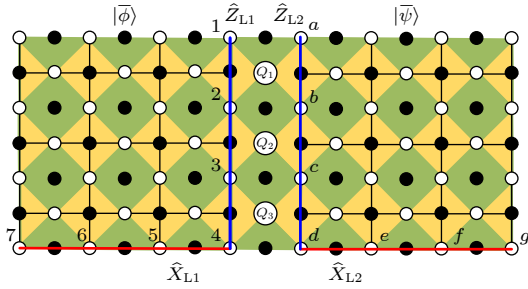


图 7 表面码的光滑融合

Fig. 7. Smooth fusion of surface codes.

同理, 对融合过程增加的第 4 个稳定子变形:

$$\begin{aligned} S_4^{Z'} &= S_4^Z \otimes S_1^Z \otimes S_2^Z \otimes S_3^Z \\ &= Z_1 Z_2 Z_3 Z_4 Z_a Z_b Z_c Z_d = Z_{L1} \otimes Z_{L2}. \end{aligned} \quad (14)$$

对量子比特 Q_3 , Q_2 和 Q_1 在 X 基下进行测量, 可以完成光滑分割操作. 在 X 基下对 Q_3 , Q_2 和 Q_1 进行测量之后, 稳定子 $S_3^Z = Z_3 Z_c Z_{Q_2} Z_{Q_3}$, $S_2^Z = Z_2 Z_b Z_{Q_1} Z_{Q_2}$ 和 $S_1^Z = Z_1 Z_a Z_{Q_1}$ 会因为测量而消失, 三端的 X 型稳定子可以通过纠正操作来满足, 这时左右两个表面码由变型过的 $S_4^{Z'}$ 稳定子连接, 如图 8 所示.

通过光滑融合和分割, 实际上是对两个逻辑态进行联合逻辑测量 $M_{Z_{L1} Z_{L2}}$, 直接对两个逻辑态进行联合测量能够满足 CNOT 门实现的需求, 因此光滑融合和分割过程可以简化为对图 8 所示的中间条形绿色稳定子的联合测量. 将稳定子测量的结果记为 M , 稳定子测量后的输出态为

$$\begin{aligned} &(\alpha|0\rangle + \beta|1\rangle)(\alpha'|0\rangle + \beta'|1\rangle) \\ &+ (-1)^M(\alpha|0\rangle - \beta|1\rangle)(\alpha'|0\rangle - \beta'|1\rangle) \\ &= (1 + (-1)^M)(\alpha\alpha'|00\rangle + \beta\beta'|11\rangle) \\ &+ (1 - (-1)^M)(\alpha\beta'|01\rangle + \beta\alpha'|10\rangle). \end{aligned} \quad (15)$$

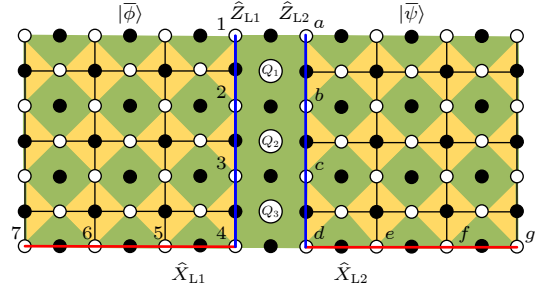


图 8 表面码的光滑分割

Fig. 8. Smooth segmentation of surface codes.

3.3 单逻辑量子比特表面码间 CNOT 门实现的优化

经过上述分析, 我们能够对基于晶格融合与分割的逻辑 CNOT 门实现方案进行优化: 在原方案实现过程中, 光滑融合对控制位和辅助位之间新产生的 Z 稳定子进行测量, 它们测量结果的异或等价于联合逻辑 $Z_{L1} Z_{L2}$ 测量的结果; 粗糙融合对目标位和辅助位之间新产生的 X 稳定子进行测量, 它们测量结果的异或等价于联合逻辑 $X_{L1} X_{L2}$ 测量的结果. 先融合后分割操作的结果是使得两个逻辑比特通过逻辑操作形成的稳定子连接成一个更大的表面码, 这个新增加的稳定子可能为正也可能为负. 因此, 基于晶格融合与分割的逻辑 CNOT 门实现方案可以简化为直接对逻辑操作形成的稳定子进行测量, 再对辅助比特进行逻辑 Z 测量, 最后添加纠正操作. 下面以码距为 2, 编码一位逻辑比特的表面码为例进行仿真实证.

图 9 中, $|CQ\rangle$ 代表控制位, 逻辑操作分别为 $X_2 X_5$ 和 $Z_1 Z_2$; $|TQ\rangle$ 代表受控位, 逻辑操作分别为 $X_{12} X_{15}$ 和 $Z_{11} Z_{12}$; $|INT\rangle$ 初始化为 $|+\rangle$ 作为辅助态, 逻辑操作分别为 $X_7 X_{10}$ 和 $Z_6 Z_7$. 第 1 步, 进行联合 Z 测量, 在 Z 基下对控制比特和辅助比特的 Z 逻辑操作涉及的比特 1, 2, 6, 7 进行联合测量,

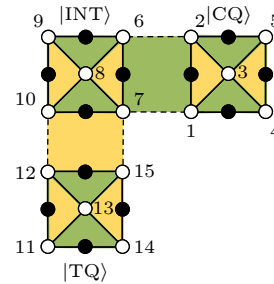


图 9 单逻辑量子比特表面码逻辑 CNOT 门的实现

Fig. 9. Implementation of logic CNOT gate for the single logical qubit surface code.

