PUC RIO

Dissertação de Mestrado

Efficient Simulation of Errors in Fusion Based Quantum Memory

Ivan Semenovich Ogloblin

Pontifícia Universidade Católica do Rio de Janeiro
Centro Técnico Científico
Departamento de Matemática

Rio de Janeiro, 21 de novembro de 2025

Pontifícia
Universidade
Católica do
Rio de Janeiro

Dissertação de Mestrado

Efficient Simulation of Errors in Fusion Based Quantum Memory

Ivan Semenovich Ogloblin

Orientação: Prof. Boyan Slavchev Sirakov
Coorientação: Prof. Sergey Tikhomirov

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Matemática pelo programa de Pós-Graduação em Matemática, no Departamento de Matemática.

Rio de Janeiro, 21 de novembro de 2025

Pontifícia
Universidade
Católica do
Rio de Janeiro

Efficient Simulation of Errors in Fusion Based Quantum Memory

Ivan Semenovich Ogloblin

Dissertação de apresentada como requisito parcial para a obtenção do grau de Mestre em Matemática. Aprovada pela Comissão examinadora abaixo:

Prof. Boyan Slavchev Sirakov
Orientador
Departamento de Matemática - PUC-Rio

Prof. Sergey Tikhomirov
Co-Orientador
Departamento de Matemática - PUC-Rio

Prof. Roberto Imbuzeiro Moraes Felinto de Oliveira
Instituto Nacional de Matemática Pura e Aplicada - IMPA

Prof. Sergey Sysoev
Saint Petersburg State University

Prof. Thiago Barbosa dos Santos Guerreiro
Departamento de Física - PUC-Rio

Rio de Janeiro, 21 de novembro de 2025

Pontifícia
Universidade
Católica do
Rio de Janeiro

Ivan Semenovich Ogloblin

Formado em Ciência da Computação e Engenharia de Software pela Saint Petersburg State University em 2023. Tem experiência nas áreas de Computação Quântica, Informação Quântica e Modelagem de Ruído Quântico. Trabalha em projetos de pesquisa focados em arquiteturas baseadas em fusão para computação quântica desde 2023, com foco em suas simulações com algoritmos clássicos.

Bibliographic data

Ogloblin, Ivan Semenovich

Efficient simulation of errors in fusion based quantum memory / Ivan Semenovich Ogloblin ; orientação: Boyan Sirakov ; coorientação: Sergey Tikhomirov. – 2025.
71 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Matemática, 2025.
Inclui bibliografia

1. Matemática – Teses. 2. Computação quântica baseada em fusão. 3. Formalismo estabilizador. 4. Qubit lógico. 5. Memória quântica. 6. Quadros de Pauli. I. Sirakov, Boyan Slavchev. II. Tikhomirov, Sergey. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Matemática. IV. Título.

CDD: 510

Acknowledgments

I would like to thank my mentor Sergey Borisovich Tikhomirov, Thiago Barbosa dos Santos Guerreiro and the rest of PUC stuff for believing in me. Then I would like to thank my mother and father for pushing me and supporting me. I would also like to thank Sam Roberts for answering my stupid questions. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

Abstract

Ogloblin, Ivan Semenovich; Sirakov, Boyan Slavchev (Advisor); Tikhomirov, Sergey Borisovich (Co-Advisor). **Efficient Simulation of Errors in Fusion Based Quantum Memory**. Rio de Janeiro, 2025. 72p. Master's Dissertation – Department of Mathematics, Pontificial Catholic University of Rio de Janeiro.

This thesis studies fault-tolerant quantum memory in a photonic Fusion-Based Quantum Computing (FBQC) architecture. Using the stabilizer formalism, we specify small entangled resource states and show how a layered six-ring fusion network implements the identity gate as a memory primitive. We embed a rotated surface code to define logical qubits and operators, linking code geometry to the fusion topology through correlation surfaces and surgery-style constructions that preserve the encoded state across layers.

We model imperfections as noisy channels applied after each fusion/measurement, focusing on outcome-flip errors, and perform purely classical decoding via Pauli-frame updates. To assess performance, we simulate Clifford-only circuits with the Pauli-frame algorithm, efficiently sampling logical error rates across network sizes and memory depths. The resulting finite-size scaling exhibits a crossing region, indicating a threshold for the studied model between 0.007 and 0.01 fusion-error probability.

The results provide theoretical and numerical evidence that a photonic FBQC memory can achieve fault tolerance under realistic noise assumptions. We outline next steps toward experimental relevance and scalability: incorporating more hardware-faithful noise processes, exploring code variants and stronger decoders, and extending beyond the identity operation to assemble a practical logical-gate set.

Keywords

fusion-based quantum computing; stabilizer formalism; logical qubit; quantum memory; pauli frames.

Resumo

Ogloblin, Ivan Semenovich; Sirakov, Boyan Slavchev; Tikhomirov, Sergey Borisovich. **Simulação Eficiente de Erros em Memória Quântica Baseada em Fusão**. Rio de Janeiro, 2025. 72p. Dissertação de Mestrado – Departamento de Mathematics, Pontifícia Universidade Católica do Rio de Janeiro.

Esta tese investiga memória quântica tolerante a falhas em uma arquitetura fotônica baseada em fusões (Fusion-Based Quantum Computing, FBQC). Partimos do formalismo de estabilizadores para descrever estados-recurso e mostramos como uma rede de fusões de seis anéis pode implementar, de forma fault-tolerant, a porta identidade — o elemento central de uma memória quântica. Em seguida, incorporamos um código de superfície rotacionado para definir qubits lógicos e operadores lógicos, conectando a geometria do código à topologia das fusões por meio de superfícies de correlação e técnicas análogas à lattice surgery.

Modelamos erros no processo de fusão e medição, adotando um canal ruidoso após cada projeção (com ênfase em "flips" de resultados) e realizamos a decodificação inteiramente no lado clássico via atualizações de "Pauli frame". Para avaliar o desempenho, simulamos circuitos somente-Clifford com o algoritmo de Pauli frames, permitindo amostragem eficiente de taxas de erro lógicas em diferentes tamanhos de rede e profundidades de memória. As curvas de erro lógico resultantes apresentam uma região de cruzamento que indica um limiar para o modelo estudado entre 0,7

Os resultados fornecem evidência teórica e numérica de que uma memória quântica fotônica baseada em FBQC pode atingir tolerância a falhas sob hipóteses realistas de ruído. Discutimos, ainda, direções para trabalhos futuros: incorporação de modelos de ruído mais fiéis ao hardware, exploração de variações do código (e decodificadores mais potentes) e extensões além da identidade para compor um conjunto de portas lógicas em larga escala.

Palavras-chave

computação quântica baseada em fusão; formalismo estabilizador; qubit lógico; memória quântica; quadros de Pauli.

Table of contents

List of figures

1 Introduction

Quantum computing is a rapidly growing field. Its fame stems from its promise of exponential acceleration in solving a certain class of problems. For example, Shor showed an algorithm for factoring large integers, which has implications for cryptographic protocols [Shor 1997]. Such an algorithm can be implemented with a device that operates on quantum states—a quantum computer. Many other algorithms and implementations have been found and discussed [Montanaro 2016], but in this document we focus on the physical implementation of a quantum computer for a specific task: quantum memory.

It is believed that in order to achieve quantum advantage we must have devices that operate with sufficiently low error rates. Unfortunately, this is very hard, so that even with conventional error suppression methods we won't see large algorithms executed in the near term [Preskill 2018]. However, the threshold theorem [Aharonov e Ben-Or 2008, Knill, Laflamme e Zurek 1998] shows that we can mitigate errors by increasing the number of qubits. With this approach we can potentially create a model that would mitigate a much higher percentage of errors, just by increasing the number of qubits. To show that it is possible, we can think of a simplest task that can be implemented with a high error threshold. Such task is an Identity gate that preserves a quantum state through time. Achieving quantum memory would be a huge milestone in the whole quantum community as it will open the door to fault-tolerant implementation of more difficult gates and in the end, to full scale implementation of quantum algorithms.

One of the techniques used to mitigate errors by increasing the number of qubits is called error-correcting codes. The idea of the code is similar to one in the classical realm of computation. We have several physical qubits that are said to encode one logical qubit as a quantum state. These physical qubits form a code, and when we apply special operations on the code, the encoded qubit state changes according to those operations. The common task is to find an encoding such that we would be able to apply arbitrary operations on the encoded qubit state and these operations would be more resilient to noise in terms of logical error, which is an error on the encoded quantum state.

When constructing such an error-correcting code it is convenient to assume that our whole system can be divided into subsystems, and each subsystem encodes one logical qubit. This way we know exactly which set of physical qubits corresponds to which logical qubit and these sets do not intersect. We then introduce operations on the subsystem of physical qubits which encode one logical qubit (one-qubit logical gates), and operations acting on two subsystems (two-qubit logical gates).

To ensure that our code indeed has the ability to tolerate more noise we want to measure the amount of noise that is being tolerated by the operation. There are different models of noise that can set on physical qubits and different measures of this noise on the output of the circuit. For one physical qubit we want to imagine that there is a value of probability that after applying a specific operation we will get expected result. And usually, the less noise we have on the physical apparatus - the bigger the value will be for one physical qubit. Same can be said about the encoded qubit state. The bigger the probability of getting expected result on the encoded quantum state - the more noise our encoding can tolerate.

This thesis studies how these values correlate for one specific operation: the identity gate, which is supposed to leave a quantum state unchanged. If we have a system with several encoded qubits, we want to preserve the state of this system over time. Quantum systems are fragile and tend to accumulate noise with time: the longer you wait, the higher the chance that the quantum state changes. However, if we have identity-gate operations that are more tolerant to noise, repeatedly applying them can keep the system in (or return it to) the initial state with higher probability. That is the idea of quantum memory—to store quantum states fault-tolerantly. Moreover, we want an encoding such that by applying the same operations to each encoded qubit we preserve the full multi-qubit state, including entanglement, not only the individual qubits.

Mitigation of errors by increasing the number of qubits is attractive for many physical models of quantum computation, but the engineering comes at a cost. Here we discuss a model called Fusion-Based Quantum Computing (FBQC). It is based on linear optics, where qubits are encoded in photonic modes. This approach is promising due to the maturity of fiber-optics technology and the potential for mass-produced photonic circuits.

With this idea in mind we show how quantum memory can be achieved, from photons up to the identity gate. This document contains a detailed description of all necessary theorems and proofs in Chapters 2 and 4 to show the theoretical possibility of quantum memory with this approach. A physical introduction describes how different errors can affect our quantum computation (Chapter 3). Finally, a threshold is estimated using numerical approximations and simulations (Chapter 7).

2
Stabilizer formalism

In this chapter we review the stabilizer formalism used throughout the thesis.

2.1
Stabilizer basics

In our description we will mostly use notations from Nielsen and Chuang book [Nielsen e Chuang 2010]. The central insight of the stabilizer formalism is easily illustrated by an example. Consider the EPR state (or the bell state) of two qubits:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

It is easy to verify that this state satisfies the identities $X_1X_2 |\psi\rangle = |\psi\rangle$ and $Z_1Z_2 |\psi\rangle = |\psi\rangle$ (recall that X_i is an X operator on the ith qubit of the system); we say that the state $|\psi\rangle$ is stabilized by the operators X_1X_2 and Z_1Z_2.

Definition 2.1.1. *A unitary operator U stabilizes $|\psi\rangle$ if $U|\psi\rangle = |\psi\rangle$.*

The group of principal interest is the Pauli group G_n on n qubits:

$$G_1 \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$$

$$G_n \equiv G_1^{\otimes n}$$

Consider S a subgroup of G_n and define V_S to be the set of n qubit states which are fixed (stabilized) by every element of S:

$$V_S = \{|\psi\rangle : s|\psi\rangle = |\psi\rangle, \forall s \in S\}$$

Note that V_S is a closed subspace. S is said to be the stabilizer of the space V_S.

Proposition 2.1.2. *Any two elements in G_n either commute or anti-commute.*

Proof. Pauli operators satisfy $\sigma_a\sigma_b = \pm\sigma_b\sigma_a$ for single qubits; tensor products inherit this property, so any two n-qubit Paulis either commute or anticommute. □

Similarly, $-I \notin S \implies \pm iI \notin S$. And if $-I \notin S$ then we have that all generators in the group commute with each other. Because otherwise we would have had both g_ig_j and $g_jg_i = -g_ig_j$ in S and $-I = (g_ig_j) * (-g_ig_j) \in S$

Now, a set of elements $g_1, \ldots, g_l$ in a group G is said to generate the group G if every element of G can be written as a product of elements from the list $g_1, \ldots, g_l$, and we write $G = \langle g_1, \ldots, g_l \rangle$.

Proposition 2.1.3. *Let $S = \langle g_1, \ldots, g_n \rangle$ be generated by n elements from G_n, and such that $-I \notin S$. Then any g_i has eigenvalues ± 1. And eigenspaces for these values will be of same dimensions.*

Proof. Since $g_i \in G_n$, we can represent this operator as a tensor product of single-qubit Pauli matrices with an overall phase in $\{\pm 1, \pm i\}$. Because $-I \notin S$, the phase cannot be $\pm i$ for any generator. Thus g_i is a tensor product of Paulis with overall factor ± 1. Each single-qubit Pauli has eigenvalues ± 1, hence so does g_i. Moreover, if g_i is not the identity, at least one tensor factor is a nontrivial Pauli with two eigenvectors e_l, v_l of opposite eigenvalues. Pairing basis states that differ only on qubit l gives a bijection between $+1$ and -1 eigenvectors, so the eigenspaces have equal dimension. □

In practice, we want our generators $g_1, \ldots, g_l$ to be independent in the sense that removing any generator g_i makes the group generated smaller.

$$\langle g_1, \ldots, g_{i-1}, g_{i+1}, \ldots g_l \rangle \neq \langle g_1, \ldots, g_l \rangle$$

Proposition 2.1.4. *Let $S = \langle g_1, \ldots, g_{n-k} \rangle$ be generated by $n-k$ independent and commuting elements from G_n, and such that $-I \notin S$. Then V_S is a 2^k-dimensional vector space.*

Proof. First let's construct a *check matrix* G. Check matrix is a $n-k$ by $2n$ matrix. Each row corresponds to a stabilizer. Remember that $Y = iXZ$. If g_i contains I on the jth qubit - then $G[i][j] = 0$ and $G[i][j+n] = 0$. If g_i contains Z on the jth qubit - then $G[i][j] = 0$ and $G[i][j+n] = 1$. If g_i contains X on the jth qubit - then $G[i][j] = 1$ and $G[i][j+n] = 0$. If g_i contains Y on the jth qubit - then $G[i][j] = 1$ and $G[i][j+n] = 1$.

For example. If we consider stabilizers of the Steane code, which are:

$$g_1 = IIIXXXX, g_2 = IXXIIXX, g_3 = XIXIXIX$$

$$g_4 = IIIZZZZ, g_5 = IZZIIZZ, g_6 = ZIZIZIZ$$

Then the check matrix G would be:

$$G = \left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]$$

Now we can call each row corresponding to a generator g_i as $r(g_i)$. We also want to define a $2n$ by $2n$ matrix

$$\Lambda = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$$

It is easy to see that if two elements of a Pauli group g and g' commute then $r(g)\Lambda r(g')^T = 0$. Vice versa, if $r(g)\Lambda r(g')^T = 0$ then two elements of a Pauli group g and g' commute.

A useful fact is that if all g_i commute - then rows of the G check matrix are linearly independent. Indeed, we have $r(g)+r(g') = r(gg')$. And generators are linearly dependent iff rows are dependent. Dependence for the rows here means that exist a_i such that they are not all zero and that $\sum_i a_i r(g_i) = 0$. If such a_i exist also for the generators, that means that one of the generators could be constructed as multiplication of others and that their set is not independent.

$$\sum_i a_i r(g_i) = 0 \Leftrightarrow \prod_i g_i^{a_i} \equiv I$$

Equivalence here means equality up to a multiplicative factor. Since $-I \notin S$ we know that multiplicative factor must be 1. That means that

$$g_j = g_j^{-1} = \prod_{i \neq j} g_i^{a_i}$$

Now follows the claim:

Claim 2.1.1. *Fix i in range from 1 to l. Then there exists $g \in G_n$ such that g anticommutes with g_i and commutes with all g_j for $j \neq i$.*

The proof is straightforward. Since rows of the check matrix G are independent there exists a $2n$-vector x such that $G\Lambda x = e_i$, where e_i is the ith standard basis vector in $\mathbb{Z}_2^{n-k}$. Let g be the Pauli with $r(g) = x^T$. Then $r(g_j)\Lambda r(g) = 0$ for $j \neq i$ (so g commutes with g_j), while $r(g_i)\Lambda r(g) = 1$ (so g anticommutes with g_i).

Now let u be a vector of $n-k$ elements of Z_2. We define a projector for u as

$$P_S^u = \frac{\prod_{j=1}^{n-k}(I+(-1)^{u_j}g_j)}{2^{n-k}}$$

Since $(I+g_j)/2$ is a projector onto the $+1$ eigenspace of g_j we see that $P_S^{(0,0,\ldots,0)}$ is a projector onto V_S. By 2.1.1 there exists g_u such that $g_u P_S^{(0,0,\ldots,0)} g_u^\dagger = P_S^u$ and therefore dimension of the range of P_S^u is the same as that of V_S. The final touch is that for distinct u the P_S^u are orthogonal and in sum they give $\sum_u P_S^u = I$. That means that V_S is of dimension 2^k.

□

Now we want to describe the action of a unitary operator on the system in terms of stabilizer operators. Suppose we apply a unitary operation U to a vector space V_S stabilized by the group S. Let $|\psi\rangle$ be any element of V_S. Then for any element g of S,

$$U|\psi\rangle = Ug|\psi\rangle = UgU^\dagger U|\psi\rangle$$

and thus the state $U|\psi\rangle$ is stabilized by $UgU^\dagger$, from which we deduce that the vector space UV_S is stabilized by the group $USU^\dagger \equiv \{UgU^\dagger | g \in S\}$.

Now we want to describe how the generator group changes with measurements. Imagine we make a measurement of $g \in G_n$. g is a Hermitian operator, and can thus be regarded as an observable. For convenience we assume without loss of generality that g is a product of Pauli matrices with no multiplicative factor of -1 or $\pm i$ out the front. The system is assumed to be in a state $|\psi\rangle$ with stabilizer $\langle g_1, \ldots, g_n\rangle$. How does the stabilizer of the state transform under this measurement?

Proposition 2.1.5. *One of two cases apply:*

- *If g commutes with all the generators of the stabilizer: if $g \in \langle g_1, \ldots, g_n\rangle$ - the measurement outcome is $+1$ with probability 1 and the state is not disturbed, if $-g \in \langle g_1, \ldots, g_n\rangle$ - the measurement outcome is -1 with probability 1 and the state becomes $-|\psi\rangle$.*
- *If g anti-commutes with one of the generators of the stabilizer g_1: if the measurement result is $+1$ - new state can be described with the generator group $\langle g, g_2, \ldots, g_n\rangle$, if the measurement result is -1 - new state can be described with the generator group $\langle -g, g_2, \ldots, g_n\rangle$.*

Proof. First we want to notice that indeed one of two cases is always applicable. If g anti-commutes with one or more of the generators of the stabilizer, suppose

g anti-commutes with g_1 without loss of generality. Without loss of generality we may assume that g commutes with $g_2, \ldots, g_n$, since if it does not commute with one of these elements (say g_2) then it is easy to verify that g does commute with $g_1 g_2$, and we simply replace the generator g_2 by $g_1 g_2$ in our list of generators for the stabilizer.

