



TOWARDS PRACTICAL LINEAR OPTICAL QUANTUM COMPUTING

MERCEDES GIMENO-SEGOVIA

DEPARTMENT OF PHYSICS
IMPERIAL COLLEGE LONDON

Thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

November 2015

To my parents

ABSTRACT

Quantum computing promises a new paradigm of computation where information is processed in a way that has no classical analogue. There are a number of physical platforms conducive to quantum computation, each with a number of advantages and challenges. Single photons, manipulated using integrated linear optics, constitute a promising platform for universal quantum computation. Their low decoherence rates make them particularly favourable, however the inability to perform deterministic two-qubit gates and the issue of photon loss are challenges that need to be overcome.

In this thesis we explore the construction of a linear optical quantum computer based on the cluster state model. We identify the different necessary stages: state preparation, cluster state construction and implementation of quantum error correcting codes, and address the challenges that arise in each of these stages. For the state preparation, we propose a series of linear optical circuits for the generation of small entangled states, assessing their performance under different scenarios. For the cluster state construction, we introduce a ballistic scheme which not only consumes an order of magnitude fewer resources than previously proposed schemes, but also benefits from a natural loss tolerance. Based on this scheme, we propose a full architectural blueprint with fixed physical depth. We make investigations into the resource efficiency of this architecture and propose a new multiplexing scheme which optimises the use of resources. Finally, we study the integration of quantum error-correcting codes in the linear optical scheme proposed and suggest three ways in which the linear optical scheme can be made fault-tolerant.

DECLARATION OF ORIGINALITY

The work presented in this thesis is the result of the research performed during my graduate studies in collaboration with my supervisors and other researchers who are adequately acknowledged. The thesis has been written solely by me and it presents my own work. Results from the literature mentioned in the thesis are appropriately referenced. Parts of chapter 5 have already been published in

From Three-Photon Greenberger-Horne-Zeilinger States to Ballistic Universal Quantum Computation

Mercedes Gimeno-Segovia, Pete Shadbolt, Dan E. Browne, and Terry Rudolph
Phys. Rev. Lett. 115, 020502 (2015)

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

ACKNOWLEDGEMENTS

First I would like to thank my supervisors Dan Browne, Terry Rudolph and Jeremy O'Brien for giving me the opportunity to work in such an interesting project. It has been an extraordinary experience to learn from them, they have all contributed in different ways to my development as a scientist but together they have equipped me well. I would like to thank Dan for his patience and encouragement, our meetings were always equally instructive and inspiring. I would like to thank Terry for continually challenging me with new interesting problems that have taught me such different new things. Finally, I would like to thank Jeremy for his support and good advice.

I have been very lucky to collaborate with many researchers: Pete Shadbolt, to whom I would like to thank for always sharing his enthusiasm and expertise, Sophia Economou, James Auger, Hussain Anwar, Tom Stace, and many researchers from the Centre for Quantum Photonics in Bristol, in particular Gabriel Mendoza, Josh Silverstone, Jacques Carolan and Hugo Cable. My understanding of physics would not have been the same without our collaborations.

I feel indebted to the people who have read parts of this thesis, Terry Rudolph, Dan Browne, Pete Shadbolt, David Jennings and Mehdi Ahmadi. Their insightful comments have helped me incredibly. I would also like to thank Naomi Nickerson, Howard Dale, Catherine Hughes, James Auger and Hussain Anwar for useful discussions.

These years as a graduate student would not have been the same without all my colleagues and friends. The members of the Controlled Quantum Dynamics CDT at Imperial College (cohort 3 in particular), the quantum group at UCL and the CQD Theory group in EEE, in particular the attendants of the Terry-less group meetings, have provided an interesting and stimulating environment. I would also like to thank all the good friends I've made in London for all the experiences that we have shared, which have help me grow personally. In particular I would like to thank the members of Wilson House wardening team 2013-2014, Naiři Usher, Dasha Zugwitz, Darren Holland, Devin Dunseith, Celia Pacheco-Moreno and Aida Moreno-Moral.

Finally I would like to thank my family for all their love and support. To my father José María, because his interest in the world around him sparked my curiosity and thirst for knowledge. To my mother Susana, for being the best example of personal integrity and good work ethic. To my sister María, for showing me to look at the world in a different way. To Mehdi, above all, for being my best friend. For continually being supportive, loving and encouraging. For being an inspiration to become a better physicist and a better person.

Mercedes Gimeno-Segovia

“It’s hopeless”, Nicholas went on. “We no longer have the learning of the ancients, the age of giants is past!”

“We are dwarfs,” William admitted, “but dwarfs who stand on the shoulders of those giants, and small though we are, we sometimes manage to see farther on the horizon than they.”

The name of the rose
UMBERTO ECO

CONTENTS

Abstract	5
Declaration	8
Acknowledgements	9
List of Figures	17
Nomenclature	19
1 Introduction	23
1.1 Introduction	23
1.2 Quantum Computing	24
1.2.1 Quantum algorithms and computational speedup	26
1.2.2 Circuit model	27
1.2.3 Measurement-based quantum computation	28
1.2.4 Adiabatic model	31
1.3 Physical requirements of a Quantum Computer	32
1.3.1 Real-world constraints	33
1.3.2 Physical implementations for quantum computers	33
1.4 Thesis outline	35
2 Linear Optical Quantum Computing	37
2.1 Introduction	37
2.2 Linear Optics	37
2.2.1 Single photons as information carriers	38
2.2.2 Implementation of any $U(N)$ in an optical multiport: Reck et al. scheme	42
2.2.3 Creating entanglement in Linear Optics	43
2.2.4 Computational complexity of linear optics	43
2.3 Integrated optics	45
2.4 First proposals for Linear Optical Quantum Computing	47
2.4.1 The Knill, Laflamme & Milburn protocol	48
2.4.2 The Yoran & Reznik protocol	50
2.5 Optical Quantum Computation with Cluster States	51
2.5.1 The Nielsen protocol	52
2.5.2 The Browne & Rudolph protocol	53
2.6 Parity-encoded optical quantum computing	55
2.6.1 The Hayes, Gilchrist, Myers & Ralph protocol	55
2.6.2 The Gilchrist, Hayes & Ralph protocol	56
2.7 Percolation-based Linear Optical Quantum Computing	57
2.7.1 Percolation theory	58
2.7.2 The Kieling, Rudolph & Eisert protocol	60

2.8	Discussion	64
3	Simulation of stabilizer computations	65
3.1	Introduction	65
3.2	Stabilizer Formalism	66
3.2.1	Gottesmann-Knill Theorem	69
3.2.2	Error correction with Stabilizers	69
3.2.3	Stabilizer circuits	70
3.2.4	Graph States	71
3.3	Simulation	72
3.3.1	Binary representation of a stabilizer code	73
3.3.2	Reduction to graph states	76
3.3.3	Local Clifford equivalence	77
3.3.4	Efficiency of the classical simulation	78
3.3.5	Improved simulation using Destabilizers	79
3.3.6	Performance comparison	81
3.4	Algorithms for the visualisation of stabilizer codes	82
3.5	Discussion and outlook	88
4	Generating photonic states	89
4.1	Introduction	89
4.2	Conventions	90
4.3	Photon sources and entanglement operations	91
4.3.1	Spontaneous parametric down-conversion sources	91
4.3.2	Bell measurements in Linear Optics	92
4.3.3	Fusion gates	96
4.4	Ballistic circuits for generation of small entangled states	100
4.4.1	Bell pair generation from single photons	100
4.4.2	3-GHZ states from single photons	102
4.4.3	Generation of larger GHZ states from single photons	103
4.4.4	Adaptation to use Bell pairs	104
4.4.5	Giving up loss tolerance for higher success probability	106
4.5	Removing stochasticity by multiplexing	107
4.5.1	Log-tree scheme	108
4.5.2	Cost of near-deterministic generation of GHZ states using a multiplexed scheme	109
4.5.3	Using probabilistic Bell pairs to generate GHZ states	111
4.6	Rotated Type-II	113
4.7	Discussion and outlook	115
5	A percolation-based scheme for linear optical quantum computing	119
5.1	Introduction	119
5.2	Boosted fusion mechanisms in the context of percolation	120
5.3	Building the percolated lattice	121
5.4	Percolation properties	124
5.4.1	Calculating the percolation threshold	125
5.5	A single qubit channel	128
5.6	Loss tolerance	129
5.7	Scaling of resources	130
5.8	Comparison with previous percolation schemes	132
5.9	QNIX: a blueprint for linear optical quantum computing	134
5.9.1	Active switching only in state preparation	135