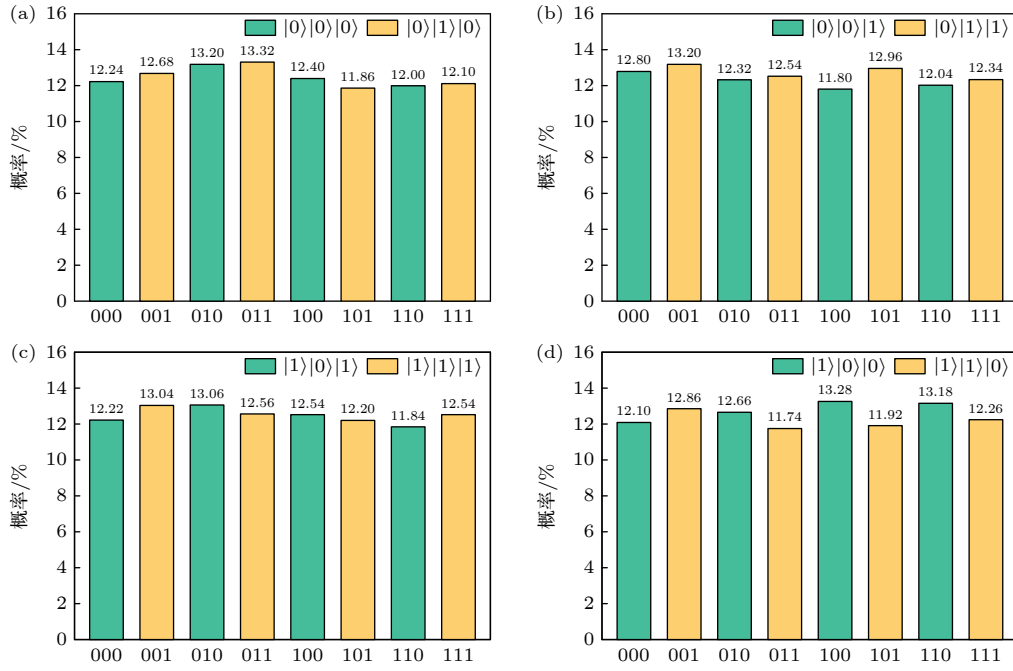


图 10 不同输入状态下的仿真输出 (a) $|CQ\rangle = |0\rangle$, $|TQ\rangle = |0\rangle$; (b) $|CQ\rangle = |0\rangle$, $|TQ\rangle = |1\rangle$; (c) $|CQ\rangle = |1\rangle$, $|TQ\rangle = |0\rangle$; (d) $|CQ\rangle = |1\rangle$, $|TQ\rangle = |1\rangle$

Fig. 10. Simulation output under different input states: (a) $|CQ\rangle = |0\rangle$, $|TQ\rangle = |0\rangle$; (b) $|CQ\rangle = |0\rangle$, $|TQ\rangle = |1\rangle$; (c) $|CQ\rangle = |1\rangle$, $|TQ\rangle = |0\rangle$; (d) $|CQ\rangle = |1\rangle$, $|TQ\rangle = |1\rangle$.

结果记为 M_1 ; 第 2 步, 进行联合 X 测量, 在 X 基下对辅助比特和目标比特的 X 逻辑操作涉及的比特 7, 10, 12, 15 进行联合测量, 结果记为 M_2 ; 第 3 步, 在 Z 基下对 $|INT\rangle$ 进行逻辑测量, 测量结果为 M_3 ; 第 4 步, 进行校正操作, 根据前 3 步的测量结果, 对 $|CQ\rangle$ 添加逻辑操作 Z^{M_2} , 当 M_2 为 1 时, 为 $Z_1 Z_2$; 在受控位 $|TQ\rangle$ 添加逻辑操作 $X^{M_1+M_3}$, 当 $M_1 + M_3 = 1$ 时, 为 $X_{12} X_{15}$; 当 $M_1 + M_3 = 1$ 且 $M_2 = 1$ 时, 整体相位进行翻转. 这里每个码字有 4 项, 最终的结果是 64 项, 是 3 个逻辑态码字直积的结果. 图 10 记录了对每一种输入进行 5000 次仿真后得到随机测量结果的概率统计以及在该种测量结果下对应的输出, 其中横坐标为 3 次测量的结果 $M_1 M_2 M_3$.

在图 10 中, 每一个测量结果的概率都约等于 12.5%, 说明两个联合逻辑测量和辅助态在 Z 基下的测量都是随机塌缩的. 对 $|CQ\rangle$ 来说, 输入和输出保持一致; 对 $|TQ\rangle$ 来说, 当 $|CQ\rangle$ 输入为 0 时, $|TQ\rangle$ 不变; 当 $|CQ\rangle$ 输入为 1 时, $|TQ\rangle$ 取反, 符合 CNOT 门的结果. 对 $|INT\rangle$ 来说, 因为 M_3 是对 $|INT\rangle$ 逻辑态的测量结果, 当 $M_3 = 0$ 时, $|INT\rangle$ 的测量结果为 $|0\rangle$, 当 $M_3 = 1$ 时, $|INT\rangle$ 的测量结果为 $|1\rangle$, 在输出结果中, 与理论分析相一致. 当 $M_1 M_2 M_3$ 测量结果

为 011 和 110 ($M_1 + M_3 = 1$ 且 $M_2 = 1$) 时, 结果中会出现一个整体的相位翻转, 因此当出现这种情况时, 可在校正操作中添加一个整体的相位翻转. 因此优化以后的过程能够实现逻辑 CNOT 操作.