If g commutes with all the generators of the stabilizer we can write $g_j g |\psi\rangle = g g_j |\psi\rangle = g |\psi\rangle$ for any j. That means that $g |\psi\rangle$ is in V_S, that means that $g |\psi\rangle$ is a multiple of $|\psi\rangle$. Since $g^2 = I$, we know that $g |\psi\rangle = \pm |\psi\rangle$, and either g or $-g$ must be in the stabilizer. If $g |\psi\rangle = + |\psi\rangle$ then the measurement of g yields $+1$ with probability one, and the measurement does not disturb the state of the system, and thus leaves the stabilizer invariant. If $g |\psi\rangle = - |\psi\rangle$ then the measurement of g yields -1 with probability one, and the stabilizer group does not change, though the state changes from $|\psi\rangle$ to $- |\psi\rangle$.

If on the other hand g_1 anti-commutes with g, Note that g has eigenvalues ± 1 and so the projectors for the measurement outcomes ± 1 are given by $\frac{(I \pm g)}{2}$. Thus the measurement probabilities are given by:

$$p(+1) = tr(\frac{(I+g)}{2} |\psi\rangle \langle\psi|)$$

$$p(-1) = tr(\frac{(I-g)}{2} |\psi\rangle \langle\psi|)$$

Using the facts that $g_1 |\psi\rangle = |\psi\rangle$ and $g g_1 = -g_1 g$ gives:

$$p(+1) = tr(\frac{(I+g)}{2} g_1 |\psi\rangle \langle\psi|) = tr(g_1 \frac{(I-g)}{2} |\psi\rangle \langle\psi|)$$

$$= tr(\frac{(I-g)}{2} |\psi\rangle \langle\psi| g_1) = tr(\frac{(I-g)}{2} |\psi\rangle \langle\psi| g_1^\dagger) = tr(\frac{(I-g)}{2} |\psi\rangle \langle\psi|) = p(-1)$$

Since $p(+1) + p(-1) = 1$ we deduce that $p(+1) = p(-1) = 1/2$. Suppose the result $+1$ occurs. In this instance the new state of the system is

$$|\psi_+\rangle \equiv \frac{(I+g) |\psi\rangle}{\sqrt{2}}$$

which has stabilizer $\langle g, g_2, \ldots, g_n \rangle$. Similarly, if the result -1 occurs the posterior state is stabilized by $\langle -g, g_2, \ldots, g_n \rangle$. □

2.2 Clifford group

Clifford group is a group of gates that can be defined in a several ways.

Definition 2.2.1. *Clifford gates are such quantum operations that map G_n onto itself by conjugation.*

Definition 2.2.2. *Clifford gates are all gates that are generated by these three: Hadamard, S, CNOT.*

Equivalence of these definitions is standard (see, e.g., [Nielsen e Chuang 2010]). Pauli matrices are Clifford gates. The T gate is not a Clifford, and neither is a controlled-Hadamard. Clifford gates map stabilizer states to stabilizer states (since stabilizers are mapped within G_n).

Another useful property: for any unitary Clifford C, $CC^\dagger = I$ and $C^{-1} = C^\dagger$. (Note that a general Clifford is unitary but not necessarily Hermitian.) Now let's prove an easy fact.

Theorem 2.2.3. *Suppose we have a quantum system in a stabilizer state $|\psi\rangle$. Then we apply any Clifford operation C onto this state. After that we measure the state with an operator $M = Z_q$ on some qubit q. The result of the system (density operator) will be the same as if we were doing the following: start with the state $|\psi\rangle$, measure with operator $M = CZ_qC^{-1}$, apply C.*

Proof. Let's look at the stabilizers of $|\psi\rangle$. Suppose they are $G = \langle g_1, g_2, \ldots \rangle$. After the application of C operation they become $CGC^\dagger$. Suppose, that we chose such g_i to generate the stabilizer group that $Cg_iC^\dagger$ are commuting with Z_q for $i > 1$. We can do that because we can imagine a C^{-1} operation being applied on the result. If $Cg_1C^\dagger$ commutes with Z_q and if Z_q is in $CGC^\dagger$ then the outcome is always 1 and the state does not change (stays $C|\psi\rangle C^\dagger$). Otherwise it's -1 and the state is $-C|\psi\rangle C^\dagger$. If $Cg_1C^\dagger$ anti-commutes with Z_q, then the outcome is random and new stabilizers are either $\langle Z_q, Cg_2C^\dagger, \ldots \rangle$ or $\langle -Z_q, Cg_2C^\dagger, \ldots \rangle$.

Now let's see what happens in the second case. Since we have chosen our g_i in a specific way, we know that $[Cg_iC^\dagger, Z_q] = 0 | \forall i > 1$. That means:

$$Cg_iC^\dagger Z_q = Z_qCg_iC^\dagger$$

$$CCg_iC^\dagger Z_qC = CZ_qCg_iC^\dagger C$$

but since $C = C^\dagger = C^{-1}$ we have:

$$g_iCZ_qC^\dagger = CZ_qC^\dagger g_i$$

which means that we also have $[g_i, CZ_qC^\dagger] = 0 | \forall i > 1$. Moreover, if $Cg_1C^\dagger$ commutes with Z_q, then g_1 commutes with $CZ_qC^\dagger$, and if $Cg_1C^\dagger$ anti-commutes with Z_q, then g_1 anti-commutes with $CZ_qC^\dagger$ (putting a minus on the right hand of the equations). So if g_1 commutes with $CZ_qC^\dagger$ then if $Z_q \in CGC^\dagger \sim CZ_qC^\dagger \in G$ then the outcome is always 1 and the state does not change. If $Z_q \notin CGC^\dagger \sim CZ_qC^\dagger \notin G$, then the outcome is always -1 and the state becomes $-|\psi\rangle$. After the application of C in this case we get the same state corresponding to the same outcomes with the same probabilities. If g_1 anti-commutes with $CZ_qC^\dagger$, then the outcome is random and new stabilizers are either $\langle CZ_qC^\dagger, g_2, \dots\rangle$ or $\langle -CZ_qC^\dagger, g_2, \dots\rangle$. Let's notice that after the application of the operation C, we get exactly the same results, which are $\langle Z_q, Cg_2C^\dagger, \dots\rangle$ or $\langle -Z_q, Cg_2C^\dagger, \dots\rangle$ corresponding to the same outcomes with the same probabilities. □

3
Fusion based quantum computation review

This chapter reviews the fusion-based quantum computation (FBQC) paradigm and the elements we use in the rest of the thesis [Bartolucci et al. 2023, Browne e Rudolph 2005, Knill, Laflamme e Milburn 2001].

3.1
Introduction to linear optics

In linear-optical quantum computing, qubits are encoded in photonic degrees of freedom (e.g., polarization or dual-rail/spatial modes). Linear elements such as phase shifters and beam splitters implement unitary mode transformations, while photodetectors implement measurements. We follow standard conventions from [Kok et al. 2007] for describing modes, creation operators, and interferometers.

3.2
Generating bell pair with optics

Entangled photon pairs can be generated probabilistically, for example via spontaneous parametric down-conversion, and routed through interferometers to prepare Bell states in the computational basis. Heralding on detector clicks conditions the state on successful pair creation and mode-matching; we will treat ideal Bell pairs as resource states at this level of abstraction.

3.3
Generating fusion networks

The general description of FBQC follows this model: first we create a number of states called resource states, which consist of a small number of entangled qubits. After that we describe a set of measurements and measure operations on these resource states and obtain a number of classical outputs. Usually we want to abstract ourselves from thinking about the ordered in which these operations should be executed. We can think that all resource states are created simultaneously as well as all measure operations are executed simultaneously, as the change in the order would not compound to the change in the output [Bartolucci et al. 2023].

In general measurement operations could be any positive operator-valued measure (POVM) but, for the purposes of achieving fault tolerance, we can consider only measurements where all outcomes are projections onto stabilizer states. This makes it possible to use existing stabilizer theory.

We now briefly outline the physical process associated with fusion in linear optics. In the idealized, error-free model we discuss here, a Bell (Type-II) fusion [Browne e Rudolph 2005] projects two input qubits onto the joint $\langle XX, ZZ\rangle$ eigenspaces using a small interferometer and photon detection.

Using Pauli operator notation we can describe a Bell fusion as measuring operators X_1X_2, and Z_1Z_2 on the two input qubits, where $X_i(Z_i)$ is the single qubit Pauli-X (Z) operator on the qubit i. In particular, we focus on stabilizer fusion networks where resource states are stabilizer states and fusion measurements are stabilizer projections (measurements where all outcomes are projections onto stabilizer states).
Now we describe stabilizer (graph) states using an unoriented graph G. The graph state $|G\rangle$ on n vertices is obtained by preparing each vertex in $|+\rangle$ and applying a controlled-Z gate along every edge. Equivalently:

Proposition 3.3.1. *The n stabilizer generators for a graph state with vertices labeled from 1 to n are $X_i \prod_{j\in\mathcal{N}(i)} Z_j$ for $i \in \{1,\ldots,n\}$, where $\mathcal{N}(i)$ is the set of neighbors of vertex i in G.*

Proof. We will prove this statement by induction on the number of qubits n. Base case for $n = 1$: We have a graph consisting of only one vertex. It is initially in the state $|+\rangle$. After we apply CZ for every edge (of which there are none) we get the exact same state $|+\rangle$. Let us notice that such a state has only one stabilizer and it is X, which aligns with the formula $X_i \prod_{j\in\mathcal{N}(i)} Z_j, j \in 1, 2, \ldots, n$. Let us assume that these stabilizers correctly describe the state for any graph state of vertices less then or equal to n. Let us show that for the $n + 1$ the statement would still hold. For that we assume that we have any graph L with n vertices. (Here we of course assume that our graph is connected, otherwise the statement is obvious). Let us enumerate these vertices in any order. Now let us look at the first vertex l_1 and the subgraph on the rest of the vertices $\overline{L} = L \setminus l_1$. Let us also denote all the edges that go from l_1 to the rest of the graph as $e_2, \ldots, e_k$, as they would go to vertices $2, \ldots, k$ without loss of generality. Now we know by the induction hypothesis that $\overline{L}$ is described by the stabilizers $X_i \prod_{j\in\mathcal{N}(i)} Z_j, j \in 2, \ldots, n + 1$, as it has n vertices. The first vertex is in the state $|+\rangle$. What is left is the application of CZ gates according to $e_2, \ldots, e_k$. This way we will get the state of the whole graph L. Let us see how these applications would change the stabilizers of the state. Initially we have stabilizers $\{X_1, X_i \prod_{j\in\mathcal{N}(i)} Z_j, i \in 2, \ldots, n + 1\}$. Let us

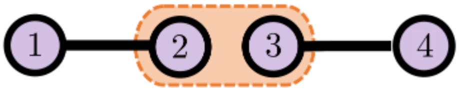


Figure 3.1: Graph representation of two graph states and fusion between them

apply first CZ corresponding to e_2. It acts on qubits l_1 and 1. Let us see how it acts on the stabilizer X_1.

$$CZ_{l_1,1}X_1CZ^{\dagger}_{l_1,1} = X_1Z_{l_1}$$

Let us see how it acts on the stabilizer $X_1\prod_{j\in\mathcal{N}(1)}Z_j$

$$CZ_{l_1,1}(X_1\prod_{j\in\mathcal{N}(1)}Z_j)CZ^{\dagger}_{l_1,1} = (X_1\prod_{j\in\mathcal{N}(1)}Z_jZ_{l_1}) = X_1\prod_{j\in\mathcal{N}'(1)}Z_j$$

where $\mathcal{N}'(1)$ is the set of vertices neighboring vertex 1 from L. Let us notice that the same way this operator would not change stabilizers of the form $X_m\prod_{j\in\mathcal{N}'(1)}Z_j$, where m is not a neighbor of l_1 in L. Now we have showed that stabilizers for this graph stay correct as we apply CZ with vertices. □

Now there is an example of fusion described in terms of stabilizer formalism. We first start with two simple graph states, which both have just two vertices connected by the edge as on the Figure 3.1. Such system is in the state that can be described by stabilizers:

$$S = \langle X_1Z_2, X_2Z_1, X_3Z_4, X_4Z_3\rangle$$

To successfully apply fusion on qubits 2 and 3 we need to measure the system with the operator $M_1 = X_2X_3$ and then with $M_2 = Z_2Z_3$. Then, since fusion is a destructive process, we will need to restrict our stabilizer group on the left out qubits, which are 1 and 4. This process is equivalent to just discarding measured qubits, as after the measurements the system will be in a product state of measured qubits and outer qubits. Proof of this fact will follow the examples. Now by the proposition 2.1.5 we know how the stabilizer group will change after the first measurement. We can see that first the measurement $M_1 = X_2X_3$ commutes with stabilizers X_2Z_1, X_3Z_4, but anti-commutes with X_1Z_2, X_4Z_3. Let's rewrite stabilizer group for our convenience:

$$S = \langle X_1Z_2, X_2Z_1, X_3Z_4, X_4Z_3\rangle = \langle X_1Z_2, X_2Z_1, X_3Z_4, X_1Z_2Z_3X_4\rangle$$

Now M_1 only anti-commutes with the first stabilizer, and we can write the result: if the first measurement outcome $m_1 = +1$, then the new stabilizer

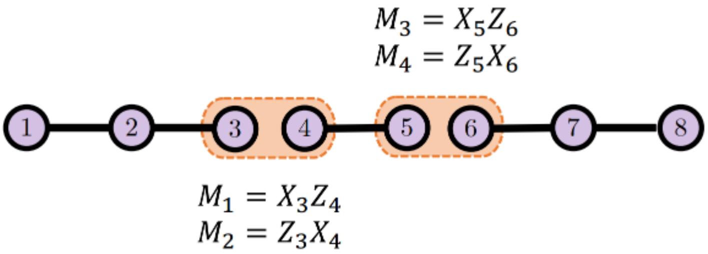


Figure 3.2: Graph representation of three graph states and fusions between them

group will be:

$$S_1 = \langle X_2X_3, X_2Z_1, X_3Z_4, X_1Z_2Z_3X_4\rangle$$

If $m_1 = -1$, then:

$$S_1 = \langle -X_2X_3, X_2Z_1, X_3Z_4, X_1Z_2Z_3X_4\rangle$$

Let us apply the second measurement $M_2 = Z_2Z_3$. M_2 commutes with $X_2X_3, X_1Z_2Z_3X_4$ and anti-commutes with X_2Z_1, X_3Z_4. Let us again rewrite S:

$$S_1 = \langle X_2Z_1, m_1X_2X_3, Z_1X_2X_3Z_4, X_1Z_2Z_3X_4\rangle$$

Now depending on $m_2 = \pm 1$ we have new stabilizer group:

$$S_2 = \langle m_2Z_2Z_3, m_1X_2X_3, Z_1X_2X_3Z_4, X_1Z_2Z_3X_4\rangle =$$

$$\langle m_2Z_2Z_3, m_1X_2X_3, m_1Z_1Z_4, m_2X_1X_4\rangle$$

We can easily see that the system is indeed in the product state and we can discard all stabilizers acting on qubits 2 and 3. By doing so we get the resulting stabilizer group:

$$S_{final} = \langle m_1Z_1Z_4, m_2X_1X_4\rangle$$

We can notice that such stabilizer group describes exactly Bell state (by the module of ± 1). It is also useful to notice that before and after fusion we don't lose the property that $-I \notin S$, because any operator from the group multiplied by itself would give I, whence any combination of operators in multiplication from the group wouldn't give I or $-I$, as they are all independent.

Next example will be with several fusions, but now fusions will be with different measurement operators i.e. $M = XZ$. For the initial state we will take three graph states and apply two fusions on them as it is depicted on the

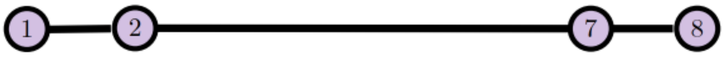


Figure 3.3: Graph representation of the result state system from example

Figure 3.2. As usual we will start with stabilizer group for the initial state:

$$S = \langle X_1Z_2, Z_1X_2Z_3, Z_2X_3, X_4Z_5, Z_4X_5, X_6Z_7, Z_6X_7Z_8, Z_7X_8\rangle$$

Now we will once again apply all the steps as in proposition 2.1.5 for our measurement to see the result. After that we will prove a useful theorem. First we apply first fusion, after the first measurement M_1 we get S_1, after the second we get S_2, and S_3 after discarding both qubits:

$$S_1 = \langle m_1X_3Z_4, X_1Z_2, Z_1X_2Z_3X_4Z_5, Z_2X_3, Z_4X_5, X_6Z_7, Z_6X_7Z_8, Z_7X_8\rangle$$

$$S_2 = \langle m_2Z_3X_4, m_1X_3Z_4, X_1Z_2, Z_1X_2Z_3X_4Z_5, Z_2X_3Z_4X_5, X_6Z_7, Z_6X_7Z_8, Z_7X_8\rangle$$

$$S_3 = \langle X_1Z_2, m_2Z_1X_2Z_5, m_1Z_2X_5, X_6Z_7, Z_6X_7Z_8, Z_7X_8\rangle$$

Next we apply second fusion. S_4 after first measurement, S_5 after second and S_6 is final result.

$$S_4 = \langle m_3X_5Z_6, X_1Z_2, m_2Z_1X_2Z_5X_6Z_7, m_1Z_2X_5, Z_6X_7Z_8, Z_7X_8\rangle$$

$$S_5 = \langle m_4Z_5X_6, m_3X_5Z_6, X_1Z_2, m_2Z_1X_2Z_5X_6Z_7, m_1Z_2X_5Z_6X_7Z_8, Z_7X_8\rangle$$

$$S_6 = \langle X_1Z_2, m_4m_2Z_1X_2Z_7, m_3m_1Z_2X_7Z_8, Z_7X_8\rangle$$

Now we can notice that if $m_4m_2 = +1$ and $m_1m_3 = +1$ then this state can be represented by a graph on the figure 3.3

There is a property of a fusion operation that it is destructive. That means that after we have done a fusion on two qubits, these two qubits will be unentangled from the system and thus can be regarded as *destroyed*. That means that no other operations will be applied on these qubits and they do not affect the rest of the system. Let's prove that it is indeed a correct statement:

Proposition 3.3.2. *Suppose that we have a system in any state $|\psi\rangle$ in an $n > 2$ qubit space. That can be any state and not just the stabilizer state. Then we apply a fusion on the first two qubits (without constraining ambiguity). Then the new state will be represented as $|\psi_{1,2}\rangle \otimes |\psi_{rest}\rangle$, where $|\psi_{1,2}\rangle$ is the state on the first two qubits and $|\psi_{rest}\rangle$ is the state of the rest of the system.*