5.9.2	A dynamical architecture with fixed physical depth	138
5.10	Discussion and outlook	140
6	Improving resource efficiency	141
6.1	Introduction	141
6.2	Relative Multiplexing	141
6.2.1	RMUX in a percolated lattice	143
6.2.2	Results	146
6.3	Effective use of resources generated in multiplexing	149
6.3.1	Surplus of entangled states	149
6.3.2	Impact of an efficient use of the generated resources	151
6.4	Discussion and outlook	152
7	Towards fault-tolerance	153
7.1	Introduction	153
7.2	Quantum Error Correction and topological codes	154
7.2.1	Foundations of the theory of quantum error correction	154
7.2.2	Error model	155
7.2.3	Topological codes	157
7.3	First linear optical proposal of a fault-tolerant quantum computer	162
7.4	Transforming a percolated lattice into a topological code	163
7.4.1	Renormalisation	164
7.4.2	Concentrating a universal lattice	166
7.4.3	Percolated Raussendorf lattice	167
7.5	Topological codes under a bond loss error model	169
7.5.1	Surface code	170
7.5.2	Raussendorf lattice	174
7.6	Discussion and outlook	174
8	Conclusion	177
A	Complexity	179
A.1	Turing machine	179
A.2	Extended Church-Turing thesis	179
A.3	Complexity classes	179
A.4	Notions of reducibility	181
A.5	Collapse of the Polynomial Hierarchy	182
B	Resource counting	185
B.1	Comparison of all proposed schemes	185
B.2	Comparison of percolation schemes	188
C	Constructive proofs	191
C.1	Proof of theorem 3	191
C.2	Assessing local Clifford equivalence	193
D	Visual-CHP	195
D.1	Internal functions	195
D.2	Cluster building commands	195
D.3	Quantum Operations	196
D.4	Cluster Operations	197
D.5	Output	198

CONTENTS

E Bosonic simulator	199
F Details on 3-GHZ and 4-GHZ state generation	201
F.1 Boosted 3-GHZ generation from Bell pairs	201
F.2 Boosted 4-GHZ generation from Bell pairs	202
F.3 3-GHZ generation from probabilistic SPDC sources	204
G Switch loss	207
G.1 Switch loss calculations in multiplexed GHZ generators	207
G.1.1 Approach A: Multiplexing 3 photon GHZ generators from single photons	207
G.1.2 Approach B: Multiplexing 3-GHZ generators from Bell Pairs	208
G.1.3 Approach C: Multiplexing Bell pair generators from single photons and 3-GHZ generators from Bell Pairs	209
G.1.4 Comparison	213
G.2 State of the art	213
Bibliography	214

LIST OF FIGURES

1.1	Bloch Sphere	25
1.2	One-way quantum computation	30
2.1	Mach-Zehnder interferometer	39
2.2	Polarisation encoding	40
2.3	Path encoding	41
2.4	Transformation between polarisation and path encoding	41
2.5	Reck <i>et al.</i> scheme	42
2.6	Directional coupler	45
2.7	Integrated Mach-Zehnder interferometer	46
2.8	Hong-Ou Mandel experiment in integrated optics	47
2.9	Non-linear phase shift	48
2.10	Conditional sign flip	48
2.11	Conditional phase shift with probability 25%	49
2.12	Encoding to protect against Z measurements.	49
2.13	Photonic chain state	50
2.14	Three qubit circuit	51
2.15	Chain states implementing three qubit circuit	51
2.16	Cluster state construction	52
2.17	Fusion gates	53
2.18	Action of type-I fusion gate	54
2.19	Action of Type-II fusion gate	54
2.20	Parity encoding	55
2.21	Implementation of Z_{90} and CNOT gates on an encoded parity state	57
2.22	Percolation subcritical regime	59
2.23	Percolation critical regime	59
2.24	Percolation supercritical regime	59
2.25	Most significant percolation signatures	59
2.26	Building a cluster state from small photonic clusters	61
2.27	Renormalisation procedure	61
2.28	Renormalisation of a percolated rectangular lattice into an hexagonal lattice	62
2.29	Dependence of renormalised block size on the size of the lattice	63
3.1	Stabilizer octahedron embedded in Bloch sphere	71
3.2	Example of local complementation	78
4.1	Colour convention for the rotated PBS and the fusion gates.	90
4.2	Optical scheme to measure in the Bell basis	93
4.3	Improved Bell-state measurement, using a Bell pair ancilla	94
4.4	Improved Bell-state measurement, using four single ancillary photons	96
4.5	Original fusion gates	97
4.6	Boosted Type-II fusion gate	100

4.7	Scheme for generating a Bell pair from single photons	101
4.8	Linear optical circuit that generates a 3-GHZ state from single photons	102
4.9	Linear optical circuit for the generation of a 4-GHZ state from single photons . .	104
4.10	Linear optical circuit to generate an n -GHZ from single photons	104
4.11	Optimised scheme for the generation of n -photon GHZ states from Bell pairs . .	105
4.12	Higher success probability schemes for the generation of GHZ states from Bell pairs	106
4.13	Schematic layout for spatial and temporal multiplexing	108
4.14	Comparison of proposed n -GHZ generation schemes	110
4.15	Comparison of strategies for the generation of 3-GHZ states	111
4.16	3-GHZ generation using Bell pairs from probabilistic SPDC sources	112
4.17	Comparison of 3-GHZ multiplexing generation schemes	112
4.18	Rotated type-II fusion gates	114
4.19	Deterministic Bell pairs consumed when multiplexing the GHZ generation scheme in figure 4.11	115
4.20	Summary of entangled states generation from single photons.	117
4.21	Summary of entangled states generation from Bell Pairs	118
5.1	Boosted Type-II fusion gate	120
5.2	Full layout of a layer of the diamond graph using 3-photon GHZ states as input .	122
5.3	Probabilistic creation of star micro-clusters	123
5.4	Fusion of 5-qubit micro-cluster to form the diamond lattice	124
5.5	Optimising connectivity of microclusters	125
5.6	Instance of percolated diamond cluster	126
5.7	Pattern of fusions success and failure	126
5.8	Percolation threshold for diamond lattice LOQC scheme	127
5.9	Percolation probabilities as a function of the length of the cluster	128
5.10	Natural loss tolerance of the diamond photonic lattice	129
5.11	Heralded loss tolerance for the photonic diamond lattice	130
5.12	Renormalised lattice	131
5.13	Probability of having a percolation path through a linear cluster made of renormalised qubits	132
5.14	Comparison of the number of Bell pairs consumed to build the $L \times L$ cluster for the two percolation schemes	134
5.15	Schematic arrangement of probabilistic sources and switching networks that allow the construction of a deterministic 3-GHZ generator	136
5.16	Structural parts of the layer which generated the percolated cluster	137
5.17	Schematic view of the QNIX architecture	139
6.1	Comparison of relative MUX with homogeneous MUX in the task of synchronising two photons for a fusion operation.	142
6.2	Classification of GHZ states and fusions in the photonic lattice	143
6.3	Arrangement of the GHZ state generators and fusion operations on chip	144
6.4	World lines of the three types of photons according to this scheme	145
6.5	Comparison of tolerable loss in the presence of no ancilla loss	147
6.6	Loss threshold trade-off for ancilla and photon loss	148
6.7	Wasted number of 3-GHZ in a multiplexing scheme	150
6.8	Number of attempts needed to obtain a “deterministic” micro-cluster	152
7.1	Planar code	159
7.2	Cycles and errors in the planar code	160
7.3	Unit cell of the Raussendorf lattice	161

7.4	Protocol for fault-tolerant LOQC	162
7.5	Renormalisation of diamond lattice into Raussendorf	164
7.6	Number of GHZ states needed per renormalised block of the Raussendorf lattice	165
7.7	Concentration of a percolated lattice into a universal lattice for MBQC	167
7.8	Mixed site-bond percolation threshold for the Raussendorf lattice	168
7.9	Percolation threshold for the photonic Raussendorf lattice	169
7.10	Bond loss rate in the photonic Raussendorf lattice as a function of the fusion success probability	170
7.11	Planar code with a bond loss	171
7.12	Probability of successful correction as a function of the computational error rate in the presence of lost bonds	172
7.13	Correction of bond loss in planar code	173
7.14	Bond loss in Raussendorf lattice	174
A.1	Cook reducibility	182
A.2	Karp reducibility	182
B.1	Comparison of the size of the renormalised qubit (k) for different cluster sizes (L)	188
B.2	Comparison of the number of Bell pairs consumed per renormalised qubits for different cluster sizes (L).	189
B.3	Comparison of the number of Bell pairs consumed to build the entire $L \times L$ cluster for different cluster sizes (L).	189
F.1	Optical circuit to generate a 3 qubits GHZ ballistically from Bell pairs with 37.5% probability	201
F.2	Optical circuit to generate a 4 qubits GHZ ballistically from Bell pairs with 28.125% probability.	203
G.1	Multiplexing approach A: ballistic generation from single photons	208
G.2	Multiplexing approach B: ballistic generation from Bell pairs	209
G.3	Multiplexing approach C1: intermediate stage where Bell pairs are generated with probability $\frac{1}{4}$	211
G.4	Multiplexing approach C1 optimised	211
G.5	Multiplexing approach C2: intermediate stage where Bell pairs are generated with probability $\frac{3}{16}$	212
G.6	Multiplexing approach C2 optimised	212
G.7	Comparison of all multiplexing approaches	213

GLOSSARY

P Complexity class encompassing the counting problems associated with the decision problems in the complexity class NP.

BPP Bounded-error probabilistic polynomial time complexity class.

BQP Bounded-error quantum polynomial time complexity class.

BSM Bell State Measurement.