4 基于联合测量和逻辑测量的多逻辑量子比特表面码间 CNOT 门的实现

本文提出的编码多个逻辑量子比特的表面码 (图 2) 依然可以基于联合测量和逻辑测量实现逻辑 CNOT 门. $|CQ\rangle$ 代表控制表面码, $|TQ\rangle$ 代表受控表面码, $|INT\rangle$ 的某一逻辑位初始化为 $|+\rangle$, 作为辅助态. 在编码一位逻辑比特的表面码中, 只有一位比特, 不需要特别关注逻辑操作的对应关系, 但在编码三位逻辑比特的表面码中, 要把信息比特和逻辑操作进行对应. 在图 2 中, 定义了表面码具体的逻辑操作, 要实现控制表面码第 i 位逻辑比特与目标表面码第 j 位逻辑比特之间的 CNOT, 就要将控制表面码第 i 位逻辑 Z 操作与辅助表面码的逻辑 Z 操作对应, 用 Z 型稳定子连接, 将目标表面码第 j 位逻辑 X 操作与辅助比特的逻辑 X 操作边界对应, 用 X 型稳定子连接. 图 11 给出了 $|CQ\rangle$ 第

一位逻辑比特利用 $|INT\rangle$ 第一位逻辑比特与 $|TQ\rangle$ 第一位逻辑比特 CNOT 的过程, Z_1Z_2 和 $Z_{13}Z_{14}$ 分别对应控制表面码和辅助表面码第一位逻辑比特的 Z 逻辑操作, $X_{13}X_{15}$ 和 $X_{25}X_{27}$ 分别对应的是辅助表面码和目标表面码第一位逻辑比特的 X 逻辑操作.

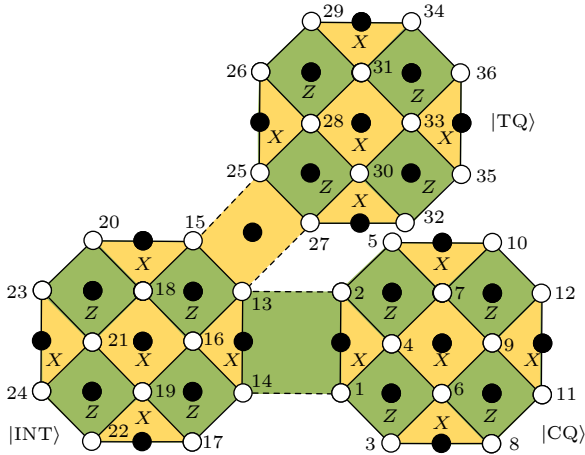


图 11 基于联合测量和逻辑测量的多逻辑量子比特表面码逻辑 CNOT 门的实现

Fig. 11. Implementation of logic CNOT gate for multiple logical qubits surface code based on joint measurement and logical measurement.

逻辑 CNOT 实现的步骤: 第 1 步, 进行联合 Z 测量, 在 Z 基下对 1, 2, 13, 14 比特进行联合测量, 结果记为 M_1 ; 第 2 步, 进行联合 X 测量, 在 X 基下对 13, 15, 25, 27 比特进行联合测量, 结果记为 M_2 ; 第 3 步, 在 Z 基下对 $|INT\rangle$ 的第一比特进行逻辑 Z 测量, 测量结果为 M_3 ; 第 4 步, 进行校正操作, 根据前 3 步的测量结果, 对 $|CQ\rangle$ 添加逻辑操作 Z^{M_2} , 当 M_2 为 1 时, 纠正操作为 Z_1Z_2 ; 在受控位 $|TQ\rangle$ 添加逻辑操作 $X^{M_1+M_3}$, 当 $M_1 + M_3 = 1$ 时, 纠正操作为 $X_{25}X_{27}$; 当 $M_1 + M_3 = 1$ 且 $M_2 = 1$ 时, 整体相位进行翻转.

在实现逻辑 CNOT 的过程中, 辅助比特 $|INT\rangle$ 初始化为逻辑 $|+\rangle$, 在该过程中 $|INT\rangle$ 不需要与 $|CQ\rangle$ 和 $|TQ\rangle$ 的拓扑结构相同, 在保证联合测量过

程中构成逻辑操作的数据比特位数能够满足对应关系的前提下, 可以将 $|INT\rangle$ 进行简化.

如图 12 所示, $|INT\rangle$ 由码距为 2 的边界表面码构成, 这时候整个过程用到的量子比特数目就会减少很多, 提高了 CNOT 门实现的效率, 并且利用普通计算机能够对此过程进行仿真. 由于 IBM 的 Qiskit 支持的量子比特数目有限, 因此在这里使用 Matlab 编写了表面码的直积、基转换、联合测量、逻辑测量等函数, 对上述 CNOT 的过程进行了仿真, 结果见表 3—表 6.