Proof. We know that a fusion operation can be modeled as applying measurements Z_1Z_2 and X_1X_2 consecutively. These are two hermitian operators and thus they can be regarded as observables. Both of them have two eigenvalues $+1$ and -1. And we know they can represented with their spectral decomposition as:

$$Z_1Z_2 = (+1)P_{Z+} + (-1)P_{Z-}; X_1X_2 = (+1)P_{X+} + (-1)P_{X-}$$

, where

$$P_{Z+} = |00\rangle\langle 00| + |11\rangle\langle 11|; P_{Z-} = |01\rangle\langle 01| + |10\rangle\langle 10|$$

$$P_{X+} = |{+}{+}\rangle\langle {+}{+}| + |{-}{-}\rangle\langle {-}{-}|; P_{X-} = |{+}{-}\rangle\langle {+}{-}| + |{-}{+}\rangle\langle {-}{+}|$$

Now, depending on the outcomes of these measurements we will have one of these final states:

$$|\psi_{+1+1}\rangle = \frac{P_{X+}P_{Z+}|\psi\rangle}{\sqrt{p(+1,+1)}}, |\psi_{+1-1}\rangle = \frac{P_{X-}P_{Z+}|\psi\rangle}{\sqrt{p(+1,-1)}};$$

$$|\psi_{-1+1}\rangle = \frac{P_{X+}P_{Z-}|\psi\rangle}{\sqrt{p(-1,+1)}}; |\psi_{-1-1}\rangle = \frac{P_{X-}P_{Z-}|\psi\rangle}{\sqrt{p(-1,-1)}}$$

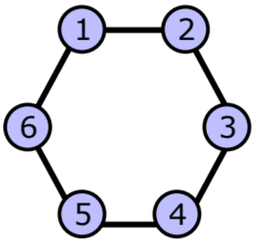
Where $p(i,j)$ is the probability to get outcomes i and j in this order. Here we must notice that projectors on the eigen-spaces are not of the same dimension as $|\psi\rangle$. When we multiply them we of course implicitly imply the construction: $P_{X+}P_{Z+} \otimes I_{rest}|\psi\rangle$, where I_{rest} acts as identity on the rest of the system but the first two qubits. Let's also write out matrices $P_{X+}P_{Z+}, P_{X-}P_{Z+}, P_{X+}P_{Z-}, P_{X-}P_{Z-}$:

$$P_{X+}P_{Z+} = (|00\rangle + |11\rangle) \otimes (\langle 00| + \langle 11|), P_{X-}P_{Z-} = (|01\rangle - |10\rangle) \otimes (\langle 01| - \langle 10|)$$

$$P_{X+}P_{Z-} = (|01\rangle + |10\rangle) \otimes (\langle 01| + \langle 10|), P_{X-}P_{Z+} = (|00\rangle - |11\rangle) \otimes (\langle 00| - \langle 11|)$$

Now it has become obvious that after applying any of these four matrices on the state we will have a separable state that can be represented as $|\psi_{1,2}\rangle \otimes |\psi_{rest}\rangle$. □

From this theorem we can understand what means for a fusion measurement to be destructive. Since the qubits included in the fusion will be unentangled from the rest of the system - we can imagine their absence from the system completely. Meaning that these qubits will no longer participate in the future operations and will no longer affect the rest of the system in any way. That is almost what happens in reality with photons (we do not yet account for any real-world noise). After the photon gets measured we can't use it anymore. Same goes for fusions with measurement operators X_1X_2, Z_1Z_2



$$\begin{aligned}
S_1 &= Z_1X_2Z_3I_4I_5I_6\\
S_2 &= I_1Z_2X_3Z_4I_5I_6\\
S_3 &= I_1I_2Z_3X_4Z_5I_6\\
S_4 &= I_1I_2I_3Z_4X_5Z_6\\
S_5 &= Z_1I_2I_3I_4Z_5X_6\\
S_6 &= X_1Z_2I_3I_4I_5Z_6
\end{aligned}$$

Figure 3.4: Graph representation of the 6-ring resource state with it's stabilizers

and X_1Z_2, Z_1X_2. In the later chapters, we will only use fusions as X_1X_2, Z_1Z_2 measurement operators.

3.4
6-ring fusion network

Description of stabilizers and fault tolerant structure of a 6-ring fusion network. A fusion network specifies an arrangement of resource states and a set of fusion measurements to be made on qubits of the resource states. After the measurements are made the qubits that were fused are removed from the state, and we learn measurement outcomes from each fusion. It is important to understand that fusion network does not specify order on the operations of creation of resource states or fusions. First creation of all resource states, then all fusions in any order. 6-ring resource state is a graph state depicted in the figure 3.4. In our fusion network, for convenience, we will represent each state with the big sphere. On this figure 3.5 the sphere is depicted. It has three axis and 6 directions. Each direction can be associated with each qubit from 6-ring state. In the same way we can associate colored directions with stabilizers of our resource state. If a direction is colored purple it is associated with X operator on the corresponding qubit. If it is colored green it is associated with Z operator on the corresponding qubit. For example on the figure 3.5 a $Z_2X_3Z_4$ stabilizer is represented. We can notice that if we multiply stabilizers

$$Z_2X_3Z_4 * Z_4X_5Z_6 * Z_6X_1Z_2 = X_3X_5X_1$$

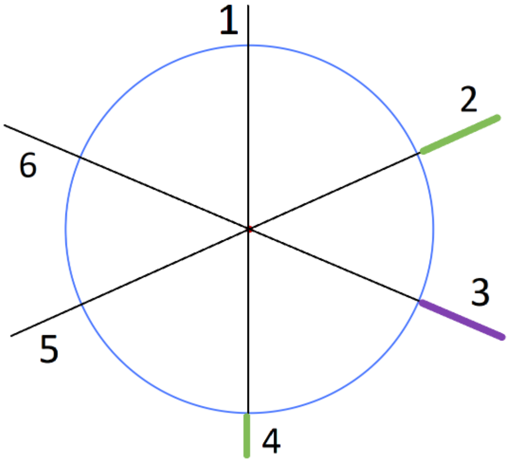


Figure 3.5: Different representations of the 6-ring resource state. Here each outcoming ray represents a qubit. If this ray is marked green (purple) it represents Z (X) operator on the corresponding qubit.

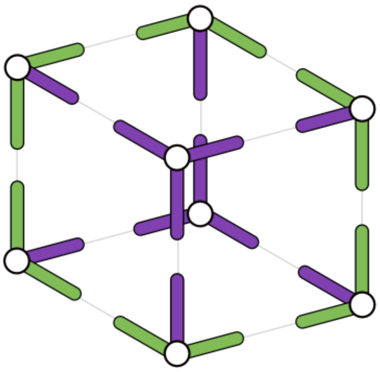


Figure 3.6: A simple 3D placement of 6-ring resource states. Each node (sphere) represents 1 resource state (6 qubits). And on each of those resource states only one stabilizer is represented.

Tha is also a viable stabilizer of a 6-ring state as well as

$$Z_3X_4Z_5 * Z_5X_6Z_1 * Z_1X_2Z_3 = X_4X_6X_2$$

But these stabilizers will only be colored purple in schemes. Now we can use this sphere in 3d construction such as cubes. For example here 3.6 are 6 resource states combined in a cube. For each one of them only one stabilizer is depicted. It is important to understand that in 3D diagrams, each node (sphere) represents 1 resource state (6 qubits).

4
Surface code and quantum memory

In this chapter we introduce the rotated surface code used for encoding, define logical operators, and connect these constructs to the fusion-based identity gate and memory.

4.1
Introduction to surface codes

Let us consider a lattice, that consists of qubits. These qubits can be thought of as physical qubits that will encode one *logical* qubit. On these qubits we define stabilizers. They will serve as the stabilizer group for our encoded state. On the Figure 4.1 we can see 25 qubits and 24 stabilizers. The lattice is equivalent to the rotated lattice introduced in [Horsman et al. 2012] only by the module of Hadamard application on each second qubit. For a general overview of surface codes and their practicality, see also [Fowler et al. 2012]. Let us notice that all of our stabilizers commute with each other and form an independent group of stabilizers. That means that by Proposition 2.1.4 we know that our stabilized state would be in a space of dimension 2, which is exactly the space of one encoded qubit.

Proposition 4.1.1. *The space of states generated by 25 physical qubits on the lattice stabilized by corresponding stabilizers will be of dimension 2.*

Proof. First we notice that each of the stabilizers commutes with any other. They either intersect in two anti-commuting Pauli matrices (XZ) or in one

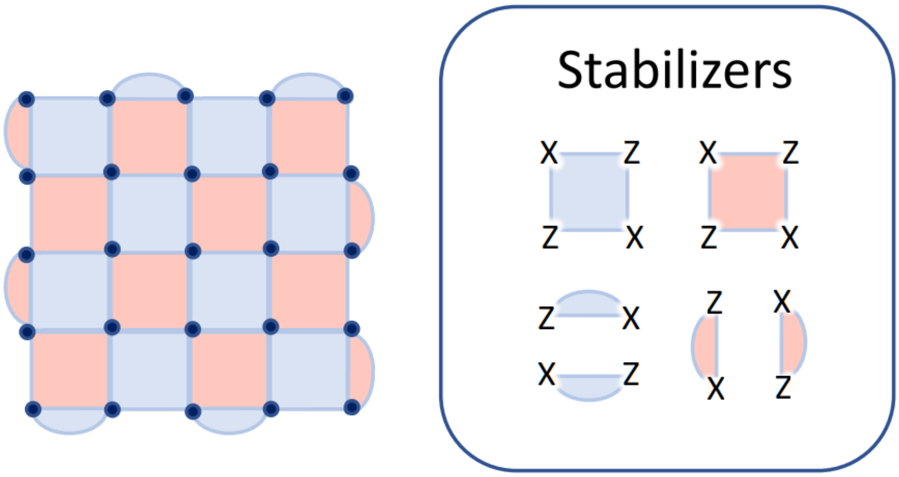


Figure 4.1: Rotated surface code on 25 qubits with stabilizers shown in red and blue.

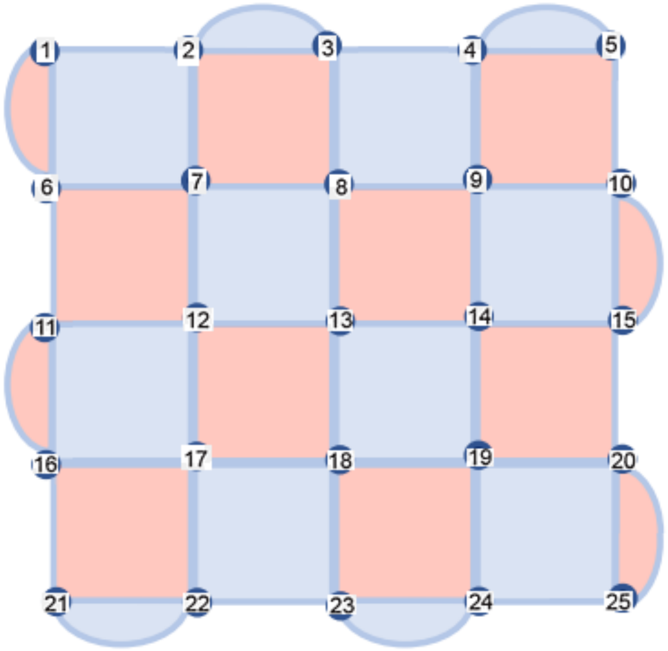


Figure 4.2: Numbered lattice

commuting (X or Z), or do not intersect at all. Second, the set of stabilizers S is independent and does not include $-I$. To see independence we argue that no subset multiplies to I. Consider the numbered lattice in Figure 4.2. By propagating parity constraints across the boundary checks (e.g., the factor of Z on qubit 20), we iteratively exclude each face-type stabilizer; the same reasoning excludes the vertex-type stabilizers. Hence no nontrivial product is I, and all stabilizers are independent. By Proposition 2.1.4 the stabilized space V_S has dimension $2^{25-24} = 2$. □

Now we want to describe how this logical qubit is encoded. Let us see the operators $\overline{Z} = X_3Z_8X_{13}Z_{18}X_{23}, \overline{X} = Z_{11}X_{12}Z_{13}X_{14}Z_{15}$. They both commute with all the stabilizers and anti-commute with each other. Moreover, they both are independent with all the other stabilizers.

Proposition 4.1.2. *$\overline{X}$ is independent of the set of 24 stabilizers. $\overline{Z}$ is independent of the set of 24 stabilizers.*

Proof. Each stabilizer covers an even number of sites; the logical strings cover an odd number. To multiply to the identity, each site must be covered an even number of times, so no product of stabilizers can equal a logical string. □

We will call $\overline{Z}$ and $\overline{X}$ the logical Z and X operators on the encoded qubit. They commute with all 24 stabilizers. We use $\overline{X}$ and $\overline{Z}$ to manipulate the degrees of freedom in the 2D array not constrained by the stabilizers. Let S_{24} denote the group generated by the 24 stabilizers. In principle we can choose any orthonormal basis of $V_{S_{24}}$ for the logical state, but a convenient choice is defined by stabilizing with $\overline{Z}$. The group $\langle S_{24}, \overline{Z}\rangle$ stabilizes a one-dimensional space (Proposition 2.1.4); call its state $|0_L\rangle$. Similarly, $|1_L\rangle$ is stabilized by $\langle S_{24}, -\overline{Z}\rangle$. The pair $\{|0_L\rangle, |1_L\rangle\}$ forms an orthonormal basis for the logical state. Acting with $\overline{X}$ swaps these states: since $\overline{X}$ commutes with S_{24} and anticommutes with $\overline{Z}$, we have $\overline{X}|0_L\rangle = |1_L\rangle$ and $\overline{X}|1_L\rangle = |0_L\rangle$. Therefore in this basis $\overline{X}$ and $\overline{Z}$ act exactly like the single-qubit X and Z.

Definition 4.1.3. *Logical qubit — a set of physical qubits encodes a single qubit state with respect to a set of stabilizers and logical operators $\overline{X}$ and $\overline{Z}$.*

If we multiply some of the surface-code stabilizers by a factor of -1, the code space is unchanged (commutation relations are preserved). Flipping the sign of a logical operator also preserves commutation relations, but corresponds to relabeling the logical basis states (e.g., exchanging the roles of $|0_L\rangle$ and $|1_L\rangle$ or flipping their eigenvalues).

4.2 6-ring lattice

In this section I will describe necessary notions for the full 6-ring lattice. As it is a 3d object and has layers, it is mostly convenient to show the structure in the pictures. Let's first look at the elementary block of the lattice - a node (a sphere). This node represents one 6-ring resource state 4.3. As we already know, it has 6 qubits and 6 stabilizers that describe its state. Each joint in the space represents one qubit. Therefore there are 6 joints coming out from a sphere.

And now that we have defined notation for physical qubits, we can move on to defining notation for fusions between them, forming a full 3d lattice. Let's look at the figure 4.4. Four six-rings are depicted. Each six ring has exactly one stabilizer highlighted. We can say that each joint going from one 6-ring to another denotes a fusion measurement on the corresponding qubits. This way we denoted four fusion measurements between these four 6-rings. Let's number each qubit in each six-ring according to the 4.3. This way a $Z_{(i,j)}$ is an operator Z on the ith 6-ring and jth qubit. Stabilizers highlighted are $Z_{(1,1)}X_{(1,2)}Z_{(1,3)}, Z_{(4,6)}X_{(4,1)}Z_{(4,2)}, Z_{(3,5)}X_{(3,6)}Z_{(3,1)}, X_{(2,1)}X_{(2,3)}X_{(2,5)}$. If we multiply all these stabilizers we get an operator

$$O = Z_{(1,1)}X_{(1,2)}Z_{(1,3)}Z_{(4,6)}X_{(4,1)}Z_{(4,2)}Z_{(3,5)}X_{(3,6)}Z_{(3,1)}X_{(2,1)}X_{(2,3)}X_{(2,5)}$$

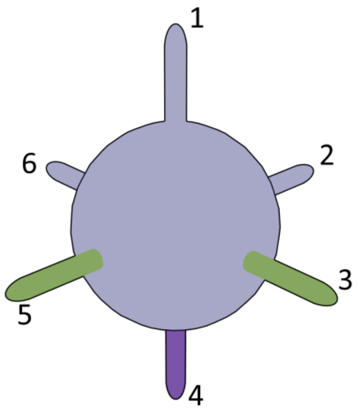


Figure 4.3: One 6 ring resource depicted in 3 dimensions. It's stabilizers have the same notations as previously discussed. Exact position of qubits is not of interest as the state is described by stabilizers. On the picture, one of the stabilizers ($Z_5X_4Z_3$) is depicted by green and purple joint.

, which is also a valid stabilizer of the state. Fusion measurements will then be:

$$M_1 = Z_{(1,3)}Z_{(4,6)}, M_2 = X_{(1,3)}X_{(4,6)}$$

$$M_3 = X_{(1,2)}X_{(2,5)}, M_4 = Z_{(1,2)}Z_{(2,5)}$$

$$M_5 = X_{(2,3)}X_{(3,6)}, M_6 = Z_{(2,3)}Z_{(3,6)}$$

$$M_7 = Z_{(3,5)}Z_{(4,2)}, M_8 = X_{(3,5)}X_{(4,2)}$$

Let's look at the stabilizers that will be left after the fusion measurements. Let's look at one specific stabilizer. Suppose that measurement M_i has the outcome m_i. Since all the measurements for fusions commute with each other, then it does not matter in which sequence to execute them (prove or reference this fact). Moreover, it means that all these measurement operators will be present in the new stabilizer group with respective multiplier of $(-1)^{m_i}$. All these measurement operators also commute with the stabilizer O, which means that it still will be present in the stabilizer group after all fusion measurements. Let's multiply O by these operators: $m_1Z_{(1,3)}Z_{(4,6)}, m_3X_{(1,2)}X_{(2,5)}, m_5X_{(2,3)}X_{(3,6)}, m_7Z_{(3,5)}Z_{(4,2)}$, which are also in the stabilizer group as we discussed. We get a simple yet intriguing stabilizer

$$m_1m_3m_5m_7Z_{(1,1)}X_{(2,1)}Z_{(3,1)}X_{(4,1)}$$

This stabilizer will play a crucial role in the description of the logical state on the lattice.

Now we can describe a 2D periodic lattice with boundary conditions.

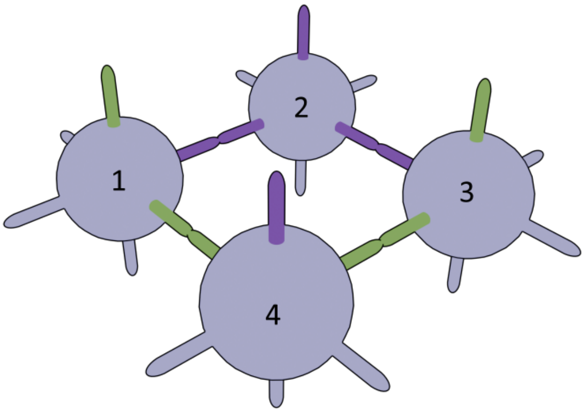


Figure 4.4: Four 6-rings forming a lattice with four fusions. Each 6-ring has one stabilizer highlighted. Green(purple) joints depict a $Z(X)$ operator on the qubit.