CNOT Controlled-Not gate.

CS Conditional sign flip.

CZ Controlled-phase gate.

GHZ Greenberger-Horne-Zeilinger state [1].

HOM Hong-Ou-Mandel.

KLM Knill, Lafflame & Milburn [2].

LC Local Clifford.

LOON Linear Optical Network.

LOQC Linear Optical Quantum Computation.

LU Local Unitary.

MBQC Measurement-Based Quantum Computation.

MUX Multiplexing.

MZI Mach-Zehnder Interferometer.

NP Nondeterministic polynomial time complexity class.

NS Non-linear phase shift.

P Polynomial time complexity class.

PBS Polarising Beam Splitter.

QEC Quantum Error Correction.

QNIX Dynamical linear optical quantum computing architecture.

Glossary

RMUX Relative Multiplexing.

SPDC Spontaneous Parametric Down Conversion.

UQC Universal Quantum Computation.

CHAPTER 1

INTRODUCTION

It's not a Turing machine, but a machine of a different kind.

RICHARD FEYNMAN

1.1 Introduction

Quantum computers have recently been subject to much interest, as they promise to harness effects at microscopic level which have no equivalence in classical physics. It was Feynman who first proposed [3] the idea of a quantum computer as a “probabilistic simulator of a probabilistic nature”. He was addressing the difficulty of simulating a quantum physical system: the problem of exponential growth when dealing with multiple particles. His proposal was for a machine that would work by following the same laws of the system simulated, so that if one repeated a certain experiment a certain number of times, one would find the same probability distribution of results as if the experiment were done in the physical system we wanted to simulate. What Feynman was really proposing was a “quantum simulator”, a controllable quantum system that would simulate the dynamics of another quantum system. A universal reprogrammable machine such as a quantum digital computer, able to perform any logical operation on quantum bits, will be an extremely capable simulator of quantum physical systems¹. Lloyd [4] formalised this intuition by formally showing how such simulation could be performed in a universal quantum computer.

A number of quantum algorithms have been proposed that show a computational advantage with respect to classical algorithms, such as Shor’s factoring algorithm [5], Grover’s unstructured data base search [6] or quantum machine learning [7]. But the most promising application of quantum computers is exactly what Feynman had in mind. Currently, 30% of the world’s supercomputing power in research facilities is being used to solve problems in quantum chemistry and material science [8]. There is a huge breadth of problems in these fields which have real-world applications, such as the design of room temperature superconductors, new medicines, an efficient catalyst to capture carbon from the atmosphere and better catalysts for nitrogen fixation, among others. New fast algorithms have been proposed [9, 8, 10, 11] which show how it is possible to accurately solve simple quantum chemistry problems in ~ 300 seconds. These

¹In analogy with classical systems, the classical digital computer has proven to be the most capable and multi-purpose simulator of classical physics.

problems are outside the scope of what can be achieved with classical computers, the ability to implement them in quantum processors gives hope for a new range of applications for quantum computers that would otherwise be outside our reach. As the authors of a recent review on the field of Quantum Computers [12] put it, the advent of a quantum computer will have a similar impact as that of the laser. The laser provides us with many technological advances but it certainly hasn't replaced light bulbs, and, in the same way, quantum computers won't replace classical computers but will allow us to perform an entire new range of computational tasks.

The necessary accurate control of quantum systems has only been demonstrated in recent years, but only for small number of qubits. A number of quantum computing architectures have recently been proposed for different physical systems, showing how the control capabilities can be scaled up sufficiently to integrate the first quantum computers. Some of the front runners in the quantum computer race are superconducting qubits [13, 14], ion traps with photonic links [15, 16], microwave ion traps [17] and photons [18].

In this thesis, we present a study of linear optics as a candidate system for quantum computing. We study the architecture in detail and bring it much closer to experimental realisation. In doing so, we provide techniques for generating entanglement, possible implementations of quantum error correction and provide a blueprint for an experimental realisation of a linear optical quantum computer.

1.2 Quantum Computing

Quantum bits, or *qubits*, are two-level quantum systems that can be used to store and process quantum information. The two levels are usually represented by $|0\rangle, |1\rangle$, which are known as the computational basis states. The main difference between bits and qubits is that the latter can be in a linear combination, usually referred to as *superposition*, of the two basis states. Hence the most general representation of a qubit is :

$$\psi = \alpha|0\rangle + \beta|1\rangle, \quad (1.1)$$

where α and β are (in general)² complex coefficients. When a qubit is measured in the computational basis, the results 0 or 1 are obtained with probability $|\alpha|^2$ and $|\beta|^2$ respectively. As these probabilities must add up to one ($|\alpha|^2 + |\beta|^2 = 1$), we have a normalisation restriction on the coefficients α and β , that can be geometrically understood as the condition that the qubit's state has length one. We can take this geometric interpretation a bit further and parametrise the quantum state in spherical coordinates

$$\psi = \cos \frac{\theta}{2}|0\rangle + e^{i\varphi} \sin \frac{\theta}{2}|1\rangle, \quad (1.2)$$

and understand the qubit as a vector in a sphere of radius one. This sphere is usually referred to as the Bloch sphere, shown in figure 1.1. Note that in this representation, orthogonal states are diagonally opposed rather than at right angles.

The three cartesian axes of the Bloch sphere form a set of three mutually unbiased bases

²Universal quantum computation can be achieved with states with only real amplitudes [19].

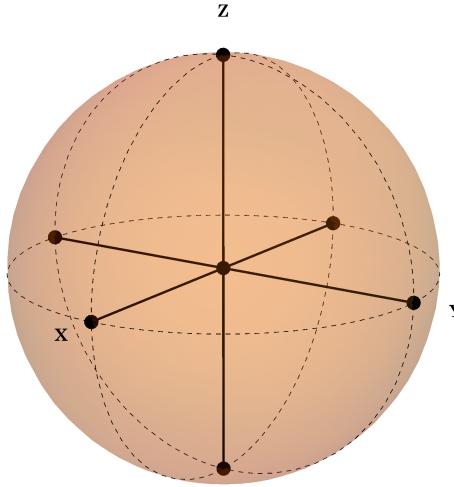


Figure 1.1: Bloch sphere. The three cartesian axes correspond to the eigenstates of the Pauli matrices.

[20], and the Bloch vectors pointing in those directions are the eigenstates of the Pauli matrices:

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \text{and} \quad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (1.3)$$

where the matrices are written in the computational basis. These operators will be referred to throughout the thesis as $\sigma_X = X$, $\sigma_Y = Y$, $\sigma_Z = Z$. The eigenstates of the Z operator are the computational basis states $\{|0\rangle, |1\rangle\}$, whereas the eigenstates of X and Y are $\{|\pm\rangle\}$ and $\{|\pm i\rangle\}$ respectively.

Single-qubit logical gates can be understood as transformations (rotations and reflections) of the states in the Bloch sphere. The most used single qubit gates are:

- Hadamard gate: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$,
- Phase gate: $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$,
- $\pi/8$ gate: $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$,
- Rotations with respect to one of the cartesian axes: $R_{\sigma_k}(\theta) = \cos \frac{\theta}{2} \mathbf{1} - i \sin \frac{\theta}{2} \sigma_k$ where $k \in \{X, Y, Z\}$.

The most commonly used two-qubit gate, controlled-NOT (CNOT), has the same truth table as the classical XOR gate, which flips the target (second) bit when the control (first) bit is in the state 1:

$$|00\rangle \rightarrow |00\rangle, \quad |01\rangle \rightarrow |01\rangle, \quad |10\rangle \rightarrow |11\rangle \quad \text{and} \quad |11\rangle \rightarrow |10\rangle. \quad (1.4)$$

The use of the CNOT gate in conjunction with some of the single qubit gates can produce *entangled* states, which show correlations with no equivalence in classical computation. For example, the action of a CNOT gate with the Hadamard gate on a pair of computational basis qubits yields:

$$|00\rangle \xrightarrow{H_1} |+0\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle). \quad (1.5)$$

This state (which is one of the four maximally-entangled states referred to as Bell pairs [21]), cannot be written as the product of two single-qubit states.

Other entangling gates such as controlled-Phase (CZ) can be obtained from combinations of CNOT gates with single qubit gates; moreover *any* multi-qubit unitary can be obtained in the same way [22], which makes this set of gates *universal*. A set of gates can perform universal quantum computation (UQC) if they are sufficient to approximate any unitary operation to an arbitrary accuracy via a quantum circuit. This universality is crucial as it ensures the equivalence of different models of quantum computation.