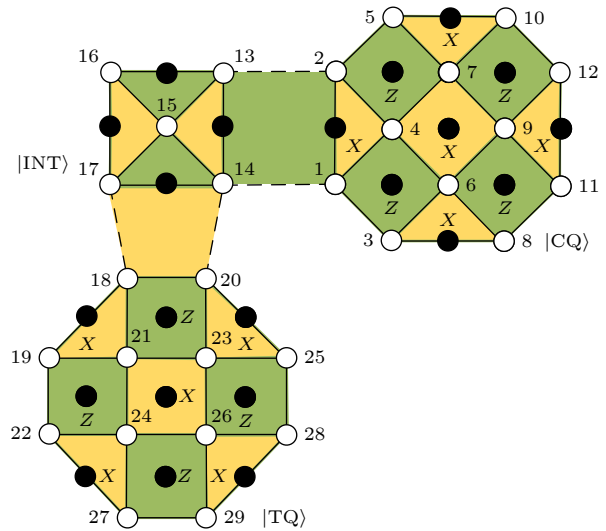


图 12 基于联合测量和逻辑测量的多逻辑量子比特表面码逻辑 CNOT 门的优化

Fig. 12. Optimization of logic CNOT gate for multiple logical qubits surface code based on joint measurement and logical measurement.

对 $|CQ\rangle$ 来说, 输入和输出保持一致, 符合 CNOT 门的结果. 对 $|TQ\rangle$ 来说, 当 $|CQ\rangle$ 为 $|AB0\rangle$, $(A, B \in (0, 1))$ 时, $|TQ\rangle$ 不变; 当 $|CQ\rangle$ 为 $|AB1\rangle$, $(A, B \in (0, 1))$ 时, $|TQ\rangle$ 第一位取反, 说明两个表面码的第 1 位逻辑比特之间实现了 CNOT 操作, 且在实现过程中没有影响到控制比特和目标比特的其他逻辑比特, 符合 CNOT 门的要求. 对 $|INT\rangle$ 来说, 因为 M_3 是对 $|INT\rangle$ 进行逻辑 Z 测量的结果, 当 $M_3 = 0$ 时, $|INT\rangle$ 塌缩为 $|0\rangle$, 当 $M_3 = 1$ 时, $|INT\rangle$

表 3 $|CQ\rangle = |AB0\rangle$, $|TQ\rangle = |CD0\rangle$, $(A, B, C, D \in (0, 1))$ 时的输出

Table 3. Output when the input is $|CQ\rangle = |AB0\rangle$, $|TQ\rangle = |CD0\rangle$, $(A, B, C, D \in (0, 1))$.

测量结果 ($M_1 M_2 M_3$)	000	001	010	011
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB0\rangle 0\rangle CD0\rangle$	$ AB0\rangle 1\rangle CD0\rangle$	$ AB0\rangle 0\rangle CD0\rangle$	$ AB0\rangle 1\rangle CD0\rangle$
测量结果 ($M_1 M_2 M_3$)	100	101	110	111
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB0\rangle 0\rangle CD0\rangle$	$ AB0\rangle 1\rangle CD0\rangle$	$ AB0\rangle 0\rangle CD0\rangle$	$ AB0\rangle 1\rangle CD0\rangle$

表 4 $|CQ\rangle = |AB1\rangle$, $|TQ\rangle = |CD1\rangle$, $(A, B, C, D \in (0, 1))$ 时的输出

 Table 4. Output when the input is $|CQ\rangle = |AB1\rangle$, $|TQ\rangle = |CD1\rangle$, $(A, B, C, D \in (0, 1))$.

测量结果($M_1 M_2 M_3$)	000	001	010	011
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB1\rangle 0\rangle CD0\rangle$	$ AB1\rangle 1\rangle CD0\rangle$	$ AB1\rangle 0\rangle CD0\rangle$	$ AB1\rangle 1\rangle CD0\rangle$
测量结果($M_1 M_2 M_3$)	100	101	110	111
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB1\rangle 0\rangle CD0\rangle$	$ AB1\rangle 1\rangle CD0\rangle$	$ AB1\rangle 0\rangle CD0\rangle$	$ AB1\rangle 1\rangle CD0\rangle$

 表 5 $|CQ\rangle = |AB1\rangle$, $|TQ\rangle = |CD0\rangle$, $(A, B, C, D \in (0, 1))$ 时的输出

 Table 5. Output when the input is $|CQ\rangle = |AB1\rangle$, $|TQ\rangle = |CD0\rangle$, $(A, B, C, D \in (0, 1))$.

测量结果($M_1 M_2 M_3$)	000	001	010	011
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB1\rangle 0\rangle CD1\rangle$	$ AB1\rangle 1\rangle CD1\rangle$	$ AB1\rangle 0\rangle CD1\rangle$	$ AB1\rangle 1\rangle CD1\rangle$
测量结果($M_1 M_2 M_3$)	100	101	110	111
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB1\rangle 0\rangle CD1\rangle$	$ AB1\rangle 1\rangle CD1\rangle$	$ AB1\rangle 0\rangle CD1\rangle$	$ AB1\rangle 1\rangle CD1\rangle$

 表 6 $|CQ\rangle = |AB0\rangle$, $|TQ\rangle = |CD1\rangle$, $(A, B, C, D \in (0, 1))$ 时的输出

 Table 6. Output when the input is $|CQ\rangle = |AB0\rangle$, $|TQ\rangle = |CD1\rangle$, $(A, B, C, D \in (0, 1))$.

