

比特的移动

比特的移动

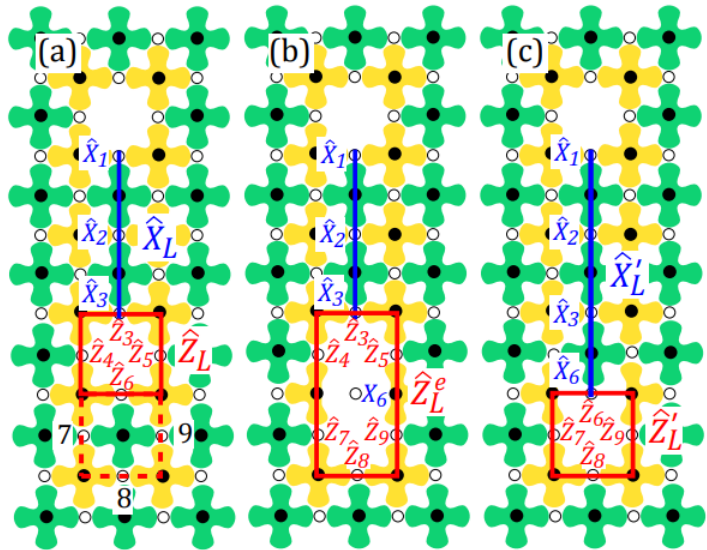
- 将挖孔移动一格
- 逻辑算符在移动中的变化
- 副产品
- 移动多格
- 移动过程中的错误

2022-11-29

实现复杂逻辑操作需要借助比特的移动。如将一个逻辑比特的挖孔在另一个逻辑比特的两个挖孔中间移动，即将两个比特编织 (braid) 了起来，以此能实现 CNOT 操作。

将挖孔移动一格

将双挖孔 Z-cut 比特的下面一个挖孔向下移动一格。逻辑算符 Z_L 定义在下面的孔上。



- a. 刚刚结束一轮镇定测量
- b. 在开始新一轮镇定测量前：
 1. 指示软件下一轮不要测量比特 6789 围绕的 Measure-Z 比特，即在此处挖孔；
 2. 下一轮中将比特 6 与两个相连的 Measure-X 比特断开，即将这两个测量比特上的镇定子改为三比特然后在此设定下进行一轮镇定测量，并测量比特 6 在 \hat{X} 方向的本征值 X_6
- c. 在开始新一轮镇定测量前：
 1. 将比特 3456 中间的挖孔恢复；
 2. 将比特 6 相连的两个测量比特上的镇定子从三比特恢复到四比特然后在此设定下继续进行镇定测量。

也即移动一格耗时两个表面码周期，算上时序方向上再等待的 $d - 1$ 个周期，移动一共耗时 $d + 1$ 个周期

逻辑算符在移动中的变化

- a. $\hat{X}_L = \hat{X}_1 \hat{X}_2 \hat{X}_3, \hat{Z}_L = \hat{Z}_3 \hat{Z}_4 \hat{Z}_5 \hat{Z}_6$
- b. $\hat{X}_L = \hat{X}_1 \hat{X}_2 \hat{X}_3, \hat{Z}_L^e = \hat{Z}_L \times \hat{Z}_6 \hat{Z}_7 \hat{Z}_8 \hat{Z}_9 = \hat{Z}_3 \hat{Z}_4 \hat{Z}_5 \hat{Z}_7 \hat{Z}_8 \hat{Z}_9$
- c. $\hat{X}_L' = \hat{X}_1 \hat{X}_2 \hat{X}_3 \hat{X}_6, \hat{Z}_L' = \hat{Z}_L^e \times \hat{Z}_3 \hat{Z}_4 \hat{Z}_5 \hat{Z}_6 = \hat{Z}_6 \hat{Z}_7 \hat{Z}_8 \hat{Z}_9$

副产品

逻辑算符的变化：

1. $\hat{X}_L \rightarrow \hat{X}_L'$, 多乘了一个 \hat{X}_6
2. $\hat{Z}_L \rightarrow \hat{Z}_L'$, 多乘了 $\hat{Z}_3 \hat{Z}_4 \hat{Z}_5 \hat{Z}_6 \times \hat{Z}_6 \hat{Z}_7 \hat{Z}_8 \hat{Z}_9$

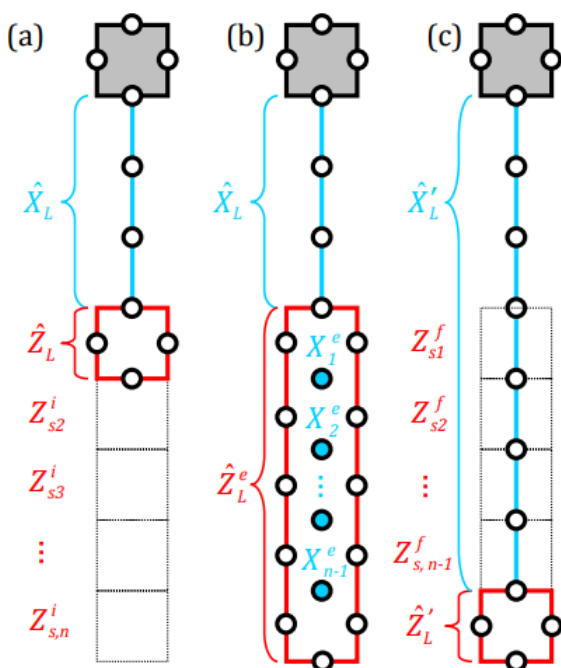
这种多乘的算符可能引起符号反转，例如当 $X_6 = -1$ 。相当于在逻辑态上引入了：

1. 多余的 \hat{Z} 错误 —— 如果 b 中测到 $X_6 = -1$
2. 多余的 \hat{X} 错误 —— 如果 a 和 c 中测到的 Z_{3456} 和 Z_{6789} 的乘积为 -1

即副产品 (byproduct)，需要在纠错软件中予以消除。

移动多格

移动多格和移动一格消耗一样的时间，也可以在两个周期中完成。



区别在于，在 b 中需要挖更长的孔来到达移动的目的地。

此外在跟踪算符变化中，需要测量更多的数据比特和 Measure-Z 比特来确定副产品算符。

移动过程中的错误

孤立出来的数据比特：

1. 不受 X 错误影响，因为上边会进行 X 方向的测量
2. 受 Z 错误影响，通过对比其 X 测量值和两个相邻的三比特 Measure-X 镇定测量值的乘积与之前的两个相邻的四比特 Measure-X 镇定测量值的乘积来确定。

孔边缘上的数据比特：

1. Z 错误可以被两个相邻的 Measure-X 镇定测量探测到
2. X 错误会改变相连的唯一的 Measure-Z 镇定测量，这个错误会和测量比特本身发生错误相混淆，不同的是该错误不会像测量错误那样倾向于自动消除，所以等待 d 个周期就能将两者分辨