1.2.1 Quantum algorithms and computational speedup

In 1936, Church and Turing [23, 24] first stated that there exists a limitation to what can be computed, not imposed by our ingenuity in designing and implementing computational technology, but universally imposed by the laws of Nature. The extended version of the Church-Turing thesis states that *any function naturally to be regarded as “efficiently” computable is “efficiently” computable by a Turing machine*³. In 1985, Deutsch [25] formulated a physical version of the thesis, which is compatible with quantum theory and a ‘universal quantum computer’: *“Every finitely realisable physical system can be perfectly simulated by a universal model computing machine operating by finite means”*. The advantage of using quantum systems to perform computational tasks was realised in the ’90s when a series of algorithms demonstrating quantum speedup appeared. Grover’s search algorithm [6] shows an improvement in scaling from $O(n)$ to $O(\sqrt{n})$ with respect to classical algorithms, which has been proven optimal [26]. Deutsch-Josza’s [27] algorithm shows exponential speedup with respect to a classical algorithm only if no margin of error is allowed, and for a rather contrived problem. In 1994 Shor proposed an algorithm that would become the best-known application for a quantum computer, as it allows to solve problems in NP that are *thought* not to be in P, i.e. prime factoring and discrete logarithm, in polynomial time⁴. Recent interest in quantum simulation [9, 8, 10, 11] has also shown how problems that are considered intractable with classical computers today can be solved efficiently using quantum processors.

The class of problems that are solvable in polynomial time by a quantum computer is usually referred to as BQP. It is *believed* that this complexity class is different from BPP, the class of problems solvable by a classical computer in polynomial time, but no proof has yet been found. Shor’s algorithm provides the strongest evidence for this to be the case, but has a big drawback

³The notions of complexity theory used in this section are explicitly defined in appendix A.

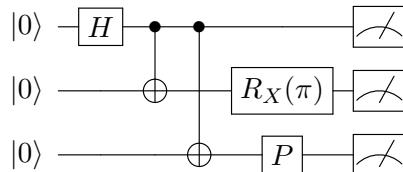
⁴Both of these problems belong to the complexity class $NP \cap coNP$ and they have a similar internal structure which is referred to as the Hidden Subgroup problem [28]. This has led to the insight that a promising approach for the development of other efficient quantum algorithms is to consider other instances of this problem.

as the problem of factoring is not *proven* to be classically hard⁵. The recently proposed problem of BOSON SAMPLING [30, 31], which we introduce in chapter 2, constitutes the strongest evidence that quantum computers have computational powers unattainable in classical systems.

1.2.2 Circuit model

The circuit model is an algorithmic model for quantum computing that closely resembles classical algorithms. Single-qubit and two-qubit operations are performed in sequence on a set of qubits initialised in a fiducial state and the results are read at the end as the outcome of single-qubit measurements. The entanglement and interference necessary for the quantum speedup is built up during the computation and if any ancillary states are used during the computation, their state must be erased so that they no longer interfere with the rest of the computation.

The following circuit diagram shows the most common representation of the quantum logic gates presented earlier in this section:



The procedure runs from left to right: preparation, single Hadamard gate, CNOT gates, rotation and phase gates, measurement.

Circuit model key facts

- **State Space:** A quantum circuit operates on a number of qubits (or two-level quantum systems), and therefore its state space is a 2^n -dimensional complex Hilbert space. The computational basis states are defined as product states of the form $|x_1, \dots, x_n\rangle$, where $x_i = 0, 1$.
- **State Preparation:** Any computational basis state $|x_1, \dots, x_n\rangle$ can be prepared in at most n steps.
- **Quantum Gates:** Gates from an universal family of gates can be applied to any subset of the qubits desired.
- **Measurements:** Measurements in the computational basis can be performed on one or more qubits.
- **Classical Computation:** In principle it is not necessary for the computation, but it can make certain tasks much easier.
- **Procedure of the computation:** Quantum algorithms are run by applying one-qubit and two-qubit gates to the quantum systems, building up the amount of entanglement, until the final measurement in the computational basis gives the result of the computation.

⁵The closely related problem of primality testing has been recently proven efficiently solvable by a classical Turing machine [29].

1.2.3 Measurement-based quantum computation

Prior to the Gottesman-Chuang [32] and Raussendorf-Briegel [33] proposals, the circuit model was commonly used for quantum computation. Measurement-based quantum computation models are radically different to the circuit model (and have no classical analogue), as the resource for the computation is prepared in advance and “offline”. This strategy has the advantage that if errors occur at the preparation stage, the prepared state can be discarded and the procedure can be repeated without any loss of information. There are two main approaches for measurement based quantum computing: the generalised teleportation model [32] and the one-way quantum computer model [33]. They have similarities such as the fact that all computation is performed by doing measurements on a pre-prepared state, and differences, as the teleportation model requires two-qubit measurements whereas the one-way model only requires single qubit measurements. A full study of their relationship can be found in [34]. It must be noted, however, than in most of the literature, and in this thesis, a reference to Measurement-Based Quantum Computing (MBQC) refers to the one-way model.

Generalised quantum teleportation

Quantum teleportation [35] is the process by which an unknown qubit can be teleported from sender to receiver by communicating only two classical bits if they share a pair of maximally entangled particles. First, a joint Bell measurement is performed on the unknown quantum state and one of the qubits of the entangled pair. Then, the outcome of the measurement is transmitted through a classical communication channel. Finally, rotations conditioned on the outcome of the measurement are applied to the other qubit of the Bell pair to recover the unknown state. Gottesman and Chuang realised that [32] if a unitary gate had been applied to the entangled pair and then the pair was used to teleport a qubit, the output of the teleportation procedure would be a transformed version of the input, where the transformation would be dictated by the gate pre-applied to the entangled pair used for the teleportation. This procedure has the advantage that a gate can be applied to *any* state. If the gate itself is difficult to apply to an unknown state, the procedure can be performed by preparing the entangled state with the applied gate directly. This procedure is a *gate teleportation* procedure, and can be used as the basis of a fault-tolerant scheme for quantum computing.

As mentioned earlier, any universal quantum computation can be performed by using a combination of CNOTs and single qubit gates. Gottesman and Chuang showed [32] how gate teleportation can be used in the case of single and two-qubit gates, in particular the CNOT gate. This implies that universal quantum computation can be performed using an alternative set of gates, which does not require two-qubit gates except for the Bell measurement used for teleportation. This scheme relies on the ability to prepare small entangled states such as Bell pairs and Greenberger-Horne-Zeilinger (GHZ) states [1] and perform deterministic Bell measurements. One of the advantages of this scheme is that it can be used to perform fault-tolerant quantum computation. The protection against error comes from the fact that the gates are pre-applied to the entangled pairs before they are ever used, therefore errors can be filtered before any quantum information is teleported through the entangled pair.

One-way quantum computer

In the one way model [33] the entire resource for the computation is supplied at the beginning of the computation, in the form of a highly entangled multi-particle state. This state is usually referred to as cluster state and it is the same for every computation, although it may vary in size. The information is then processed by carrying a series of adaptive single qubit measurements.

This highly entangled multi-particle state is prepared by applying a pattern of entangling gates to the qubits. The initial state of every qubit is $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ and they are subsequently linked by applying entangling gates to form the cluster state. A generic cluster state of n particles is not easy to write in any basis, but it can be efficiently described with a graph, where each node of the graph represents a qubit and each bond denotes that the two sites have been connected by an entangling controlled-Z gate (CZ) operation. The CZ operation when acting on two qubits in the computational basis, flips the phase of the $|11\rangle$ state:

$$|00\rangle \rightarrow |00\rangle, \quad |01\rangle \rightarrow |01\rangle, \quad |10\rangle \rightarrow |10\rangle \quad \text{and} \quad |11\rangle \rightarrow -|11\rangle. \quad (1.6)$$

The preparation of the cluster state is easy in theory. One only has to apply a CZ gate to every pair of qubits that need to be connected. It does not matter the order in which the gates are applied as they commute with each other. In practice however, the preparation of this highly-entangled cluster state can be technically difficult, not only because entangling such a big number of particles is very challenging, but also because not all entangled states make the cluster state a universal resource for quantum computation. A cluster state is a universal resource for MBQC if any quantum state can be generated from its original state solely by local (single-qubit) operations [36]. It has been shown that for the cluster state to be a universal resource, it “*must exhibit maximal (scaling of) entanglement with respect to essentially all types of entanglement*” [36, 37, 38]. For example, it can be shown that GHZ states or W-states [39] are not universal resources despite being highly entangled. In both cases, at least one type of entanglement is non-maximal [38], rendering these states inadequate as resources for MBQC.

The cluster can be shaped by applying Pauli gates. Measuring a qubit on the computational basis (Z) effectively removes the qubit and all its bonds from the cluster, while measuring a qubit in the X basis has the effect of redistributing the entanglement structure of the neighbouring qubits (in a manner that highly depends on the structure of the measured qubit’s neighbourhood). The information processing is done via sequential measurement of the qubits in a certain basis. It is assumed that the correct algorithm is performed if all the measurement outcomes are the $+1$ eigenstate, however, given the probabilistic nature of quantum mechanics, this is not always the case. We can steer the computation back to its correct subspace by applying Pauli corrections to subsequent measurements. Therefore, measurement results determine the basis of the following measurements on other qubits⁶. Finally the result of the computation is read out by one last measurement in the computational basis.

In figure 1.2 we can see an example of a quantum algorithm performed using the MBQC model. We initially start with a rectangular cluster state of 6×12 qubits, where qubits are

⁶This adaptivity of the measurements means that the computation cannot be performed instantaneously. There exists a restricted class of quantum computations that is temporally unstructured and can be performed instantaneously [40], but it is not universal.