测量结果($M_1 M_2 M_3$)	000	001	010	011
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB0\rangle 0\rangle CD1\rangle$	$ AB0\rangle 1\rangle CD1\rangle$	$ AB0\rangle 0\rangle CD1\rangle$	$ AB0\rangle 1\rangle CD1\rangle$
测量结果($M_1 M_2 M_3$)	100	101	110	111
$ CQ\rangle \otimes INT\rangle \otimes TQ\rangle$	$ AB0\rangle 0\rangle CD1\rangle$	$ AB0\rangle 1\rangle CD1\rangle$	$ AB0\rangle 0\rangle CD1\rangle$	$ AB0\rangle 1\rangle CD1\rangle$

塌缩为 $|1\rangle$, 与理论分析相一致. 经过上述分析, 基于联合测量和逻辑测量实现了多比特表面码之间的逻辑 CNOT 门.

在图 2 表面码中, 逻辑比特 2 和逻辑比特 1 类似, 因此逻辑 CNOT 的实现也类似. 当 CNOT 的控制比特和目标比特都是逻辑比特 3 时, 由于对 $|INT\rangle$ 只是进行逻辑比特联合测量和后续的逻辑测量, 逻辑操作的数据比特位数不需要一一对应, 因

此 $|INT\rangle$ Z 逻辑操作的长度并不需要为 4, 仍然可以采用最小的表面码作为中间态, 如图 13 所示. 经过仿真验证, 能够实现 CNOT 逻辑操作. 基于简化的原理, 可以设计出控制表面码的任意逻辑比特和目标表面码的任意逻辑比特之间的 CNOT 门, 实现任意逻辑比特间的逻辑 CNOT 操作.

5 讨论和结论

本文的主要创新包括: 第一, 提出了一种基于边界可以编码 3 个逻辑比特的表面码, 该表面码用 12 个数据比特编码 3 个逻辑比特; 第二, 对基于晶格融合与分割的表面码逻辑 CNOT 门实现方法进行了改进, 提出了一种基于联合测量和逻辑测量的表面码逻辑 CNOT 门实现方法, 并在提出的多逻辑比特表面码上进行了仿真验证.

编码多个逻辑比特的表面码的优点. 第一, 虽然本文提出的表面码结构与 3 个逻辑长度相同编码单逻辑比特的表面码所用的物理比特一样多, 但是在本文提出的表面码结构中有一个更长的逻辑操作, 增加了表面码的容错能力. 第二, 对于编码单个逻辑比特的表面码, 它们结构上是分离的, 两个表面码之间的物理比特不能充分利用. 本文的编码多逻辑比特的表面码相比于编码单逻辑比特的表面码, 物理比特能够形成密铺, 结构上更紧凑,

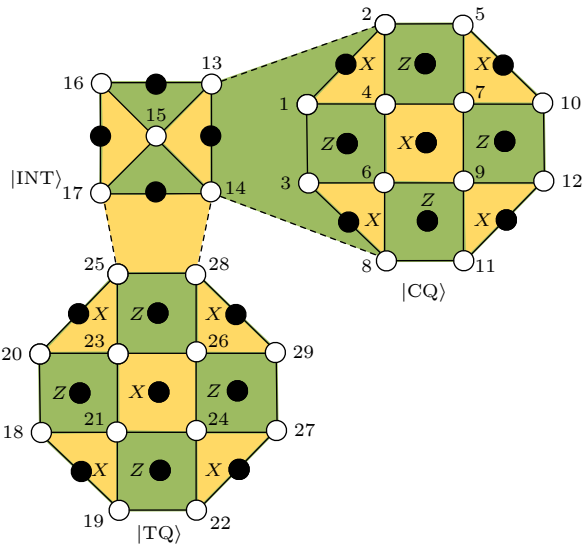


图 13 控制比特和目标比特都为逻辑比特 3 时, 逻辑 CNOT 门的优化

Fig. 13. Optimization of logic CNOT gate when both control qubit and target qubit are the 3rd logical qubits.

在量子计算的实现过程中能更充分地利用每一个物理量子比特. 第三, 虽然本文给出的是长度为 2 的表面码, 只有检错能力没有纠错能力, 但按照交替边界设计的思想能够设计出具有更长逻辑操作的表面码. 这里是因为仿真的限制, 因此研究长度为 2 的表面码.

基于联合测量和逻辑测量的表面码 CNOT 门实现方法过程更简洁, 能够节省资源. 这里以实现图 5 所示码距为 4 的表面码之间的 CNOT 操作为例进行资源需求分析, 基于融合和分割的方法辅助态表面码结构与图 5 中的结构一致, 而基于联合测量和逻辑测量的方法对辅助态没有具体要求, 这里选择有纠错能力的码距为 3 的表面码作为辅助态. 两种方法需要的资源如表 7 所列, 可以发现基于联合测量和逻辑测量的方法性能优于基于晶格融合与分割的方法. 逻辑 CNOT 门优化方案的优点: 第一, 中间辅助态的选择更灵活, 可以利用码距比较小的表面码来实现, 因此辅助表面码的码距和数据量子比特数目都比较小. 第二, 更节省量子门的数目. 在基于晶格融合与分割的方法中, 通过融合和分割操作, 将两个表面码形成纠缠态, 在基于联合测量和逻辑测量的方法中, 联合测量直接完成了这个过程, 联合测量的结果等价于融合与分割的结果. 因此不仅简化了操作, 还减少了量子门的数目. 第三, 在基于晶格融合与分割的方法中, 需要多次对新产生的稳定子进行测量, 且每次根据不同的测量结果需要进行纠正操作, 还需要进行单比特的测

量来消除稳定子; 而优化方法中, 不需要进行稳定子的测量和单比特测量, 因此测量次数和纠正次数都减少. 融合分割操作的优点是量子比特仅需要与相邻比特进行作用, 在大码距表面码的物理实现上更容易.