Let's look at 4.5. Nine 6-rings are depicted with fusions between some of them. Eight of them on the boundary has one stabilizer highlighted. Potentially we could form a bigger square lattice from 16 or 25 resource states. In any of such examples we would want to look at the boundary, at the resource states that are at the edge of the square. We can notice specific kind of stabilizers there. Taking an example of stabilizers $X_{(1,1)}X_{(1,3)}X_{(1,5)}$ and $Z_{(4,1)}X_{(4,6)}Z_{(4,5)}$. To have a desired stabilizer after all the measurement we want to introduce a series of one-qubit measurements. They should also commute with the rest of the fusion measurement operators. Let's define them as

$$M_1 = X_{(1,5)}; M_2 = Z_{(4,5)}; M_3 = Z_{(7,3)}; M_4 = X_{(8,3)};$$

$$M_5 = Z_{(9,2)}; M_6 = X_{(6,2)}; M_7 = X_{(3,6)}; M_8 = Z_{(2,6)}$$

Let's also denote some fusion measurements as

$$M_9 = X_{(1,3)}X_{(4,6)}; M_{10} = x_{(7,2)}X_{(8,5)}; M_{11} = Z_{(9,6)}Z_{(6,3)}; M_{12} = Z_{(3,5)}Z_{(2,2)}$$

Same way as above we can find that after all the measurements are done (fusions and single qubit) we will have a special class of boundary stabilizers. In the case of a distance three square lattice these stabilizers will be:

$$m_1m_2m_9X_{(1,1)}Z_{(4,1)}, m_3m_4m_{10}Z_{(7,1)}X_{(8,1)},$$

$$m_5m_6m_{11}X_{(9,1)}Z_{(6,1)}, m_7m_8m_{12}Z_{(3,1)}X_{(2,1)}$$

These are boundary stabilizers that are also important.

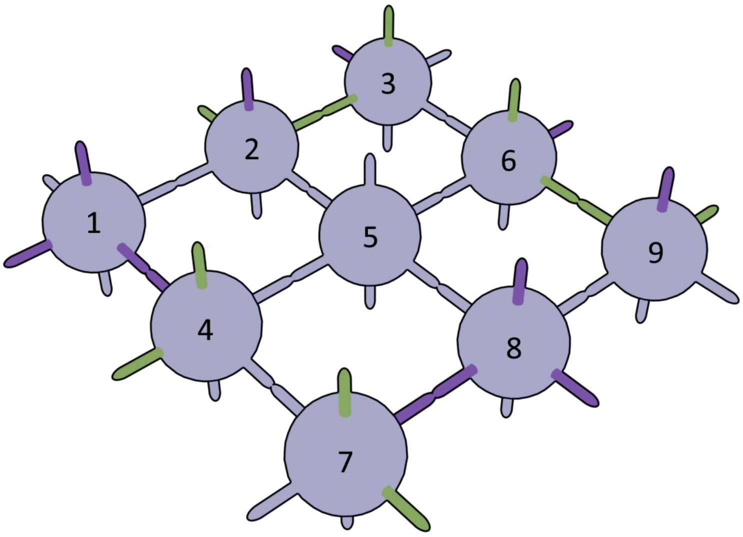


Figure 4.5: Nine 6-rings forming a lattice with boundary conditions.

Now we can associate qubits of these resource states that are facing up with the qubits of the rotated surface code of distance three. We defined rotated surface code as a number of qubits with specific stabilizers. Let's look at our nine qubits facing up. They have all the necessary stabilizers (without respect to a sign)! We have square stabilizers forming as in 4.4:

$$Z_{(1,1)}X_{(2,1)}Z_{(3,1)}X_{(4,1)}, Z_{(4,1)}X_{(5,1)}Z_{(7,1)}X_{(8,1)}$$

$$Z_{(2,1)}X_{(3,1)}Z_{(5,1)}X_{(6,1)}, Z_{(5,1)}X_{(6,1)}Z_{(8,1)}X_{(9,1)}$$

And we have the necessary boundary stabilizers:

$$X_{(1,1)}Z_{(4,1)}, X_{(8,1)}Z_{(7,1)}, X_{(9,1)}Z_{(6,1)}, X_{(2,1)}Z_{(3,1)}$$

That means that if we are only looking at the qubits facing up - they are forming a rotated surface code and thus encoding one logical qubit.

But it's not clear how to look at them separately. They might also be entangled with other qubits. Let's imagine that we do more measurements. Specifically we add this group of measurements to already existing fusions and one-qubit measurements:

$$M_i = Z_{(i,4)} | \forall i \in \{1, \ldots, 9\}$$

These measurements are of the qubits facing down and we need them only to disentangle them from the rest of the system. Same way we disentangle the rest of the qubits with measurements:

$$M_1 = X_{(1,6)}, M_2 = Z_{(3,2)}, M_3 = Z_{(9,3)}, M_4 = X_{(7,5)}$$

After them we will have all the qubits unentangled from the facing-up qubits. But all of these measurements commute with the stabilizers of the surface code that we defined on the up-facing qubits. That means that the final state of the system will still have surface code stabilizers and thus can be regarded as encoding a logical qubit.

4.3 Identity gate

All the physical qubits which are being measured with one-qubit measurement or which are involved in fusion - will be unentangled from the rest of the system and will be effectively destroyed. Identity gate in this discussion means having one logical qubit encoded in a number of physical qubits, then applying a set of operations and having another logical qubit encoded on different physical qubits, but in the same logical one-qubit state. Mind that all available operations are creation of the resource states, fusions and one-qubit measurements.

Now let's define a logical block that we could use to perform a logical identity gate.

Proposition 4.3.1. *Let's assume a network of 6-ring as was described previously in 4.5. We will make appropriate fusions and measurements described except for the qubits facing down and qubits facing up. After that we are left with* $2 \times 6 \times 9$ *qubits. Qubits facing down will be in the encoded state of a surface code. Qubits facing up will be in the encoded state of a surface code. And the composite system will be in a logical bell state. That means it will be one of*

$$\frac{\overline{|00\rangle} + \overline{|11\rangle}}{\sqrt{2}}, \frac{\overline{|00\rangle} - \overline{|11\rangle}}{\sqrt{2}}, \frac{\overline{|01\rangle} + \overline{|10\rangle}}{\sqrt{2}}, \frac{\overline{|01\rangle} - \overline{|10\rangle}}{\sqrt{2}}$$

where $\overline{|00\rangle}$ *is a state describing qubits facing down in a logical zero state and qubits facing up in a logical zero state. Same for* $\overline{|11\rangle}$*.*

Proof. First let's prove that qubits facing down will be in an encoded state of the surface code. The same way we did with qubits facing up, let's first look at the square inner stabilizers of the code. Taking 4.5 as a reference, we can see a set of valid stabilizers of the system after all fusions and measurements:

$$S = Z_{(1,2)}X_{(1,2)}Z_{(1,4)}X_{(4,6)}X_{(4,4)}X_{(4,2)}Z_{(5,4)}X_{(5,5)}Z_{(5,6)}Z_{(2,3)}X_{(2,4)}Z_{(2,5)}$$

$$M_1 = X_{(1,2)}X_{(4,6)}, M_2 = X_{(4,2)}X_{(5,5)}, M_3 = Z_{(5,6)}Z_{(2,3)}, M_4 = Z_{(2,5)}Z_{(1,2)}$$

From that we can conclude that a $Z_{(1,4)}X_{(4,4)}Z_{(5,4)}X_{(2,4)}$ will also be a valid stabilizer up to a ±1 factor due to measurement results from M_1, M_2, M_3, M_4. That is exactly one of the inner stabilizers of the surface code on the facing down qubits. In the same way the other three will be:

$$Z_{(4,4)}X_{(7,4)}Z_{(8,4)}X_{(5,4)}, Z_{(2,4)}X_{(5,4)}Z_{(6,4)}X_{(3,4)}, Z_{(5,4)}X_{(8,4)}Z_{(9,4)}X_{(6,4)}$$

Now let's look at the boundary stabilizers of the facing down qubits. Taking 4.5 as a reference, we can see a set of valid stabilizers of the system after all fusions and measurements:

$$S = Z_{(4,5)}X_{(4,4)}Z_{(4,3)}Z_{(7,6)}X_{(7,5)}Z_{(7,4)}, M = Z_{(4,3)}Z_{(7,6)}$$

With that we know that a $X_{(4,4)}Z_{(7,4)}$ will also be a valid stabilizer up to a ±1 factor due to measurement result from M. That is exactly one of the boundary stabilizers of the surface code on the facing down qubits. In the same way the other three will be:

$$Z_{(8,4)}X_{(9,4)}, X_{(6,4)}Z_{(3,4)}, Z_{(2,4)}X_{(1,4)}$$

That way we have defined a surface code on the facing down qubits and showed that after all fusions and measurements they will be encoding a valid logical state. The zero logical state on those qubits will be defined with respect to a logical Z operator, which is $\overline{Z} = Z_{(2,4)}X_{(5,4)}Z_{(8,4)}$. The logical qubit on the up-facing qubits will be defined with respect to the $\overline{Z} = X_{(2,1)}Z_{(5,1)}X_{(8,1)}$. Now let's define a correlator. Correlator is a special kind of a stabilizer of the system. It is preserved after all fusions and measurements are done and it tells us about the correlation between the logical operators on the logical qubits of two surfaces. For our case let's look at this stabilizer:

$$Stab_Z = Z_{(2,4)}X_{(5,4)}Z_{(8,4)}X_{(2,3)}X_{(5,6)}Z_{(5,3)}Z_{(8,6)}Z_{(2,6)}X_{(8,3)}X_{(2,1)}Z_{(5,1)}X_{(8,1)}$$

It is indeed a valid stabilizer of the system after all measurements and fusions. That means that we have a correlator that will also be a valid stabilizer of the system:

$$Corr_Z = m_1m_2m_3m_4Z_{(2,4)}X_{(5,4)}Z_{(8,4)}X_{(2,1)}Z_{(5,1)}X_{(8,1)}$$

with respect to measurement outcomes from:

$$M_1 = X_{(2,3)}X_{(5,6)}, M_2 = Z_{(5,3)}Z_{(8,6)}, M_3 = Z_{(2,6)}, M_4 = X_{(8,3)}$$

Let's notice that $Corr_Z = \pm\overline{ZZ}$. Which means that up to a factor it acts as a logical Z on the down-facing logical qubit and it acts as a logical Z on the up-facing logical qubit. Same way we define logical X operators on the down-facing and up-facing logical qubits.

$$\overline{X_{up}} = Z_{(4,1)}X_{(5,1)}Z_{(6,1)}, \overline{X_{down}} = X_{(4,4)}Z_{(5,4)}X_{(6,4)}$$

The stabilizer will be

$$Stab_X = X_{(4,4)}Z_{(5,4)}X_{(6,4)}Z_{(4,5)}X_{(4,2)}X_{(5,5)}Z_{(5,2)}Z_{(6,5)X_{(6,2)}}Z_{(4,1)}X_{(5,1)}Z_{(6,1)}$$

The correlator will be:

$$Corr_X = m_1m_2m_3m_4X_{(4,4)}Z_{(5,4)}X_{(6,4)}Z_{(4,1)}X_{(5,1)}Z_{(6,1)}$$

with respect to measurement outcomes from:

$$M_1 = Z_{(4,5)}, M_2 = X_{(4,2)}X_{(5,5)}, M_3 = Z_{(5,2)}Z_{(6,5)}, M_4 = X_{(6,2)}$$

Let's notice that $Corr_X = \pm\overline{XX}$. Since the logical state has two stabilizers of the form $\pm\overline{XX}$ and $\pm\overline{ZZ}$ it's easy to conclude that the composite logical state will be a logical bell state. □

An important notice is that we have access to the measurement outcomes. That means that depending on the measurement outcomes we know exactly in which state the system is. For example if $m_1m_2m_3m_4 = +1$ for $Corr_X$ and $m_1m_2m_3m_4 = +1$ for $Corr_Z$ that means that the system is in a state stabilized by $\langle +\overline{XX}, +\overline{ZZ}\rangle$. It also means that depending on the measurement outcomes we can choose different basis for the logical qubit encoded in the up-facing surface. For example, choosing the basis operator $-\overline{Z}$ instead of $\overline{Z}$ will change the state in a way that $\overline{|0\rangle}$ goes to $\overline{|1\rangle}$ and $\overline{|1\rangle}$ goes to $\overline{|0\rangle}$. The same way we can choose a basis $\{\overline{|0\rangle}, -\overline{|1\rangle}\}$ instead of $\{\overline{|0\rangle}, \overline{|1\rangle}\}$, just by saying so. That means that we can choose such a basis so that the state of our whole system is always exactly $\frac{\overline{|00\rangle}+\overline{|11\rangle}}{\sqrt{2}}$.

Now, suppose we already have 9 physical qubits that are forming a surface code and are encoding one logical qubit. We will then fuse them with the down-facing qubits from one layer of our lattice as it is shown in the 4.6. We want to show that after all fusions and measurements are done, up-facing qubits will encode the same state of the logical qubit as was encoded into the initial state. That process is called projecting a logical state onto a logical block. Since our block describes an Identity gate - it is only logical to assume that the output

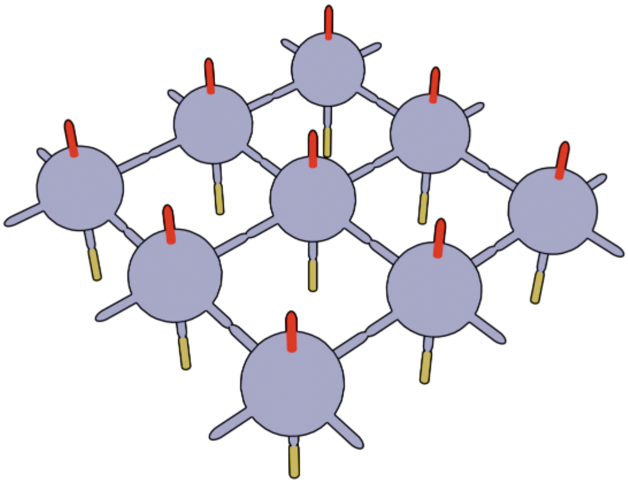


Figure 4.6: Fusing 9 qubits encoding a logical qubit with one layer of lattice. Yellow joints represent initial qubits encoding a logical qubit in a specific state. Red joints represent qubits that will be encoding the same state on the logical qubit as was encoded on yellow qubits after all fusion and measurements are done.

surface will encode the same state that was projected on the input surface. We will prove this fact using a trick called quantum teleportation.

Proposition 4.3.2. *In the described scenario, for the defined lattice of resource states, after all the fusions and measurements are done, up-facing qubits will encode the same state as was encoded in the initial qubits, with respect to corresponding logical operators and stabilizers of the codes.*

Proof. Let's first have a notation for the initial qubits. We will call an initial qubit that that is fused with the resource state i as an ith initial qubit and operators on it can be written as O_i. Notice that this notation is different from any other qubit in the system and thus unique.

Now let's define a surface code on the initial (yellow) qubits. For the initial qubits we implicitly assumed analogous set of stabilizers to that of the up-facing qubits. Let's name them:

$$m'_1X'_1Z'_2, m'_2Z'_3X'_6,$$

$$m'_3X'_9Z'_8, m'_4Z'_7X'_4$$

$$m'_5Z'_1X'_2Z'_3X'_4, m'_6Z'_4X'_5Z'_7X'_8$$

$$m'_7Z'_2X'_3Z'_5X'_6, m'_8Z'_5X'_6Z'_8X'_9$$

(Notice that boundary stabilizers are mirrored). And the Corresponding logical gates would be $\overline{Z'} = Z'_2X'_5Z'_8, \overline{X'} = X'_4Z'_5X'_6$.

Now let's first prove a simpler fact. Suppose that we have already done all fusions and measurements. Then measuring with respect to operators $\overline{Z'Z_{down}}$ and $\overline{X'X_{down}}$ will not affect the system in any way (up to a global phase). Here $\overline{Z_{down}}$ is a logical Z operator on the down-facing qubits and $\overline{X_{down}}$ is a logical X operator on the down-facing qubits. This fact is easy to see. Both $\overline{Z'Z_{down}}$ and $\overline{X'X_{down}}$ operators will be inside a stabilizer group with some factors. That means that these measurements won't affect the system except for maybe changing a global sign.

Since these measurements do not affect our system and they also commute with all the fusions and measurements that we are performing - we can assume that we are measuring the operators $\overline{Z'Z_{down}}$ and $\overline{X'X_{down}}$ first, and then applying fusions and measurements of the network.