1. INTRODUCTION

located at the vertices of the grid (grid lines are not shown). Measuring qubits in the Z basis allows us to shape the grid into the shape we need for the performance of the algorithm. Despite starting with 72 physical qubits, this algorithm is actually performed on only 3 logical qubits, which are shown by the three horizontal yellow arrows. The yellow paths between the logical qubits represent two-qubit gates as explained in the circuit model and the rest of the single-qubit gates are performed by measuring the qubits in the X-Y plane.

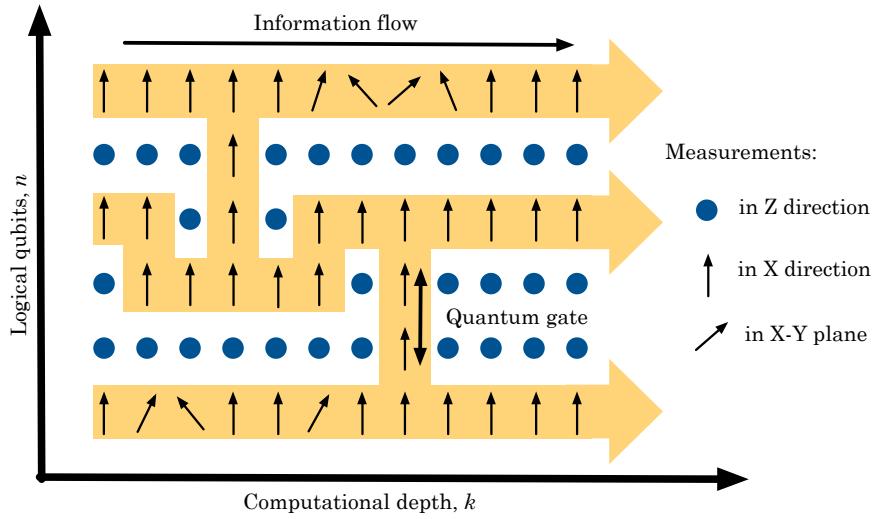


Figure 1.2: One-way quantum computation. Horizontal arrows indicate the information flow and each of them represent a logical qubit. Vertical lines correspond to quantum gates performed between the logical qubits. The small arrows show the direction in which the qubit is to be measured in. Image from [33], copyright (2001) by the APS.

This model of computation has a huge technical advantage over the classic circuit model, which makes it very appealing to implement quantum computation using certain physical systems. The cluster states can be produced offline and only when the resource is prepared correctly, the computation is performed. For many physical systems, performing entangling gates is the most challenging part of the computation, but if we post-select on the successful preparations of the resource state, this model substantially increases the probability of a successful computation. All that is required then is to be able to successfully perform single qubit gates with high fidelity, which is less technologically demanding for many physical systems such as superconducting qubits, ion traps and linear optics [41, 16, 42, 43].

MBQC model key facts

- **State Space:** A cluster state computation operates on $n \times k$ physical qubits (or two-level quantum systems) for a quantum algorithm with $\text{len}n$ logical qubits. Its state space is a $2^{n \times k}$ -dimensional complex Hilbert space. The computational basis states are defined as product states of the form $|x_1, \dots, x_{n \times k}\rangle$, where $x_i = 0, 1$. The graph state representation is usually used.
- **State Preparation:** Before the computation starts a resource state must be prepared.

This resource state is a highly entangled multi-particle state (cluster state), which is the same (except for its size) for every computation.

- **Quantum Gates:** An entangling gate is performed on qubits in a product state to build the resource state. Afterwards only single qubit measurements are needed.
- **Measurements:** Single qubit measurements can be performed in an arbitrary basis. (Or if only measurements of the computational basis can be performed, single qubit rotations must be performed alongside.)
- **Classical Computation:** Classical computation alongside the quantum computations is a key feature of this model, as the basis of the measurements performed sequentially depends on the results of previous measurements.
- **Procedure of the computation:** The entire resource for the quantum computation is supplied at the beginning of the computation. The information is then processed by carrying a series of adaptive single qubit measurements.

1.2.4 Adiabatic model

Although less relevant for the purpose of this thesis, we briefly mention for completeness a new paradigm for quantum computation that was recently proposed [44] based on *quantum adiabatic evolution*. Computation in this model is not performed by applying gates or measurements to qubits, but rather the algorithm starts from a disordered state and it arrives at a solution to the problem by performing what can be understood as a quantum local search. The procedure for the computation is as follows:

- At time $t = 0$, the quantum mechanical system is described by a Hamiltonian H_E , whose eigenstates are easy to compute.
- The system is slowly transformed to the final Hamiltonian at time $t = T$, whose ground eigenstates are the solution to the problem that needs to be solved. This process can be described by a time-dependent Hamiltonian

$$H(t) = A(t)H_E + B(t)H_P. \quad (1.7)$$

$A(t)$ and $B(t)$ are slowly varying monotonic functions such that $A(0) = 1, B(0) = 0$ and $A(T) = 0, B(T) = 1$. According to the *adiabatic theorem* [45], if the evolution is slow enough, i.e. T is long enough, and if there is a gap between the ground state eigenvalue and the rest of the Hamiltonian's spectrum, the state of the system at time $t = T$ would correspond to the ground state of H_P , thus producing the solution to the problem.

- Measurement of the ground state allows for the extraction of the solution.

This model for quantum computation has been proven to be universal for quantum computation [46].

1.3 Physical requirements of a Quantum Computer

A quantum computer is a multi-purpose quantum processor on which a variety of quantum algorithms can be performed. The first characterisation of the physical requirements for an implementation of a fault-tolerant computer was made by DiVincenzo [47]. He posed five requirements for the implementation of fault-tolerant quantum computation, plus two additional requirements for quantum communications:

1. **A scalable physical system with well characterised qubits (two level systems representing the states $|0\rangle$ and $|1\rangle$):** By well characterised we mean that its physical parameters must be accurately known, as well as the presence of couplings to other systems or external fields.
2. **The ability to initialise the state of the qubits in a simple fiducial state, such as $|000\dots\rangle$:** This arises from the computing requirement that the initial state of any computation must be known in advance.
3. **Long relevant decoherence times, much longer than the gate operation time:** Decoherence has been identified as one of the principal mechanisms for the emergence of classical behaviour, hence the requirement of long relevant decoherence times for quantum computers. By relevant we mean that they should be the ones that apply to the particular degree of freedom in which the qubit is stored. Shorter decoherence times can be tolerated by using quantum error correction techniques, which make experimental quantum computing more feasible.
4. **A “universal” set of quantum gates:** The ability to perform any quantum computation can be reduced to the ability of performing a universal set of gates.
5. **A qubit-specific measurement capability:** The result of a computation must be read out, which requires the ability to read the state of specific qubits. While 100% measurement efficiency is desirable, it is possible to trade efficiency for resource consumption.

The main challenge is to build a quantum computer that simultaneously maintains the abilities of controlling quantum systems and measuring them, while at the same time preserving their isolation from the controlled parts of their environment. Quantum communications can be the key to solve this, the disturbance can be kept to a minimum if the different components are interconnected in such a way that measurements are made far away from the memories that need to be kept isolated. DiVincenzo proposed two extra requirements for quantum communications: The ability to interconvert stationary and flying qubits; and the ability to faithfully transmit flying qubits between specified locations.

These physical requirements for a quantum computer were specifically formulated for the circuit model and don't fit well other models of computation such as the MBQC or adiabatic models. In [48], a formal operational definition of a quantum computer is introduced as well as general criteria for its implementation. In this formulation, a quantum computer is a device that consists of a quantum memory whose quantum evolution can be controlled and from which entropy can be extracted using an information-theoretic procedure (i.e. cooling). A readout

mechanism allowing the extraction of subsets of quantum memory must exist also. These criteria are met when the quantum computers is a scalable device operating fault-tolerantly.

1.3.1 Real-world constraints

When considering implementations of a quantum computer, most proposals (including the proposal in chapter 5 of this thesis) are mainly concerned about efficiency. An implementation is deemed *efficient* if the number of resources and time (accounting for both quantum and classical processes) necessary to perform a computation on n qubits scales as $\text{poly}(n)$. Theoretically, this is all that is needed. However not every efficient proposal can have a *feasible* implementation in practice. For example, the first proposal for a linear optical quantum computer [2], which we will review in detail in chapter 2, was theoretically efficient with a polynomial scaling of resources, but the overhead was so large that it is impossible to build for all practical purposes.

Choosing the physical system that will ultimately be the main platform for quantum computers is not easy. Technological problems that may seem unsurmountable today might be solved in a few years time. However, as quantum computers are physical devices, the laws of physics ultimately dictate what they can do or not [49]. The amount of information that classical computers are capable of processing and the rate at which they do so has doubled every 18-months for the last 40 years, which is known as Moore's law [50]. However, Moore's law is not a law of Nature and rather an observation of human ingenuity (and economic power), and it is expected it will soon reach saturation: Intel has already confirmed that their cadence in chip production has slowed.