表 7 两种逻辑 CNOT 门实现方法的资源消耗对比
Table 7. Comparison of the resource consumption of the two logic CNOT gate implementation methods.

	基于联合测量和逻辑测量的方法	基于晶格融合与分割的方法
辅助表面码的码距	3	4
辅助表面码的数据量子比特数目	13	25
量子门数目 (不含纠正操作)	19	40
测量次数	3	15
最大纠正次数	2	15

本文设计的多逻辑比特表面码克服了单比特表面码不能密铺于量子芯片的缺点, 能够充分利用物理量子比特, 且提高了某些逻辑操作的长度, 增加了容错能力. 基于联合测量的思想降低了对辅助比特的要求, 简化后的实现过程减小了对量子资源的需求, 高效地实现了逻辑量子比特之间的 CNOT 门. 本文的研究成果对通用容错量子计算的研究和实现都具有重要的意义.

附录 A

图 3 所示的编码线路的 8 种输出码字见表 A1.

表 A1 图 3 所示的编码线路的 8 种输出码字
Table A1. Eight output codewords for the encoding circuit shown in Fig. 3.

000	001	010	011	100	101	110	111
000000000000>	010000110011>	000000000101>	010000110110>	000000010010>	010000100001>	000000010111>	010000100100>
000000001011>	010000111000>	000000001110>	010000111101>	000000011001>	010000101010>	000000011100>	010000101111>
000010100100>	010010010111>	000010100001>	010010010010>	000010110110>	010010000101>	000010110011>	010010000000>
000010101111>	010010011100>	000010101010>	010010011001>	000010111101>	010010001110>	000010111000>	010010001011>
000101100011>	010101010000>	000101100110>	010101010101>	000101110001>	010101000010>	000101110100>	010101000111>
000101101000>	010101011011>	000101101101>	010101011110>	000101111010>	010101001001>	000101111111>	010101001100>
000111000111>	010111110100>	000111000010>	010111110001>	000111010101>	010111100110>	000111010000>	010111100011>
000111001100>	010111111111>	000111001001>	010111111010>	000111011110>	010111101101>	000111011011>	010111101000>
001001010000>	011001100011>	001001010101>	011001100110>	001001000010>	011001110001>	001001000111>	011001110100>
001001011011>	011001101000>	001001011110>	011001101101>	001001001001>	011001111010>	001001001100>	011001111111>
001011110100>	011011000111>	001011110001>	011011000010>	001011100110>	011011010101>	001011100011>	011011010000>

表 A1 (续) 图 3 所示的编码线路的 8 种输出码字
Table A1 (continued). Eight output codewords for the encoding circuit shown in Fig. 3

000	001	010	011	100	101	110	111
001011111111>	011011001100>	001011111010>	011011001001>	001011101101>	011011011110>	001011101000>	011011011011>
001100110011>	011100000000>	001100110110>	011100000101>	001100100001>	011100010010>	001100100100>	011100010111>
001100111000>	011100001011>	001100111101>	011100001110>	001100101010>	011100011001>	001100101111>	011100011100>
001110010111>	011110100100>	001110010010>	011110100001>	001110000101>	011110110110>	001110000000>	011110110011>
001110011100>	011110101111>	001110011001>	011110101010>	001110001110>	011110111101>	001110001011>	011110111000>
110001100011>	100001010000>	110001100110>	100001010101>	110001110001>	100001000010>	110001110100>	100001000111>
110001101000>	100001011011>	110001101101>	100001011110>	110001111010>	100001001001>	110001111111>	100001001100>
110011000111>	100011110100>	110011000010>	100011110001>	110011010101>	100011100110>	110011010000>	100011100011>
110011001100>	100011111111>	110011001001>	100011111010>	110011011110>	100011101101>	110011011011>	100011101000>
110100000000>	100100110011>	110100000101>	100100110110>	110100010010>	100100100001>	110100010111>	100100100100>
110100001011>	100100111000>	110100001110>	100100111101>	110100011001>	100100101010>	110100011100>	100100101111>
110110100100>	100110010111>	110110100001>	100110010010>	110110110110>	100110000101>	110110110011>	100110000000>
110110101111>	100110011100>	110110101010>	100110011001>	110110111101>	100110001110>	110110111000>	100110001011>
111000110011>	101000000000>	111000110110>	101000000101>	111000100001>	101000010010>	111000100100>	101000010111>
111000111000>	101000001011>	111000111101>	101000001110>	111000101010>	101000011001>	111000101111>	101000011100>
111010010111>	101010100100>	111010010010>	101010100001>	111010000101>	101010110110>	111010000000>	101010110011>
111010011100>	101010101111>	111010011001>	101010101010>	111010001110>	101010111101>	111010001011>	101010111000>
111101010000>	101101100011>	111101010101>	101101100110>	111101000010>	101101110001>	111101000111>	101101110100>
111101011011>	101101101000>	111101011110>	101101101101>	111101001001>	101101111010>	111101001100>	101101111111>
111111110100>	101111000111>	111111110001>	101111000010>	111111100110>	101111010101>	111111100011>	101111010000>
111111111111>	101111001100>	111111111010>	101111001001>	111111101101>	101111011110>	111111101000>	101111011011>