Now let's see what happens when we perform $\overline{Z'Z_{down}}$ and $\overline{X'X_{down}}$ measurements. Suppose that the initial qubits were in the logical state $a\,|0\rangle + b\,|1\rangle$. That means that the whole system was in the state

$$(a\overline{|0\rangle} + b\overline{|1\rangle}) \otimes (\overline{\frac{|00\rangle + |11\rangle}{\sqrt{2}}})$$

We know that $\overline{Z} = \overline{|0\rangle\langle 0|} - \overline{|1\rangle\langle 1|}$ and $\overline{X} = \overline{|0\rangle\langle 1|} + \overline{|1\rangle\langle 0|}$. Same way we know that

$$\overline{Z'Z_{down}} = (+1) * (\overline{|00\rangle\langle 00|} + \overline{|11\rangle\langle 11|}) + (-1) * (\overline{|01\rangle\langle 01|} + \overline{|10\rangle\langle 10|})$$

for the respected logical qubits. And the same statement for X:

$$\overline{X'X_{down}} = (+1) * (\overline{|{+}{+}\rangle\langle{+}{+}|} + \overline{|{-}{-}\rangle\langle{-}{-}|}) + (-1) * (\overline{|{+}{-}\rangle\langle{+}{-}|} + \overline{|{-}{+}\rangle\langle{-}{+}|})$$

With the first measurement we have:

$$p(+1) = (a^*\overline{\langle 0|} + b^*\overline{\langle 1|}) \otimes (\overline{\frac{\langle 00| + \langle 11|}{\sqrt{2}}})(\overline{|00\rangle\langle 00|} + \overline{|11\rangle\langle 11|})(a\overline{|0\rangle} + b\overline{|1\rangle}) \otimes (\overline{\frac{|00\rangle + |11\rangle}{\sqrt{2}}}) =$$

$$= (a^*\overline{\langle 0|} + b^*\overline{\langle 1|}) \otimes (\overline{\frac{\langle 00| + \langle 11|}{\sqrt{2}}})(\frac{a\overline{|000\rangle} + b\overline{|111\rangle}}{\sqrt{2}}) = \frac{|a|^2 + |b|^2}{2} = \frac{1}{2}$$

And the corresponding states will be

$$+1 \to a\overline{|000\rangle} + b\overline{|111\rangle}; -1 \to (\overline{|01\rangle\langle 01|} + \overline{|10\rangle\langle 10|})(a\overline{|0\rangle} + b\overline{|1\rangle}) \otimes (\overline{\frac{|00\rangle + |11\rangle}{\sqrt{2}}}) =$$

ca

$$a\overline{|011\rangle} + b\overline{|100\rangle}$$

Let's deal with the $+1$ case first. Useful to notice that $\overline{|{+}{+}\rangle\langle{+}{+}|} + \overline{|{-}{-}\rangle\langle{-}{-}|} = (\overline{|00\rangle\langle 00|} + \overline{|01\rangle\langle 01|} + \overline{|10\rangle\langle 10|} + \overline{|11\rangle\langle 11|} + \overline{|00\rangle\langle 11|} + \overline{|01\rangle\langle 10|} + \overline{|10\rangle\langle 01|} + \overline{|11\rangle\langle 00|})/2$ and $\overline{|{+}{-}\rangle\langle{+}{-}|} + \overline{|{-}{+}\rangle\langle{-}{+}|} = (\overline{|00\rangle\langle 00|} + \overline{|01\rangle\langle 01|} + \overline{|10\rangle\langle 10|} + \overline{|11\rangle\langle 11|} - \overline{|00\rangle\langle 11|} - \overline{|01\rangle\langle 10|} - \overline{|10\rangle\langle 01|} - \overline{|11\rangle\langle 00|})/2$. Applying the $\overline{XX}$ we get:

$$p(+1) = (a^*\overline{\langle 000|} + b^*\overline{\langle 111|})(\overline{|{+}{+}\rangle\langle{+}{+}|} + \overline{|{-}{-}\rangle\langle{-}{-}|})(a\overline{|000\rangle} + b\overline{|111\rangle}) =$$

$$= (a^*\overline{\langle 000|} + b^*\overline{\langle 111|})((\overline{|00\rangle} + \overline{|11\rangle}) \otimes (a\overline{|0\rangle} + b\overline{|1\rangle}))/2 = \frac{1}{2}$$

The corresponding state will be $(\frac{\overline{|00\rangle}+\overline{|11\rangle}}{\sqrt{2}}) \otimes (a\overline{|0\rangle} + b\overline{|1\rangle})$. And the corresponding state of the -1 outcome will be:

$$(\overline{|{+}{-}\rangle\langle{+}{-}|} + \overline{|{-}{+}\rangle\langle{-}{+}|})(a\overline{|000\rangle} + b\overline{|111\rangle})/(\sqrt{2}) = (\frac{\overline{|00\rangle} - \overline{|11\rangle}}{\sqrt{2}}) \otimes (a\overline{|0\rangle} - b\overline{|1\rangle})$$

Now what's left is to look at the -1 outcome of the ZZ measure:

$$p(+1) = (a^*\overline{\langle 011|} + b^*\overline{\langle 100|})(\overline{|{+}{+}\rangle\langle{+}{+}|} + \overline{|{-}{-}\rangle\langle{-}{-}|})(a\overline{|011\rangle} + b\overline{|100\rangle}) =$$

$$= (a^*\overline{\langle 000|} + b^*\overline{\langle 111|})((\overline{|01\rangle} + \overline{|10\rangle}) \otimes (a\overline{|1\rangle} + b\overline{|0\rangle}))/2 = \frac{1}{2}$$

And the corresponding state will be $((\frac{\overline{|01\rangle}+\overline{|10\rangle}}{\sqrt{2}}) \otimes (a\overline{|1\rangle} + b\overline{|0\rangle}))$. And the corresponding state for the -1 outcome will be:

$$(\overline{|{+}{-}\rangle\langle{+}{-}|} + \overline{|{-}{+}\rangle\langle{-}{+}|})(a\overline{|011\rangle} + b\overline{|100\rangle})/(\sqrt{2}) = ((\frac{\overline{|01\rangle} - \overline{|10\rangle}}{\sqrt{2}}) \otimes (a\overline{|1\rangle} - b\overline{|0\rangle}))$$

Now we can clearly see that the output qubit is unentangled from the rest of the system in any of those four outcomes. Moreover, depending on the measurement outcomes we can choose such a basis that the output state will be $a\overline{|0\rangle} + b\overline{|1\rangle}$. After that we need to apply the rest of the fusions and measurements. None of them act on the output qubits (the qubits that encode output logical state). And since the output qubits are unentangled - these fusions and measurements will not affect the output qubit in any way.

□

4.4 Quantum memory

Quantum memory experiment is about conserving the logical state in time. That we can achieve by repeatedly creating layers of fusion lattice and fusing them with each other. We already have shown how it can be done for

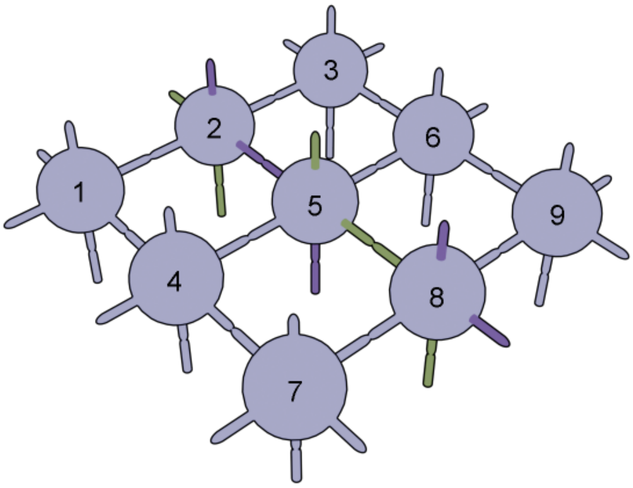


Figure 4.7: Correlator between the initial qubit's logical operator and final's.

one layer and one prepared logical qubit. Let's see how we can operate when there is more than one layer.

We can look at 4.8 and see that now there are three layers to our fusion network. Initial qubits are taken as the up-facing qubits of the first layer and final qubits are the up-facing qubits of the last layer. We want for the initial qubits to have the same state before any measurements and fusions as the final qubits after all measurements and fusions. To define the logical state on initial and final qubits we need to account for the stabilizers of the surface code on them. As in the section above, we saw that stabilizers of the surface code change with the new layer. Boundary stabilizers are being shifted due to the nature of the correlator defined. Without this shift the correlator would not be commuting with the rest of the stabilizers of the code and thus would not be able to define a logical state. That is why we can see a periodical shift in the boundary stabilizers for each layer.

This way we have shown how to define a fusion network for the quantum memory experiment. Note that, as the surface code, our network can be of different size and not only size. For example, it is easy to define a surface on the qubit lattice of size $(2d+1)^2$, for any integer d greater than 1. The same way we can define a layer of the 6-ring resource states of the size $(2d+1)^2$, for any integer d greater than 1. And the number of layers in the system can be arbitrary, the only condition is periodically changing boundary stabilizers. This way we can define a fusion network of size n by n by m for m is the number of layers.

From the physical perspective, we create each layer iteratively. First, we create the first layer and measure all qubits except initial. Then we create the second layer, then we execute all measurements and fusions between the first and the second layer. After that we are once again left with 9 qubits encoding a logical state. The process can be repeated any number of times. It is hard to preserve a state encoded in the same photons, since photons have a tendency to dissipate in time. However, if we introduce a new set photons each time - that can potentially alleviate these errors due to dissipation. That is the notion of MBQC and specifically FBQC. We only use three operations: single qubit measurements, fusions, creation of the resource states(6-rings). That means that this type of computation is convenient from the operational point of view. Next section will describe errors that could occur during our computation and how to deal with them.

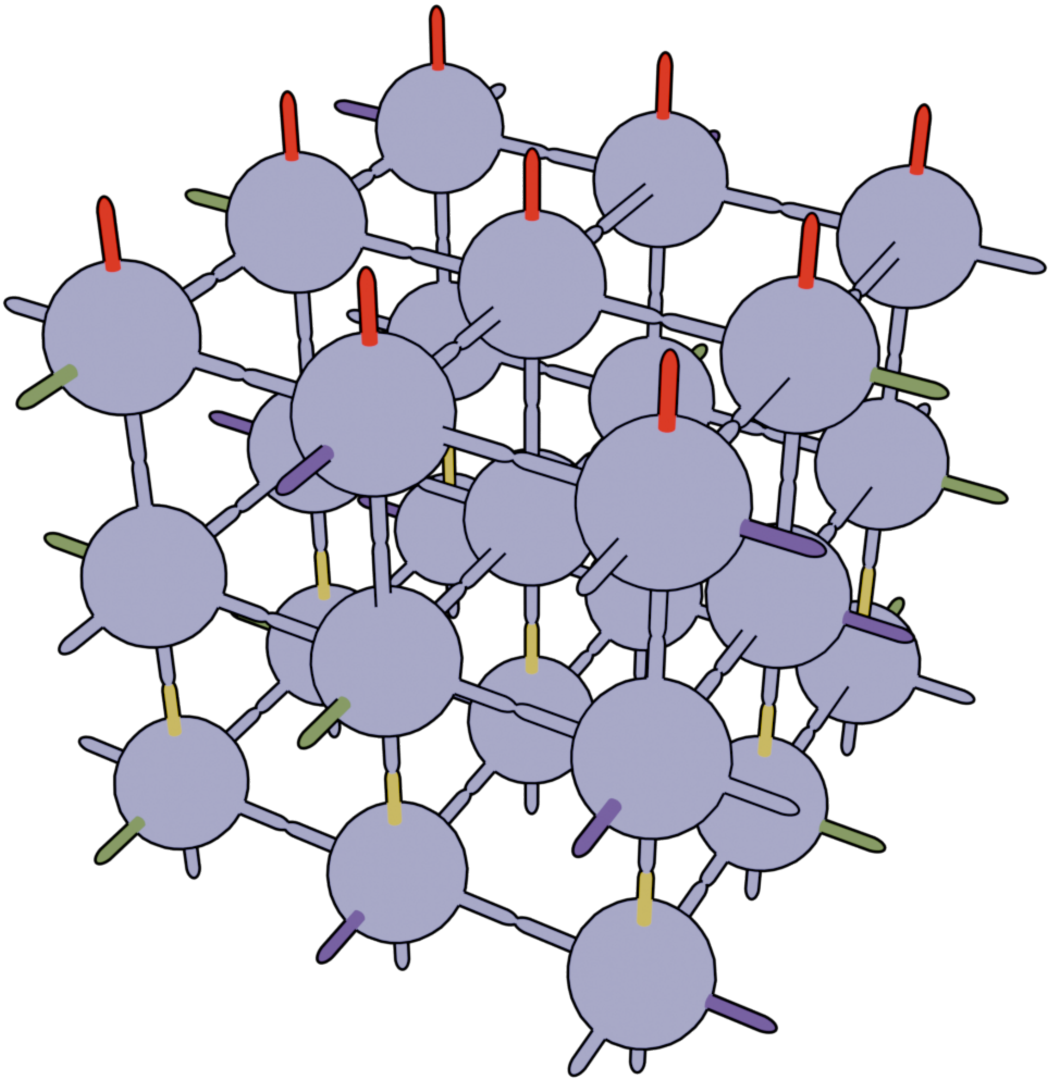


Figure 4.8: Fusion network that consists of 3 layers. Initial qubits encoding initial quantum state are colored yellow. Final are colored red. Purple(green) joints represent qubits on which the $X(Z)$ measurement is to be applied. Uncolored boundary joints will need a Z measurement as default. Touching joints will have a fusion on them.

5
Error Models and Threshold Estimation

In this chapter, we will familiarize ourselves with the error models that could occur on the fusions. We will see how these errors can be detected and accounted for. How they can be corrected and how they affect a logical state of the system. Finally, we will see how we can estimate the error rate of the system and calculate the threshold for the 6-ring fusion network. I don't cover the topic of the "right" physical model for the noise on purpose. This question is quite complex and deep. It is partially covered in the literature [Bartolucci et al. 2023].

5.1
Computational Framework

In this section we will describe the steps that are needed from the start to end to perform a quantum memory experiment. During the process we will work with quantum data, that has a physical part, and with classical data, that will be used to control the process.

We start by preparing all the necessary resource states that will later be fused to create a fusion network and measured to propagate the quantum state. That part is usually straightforward and we will not focus on it. Resource states are prepared by specialized devices called resource-state generators (RSGs). As physical instruments, they are subject to errors and are discussed in the literature [Li et al. 2022, Bartolucci et al. 2023]. However, we will not discuss them in this paper and assume perfect RSGs.

The fusion network itself consists of two parts: logical blocks that perform Clifford operations, and a process called state projection. Universal quantum computation requires non-Clifford operations, which are not native in this fusion framework. To achieve universality one uses state injection (projection of additional states), e.g., T-state injection [Bravyi e Kitaev 2005]. We will not discuss universality further here, but note that logical operations (represented by fusion networks) and state injection are separate processes. In our case, we only have an identity-gate operation represented by one logical block, and the state is projected once. As described in the previous chapter, the projection is done by applying fusions between the projected state and the logical block. In principle this projection can be done before or after the fusions within the logical block; in practice it is done before because fusions are applied sequentially between adjacent layers. For our purposes, we do not focus on state projection itself and its errors; we study the logical block and its quantum error-correction properties.

After all resource states are prepared we perform the required fusions and measurements. Physically, this is also straightforward, and all fusions (two-qubit projections built from one-qubit measurements) can be modeled as being performed by identical devices. Therefore we consider a single error model for all fusions/measurements. An error model describes how errors occur during the process. It is convenient to model errors as being applied after the process, i.e., as a noisy channel applied after every fusion or measurement. Some errors cannot be described this way, but we can still account for them as noted below.

After all the noisy fusions and measurements are done we are left with a logical state and a set of classical outcomes to be processed. From this data we can infer the behavior of the noise during our computation. If our inference is sufficient, we can deduce an error pattern and account for it. Importantly, once a specific error pattern is predicted, the correction is applied on the classical side only (Pauli frame updates): there is no physical correction on the photonic system. This is possible because the errors we consider and the logical blocks are Clifford.

After we have processed the classical data and accounted for a correction, the last step is to interpret the quantum data consistently. If the predicted correction (and interpretation) matches what actually occurred, decoding succeeds and the final logical state is correct; otherwise decoding fails and the final logical state is incorrect. The logical error rate (LER) is the probability of decoding failure.

It is useful to define the LER for an isolated logical block. In that case we still apply fusions and measurements, but we evaluate whether the block's correlation surfaces (logical operators propagated through the block) match the corrected prediction. This allows us to put aside state projection and focus experiments on the fusion network implementing the memory.

For different error models and different fusion networks there can be different LERs. The hope of quantum error correction code is to suppress the LER. That means that by increasing the amount of qubits that encode a logical state we hope to achieve a better chance of correctly detecting an error that occured during the computation. Same goes for the incresing amount of layers in the fusion network. The threshold is the point of an error parameter before which increasing the amount of qubits(layers) in the network will decrease the LER, and after which it will increase it. There is a threshold theorem that states that such point exists [Aharonov e Ben-Or 2008, Knill, Laflamme e Zurek 1998]. Such statement is important and relies on the physical arguments. For our case of a 6-ring fusion network and surface code encoding a logical state such point should exist as well. Finding it can be done by accurate simulations of the system.

5.2 Error Model for the Fusion Process

We have defined fusion as a $\langle XX, ZZ\rangle$ measurement. Now let's look at how it is done and which errors can occur during it.

As FBQC is primarily dictated by photonic architectures - the main physical object in our story will be a photon. Now, we can consider different ways of encoding a quantum state in a photon be it in the polarization or in a special mode (dual-rail encoding) [Kok et al. 2007, Pan et al. 2012]. In both cases a fusion operation can be defined [Browne e Rudolph 2005, Bartolucci et al. 2023, Knill, Laflamme e Milburn 2001]. And in both cases there is the same pattern for the error model that can be observed during the fusion. Fusion can have more than two bits of information outputted associated with it. Let's consider the scenarios in which the fusion can result:

- **Fusion success:** in this option the qubits are measured in the intended basis of X_1X_2, Z_1Z_2.
- **Fusion failure:** here the qubits are measured in a separable basis of Z_1I_2, I_1Z_2. In this case we can still obtain the outcome of the Z_1Z_2 measurement, but the entanglement is not created.
- **Fusion erasure:** in this case none of the intended outcomes can be retrieved and the mathematical description of the process can vary by hardware. The qubits may leave the computational subspace; different effective models are possible, as discussed in the literature.

A lucky thing is, that the loss can usually be heralded. That means that from the outcomes of the fusion process we know if it was successful, failed or erased.

We are not accounting for the single-qubit measurements errors. Because they are always only applied on the resource states and so we can think of them as just reduced resource states that were generated in this way.

Moreover in our simulations we will only look at the errors on obtaining the outcome from Fusion success. That means that each bit obtained from fusions will be flipped with some small probability, implementing an error on obtaining this particular outcome. This error model is possibly the easiest to think of. It is easy to account for in the decoding process and in the correction process.

That way we described the error model on the fusion and which classical data it outputs.

5.3 Decoding process

We now describe the decoding process. It involves no physical intervention or basis changes; we only interpret the classical outcomes and derive a correction. We first fix notation.

5.3.1 Checks

Once we have a code, we know how the fusion network will look like. It consists of resource states and Fusion Measurements. Let's name the group of all stabilizers for all the resource states as R. It can be generated by the union of generators for each individual resource state. We will also have a notation for the stabilizer group of all the fusion measurements. F - is generated by the union of generators for all fusions that will be executed during the computation.

We could notice from previous chapters that the outcomes of fusions are usually non-deterministic. However in the decoding process we need to rely on some invariants in the classical data. Classical data must not be completely random. For that let's define a check operator:

Definition 5.3.1. *Check operator is an operator from a group C. Group C is called checks and is defined as $C := R \cap F$.*

The idea of the check operator is fairly similar to that of a stabilizer in a surface code. We use them to detect if an error has occured. Now let's prove a similar statement that will help us understand the role of the checks in the error detection process.

Proposition 5.3.2. *In the computation without any errors, the outcomes of the checks will always be $+1$.*

Proof. Let's first clarify what does it mean "the outcome of the check", since we are not directly measuring any of those check operators. But since $C = R \cap F$, that means that $C \subset F$. That means that for an arbitrary $c \in C$ we have a number of fusions $\{f_1, f_2, \dots f_n\} \subset F$ that form c by multiplication $f_1 \cdot f_2 \cdot ... \cdot f_n = c$. That means that after measuring all fusions we will have an outcome associated with each measurement in $\{f_1, f_2, \dots f_n\}$. Multiplying these results is interpreted as an outcome of the check.

c - is a stabilizer of the system before any fusions. c - commutes with all the fusions. That means that c will be a valid stabilizer even after all the fusions. Now it is enough to notice that if we were to measure c on the final state the outcome will be the product of the outcomes for each fusion measurement that makes it up (since obviously we can make this measurements on the final state and results will be the same). It is also true, that when we measure a stabilizer - we get $+1$. □

5.3.2 Matching graph

A usual description of the decoding problem can be reformulated in the graph theory. The graph is weighted and bipartite. The verticies are error mechanisms and checks. Each vertex corresponds to a single error mechanism that can occur or to a check. For example in the fusion procces, a flip of one of the measruements would be an error mechanism. The edges of the graph will connect an error mechanism with a check iff when this error triggers - the check changes parity (from $+1$ to -1 or the other way). The weight of the edge corresponds to the probability of adjacent error mechanism occuring. Such graph is called matching graph.