The largest transistor count in a single CPU today is of 5.5 billion transistors, with current transistors being of the size of $\sim O(10)\text{nm}$, we can imagine that even if we have quantum processors, machines with more than a trillion components do not seem physically feasible. There are other types of constraints too: if all our components need to be at mK temperature, the size of the quantum computer will be restricted by cooling ability⁷, the clock speed (number of operations per second) will be limited by the amount of available energy in the system [49], but more energy means more noise and entropy limits the amount of information that can be processed. The ultimate limits for computation are given by the laws of physics [49], but there is no guarantee that these limits can really be reached.

1.3.2 Physical implementations for quantum computers

Various quantum technologies have been considered as good candidates for building quantum computers. They each have their own advantages and challenges, and it is not clear today which will be the final technology; it might not even be just one but a combination of several. In this section we briefly mention the three technologies that (in our view) are most promising⁸. Despite their differences, they have one significant factor in common: they are compatible with microfabrication techniques which will allow each architecture to become modular and be made from regular-sized chips.

⁷It is true that there exist large scale machines which operate at $\sim 2K$ such as CERN, but they would not be considered efficient in the sense we describe here.

⁸A review of different technologies that are being developed for quantum computation can be found in [12].

Ion traps with photonic links

Ions can be controlled and manipulated in macroscopic traps with a very high degree of accuracy [16]. Excellent control has been achieved in macroscopic ion traps with nearly 100% [15] fidelity in all gates, however for current implementations there exists a harsh scalability factor: only a bounded number of ions can be trapped and individually addressed in the chain. The networked model for quantum computation [51], in which cells with a small number of qubits are interconnected to perform quantum computation, is particularly well suited for this technology and full-scale architectures have been proposed [15]. The entanglement between different cells is obtained via entangling operations on photons emitted by ions in the different traps. This operation is very slow however (~ 300 times slower than any other gate [16]) and uses large photonic switching networks which rapidly increase the photon loss rate. New very low-loss photonic switches and better entangling operations are needed for this technology to be feasible on a large scale. A new approach to overcome the scalability factor is that of integrated ion traps [15], in which standard semi-conductor processing techniques can be used to fabricate micrometer-scale surface-chip traps. Having integrated elements implies a scale reduction in the size of the experimental components, however in these microscopic traps, multi-qubit entangling operations become more challenging. This problem however does not appear to be a fundamental limitation as it can be suppressed at cryogenic temperature or with an adequate treatment of the trap surface.

Superconducting qubits

Superconducting systems exhibit generic quantum properties commonly associated with atoms, such as quantised energy levels, entanglement and superposition of states [52]. As such, artificial-atoms can be engineered from these systems and exquisite control can be achieved by using electromagnetic pulses. Recent demonstrations [41] show the ability to perform single qubit gates with 99.92% fidelity and two-qubit gates with 99.4% fidelity. Moreover, these fidelities are within the fault-tolerant threshold [53] for the surface code [54] which has allowed the experimental implementation of a small surface code implementation of five qubits [41]. Although this implementation of quantum computing benefits from microfabrication of the devices, it has a number of shortcomings. The most important are the cross-talk between nanowires, which hinders the construction of three dimensional qubit structures⁹ and the fact that they operate at mK temperatures, which limits the number of qubits that can be implemented due to the limited cooling capacity.

Linear optics

Single photons are very good carriers of information with low decoherence rates and very high single-qubit gate fidelity [18, 56, 43, 57]. Non-deterministic two-qubit operations and photon loss are a challenge for current technologies, but a series of theoretical advances in recent years (which will be explained in detail in chapter 2) together with technological advances make this physical system a competitive candidate for quantum computing. Throughout this thesis we will explore

⁹These are considered more advantageous for the implementation of fault-tolerance [55].

in detail theoretical techniques that allow the optimisation of linear optical schemes for quantum computing. Experimentally, there have been many technological advances [58, 56, 43, 59, 60] which make this technology ever more feasible. In particular, integrated optical devices can be nano-fabricated. The ability to miniaturise $O(10^6)$ linear optical elements on a single chip [61], is a very promising sign for the construction of linear optical quantum computers with millions of elements per silicon chip in the future.

1.4 Thesis outline

This chapter has introduced the concept of quantum computers, the different computational models, the physical requirements needed for a feasible implementation and the most promising physical systems for the implementation of a quantum computer. In chapter 2, we focus on linear optics (in particular integrated linear optics) as a physical system suitable for quantum computing. We also give a detailed account of the most important proposals for Linear Optical Quantum Computing (LOQC). We compare the resource efficiency of all proposed protocols according to the number of entangled pairs that are consumed during the computation. In chapter 3 we introduce the stabilizer formalism and explain how we can use it to build a simulator for certain classes of quantum computations. This simulator (its main functions are detailed in appendix D) not only allows to perform quantum error-correcting protocols, but given its ability to visualise the quantum computations as transformations on a graph, it allows to build a better intuition. This has allowed for the design of the LOQC protocol presented in chapter 5. In chapter 4, we focus on the creation of entanglement in linear optics. We review literature results and propose new schemes to generate small entangled states. In chapter 5 we present a new protocol for LOQC which is shown to be at least one order of magnitude more efficient than previous proposals. It is a percolation-based protocol with constant depth which only requires 3-photon GHZ states and Bell pairs as resources, an improvement on previous protocols that required larger entangled states. We conclude the chapter by presenting QNIX, which is a linear optical architecture for quantum computing. We outline the necessary experimental stages required for its construction and the overall structure. In chapter 6 we present a full analysis of the resource efficiency of QNIX and introduce a new multiplexing scheme that allows an optimal utilisation of resources. In chapter 7 we introduce basic concepts of fault-tolerance and quantum error-correction (QEC) and outline how they can be implemented in the LOQC protocol presented in chapter 5. Furthermore, three main ways of implementing fault-tolerance and preliminary results on the efficiency are presented. Chapter 8 concludes this thesis with a summary of the work presented and provides an outline of possible directions for future research.

CHAPTER 2

LINEAR OPTICAL QUANTUM COMPUTING

2.1 Introduction

Having reviewed the computational advantages of a quantum computer and the necessary elements to build one, we now turn our attention to a particular physical system that can be used for quantum computing: linear optics. To realise a scalable quantum computer using photons as our physical system we will require low-loss optical networks and highly efficient single-photon sources and detectors. This set up will yield the accurate controlled manipulation, interference and measurement of single photons required to perform quantum computation

In this chapter, we review qubit encoding and operations in linear optics. We focus on integrated systems, as our own proposal for a linear optical quantum computer (chapter 5) is intended for such systems. We give a detailed overview of the proposals for a linear optical quantum computer to date¹. In particular, we present a full account of the resources consumed by each proposal, as for a quantum computer to be viable, the resources required must scale polynomially with the system size. Different proposals account for their resource consumption in different ways: counting entangled states used, number of particular gates applied, number of optical elements or level of encoding. In order to have a unified view of all these proposals, as well as having an understanding of the computational costs of different strategies, we map the resource consumption of each protocol to a single figure of merit: Bell pairs used per single successful entangling gate. A full account of these calculations can be found in appendix B.

2.2 Linear Optics

Photonic systems are favourable candidates as qubits in a quantum computer [18]. There are many degrees of freedom of the photon that can be used to encode qubits, and they are relatively free of the decoherence that plagues other systems (fulfilling DiVincenzo's criteria 1 & 3). Measurement is done using photon detectors (DiVincenzo's 5), although photon loss still poses a significant challenge for current technologies². We require the ability to perform single and two-qubit gates in order to be able to implement universal quantum computing

¹There is one recent proposal [62] which we postpone until chapter 7, as unlike the proposals in this chapter, it is mainly focused on fault-tolerance rather than scalability.

²In chapter 6 we compare the maximum photon loss per physical component required by some specific architectures with current state of the art component specifications.

(DiVincenzo's 4). Single-qubit gates can be easily performed, as will be shown later in the chapter. These single-qubit gates are all we need to initialise the state to a fiducial state as required by DiVincenzo's criteria 2 provided access to single photon sources. However, two-qubit gates are more challenging to implement. Nonetheless, they can be achieved and several proposals for a linear optical quantum computer have been put forward.

2.2.1 Single photons as information carriers

There are several representations of single photons as information carriers. The two main implementations are single-rail and dual-rail qubits. In the single-rail representation, the qubit is carried by a single optical mode and the qubit degree of freedom is the photon number. This implementation is not very common as the single qubit operations do not preserve photon number and are therefore much more experimentally challenging. Also, from an error-correction point of view, loss of a photon could be confused with an Pauli-X error in some instances. Much more widely used is the dual-rail implementation, which encodes a qubit using two optical modes. These modes can be polarisation, spatial modes, orbital angular momentum or time-bins [63]. Most of the work in this thesis will be targeted towards polarisation and path encodings, these two are treated similarly as one can be easily converted into the other.

In order to properly define the optical modes, we assume the electromagnetic field to be quantised in terms of plane waves that extend to infinity in all directions³. Plane waves provide a relatively simple quantisation in terms of the wave vector, but are not physical solutions. Localised optical modes can be defined as a superposition of these plane waves, they have well defined bosonic commutation relations and will be hereafter referred to as the physical modes representing the qubits.