参考文献

- [1] Feynman R P 1982 *Int. J. Theor. Phys.* **21** 467
- [2] Shor P W 1999 *SIREV* **41** 303
- [3] Preskill J 2012 arXiv: 1203.5813v3 [quant-ph]
- [4] Zhang S H, Zhang X D, Li L Z 2021 *Acta Phys. Sin.* **70** 210301 (in Chinese) [张诗豪, 张向东, 李绿周 2021 物理学报 **70** 210301]
- [5] Zhou W H, Wang Y, Weng W K, Jin X M 2022 *Acta Phys. Sin.* **71** 240302 (in Chinese) [周文豪, 王耀, 翁文康, 金贤敏 2022 物理学报 **71** 240302]
- [6] Song K H 2005 *Acta Phys. Sin.* **54** 4730 (in Chinese) [宋克慧 2005 物理学报 **54** 4730]
- [7] Grover L 1996 *Proc. 28th ACM Symp. Theo. Comp.* 212
- [8] Fan H 2023 *Acta Phys. Sin.* **72** 070303 (in Chinese) [范桁 2023 物理学报 **72** 070303]
- [9] Shor P W 1995 *Phys. Rev. A* **52** 2493
- [10] Steane A M 1996 *Phys. Rev. Lett.* **77** 793
- [11] Frank A, Kunal A, Ryan B, et al. 2019 *Nature* **574** 505
- [12] Davide C 2023 *Nature* **618** 656
- [13] Deng Y H, Gu Y C, Liu H L, Gong S Q, Su H, Zhang Z J, Tang H Y, Jia M H, Xu J M, Chen M C, Qin J, Peng L C, Yan J R, Hu Y, Huang J, Li H, Li Y X, Chen Y J, Jiang X, Gan L, Yang G W, You L X, Li L, Zhong H S, Wang H, Liu N L, Renema J J, Lu C Y, Pan J W 2023 *Phys. Rev. Lett.* **131** 150601
- [14] Huang J S, Chen X J, Li X D, Wang J W 2023 *AAPPS Bull.* **14** 33
- [15] Fowler A G, Mariantoni M, Martinis J M, Cleland A N 2012 *Phys. Rev. A* **86** 032324
- [16] Horsman C, Fowler A G, Devitt S, van Meter R 2012 *New J. Phys.* **14** 123011
- [17] Kitaev A Y 1997 *Quantum Communication, Computing, and Measurement* (New York: Plenum Press) pp181–188
- [18] Kitaev A Y 1997 *Russ. Math. Surv.* **52** 1191
- [19] Kitaev A Y 2003 *Ann. Phys.* **303** 2
- [20] Bravyi S B, Kitaev A Y 1998 arXiv: 9811052 v1 [quant-ph]
- [21] Freedman M H, Meyer D A 2001 *Found. Comput. Math.* **1** 325
- [22] Wang C Y, Harrington J, Preskill J 2003 *Ann. Phys.* **303** 31
- [23] Raussendorf R, Harrington J, Goyal K 2006 *Ann. Phys.* **321** 2242
- [24] Xing L J, Li Z, Bai B M, Wang X M 2008 *Acta Phys. Sin.* **57** 4695 (in Chinese) [邢莉娟, 李卓, 白宝明, 王新梅 2008 物理学报 **57** 4695]
- [25] Fowler A G, Stephens A M, Groszkowski P 2009 *Phys. Rev. A* **80** 052312
- [26] DiVincenzo D P 2009 *Phys. Scr.* **137** 014020
- [27] Tomita Y, Svore K M 2014 *Phys. Rev. A* **90** 062320
- [28] Brown B J, Laubscher K, Kesselring M S, Wootton J R 2017 *Phys. Rev. X* **7** 021029
- [29] Litinski D, von Oppen F 2018 *Quantum* **2** 62