Important to notice that in the definition of the matching graph the correlation between different errors is not captured. Meaning that if two errors occur - they might flip a different set of checks that just a xor of their individual check sets. However, such construction still can serve as a good approximation of how errors affect checks.

After all the fusions are done, we can calculate a syndrome of the computation:

Definition 5.3.3. *A syndrome is a set of flips for all checks. Can be thought of as a vector $v \in \{0,1\}^n$, where n is the number of all checks.*

Given a syndrome and a matching graph the task of decoding becomes simple. Find the most plausible set of error mechanisms that triggered a given syndrome.

Definition 5.3.4. *The decoding problem can be formulated as the following optimization problem:*

$$\hat{E} = \arg\max_{E} P(E,S)$$

where $\hat{E}$ is the most likely set of error mechanisms, E represents a possible set of error mechanisms, and S is the observed syndrome. The probability of a set of error mechanisms $P(E,S)$ is the product of all probabilities of all error mechanisms in the set iff E gives a syndrome S, and is $-INF$ otherwise.

It might be that no set of error mechanisms could create a given syndrome. In such case we can either say that decoding has failed, or find the most plausible answer in any definition of what might *plausible* mean. In practice decoding does not fail that often and it does not make sense to complicate the problem.

5.3.3 Correction process

The only error model that we are going to be considering during simulations only allows for the errors on the obtaining of the measurement outcomes. This error model does not disturb the physical state of the system. However, in our framework there is also a classical counterpart of the simulation that describes interpretation of the physical data and potentially can influence the evolution of the simulation itself. In the Identity gate experiment we don't have gates applied on the physical system depending on the classical data, but the output state can be different depending on the classical data because in the definition of logical qubit it is used. This way the output state can change, just from a wrong interpretation of the data. In a general case, usage of classical data can be postponed until the end of the computation if certain criteria is met (Pauli tracking) [Paler et al. 2014].

For example, when we calculate $m_1 m_2 m_3 m_4$ for $Corr_Z$ the erroneous $m1$ will directly affect a logical Z operator, in respect to which we will define the output logical qubit.

Correction in such scenario is straight forward. A set of measurements M_r correlates to the actual state evolution of the system. A set of measurements M_1 were flipped during the simulation and recorded. A possibly different set of measurements M_2 was decoded to be flipped. The only thing left to do is to interpret classical data in a way that already accounts for the decoded result. Instead of the using correct measurements, which are $M_r \oplus M_1$, we will be using $M_r \oplus M_2$. Thus there is still a chance for a computation to have a logical error.

Logical error would be counted if the observable Z or X on the logical qubit was flipped with respect to the expected outcome. That does not necessarily means that we must measure both observables on the logical qubits. It is enough to understand the state in which they are and the state in which we think they are. That is the essential description of the correction process.

6
Simulation of stabilizer circuits

Stabilizer circuits can be simulated efficiently using tableau- and stabilizer-based representations [Gottesman 1998, Nielsen e Chuang 2010, Aaronson e Gottesman 2008]. We prove correctness of one very important optimization of this approach, referred to as Pauli frames [Gidney 2021].

6.1
Proof of Pauli frames

Here I prove that the Pauli frame algorithm described in [Gidney 2021] indeed reproduces the correct measurement statistics. A Pauli frame is a structure that contains one Pauli matrix for each qubit; it can be represented with $2n$ bits and thought of as a Pauli operator.

6.1.1
Algorithm

In our circuit we only allow Clifford gates, Pauli errors, reset operations, measurement operations in the basis Z and all qubits are assumed to be initialized in the state $|0\rangle$. Here Clifford gates are any gates from the group generated by the set: Hadamard gate, S gate, CNOT gate. One important property of this group is that it consists exactly of all operations that send a stabilizer state into stabilizer state. Stabilizer state is a state that is stabilized by Pauli operators. The algorithm itself can be described with these steps:

1. A reference sample is collected from the target circuit, with all noise processes disabled, using some other method. The reference sample can be used and reused as many times as needed. For the measurement with index m we will have a r_m as a result of this measurement in the reference frame. It can only be 0 or 1.

2. A Pauli frame P is initialized with a randomly chosen I or Z term on each qubit.

3. The Pauli frame is advanced through the circuit.

   - When the frame crosses a Clifford operation C, the frame undergoes the update $P \longrightarrow CPC^\dagger$
   - When the frame crosses a reset, with 50% probability the qubit's term in the frame is set to the identity operation and with 50% probability it is set to Z.

- When the frame crosses a Pauli error channel, a Pauli product is sampled from the error channel and multiplied into the frame.
- When the frame crosses a measurement, the result of that measurement is reported as $r_m \bigoplus x_q$, where x_q is true iff the Pauli frame has X or Y on the target qubit. Then a Z on the target qubit is multiplied into the frame with 50% probability.

4. For more samples - repeat 2) . . . 3).

6.1.2 Statement

The probability of observing measurement pattern ρ from Pauli Frame algorithm is the same as observing the same pattern with the actual perfect quantum computation of the given circuit (including resets, measurements and probabilistic noise).

6.1.3 Proof

For our convenience we can imagine running three simulations simultaneously. First is the reference frame simulation, second is the Pauli Frame simulation (according to the algorithm), third is the perfect simulation. We can have a notion of time, meaning that at each time we have passed a specific amount of gates, operators, errors. This way we can know a state of the system at a specific time t.

In the algorithm, we use randomness for determining PauliFrame initialization and random Z application on it after resets and measurements, measurement outcomes, sampling pauli operators from probabilistic Pauli errors. We can say that we determine a random outcome from a predetermined sequence of 0s and 1s. Let's say that we determine PauliFrame initialization and Z application on it from $RPF = \{0, 1, 0, 0, 1, \dots\}$. The sequence itself can be any sequence of zeros and ones, but for the same sequences we will have the same results of PauliFrame initialization and random Z application. Same way we can determine measurement outcomes of perfect simulation from $RMeas = \{0, 1, \dots\}$. Same way we can determine Pauli operators at probabilistic Pauli errors from $RErrors = \{\dots\}$

Here we should notice that Probabilistic Pauli noise does not depend on the evolution of the simulation. For the perfect simulation or for the Pauli Frames algorithm, which Pauli noise will be applied does not depend on the $RMeas$ or RPF. It only depends on it's own randomness, which is drawn from $RErrors$. In the end we want to prove that the probability of obtaining a measurement pattern m is the same for both algorithms. That means that the ratio of $RMeas$ and $RErrors$ that generate m on the perfect simulation to all $RMeas$ and $RErrors$ is the same as the ration of RPFs and $RErrors$ that generate m on the Pauli Frame simulation to all RPFs and $RErrors$. For that it is enough to show that for any given $RErrors = R_1$ the ratio of $RMeas$ that generate m on the perfect simulation to all $RMeas$ is the same as the ration of RPFs that generate m on the Pauli Frame simulation for the same $RErrors = R_1$ to all RPFs. In the following discussion we assume that the noise is deterministic. We fixed $RErrors = R_1$ and will be proving the main statement just for these errors and count them as deterministic.

Let's also recall that for Clifford circuits a measurement in the Z basis is either deterministic or has probability 1/2 of returning 1 (for a fixed time), because the state is a stabilizer state (see, e.g., [Nielsen e Chuang 2010, Aaronson e Gottesman 2008]). However for Pauli frames that might not be obvious a priori; we prove below that this property still holds for the frame simulation.

Now let's see the state of the system at any time as

$$\{P_{\rho,t}, |\psi_t\rangle, S_t = \langle S_1, \dots, S_n\rangle, PFMP_t, PSMP_t\}$$

Here $|\psi_t\rangle$ is the state of the perfect quantum simulation. S_t is it's generators for the stabilizer group. And $PFMP_t$ is a set of zeros and ones that correspond to a measurement pattern observed on Pauli Frame simulation from RPF before the time t. And $PSMP_t$ is a set of zeros and ones that correspond to a measurement pattern observed on the perfect simulation from $RMeas$ before the time t. Now, $P_{\rho,t}$ is the set of all possible Pauli Frames at this time (meaning Pauli Frames that could have occurred at time t for any RPF that produces the $PFMP_t$). If we just use $PFMP$ or $PSMP$ notation - it means that they represent a vector of bits returned by the whole circuit ($PFMP_t, PSMP_t$ for the last moment of time).

Before proving the main statement, let's prove another one, which will be a nice additional knowledge about Clifford simulations:

Lemma 6.1.1. *Suppose we are at a time t, and next operation is a measurement operation. Then independently of $PSMP_t$ this measurement will either be random or deterministic. Meaning that if this measurement is deterministic for some $PSMP_t$, then it will be deterministic for all $PSMP_t$s, if it is random for some $PSMP_t$, it will be random for all $PSMP_t$s.*

Proof. Let's prove this fact by induction on the number of operations before the the measurement under consideration. Let's suppose that there was

$n-1$ measurements before our measurement at a time t. Then let's name our measurements as $M_1, M_2, \ldots, M_{n-1}, M_n$. And so we want to prove that determinism of the measurement M_n is irrespective of the outcomes of the measurements $M_1, M_2, \ldots, M_{n-1}$ in a perfect quantum evolution.

Let's also denote the stabilizers of the state at some points in time. We will denote S^{2i} a stabilizer group for a state of the system at a time right before the measurement M_i, and S^{2i+1} right after the measurement M_i. Be careful as those are groups and not stabilizers themselves.

By the induction's assumption, all the other measurements already follow the statement, as for $M_1, M_2, \ldots, M_{n-1}$ measurements, their determinism is not conditioned on the previous measurements. It is also not conditioned on the measurements in the future (since no one can know what is going to happen in the future), and thus they are always either random or deterministic.

Let's prove that M_n is either random or deterministic. To achieve that we will look at the evolution of the stabilizer group in time. Suppose we already ran one perfect simulation. We got measurement results $m_1, \ldots, m_{n-1}$, and we got specific states at each point of time that correspond to the groups $S^i, i \leq 2n-2$. Now we run a second perfect evolution and track how it's stabilizer groups are different from the first ones. It's easy to see that the first stabilizer group will be absolutely the same, since no measurements came before it. We will denote it as S'^1, and say $S'^1 = S^1$. We know that after the first measurement S^1 becomes S^2. Let's look at what will happen to S'^1. If M_1 was deterministic, then $S'^2 = S^2$. If it was random, then might have changed the group in a non-obvious way. Specifically we know that depending on the measurement outcome - one generator of the stabilizer group either acquires a factor of -1 or it does not. That means that depending on the measurement outcome, some stabilizers in the group will acquire a factor of -1. We don't want to track exactly which stabilizers those were. Instead, let's have a convenient notation to say that one group of stabilizers is different from another in that some stabilizers are different by a factor of -1.

We define R - equivalence relation on the stabilizer groups. We say that $P \sim Q \leftrightarrow \forall p \in P \exists q \in Q : p = \pm q$, where $P \subset G_n, Q \subset G_n$.

And it means that $S'^2 \sim S^2$. We don't know if it actually has any stabilizers multiplied by -1, it might also be that $S'^2 = S^2$, but we don't know that, we don't care about that and we don't specify that in the $S'^2 \sim S^2$ notation. Now we know that $S^3 = CS^2C^\dagger$ for some Clifford operation that acts between those two measurements. We also know that $S'^3 = CS'^2C^\dagger \sim CS^2C^\dagger = S^3$, since conjugation by C maps one equivalence class to another. Now let's look at M_2, if it's deterministic than it does not change the state in both cases and we can write $S'^4 \sim S^4$. If it is random, then it is random in both cases. But the set of the stabilizers which anti-commute with M_2 is the same in S^3 and S'^3, which means that the act of the measurement will stay the same up to -1 factors on some of the stabilizers (since -1 factors do not change commutator relationship with M_n and thus do not affect the act of the M_n on the stabilizer group). That means that we can once again write $S'^4 \sim S^4$. By now we notice that our invariants $S'^{2i} \sim S^{2i}$ and $S'^{2i+1} \sim S^{2i+1}$ will be holding through out the whole computation until the M_n. But if it holds right up until M_n - that means that it will be either random in both cases, or deterministic.

We still have not discussed the reset operation. It is not Clifford and it might not produce explicit measurement outcome in some architectures. But for the proof of this theorem it is enough to see a reset operation as a measurement followed by a conditional X gate. Let's show that our invariant holds for the reset operation as well. Suppose that S^i is a stabilizer group before the reset, S^{i+1} - after the measurement and S^{i+2} - after the conditional application of X gate. We know that $S'^i \sim S^i$, let's prove that $S'^{i+2} \sim S^{i+2}$. It's easy to see that $S'^{i+1} \sim S^{i+1}$. Let's see what happens on the conditional gate. If we apply it in both simulations - the statement is correct. If we don't apply it in both simulations - the statement is obviously correct too. Suppose we apply it only on S^{i+1} simulation and not on the S'^{i+1}. We know that either Z_q or $-Z_q$ is inside the S^{i+1} group after the measurement. And Z_q also commutes with all the stabilizers inside S^{i+1}. That means that applying an X_q can only change the sign of any stabilizer in S^{i+1}. And thus: $S^{i+2} \sim S^{i+1} \sim S'^{i+1} = S'^{i+2}$. If we apply X only on the S'^{i+1} simulation the story is the same. Now we showed that invariants $S'^{2i} \sim S^{2i}$ and $S'^{2i+1} \sim S^{2i+1}$ hold for a reset operation as well. The last thing to check is that this property holds for Pauli probabilistic errors. Since we are only looking on the deterministic noise - it affects both simulations as Clifford gate. □

Theorem 6.1.2.

$$\forall t, \forall PSMP_t, \forall PFMP_t : P_{\rho,t} = \{ \prod_{0<i\leq n} (S_i^{\delta_i})F \Big| \delta \in \{0,1\}^n \}$$

Where F is any Pauli Frame from $P_{\rho,t}$. We can think of PauliFrames as matrices from Pauli group with factor 1(since actual Pauli frame does not store information about the factor). This way we can multiply stabilizers into PauliFrames. And we can think of $P_{\rho,t}$ as a set of matrices, or as a set of PauliFrames which would be the same matrices from Pauli group, but with factors 1. (Mind that the theorem is true only for the Pauli Frames and not the matrices). We can notice right away that the statement is ignorant to the *specific generators of the stabilizer group that we choose. Meaning that we could have chosen different generators for $|\psi_t\rangle$ and the set of Pauli Frames would still be the same (because we multiply each operator that can be generated by generators).*

Moreover: Pauli frame $\prod_{0<i\leq n}(S_i^{\delta_i})F$ corresponds to exactly $\frac{1}{2^n}$ fraction of all RPFs, that produce the $PFMP_t$. That means that we are getting this specific Pauli Frame on time t with probability $\frac{1}{2^n}$ for a specific measurement pattern observed. More specifically the fraction of RPF_ts that generate $PFMP_t$ and that correspond to a specific Pauli Frame $\prod_{0<i\leq n}(S_i^{\delta_i})F$ to all RPF_ts that generate $PFMP_t$ is equal to $\frac{1}{2^n}$ for any such Pauli Frame.

Also important to note that $RMeas$ and RPF are independent and are not correlated. That means that there is no correlation between $PSMP$ and $PFMP$.

Proof: 6.1.2. Now let's make sure that this property holds in the first instance of time - right after the initialization of all qubits. All qubits are initialized in $|0\rangle$ state, Pauli Frame initialized as I and Z terms are multiplied into Pauli Frame with probability 50%. Which means that $P_{\rho,t} = \{\prod_{0<i\leq n}(Z_i^{\delta_i})I\}$, for all δ vectors. But this is exactly what we wanted, because Z_i are stabilizers of the $|0\rangle$ state and δ is uniformly distributed across RPF. We will be proving the statement by induction on t. Since we have already proved the base case, we only need to formulate the Induction Hypothesis and Induction Step. Induction Hypothesis: the statement holds for a time t. Induction Step: If the property holds for t, then it will hold for $t+1$. We only need to show that this property holds after application of any Clifford operation, measurement, reset or noise operation.

1. Let's show that it holds after a Clifford operation C. New generators are $\langle CS_1C^\dagger, \ldots, CS_nC^\dagger\rangle$, and new Pauli Frames are $CPC^\dagger = \prod_{0<i\leq n}(CS_i^{\delta_i}C^\dagger)CFC^\dagger$. And the distribution of δ on RPF does not change. The property holds.

2. Let's show that it holds after a measurement on the qubit q. The measurement is either deterministic or random on both the perfect simulation and PauliFrame simulation. This statement requires a proof.