The computational basis states are defined for two optical modes \hat{a}_1 and \hat{a}_2 as follows:

$$|0\rangle_L = \hat{a}_1^\dagger |0,0\rangle_{1,2} = |1,0\rangle_{1,2} \quad \text{and} \quad |1\rangle_L = \hat{a}_2^\dagger |0,0\rangle_{1,2} = |0,1\rangle_{1,2} \quad (2.1)$$

We will assume that these two modes are completely indistinguishable in any other degrees of freedom and hence we suppress other mode information from our description.

Any linear optical operation on two modes can be described by a *passive* Bogoliubov transformation of the optical modes, which is a transformation that does not mix the creation and annihilation operators and can therefore be described as

$$\hat{a}_1^\dagger \rightarrow \hat{a}'_1^\dagger = \alpha_{11}\hat{a}_1^\dagger + \alpha_{12}\hat{a}_2^\dagger \quad (2.2)$$

$$\hat{a}_2^\dagger \rightarrow \hat{a}'_2^\dagger = \alpha_{21}\hat{a}_1^\dagger + \alpha_{22}\hat{a}_2^\dagger \quad (2.3)$$

where the transformation $\begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix}$ is a unitary matrix.

All single qubit rotations can be written as mode transformations on \hat{a}_1 and \hat{a}_2 , which can be performed by the generalised beam-splitter and phase-shift transformations. These two-mode operations can be described by the transformations of the creation and annihilation operators.

³For the full derivation of the quantisation, see [63].

The most general $U(2)$ transformation is given by

$$\hat{a}_1^\dagger \rightarrow \cos \theta \hat{a}_1^\dagger - i e^{i\varphi} \sin \theta \hat{a}_2^\dagger \quad (2.4)$$

$$\hat{a}_2^\dagger \rightarrow -i e^{-i\varphi} \sin \theta \hat{a}_1^\dagger + \cos \theta \hat{a}_2^\dagger \quad (2.5)$$

Here, the parameter θ describes the reflectivity of the beam-splitter and φ determines the phase shift on the reflected mode of the beam-splitter. A beam-splitter with variable reflectivity can also be implemented by using a Mach-Zehnder interferometer (MZI) with two balanced beam-splitters and a phase-shifter, as it is shown in figure 2.1.

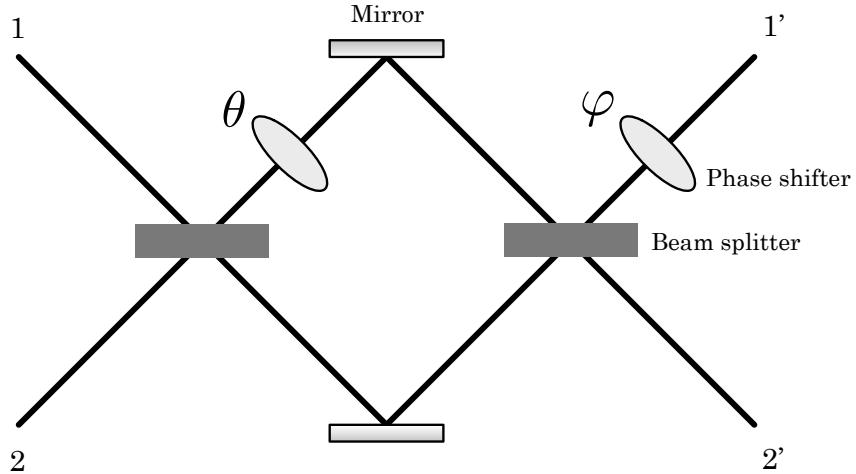


Figure 2.1: Most general transformation between two optical modes, implemented by a Mach-Zehnder interferometer (MZI) and a phase-shifter on one output mode. Figure adapted from [64], copyright (1994) by the APS.

It is important to note that not all qubit transformations are rotations in the Bloch sphere. Some operations, like the Hadamard or Phase gates are reflections with respect to a particular axis. These transformations can be achieved by adding by an extra mode-dependent phase shift to a rotation.

Polarisation qubits

Polarisation is a natural degree of freedom to encode qubits as the polarisation transformations are also generated by the Pauli matrices. It has historically been the most commonly chosen degree of freedom in photonic systems as it is extremely stable in bulk optics [65]. Two orthogonal polarisation modes are chosen as the computational basis states (eigenstates of the σ_Z matrix), usually the horizontal and vertical polarisations (in some chosen reference frame) in the plane parallel to the direction of the wave vector \vec{k} . The diagonal and anti-diagonal polarisations then correspond to the eigenstates of the σ_X matrix and the right and left circularly polarised states correspond to the eigenstates of the σ_Y matrix. They thus form a complete qubit basis, as they cover the three orthogonal axes of the Bloch sphere.

In the notation used previously to define qubits (logical modes) in optical modes we have

$$|0\rangle_L = \hat{a}_H^\dagger |0,0\rangle_{H,V} = |1,0\rangle_{H,V} = |H\rangle \quad \text{and} \quad |1\rangle_L = \hat{a}_V^\dagger |0,0\rangle_{H,V} = |0,1\rangle_{H,V} = |V\rangle \quad (2.6)$$

With respect to the wave vector \vec{k} , the qubit basis states are defined as shown in figure 2.2.

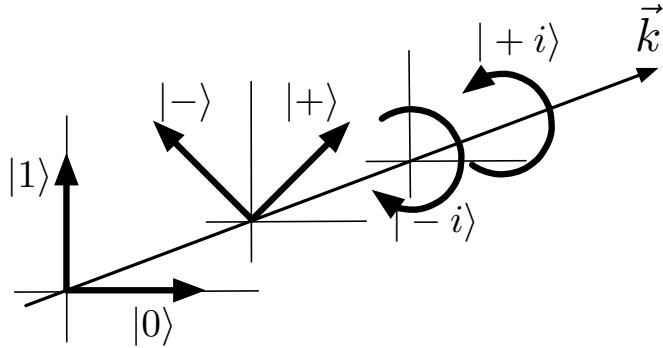


Figure 2.2: Polarisation encoding of qubit basis states.

Transformations between optical states (i.e. polarisation rotations) can be implemented using plates of a birefringent material, in which light of different polarisation travels at different speed through the material [66].

A linear optical element that we will use often in this thesis is the *Polarising Beam Splitter* (PBS). This element is made of a birefringent material, and aligned so that the angle of transmission for two orthogonal polarised beams is different, thus it separates an incident beam of light in two orthogonal beams of polarised light. For example, a PBS oriented in the H-V basis will always transmit horizontally-polarised photons and reflect vertically-polarised photons. This corresponds to the mode transformation:

$$\hat{a}_{H,1}^\dagger \rightarrow \hat{a}'_{H,1} \quad \hat{a}_{H,2}^\dagger \rightarrow \hat{a}'_{H,2} \quad (2.7)$$

$$\hat{a}_{V,1}^\dagger \rightarrow \hat{a}'_{V,2} \quad \hat{a}_{V,2}^\dagger \rightarrow \hat{a}'_{V,1} \quad (2.8)$$

We can observe that the effect of this optical element on two indistinguishable qubits is to exchange the vertical mode amplitudes.

In chapter 4 we will use the polarisation encoding to review the creation of entanglement with linear optical elements and introduce new schemes for the generation of small entangled states. We will solely use the creation operators and their transformations to describe the states and operations, as this is the notation used in the BOSONIC SIMULATOR⁴ use to perform calculations. The creation operators will be written as $h_i^m v_j^n$, where the subscript indicates the spatial optical mode, the letter of the operator (*h* or *v*) indicates the logical (polarisation) state and the superscript indicates the number of photons with said polarisation in that optical mode.

⁴See appendix E for a brief description of this simulator.

Path encoded qubits

In this encoding the logical computational basis states are represented by the occupation of one of two paths:

$$|0\rangle_L = \hat{a}_1^\dagger |0\rangle_{1,2} = |10\rangle \quad \text{and} \quad |1\rangle_L = \hat{a}_2^\dagger |0\rangle = |01\rangle \quad (2.9)$$

where the notation $|ij\rangle$ means i photons in the upper waveguide and j photons in the lower waveguide (see figure 2.3). The control and manipulation of the photons in this encoding is accomplished using combinations of beam-splitters and phase-shifters. In section 2.3 we will explain in more detail how these optical operations are implemented in integrated optical systems that use this encoding.

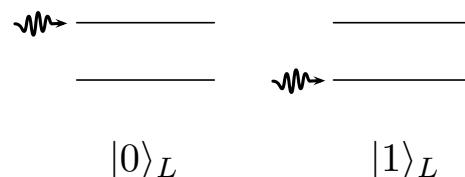


Figure 2.3: Path encoding of the logical computational basis states.

Note that even though this type of encoding would imply always having an even number of waveguides (as we would use two waveguides for each mode) having extra (ancillary) waveguides as vacuum can be useful, as shown in the probabilistic implementation of a CNOT gate in the coincidence basis by Ralph *et al.* [67].

Transformation between polarisation and path encodings

The polarisation and path encodings can be easily and deterministically converted one into the other using a PBS as can be seen in figure 2.4.