- [30] Krylov G, Lukac M 2018 arXiv: 1809.11134v1 [quant-ph]
 [31] Beaudrap de N, Horsman D 2020 *Quantum* **4** 218
 [32] Camps D, van Beeumen R 2020 *Phys. Rev. A* **102** 052411
 [33] Shirakawa T, Ueda H, Yunoki S 2021 arXiv: 2112.14524v1 [quant-ph]
 [34] Wang H W, Xue Y J, Ma Y L, Hua N, Ma H Y 2022 *Chin. Phys. B* **31** 010303
 [35] Marques J F, Varbanov B M, Moreira M S, Ali H, Muthusubramanian N, Zachariadis C, Battistel F, Beekman M, Haider N, Vlothuizen W, Bruno A, Terhal B M, DiCarlo L 2022 *Nat. Phys.* **18** 80
 [36] Kumari K, Rajpoot G, Ranjan Jain S 2022 arXiv: 2211.12695v4 [quant-ph]
 [37] Chen P H, Yan B W, Cui S X 2022 arXiv: 2210.01682v2 [cond-mat.str-el]
 [38] Chen X B, Zhao L Y, Xu G, Pan X B, Chen S Y, Cheng Z W, Yang Y X 2022 *Chin. Phys. B* **31** 040305
 [39] Xue Y J, Wang H W, Tian Y B, Wang Y N, Wang Y X, Wang S M 2022 *Quantum Eng.* **2022** 9
 [40] Ding L, Wang H W, Wang Y N, Wang S M 2022 *Quantum Eng.* **2022** 8
 [41] Siegel A, Strikis A, Flatters T, Benjamin S 2023 *Quantum* **7** 1065
 [42] Quan D X, Liu C S, Lü X J, Pei C X 2022 *Entropy* **24** 1107

Structure design and logical CNOT implementation of multi-logical-qubits surface code^{*}

Quan Dong-Xiao^{1)2)†} Lü Xiao-Jie¹⁾ Zhang Wen-Fei¹⁾

¹⁾ (School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)

²⁾ (Collaborative Innovation Center of Quantum Information, Xidian University, Xi'an 710071, China)

(Received 14 July 2023; revised manuscript received 21 November 2023)

Abstract

As its parallel processing ability, quantum computing has an exponential acceleration over classical computing. However, quantum systems are fragile and susceptible to noise. Quantum error correction code is an effective means to overcome quantum noise. Quantum surface codes are topologically stable subcodes that have great potential for large-scale fault-tolerant quantum computing because of their structural nearest neighbor characteristics and high fault-tolerance thresholds. The existing boundary-based surface codes can encode one logical qubit. This paper mainly studies how to implement multi-logical-qubits encoding based on the boundary, including designing the structure of the surface code, finding out the corresponding stabilizers and logical operations according to the structure, and further designing the coding circuit based on the stabilizers. After research on the single qubit CNOT implementation principle based on measurement and correcting and the logic CNOT implementation based on fusion and segmentation, we further optimized implementation scheme of the logic CNOT implementation based on fusion and segmentation. The scheme is extended to the designed multi-logical-qubits surface code to realize the CNOT operation between the multi-logical-qubits surface codes, and the correctness of the quantum circuit is verified by simulation. The multi-logical-qubits surface code designed in this paper overcomes the disadvantage that the single-logical-qubit surface code can not be densely embedded in the quantum chip, improves the length of some logical operations, and increases the fault tolerance ability. The idea of joint measurement reduces the requirement for ancilla qubits and reduces the demand for quantum resources in the implementation process.

Keywords: quantum surface code, multi-logical-qubits encoding, logic CNOT gate, fusion operation, segmentation operation

PACS: 03.67.Lx, 03.67.Hk

DOI: 10.7498/aps.73.20231138

^{*} Project supported by the National Natural Science Foundation of China (Grant No. 62001351) and the Key Research and Development Program of Shaanxi Province, China (Grant No. 2019ZDLGY09-02).

[†] Corresponding author. E-mail: dxquan@xidian.edu.cn



多逻辑比特表面码结构设计及其逻辑CNOT门实现

权东晓 吕晓杰 张雯菲

Structure design and logical CNOT implementation of multi-logical-qubits surface code

Quan Dong-Xiao Lü Xiao-Jie Zhang Wen-Fei

引用信息 Citation: *Acta Physica Sinica*, 73, 040304 (2024) DOI: 10.7498/aps.73.20231138

在线阅读 View online: <https://doi.org/10.7498/aps.73.20231138>

当期内容 View table of contents: <http://wulixb.iphy.ac.cn>

您可能感兴趣的其他文章

Articles you may be interested in

离子阱中以声子为媒介的多体量子纠缠与逻辑门

Phonon-mediated many-body quantum entanglement and logic gates in ion traps

物理学报. 2022, 71(8): 080301 <https://doi.org/10.7498/aps.71.20220360>

飞秒激光直写光量子逻辑门

Femtosecond laser direct writing of optical quantum logic gates

物理学报. 2019, 68(10): 104205 <https://doi.org/10.7498/aps.68.20190024>

基于共振里德伯偶极-偶极相互作用的双反阻塞机制及量子逻辑门的实现

Implementation of the Rydberg double anti-blockade regime and the quantum logic gate based on resonant dipole-dipole interactions

物理学报. 2021, 70(13): 134202 <https://doi.org/10.7498/aps.70.20210059>

基于倾斜纳磁体翻转倾向性的与(或)逻辑门应力模型

Modeling of stress-regulated AND (OR) logic gate based on flipping preference of tilted nanomagnet

物理学报. 2019, 68(1): 017501 <https://doi.org/10.7498/aps.68.20181621>

基于人工表面等离激元的双通带频率选择结构设计

Design of dual-band-pass frequency selective structure based on spoof surface plasmon polariton

物理学报. 2018, 67(20): 204101 <https://doi.org/10.7498/aps.67.20180696>

电路量子电动力学中基于超绝热捷径的控制相位门实现

Implementation of controlled phase gate based on superadiabatic shortcut in circuit quantum electrodynamics

物理学报. 2022, 71(15): 150301 <https://doi.org/10.7498/aps.71.20220248>