   Lemma 6.1.3. *If the following measurement (meaning the measurement that will be applied on the state right after time t) is deterministic in perfect simulation, then it will be deterministic in PauliFrame simulation as well (given an instance of $PFMP_t$, we will get the same result for any RPF that produces $PFMP_t$). If it is random in perfect simulation, it will* *be random in PauliFrame simulation as well and have a 50% chance of outputting 1 (meaning that for half of all RPFs that produce $PFMP_t$ we'd get 1 and for other half we'd get 0).*

Proof. We know that the measurement is only deterministic in a perfect simulation if Z_q commutes with all the stabilizers of a state $|\psi_t\rangle$. But in this case, all Pauli Frames from $P_{\rho,t}$ will have the same X bit on the qubit q(0 if Z or I, 1 if X or Y) since $P_\delta = \prod_{0<i\leq n}(S_i^{\delta_i})F$, and all S_i do not have an X multiplier on the qubit q(they are either I or Z). That means that we will always get the same outcome on the measurement(no matter the Pauli Frame) and this measurement can be considered deterministic. Same way we show, that if this measurement is random in a perfect system then it is random in PauliFrame simulation as well. Z_q must anti-commute with at least one stabilizer. Let's say that this stabilizer is S_l. Then we can understand that $\prod_{0<i\leq n}(S_i^{\delta_i})F$ and $S_l(\prod_{0<i\leq n}(S_i^{\delta_i}))F$ are two Pauli Frames from $P_{\rho,t}$ and they will produce different outcomes and they occur with the same probability. This way we showed a one-to-one correspondence between Pauli Frames which produce output 1 and 0. That means that the probability of getting 1 would be 50%. □

Now back to the proof. If the measurement is deterministic that means that all generators commute with Z_q operator. It also means that the state will not be changed as the result of the operation. It also means that either Z_q or $-Z_q$ is already inside the S. That means that multiplying Z_q term into our Pauli Frames $P_{\rho,t}$ will not change the Pauli Frames set and it's probabilistic distribution (the $PFMP_t$ got a deterministic bit and thus the $RPFs$ that generate our $PFMP_t$ are the same as those that generate $PFMP_{t+1}$). That means that theorem will still hold. Now what if the measurement is not deterministic, but random? That means that at least one of the generators anti commutes with Z_q. If it anti commutes with at least two (S_i, S_j) - we can change S_j for S_jS_i, which will commute with Z_q. We can do this operation until Z_q only anti-commutes with the first generator. Let's call new generators as $\langle G_1, \ldots, G_n\rangle$. We can do the same operations on generators inside our Pauli Frames set (meaning that we just substitute S_i for G_i in definition of $P_{\rho,t}$), because they all commute with each other and because $S_i^2 = I$ for any stabilizer. This way we get a new representation of a state as

$$\{\prod_{0<i\leq n}(G_i^{\delta_i})F, |\psi_t\rangle, S = \langle G_1, \ldots, G_n\rangle\}$$

, where Z_q only anti-commutes with G_1 without the loss of generality. After the measurement we have new stabilizers as $\langle Z_q, G_2, \ldots, G_n \rangle$ or $\langle -Z_q, G_2, \ldots, G_n \rangle$, depending on a specific bit of $RMeas$, generating last bit of $PSMP_{t+1}$. Let's see how the $P_{\rho,t}$ changes. The last bit in $PFMP_{t+1}$ is determined randomly, since we don't track the exact Pauli Frame that is carried out during the simulation. We only track $P_{\rho,t}$, and the probability distribution of it's Pauli Frames. Since the measurement is random - that means that $P_{\rho,t}$ contains some Pauli Frames which return 0 and some that will return 1. The final choosing of the actual outcome that occurs during our Pauli Frame simulation is left on RPFs. We have a set of RPFs that generate $PFMP_t$, and we know that half of them will produce 1, and half of them will produce 0. Choosing the outcome is random, because probability distribution on the RPFs is uniform. We choose a random value, which determines the measurement outcome, and thus narrows down the RPFs that generate $PFMP_{t+1}$. Since we got a specific value out of the measurement and the RPFs are narrowed down - that narrows down which Pauli Frames could be in the state to produce such specific result. More accurately - this value determines the value of δ_1, because it should be either 1 for all Pauli Frames or 0 for all Pauli Frames in $P_{\rho,t}$. Because otherwise we could have gotten a different result from Pauli Frame simulation (because the term in G_1 on the q'th qubit is either X or Y). After the determined value of the measurement outcome, our set of Pauli Frames is either $\prod_{1<i\leq n}(G_i^{\delta_i})F$ or $\prod_{1<i\leq n}(G_i^{\delta_i})G_1F$. After the multiplication of Z-error into Pauli Frame with probability 50% we have the same corresponding set of narrowed down RPFs and a new P_ρ as

$$P_{\rho,t+1} = \{(Z_q^{\delta_{new}}) \prod_{1<i\leq n} (G_i^{\delta_i})G_1F\}$$

in the case of $\delta_1 = 1$ and

$$P_{\rho,t+1} = \{(Z_q^{\delta_{new}}) \prod_{1<i\leq n} (G_i^{\delta_i})F\}$$

in the case of $\delta_1 = 0$. The bit δ_{new} is determined by the next bit from the corresponding RPF. Bits in the RPF have 50% chance of spitting out 0. That means that we can say that a new δ' with uniform distribution can be formed as $\delta_i' = \delta_i | \forall i > 0$ and $\delta_1' = \delta_{new}$. And in both cases $(\delta_1 = 0, \delta_1 = 1)$ we can write P_ρ as

$$P_{\rho,t+1} = \prod_{0<i\leq n} (G_i'^{\delta_i})F'$$

as G_i from 1 to n are the same and $G'_1 = Z_q$ or $G'_1 = -Z_q$ which are equivalent since we will be multiplying it into the Pauli Frame. And the theorem statement holds.

3. We can think of reset operation as a measurement in Z basis and a conditional application of X or not application of X. Just as in the item above, let's see the stabilizers of the new state. After the measurement they will be either $\langle Z_q, G_2, \ldots, G_n \rangle$ or $\langle -Z_q, G_2, \ldots, G_n \rangle$. And after the conditional application of X they will be $\langle Z_q, G_2, \ldots, G_n \rangle$ always. Let's look at what happens with the Pauli Frame set P_ρ. Depending on the measurement value we get either $\prod_{1<i\leq n}(G_i^{\delta_i})F$ or $\prod_{1<i\leq n}(G_i^{\delta_i})G_1F$. And in one of the cases we conjugate the frames with X_q getting either

$$X_q(\prod_{1<i\leq n}(G_i^{\delta_i})F)X_q; \prod_{1<i\leq n}(G_i^{\delta_i})G_1F$$

or

$$\prod_{1<i\leq n}(G_i^{\delta_i})F; X_q(\prod_{1<i\leq n}(G_i^{\delta_i})G_1F)X_q$$

For the different cases of measurement result values. After that in any of those cases we multiply Z_q into the Pauli Frame with probability 50% getting either

$$(Z_q^{\delta_{new}})X_q(\prod_{1<i\leq n}(G_i^{\delta_i})F)X_q; (Z_q^{\delta_{new}})(\prod_{1<i\leq n}(G_i^{\delta_i})G_1F)$$

or

$$(Z_q^{\delta_{new}})(\prod_{1<i\leq n}(G_i^{\delta_i})F); (Z_q^{\delta_{new}})X_q(\prod_{1<i\leq n}(G_i^{\delta_i})G_1F)X_q$$

Now we can rewrite them as

$$-X_q((Z_q^{\delta_{new}})\prod_{1<i\leq n}(G_i^{\delta_i})F)X_q; ((Z_q^{\delta_{new}})\prod_{1<i\leq n}(G_i^{\delta_i})G_1F)$$

or

$$((Z_q^{\delta_{new}})\prod_{1<i\leq n}(G_i^{\delta_i})F); -X_q((Z_q^{\delta_{new}})\prod_{1<i\leq n}(G_i^{\delta_i})G_1F)X_q$$

What is left is to notice that all of them can be represented as

$$P_{\rho,t+1} = \prod_{0<i\leq n}(G_i'^{\delta_i})F'$$

where G_i from 1 to n are the same and $G'_1 = Z_q$, and F' is some Pauli Frame. And as such the main statement holds.

4. Since noise is Clifford, for the noise operation E the proof is easy.

New stabilizers are $\langle ES_1E^\dagger, \ldots, ES_nE^\dagger \rangle$, and new Pauli Frames are $EP = \prod_{0<i\leq n}(ES_i^{\delta_i}E^\dagger)EF$. The property holds.

□

Let's notice that in the process we also proved a very convenient and strong fact. The fact is that Pauli Frame simulation's measurements are also always either deterministic or random. Meaning that for a given measurement, either for all $PFMP_t$s half of the RPFs that produce that $PFMP_t$ give 0 on the measurement, or for all $PFMP_t$s all RPFs that produce that $PFMP_t$ will give the same outcome on the measurement under consideration. Moreover, if a measurement is always deterministic in Pauli Frame simulation - then it is also always deterministic in the perfect simulation. And if a measurement is always random in Pauli Frame simulation - then it is also always random in the perfect simulation.

Now let's show the next statement. Suppose we have a quantum computation running at time t with all three systems simultaneously (reference frame, Pauli Frame, perfect simulation). Let's note $|\psi_{r,t}\rangle$ as a state of the system from the reference run at time t, and P_t as a Pauli Frame of our algorithm simulation running at a time t, and $|\psi_t\rangle$ as a state of the perfect simulation system at a time t.

Theorem 6.1.4. *Suppose we have a circuit $Circ$. Suppose we are running three simulation on that circuit simultaneously. Suppose the circuit outputs k bits. Then the statements hold:*

- *For any vector of bits v of length k - the fraction of $RPFs$ that give $PFMP = v$ is equal to the fraction of $RMeas$ that give $PSMP = v$*
- *If the probability of vector v occurring on the outcomes is not 0 and it so happened that we got $PSMP = PFMP$ - then the similarity holds:*

$$|\psi_{t_l}\rangle \sim P_{t_l}\,|\psi_{r,t_l}\rangle$$

Here t_l is the last instance of time (after all circuit operations have been done). The sign of similarity here means that the states are equal up to a complex multiplier. Also note, that here the multiplication is a matrix multiplication.

Proof. We will prove this statement by induction on the size of the circuit $Circ$. For the zero size circuit the statement is obviously true. Induction's assumption: the statement is correct for any circuit of the size w. Let's prove that the statement will be correct for any circuit of size $w + 1$.

Let's look at the last instruction of the circuit of size $w+1$. We know that the statements hold for the circuit without this instruction. Now let's check that properties hold after the application of any Clifford operation, measurement, reset or noise operation. In this discussion t_l is the final instance of time for the circuit of the size w, and t_l+1 is the last instance of time for the circuit of size $w+1$.

1. Let's show that the first statement holds for the circuit of size $w+1$. It follows from the induction's assumption immediately as we realize that the number of measurements is the same and since it does not change with the last operation - the distribution of measurement patterns will stay the same. Now Let's show that the similarity will hold after any Clifford operation C for any possible vector of measurement pattern. $|\psi_{t_l}\rangle$ changes to $|\psi_{t_l+1}\rangle = C|\psi_{t_l}\rangle$ and the reference state changes to $|\psi_{r,t_l+1}\rangle = C|\psi_{r,t_l}\rangle$ and the Pauli Frame changes to $P_{t_l+1} = CP_{t_l}C^\dagger$. All that is left is to remember that $CC^\dagger = I$, because

$$P_{t_l+1}|\psi_{r,t_l+1}\rangle = CP_{t_l}C^\dagger C|\psi_{r,t_l}\rangle = CP_{t_l}|\psi_{r,t_l}\rangle = C|\psi'_{t_l}\rangle = |\psi'_{t_l+1}\rangle$$

   for the same δ'. Here we used the induction's assumption.

2. Now let's show that the statements hold if the last operation of the circuit is Z_q. We will start with showing that the probability distribution on the measurement patterns will be the same.

   We already have induction's assumption that before the last measurement probability distributions of measurement patterns occurring will be the same for Pauli Frame simulation and for the perfect simulation. Then it is sufficient to show, that for any case when $PFMP_{t_l} = PSMP_{t_l}$ the probability of getting the same outcome will be the same. But for any case when $PFMP_{t_l} = PSMP_{t_l}$, we also have a similarity holding. As it goes, we know that the last measurement is either deterministic or random in Pauli Frame simulation and in perfect simulation. If it is indeed random, for both simulations we have a 50% chance of obtaining 0 and 50% chance of obtaining 1. That means that the probability distribution on the measurement patterns will stay the same. However, if the measurement is deterministic, then we want to use the similarity to prove that the result of the Pauli Frame will be the same as for the perfect simulation (for a specific $PSMP_{t_l} = PFMP_{t_l}$). Seeing as

$$|\psi_{t_l}\rangle \sim P_{t_l}|\psi_{r,t_l}\rangle$$

We know, that if we were to measure deterministically Z_q, then both the $|\psi_{t_l}\rangle$ and $P_{t_l}|\psi_{r,t_l}\rangle$ would give the same result. And $P_{t_l}|\psi_{r,t_l}\rangle$ would give exactly the outcome from the Pauli Frame algorithm. Because: suppose $|\psi_{r,t_l}\rangle$ has stabilizers $\langle A_1, \dots, A_n\rangle$. Then $P_{t_l}|\psi_{r,t_l}\rangle$ has stabilizers $\langle P_{t_l}A_1P_{t_l}, \dots, P_{t_l}A_nP_{t_l}\rangle$. We know that $P_{t_l}A_iP_{t_l}$ has I or Z on the qth place. We also know that $P_{t_l}A_iP_{t_l} = cA_i$ for a complex factor c. That means that A_i has I or Z on the qth place. That means the $|\psi_{r,t_l}\rangle$ will also produce a deterministic result on the Z_q measurement. Moreover, if P_{t_l} had I or Z on the qth place - the outcome of the measurement on the state $|\psi_{r,t_l}\rangle$ will be the same as on the state $P_{t_l}|\psi_{r,t_l}\rangle$. And conversely, if P_{t_l} had X or Y on the qth place - the outcome of the measurement on the state $|\psi_{r,t_l}\rangle$ will be different from one on the state $P_{t_l}|\psi_{r,t_l}\rangle$. That way we proved that probability distribution on the measurement outcomes will be the same for perfect simulation and for Pauli Frame algorithm.

Let's prove the similarity now for the case when $PFMP_{t_l+1} = PSMP_{t_l+1}$. If the measurement Z_q is deterministic - $|\psi_{t_l}\rangle$ does not change (up to a global phase $|\psi_{t_l}\rangle \sim |\psi_{t_l+1}\rangle$). That means that $P_{t_l}|\psi_{r,t_l}\rangle$ does not change as well. We've already shown that measurement Z_q will also produce a deterministic outcome on the state $|\psi_{r,t_l}\rangle$. That means that the state $|\psi_{r,t_l}\rangle$ does not change as well (up to a global phase $|\psi_{r,t_l}\rangle \sim |\psi_{r,t_l+1}\rangle$). Lastly we know that Pauli Frame P_{t_l} might not change or we might multiply Z_q into it. But it's obvious that the similarity holds in both cases

$$|\psi_{t_l+1}\rangle \sim P_{t_l}|\psi_{r,t_l+1}\rangle\,;|\psi_{t_l+1}\rangle \sim Z_qP_{t_l}|\psi_{r,t_l+1}\rangle$$

since $Z_qP_{t_l}|\psi_{r,t_l+1}\rangle \sim P_{t_l}|\psi_{r,t_l+1}\rangle$, because the state $P_{t_l}|\psi_{r,t_l+1}\rangle$ already contains either Z_q or $-Z_q$.

Now let's show for the random measurement case. We will prove that the similarity holds for the case when $PFMP_{t_l+1} = PSMP_{t_l+1}$. Since the measurement is random, there are two cases which we have to look into. First case is when both simulations output 0 on the last measurement, and when they both output 1. The probabilities of obtaining either one of these cases is 50%. Let's call m_a - result of the measurement of the perfect simulation, m_r - result of the measurement on the reference system and x_q - X bit in Pauli Frame P_{t_l}. There are four possibilities for these values: $(m_a = 1, m_r = 0, x_q = 1), (m_a = 1, m_r = 1, x_q = 0), (m_a = 0, m_r = 0, x_q = 0), (m_a = 0, m_r = 1, x_q = 1)$. Let's look at the first case. In this case the state of the perfect simulation after the measurement will be $|\psi_{t_l+1}\rangle = \sqrt{2}\,|1\rangle\langle 1|\,|\psi_{t_l}\rangle$ (these one qubit states of course act on the same qubit q as the measurement. The projector can be interpreted as projecting that qubit's subspace onto a $|1\rangle\langle 1|$ and acting as Identity on the rest of the space). And the new reference state will be $|\psi_{r,t_l+1}\rangle = \sqrt{2}\,|0\rangle\langle 0|\,|\psi_{r,t_l}\rangle$. And the equation will still hold (note that $x_q = 1$ means that $|1\rangle\langle 1|\,P_{t_l} = P_{t_l}\,|0\rangle\langle 0|$):

$$|\psi_{t_l+1}\rangle = \sqrt{2}\,|1\rangle\langle 1|\,|\psi_{t_l}\rangle \sim \sqrt{2}\,|1\rangle\langle 1|\,P_{t_l}\,|\psi_{r,t_l}\rangle = P_{t_l}\sqrt{2}\,|0\rangle\langle 0|\,|\psi_{r,t_l}\rangle = P_{t_l}\,|\psi_{t,t_l+1}\rangle$$

If we don't multiply Z_q into P_{t_l} and $P_{t_l+1} = P_{t_l}$, then we have showed the required similarity. However if we do multiply Z_q into P_{t_l} and $P_{t_l+1} = Z_q P_{t_l}$, then we know that $P_{t_l}\,|\psi_{t,t_l+1}\rangle = Z_q P_{t_l+1}\,|\psi_{t,t_l+1}\rangle \sim P_{t_l+1} Z_q\,|\psi_{t,t_l+1}\rangle = P_{t_l+1}\,|\psi_{t,t_l+1}\rangle$. And that way we have showed that the similarity holds for the case $(m_a = 1, m_r = 0, x_q = 1)$. For other cases proofs are the same, except when $x_q = 0$ then $|0\rangle\langle 0|\,P = P\,|0\rangle\langle 0|$ and $|1\rangle\langle 1|\,P = P\,|1\rangle\langle 1|$. Now we have showed that the similarity holds in all the cases.

3. Let's prove both statements when the last operation is reset. We can think of a reset operation as a measurement and a conditional X gate applied sequentially on the same qubit. We have already shown that after such a measurement the probability distribution on the measurement pattern will be the same for Pauli Frame and perfect simulations. We have showed the similarity as well. Let's look at what happens after the conditional X gate. First let's show the first statement. It's obvious that after this conditional gate the probability distribution on the measurement pattern will still be the same in both simulations, since conditional X gate does not produce any outcomes. Now let's show that the similarity will still hold on the same measurement pattern. Since the measurement pattern is the same - that means that the last measurement outputted the same result and we either apply X gate in both simulations or we don't in both simulations. For both cases it's trivial to show that the similarity still holds.

4. For noise operation E it is trivial, because deterministic noise is Clifford. And we don't apply E on the reference frame and instead of conjugating Pauli Frame with this operation we just multiply it into the Frame:

$$|\psi_{t_l+1}\rangle = E\,|\psi_{t_l}\rangle \sim EP_{t_l}\,|\psi_{r,t_l}\rangle = P_{t_l+1}\,|\psi_{r,t_l+1}\rangle$$

□

This way we have proved that it is with the same probability that we get a specific measurement pattern on our Pauli Frame simulation and perfect simulation.

7
Simulation results and a Threshold plot

With the Pauli Frame algorithm we can now simulate Clifford circuits with probabilistic Pauli noise with a good complexity of $O(1)$ per operation (further complexity considerations are described in [Gidney 2021]). Fusion network described in the previous chapters lies inside of this computational model. Creation of the resource states can be described just by Clifford operations, since we are not considering any errors during this process. Application of single qubit measurements and fusions can also be described inside our model. To do a single qubit measurement in any Pauli basis we first apply an appropriate Pauli transformation, then measure in a Z basis, then apply the transformation again. To apply a fusion - we can use an ancilla qubit. Ancilla qubit is an additional qubit that is used only to execute a specific measurement. For example they are often used in the theory of surface codes to execute stabilizer measurements. In reality they may have an actual physical representation (for example in superconducting architectures). But in our discussion of fusions in photonic architectures such qubits do not exist and fusion operation is executed without this notion. However for our simulation this notion is necessary, because Pauli Frame algorithm does not support multiple qubit measurements by default.

Theorem 7.0.1. *Circuit represented in the Fig.7.1 produces the fusion operation on the target qubits i and j.*

Proof. First let's notice that the first part of the circuit implements Z_iZ_j measurement. Using the theorem 2.2.3 we can move the first measurement before both CX gates. The measurement will then become CZ_aC^{-1}, where C is a Clifford gate composed of both CXs, and Z_a is a Z operator on an ancilla qubit.

$$M_1 = CZ_aC^{-1} = CX_{i,a}CX_{j,a}Z_aCX_{j,a}CX_{i,a} = Z_iZ_jZ_a$$

Since the ancilla qubit is initialized in the $|0\rangle$ state and is unentangled from the rest of the system, the measurement $Z_iZ_jZ_a$ will be equivalent to Z_iZ_j. Since ancilla qubit is unentangled from the rest of the system right before the reset - it will still be unentangled from the rest of the system right before the reset in our modified scheme. Therefore, reset does not collapse the state. Now let's notice the same behavior in the second part of the circuit. We move the measurement to the start of the second part.

$$M_2 = CZ_aC^{-1} = H_aCX_{a,i}CX_{a,j}H_aZ_aH_aCX_{a,j}CX_{a,i}H_a = X_iX_jZ_a$$

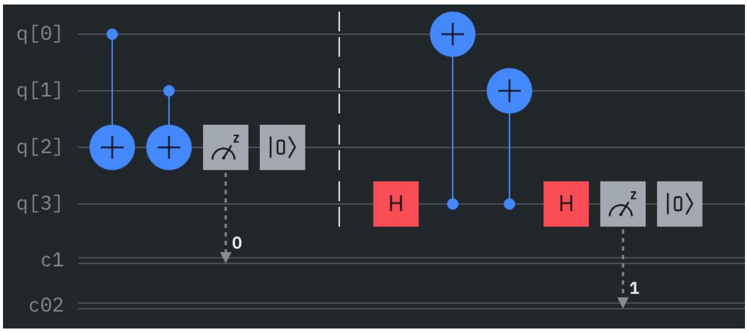


Figure 7.1: This is a fusion circuit. Qubits q[0] and q[1] represent target qubits of the full system with indices i and j. q[2], q[3] are ancilla qubits. c1 and c02 are classical registers. The circuit is two CNOTs, followed by a measurement in a Z basis, followed by a reset. Then the same scheme is applied, only CNOTs are conjugated with H. First part of the circuit (before the tick) implements a Z_iZ_j measurement. And the second part implements X_iX_j measurement.

But since the ancilla qubit is unentangled and in the $|0\rangle$ state - the measurement is equivalent to X_iX_j. Now, for our convenience, let's move the second measurement to the start of the circuit (right after the first measurement and just before the first CX)

$$CX_{a,0}CX_{a,1}X_0X_1CX_{a,1}CX_{a,0} = X_0X_1$$

And thus it's the same measurement. Now we have transformed our circuit such that first there are two measurements Z_0Z_1 and X_0X_1 and then there are all the other operations. But, from the proposition 3.3.2 we know that both target qubits will already be unentangled from the system and ancilla qubit will stay unentangled as well. That means that all future operations on those qubits are irrelevant for the rest of the computation and we have successfully implemented fusion operation in the Clifford circuit instructions notation. □

Now it is easy to understand the process of running the simulation. The process of decoding is also straightforward and have been described in the previous chapter. Following are the figures with the results of these simulations.

What Is Simulated

In the simulations reported here we instantiate the six-ring fusion network as a single logical memory block and evolve it under the stabilizer-only model. Concretely:

- Network geometry: a layered six-ring FBQC architecture with non-periodic boundaries. We simulate three linear sizes (3, 5, 7), corresponding to $3 \times 3 \times 3$, $5 \times 5 \times 5$, $7 \times 7 \times 7$, to understand the scaling of the logical error rate on a smaller scale.

- Circuit representation: each fusion is realized as a pair of commuting two-qubit parity measurements (XX, ZZ), which we implement in the Clifford circuit model via the ancilla construction discussed above (and compatible with the Pauli-frame formalism of Chapter 6). Resource-state preparation is assumed ideal, as argued in Chapter 5 (Computational Framework).
- Error application: we use the fusion model of Chapter 5 (Error Model for the Fusion Process). By a *fusion-outcome flip* we mean a classical bit flip applied to the reported outcome of a *successful* fusion. We model independent flips on the two parity outcomes, with probabilities p_{XX} for the XX outcome and p_{ZZ} for the ZZ outcome. Other fusion-related terms (e.g., failure/erasure) are held zero in the sweeps below.
- Decoding and success criterion: classical outcomes are combined into checks (Chapter 5, Decoding process). A correction is inferred purely classically (Pauli-frame update) without acting back on the photonic system. Decoding succeeds if the corrected outcomes are consistent with the logical correlation surfaces through the block; the logical error rate (LER) is the empirical failure probability over repeated trials.
- Sampling and scaling: for each size and each value of the swept parameter we perform Monte Carlo sampling to estimate the LER, then aggregate the curves across sizes and locate the crossing region. This produces the threshold plots shown below.

Objective and Rationale

Our goal in these plots is to identify, for fixed p_{XX}, the range of p_{ZZ} where increasing the code size ($3 \to 5 \to 7$) improves the logical error rate (below threshold) versus where it worsens (above threshold). We therefore sweep p_{ZZ} while holding p_{XX} constant.

The choice to vary p_{ZZ} with p_{XX} fixed is motivated by architectural asymmetries: in some photonic FBQC realizations the effective noise on the two parity outcomes is not symmetric, and system design can often trade a lower error on one parity for a higher error on the other. Studying one axis at a time (here, p_{ZZ}) therefore provides understanding of the behavior of the system.

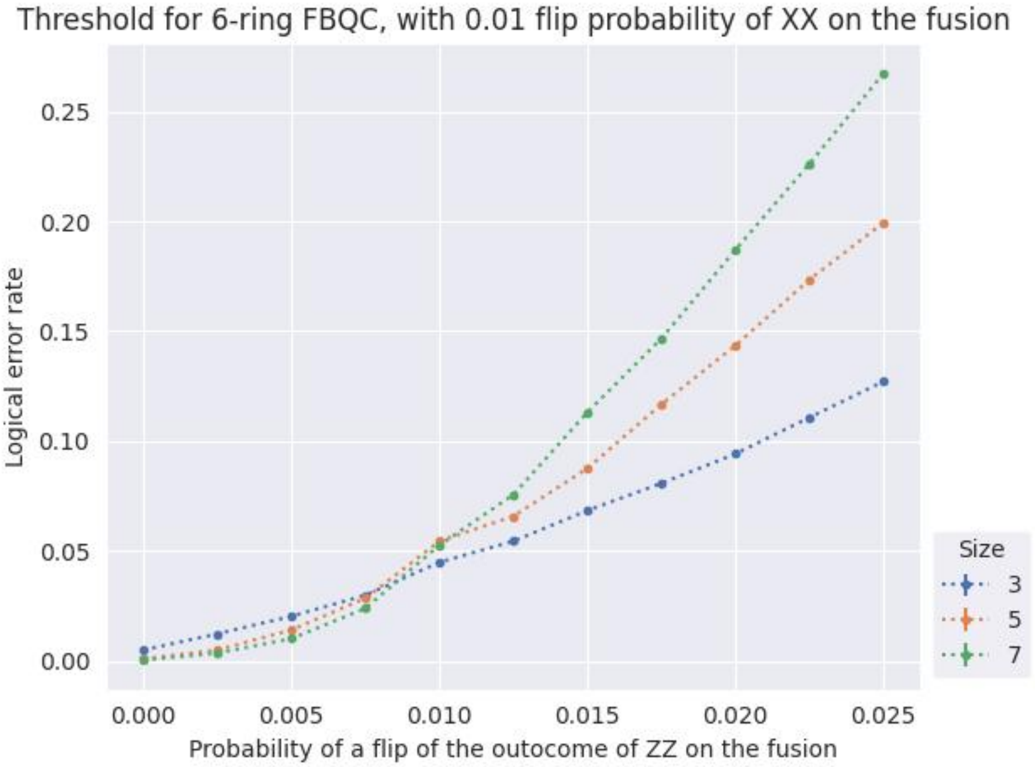


Figure 7.2: Threshold plot for fusion networks of sizes 3 × 3 × 3, 5 × 5 × 5, and 7 × 7 × 7 under an asymmetric outcome-flip channel. Here we sweep p_{ZZ} and hold $p_{XX} = 0.01$ fixed. The curves start near zero and exhibit different slopes; the finite-size crossing indicates the threshold region.

We did not perform a formal asymptotic or analytic threshold derivation for this specific asymmetric channel. Qualitatively, percent-level crossing points are consistent with memory-like, circuit-level noise models reported for surface-code-style systems. The observed crossing region near $p_{ZZ} \approx 0.008-0.010$ at $p_{XX} = 0.01$ is therefore in line with expectations; a tighter estimate would require larger sizes and/or a dedicated heuristic analysis, which we leave for future work.

Results and Interpretation

The three size curves exhibit a single, well-defined crossing when sweeping p_{ZZ} at fixed $p_{XX} = 0.01$.

- Crossing location: the intersection lies in $p_{ZZ} \approx 0.008-0.010$. Finite-size drift is mild between sizes 3, 5, and 7, suggesting that the asymptotic threshold for this asymmetric channel is close to this interval.
- Ordering below/above crossing: for p_{ZZ} below the crossing, the LER decreases monotonically with size (evidence of effective error suppres-

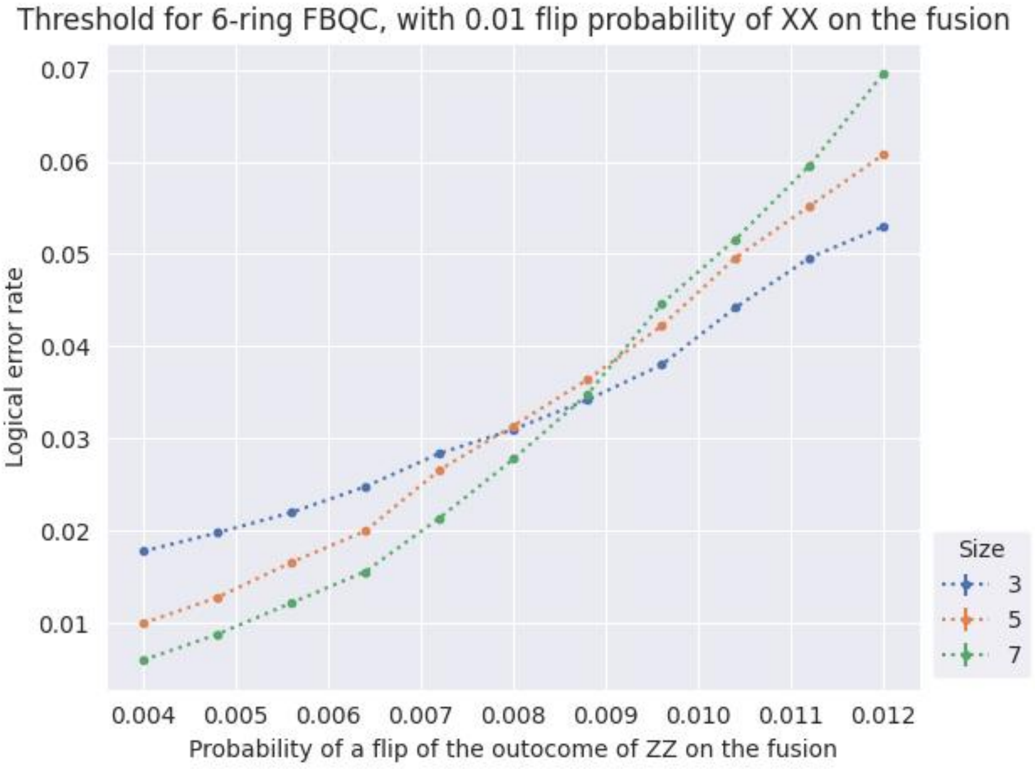


Figure 7.3: Same data as Figure 7.2, highlighting the crossing region. With $p_{XX} = 0.01$ fixed and p_{ZZ} swept, the three curves cross in the interval $p_{ZZ} \in [0.007, 0.010]$. Below this range the logical error decreases with size $(3 \to 5 \to 7)$, and above it the ordering reverses.

sion); above the crossing, the ordering reverses, indicating we are beyond threshold.

- Slopes and curvature: the larger sizes show a steeper dependence on p_{ZZ}, consistent with increasing code distance amplifying the distinction between below- and above-threshold behavior.
- Asymmetry note: because p_{XX} is held at 0.01, the reported threshold is conditional on that choice. Different fixed p_{XX} values would move the crossing along the p_{ZZ} axis, as expected under an asymmetric noise trade-off.

Reproducibility and Setup

The results in Figures 7.2 and 7.3 were produced with my local installation of the *plaquette* tooling [QC Design GmbH 2025], using a small, private Python driver script (no code is included here). The workflow generates six-ring fusion-based codes of varying size, sweeps a single fusion-error parameter, estimates logical error rates by sampling, and renders threshold plots.

Simulation configuration used for the plots:

- Fusion-network geometry: six-ring FBQC, non-periodic boundaries;
- System sizes: 3, 5, and 7 (standing for $3 \times 3 \times 3$; $5 \times 5 \times 5$; $7 \times 7 \times 7$).
- Parameter sweep: vary the ZZ fusion-outcome flip probability from 0.0 to 0.025 in 11 evenly spaced points; hold the XX fusion-outcome flip probability fixed at $p_{XX} = 0.01$.
- Sampling: 5000 Monte Carlo samples per sweep point.
- Deterministic seed set to ensure run-to-run reproducibility.
- Decoder used is a graph-based MWPM decoder consistent with the check construction in Chapter 5 (Decoding process).

Environment and hardware:

- Operating system: macOS 15.6.1.
- CPU: 11 logical cores available to the process.

Runtime: the total wall-clock time scales linearly with the number of sweep points and samples per point; the configuration above (three sizes, 11 sweep points, 5000 samples per point) was used to produce the figures shown in about 300 seconds.

8
Conclusion and future work

Although the key pieces are now in place, moving from theoretical evidence to a laboratory demonstration and ultimately to a scalable architecture opens several promising research directions:

1. **More realistic noise and hardware feedback.** We have only looked at one error model that was the easiest to simulate. Hardware imperfections can be approximated with better accuracy using more sophisticated models. They might not be simulated as efficiently, but would give more accurate logical error rates. We are working hard in QC Design to bring a library that can handle diverse, cutting edge error models with the best speed possible.

2. **Larger codes and better decoders.** The distance-d surface codes we embedded are not unique. Hybrid X/Z-boundary tilings and XZZX variants could raise the threshold or reduce the number of rings per logical qubit. We used minimum-weight perfect matching with uniform edge weights. Belief-propagation or neural decoders that exploit the full fusion syndrome graph may suppress logical error further, postponing the need for larger lattices. Systematic finite-size scaling up to $d = 11$ and many-layer memories will pin down the asymptotic threshold more accurately and reveal whether the 3-D fusion lattice obeys the same universality class as conventional surface codes.

3. **Beyond the identity gate.** A memory is only a first step. By modifying boundary conditions between layers one can braid logical defects and implement a universal Clifford set entirely in the fusion picture. Extending our stabilizer proofs to these braids will quantify the gate-level overhead.

   State-injection protocols (e.g. T-state or Y injection) remain to be re-optimised for photonic FBQC. Analysing their distillation cost under the same noise assumptions is necessary before claiming a full universal computer.

   Fault-tolerant read-write interfaces that swap a matter qubit (ion, spin, Rydberg atom) into and out of the photonic memory would connect the present work to heterogeneous architectures.

4. **Resource-state engineering.** Our 6-ring is only one possibility. Larger but cheaper-to-make graph states (e.g. doped cluster “snowflakes”) might trade extra qubits for reduced fusion depth. An automated search over small graphs combined with our simulator can identify optimal candidates. Investigating time-multiplexed generation—reusing a single physical source in a loop to emit the six photons sequentially—may drastically lower the footprint per ring and relax synchronization demands.

Demonstrating a fault-tolerant quantum memory is the acid test for scalable quantum information. The path traced in this thesis—from abstract stabilizer proofs to a concrete photonic layout and a realistic error budget—brings that milestone tangibly closer. While substantial engineering remains, the theoretical and numerical groundwork laid here provides a clear quantitative target and a flexible simulation toolbox for the community to refine, extend and ultimately beat.

9
Bibliography

[Aaronson e Gottesman 2008]AARONSON, S.; GOTTESMAN, D. *Improved Simulation of Stabilizer Circuits*. 2008. Disponível em: <https://arxiv.org/abs/quant-ph/0406196>.

[Aharonov e Ben-Or 2008]AHARONOV, D.; BEN-OR, M. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 2008.

[Bartolucci et al. 2023]BARTOLUCCI, S. et al. Fusion-based quantum computation. *Nature Communications*, 2023.

[Bravyi e Kitaev 2005]BRAVYI, S.; KITAEV, A. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A*, 2005.

[Browne e Rudolph 2005]BROWNE, D. E.; RUDOLPH, T. Resource-efficient linear optical quantum computation. *Physical Review Letters*, APS, v. 95, n. 1, p. 010501, 2005.

[Fowler et al. 2012]FOWLER, A. G. et al. *Surface codes: Towards practical large-scale quantum computation*. 2012. Disponível em: <https://arxiv.org/abs/1208.0928>.

[Gidney 2021]GIDNEY, C. Stim: a fast stabilizer circuit simulator. *Quantum*, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 5, p. 497, jul 2021. ISSN 2521-327X. Disponível em: <https://doi.org/10.22331/q-2021-07-06-497>.

[Gottesman 1998]GOTTESMAN, D. Theory of fault-tolerant quantum computation. *Physical Review A*, 1998.

[Horsman et al. 2012]HORSMAN, D. et al. Surface code quantum computing by lattice surgery. *New Journal of Physics*, v. 14, p. 123011, 2012.

[Knill, Laflamme e Milburn 2001]KNILL, E.; LAFLAMME, R.; MILBURN, G. J. A scheme for efficient quantum computation with linear optics. *Nature*, 2001.

[Knill, Laflamme e Zurek 1998]KNILL, E.; LAFLAMME, R.; ZUREK, W. H. Resilient quantum computation: error models and thresholds. *Proceedings of the Royal Society A*, 1998.

[Kok et al. 2007]KOK, P. et al. Linear optical quantum computing with photonic qubits. *Reviews of Modern Physics*, 2007.

[Li et al. 2022]LI, B. et al. Photonic resource state generation from a minimal number of quantum emitters. *npj Quantum Information*, 2022.

[Montanaro 2016]MONTANARO, A. Quantum algorithms: an overview. *npj Quantum Information*, 2016.

[Nielsen e Chuang 2010]NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information*. [S.l.]: Cambridge University Press, 2010.

[Paler et al. 2014]PALER, A. et al. *Software Pauli Tracking for Quantum Computation*. 2014. Disponível em: <https://arxiv.org/abs/1401.5872>.

[Pan et al. 2012]PAN, J.-W. et al. Multiphoton entanglement and interferometry. *Reviews of Modern Physics*, 2012.

[Preskill 2018]PRESKILL, J. Quantum computing in the nisq era and beyond. *Quantum*, 2018.

[QC Design GmbH 2025]QC Design GmbH. *plaquette: An all-encompassing quantum error-correction software package*. 2025. Software, version 2025.7.6.

[Shor 1997]SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 1997.