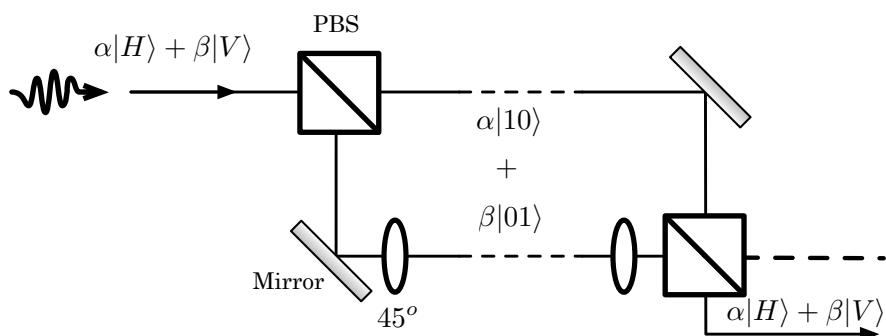


Figure 2.4: Linear optical circuit showing the transformation between polarisation and path encodings, using polarising beams splitters (PBS) and half wave plate that implement a 45° rotation on the state. Figure adapted from [18], copyright (2007) by the AAAS.

2.2.2 Implementation of any $U(N)$ in an optical multiport: Reck et al. scheme

Any lossless experimental linear optical setup can be described by a unitary operator, obtained by multiplying together the matrix representations of each individual component. However, it is not so obvious to see that the opposite is true, i.e. that any unitary operator $U(N)$ on N modes can be implemented by an optical interferometer⁵. This question was resolved by the proposal of Reck *et al.*[64] of an algorithmic proof which shows that any discrete finite-dimensional unitary operator can be constructed as an $n \times n$ multiport. Any unitary operator can be implemented experimentally in this way as well, as can the measurement of any observable that corresponds to any discrete Hermitian matrix.

The $n \times n$ unitary operator is reduced to a series of 2×2 matrices, representing the most general element of $U(2)$. Recall that this most general element corresponds to a lossless beam-splitter with a phase-shifter in one output port. Reck *et al.* show that they can construct an experiment equivalent to any $U(N)$ matrix by using the generalised beam-splitter in successive $U(2)$ transformations that together span the N -dimensional Hilbert space. Their proof shows the equivalence between the tasks of designing an optical experiment that implements an arbitrary $U(N)$ matrix and factorising said unitary matrix into a product of block matrices, each of which can be implemented using beam-splitters and phase-shifters. The successive decomposition of the $U(N)$ unitary into the series of $U(2)$ operations is equivalent to setting up the 2×2 beam-splitters in series. The algorithm proposed is recursive and therefore valid in any finite dimension. The maximum number of beam-splitters required is $\binom{N}{2} = \frac{N(N-1)}{2}$, which is quadratic in the number of modes. This decomposition is optimal as it has the exact same number of parameters as needed to fully describe a $U(N)$ matrix.

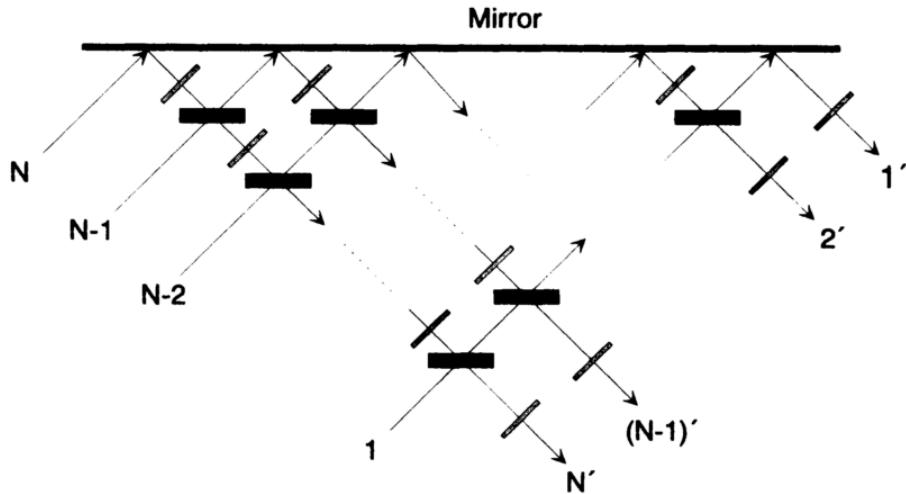


Figure 2.5: Reck *et al.* scheme: triangular arrangement of beam-splitters that implements a $U(N)$ matrix on an optical multiport. Image from [64], copyright (1994) by the APS.

The implementation of all $U(N)$ matrices makes the measurement of any discrete Hermitian operator possible. To do so, it is only required to implement the unitary that relates the

⁵The procedure to decompose a unitary matrix into two-level matrices was already well known, but not its realisation as an optical circuit.

eigenbasis of the Hermitian operator to the single mode occupation basis. Following the unitary operator, we will have an array of N detectors, one per optical mode, each corresponding to an orthogonal eigenstate. Detection of a photon on one of the output modes will correspond to measurement of an eigenstate of the Hermitian matrix.

Recently, a reprogrammable version of the Reck scheme has been experimentally implemented in integrated optics [43], with variable phase-shifters that allow the implementation of all possible linear optical protocols up to the size of the circuit (6 modes) involving 35 adjustable parameters. A series of six-mode experiments are shown [43], they showcase the versatility of this construction and it is highlighted as a fundamental piece of technology for future linear optical schemes.

2.2.3 Creating entanglement in Linear Optics

Creating entanglement between qubit modes using only linear optical elements is extremely difficult. Without the use of extra resources⁶, the best we can do [68] is a probabilistic entangling gate with 50% success probability. The difficulty of entangling two single photons can be explained physically because photons don't interact. The probabilistic entangling gate is achieved through interference, not interaction [69].

Mathematically, this impossibility of creating entanglement can be explained due to the linearity of the passive Bogoliubov transformations [63]. If the task of creating entanglement were possible, the transformation $|HH\rangle \rightarrow \frac{1}{\sqrt{2}}(|HH\rangle + |VV\rangle)$ would be a linear transformation of the Bogoliubov coefficients. Using the previously defined Bogoliubov transformations we have:

$$\hat{h}_1^\dagger \hat{h}_2^\dagger \rightarrow \hat{h}_1^\dagger \hat{h}_2^\dagger + \hat{v}_1^\dagger \hat{v}_2^\dagger \quad (2.10)$$

which cannot be written as the product of two passive Bogoliubov transformations.

However, we have seen that the Reck scheme [64] gives us a recipe to build any unitary operation on N optical modes using $O(N^2)$ beams splitters and phase-shifters. It seems therefore that we should be able to implement a quantum computation using such interferometers, as the gates in a quantum computer can also be described as unitary operations. In particular, the CZ operation is a 4×4 unitary operation on two *qubit* modes, can't it be expressed in terms of two *optical* modes? The subtlety lies in that the basis of the unitary matrices we are trying to compare is quite different. A usual CZ gate is written on the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, which translated to the encoding in optical modes would be $\{\hat{h}_1^\dagger \hat{h}_2^\dagger, \hat{h}_1^\dagger \hat{v}_2^\dagger, \hat{v}_1^\dagger \hat{h}_2^\dagger, \hat{v}_1^\dagger \hat{v}_2^\dagger\}$. However, the $N = 4$ interferometer operates on the basis $\{\hat{h}_1^\dagger, \hat{v}_1^\dagger, \hat{h}_2^\dagger, \hat{v}_2^\dagger\}$ and in this basis, the mode transformation cannot be written as a unitary matrix without mixing creation and annihilation operators.

2.2.4 Computational complexity of linear optics

Entanglement is a fundamental component for quantum computing [70, 71], the fact that a two-qubit gate cannot be realised deterministically in linear optics might lead to the *wrong* conclusion that a linear optical system is easy to simulate classically. A recent result by Aaronson and

⁶In chapter 4 we explore how efficiently we can create entanglement using non-vacuum ancilla states.

Arkhipov [30, 31] explores the computational complexity of classically simulating a linear optical system and concludes that a model for non-interacting photons not only cannot be efficiently simulable by a classical computer, but its complexity is far greater than that of NP problems⁷.

Aaronson and Arkhipov propose the problem of **BOSONSAMPLING** which is the problem of sampling, either exactly or approximately, from the output distribution of a boson computer. A boson computer is a model for non-universal quantum computation with non-interacting bosons, which can be physically implemented using a linear optical network on m modes, into which we input n identical photons. Four experimental realisations of the **BOSONSAMPLING** problem have been shown [72, 73, 74, 75].

It has been shown that linear optics with adaptive measurements is universal [2] for BQP. However, the boson computer is a quantum computing model that cannot implement a standard quantum algorithm (Shor's factoring algorithm [76], Grover's search algorithm [6], etc) and cannot even do universal *classical* computation. Yet, Aaronson and Arkhipov's result provides evidence that quantum computers have capabilities outside the entire polynomial hierarchy. If a classical **BOSONSAMPLING** algorithm existed, then the polynomial hierarchy would collapse, an event regarded as highly unlikely⁸.

Work by Caianiello [77] first showed that the amplitudes for n -boson processes can be written as the permanents of $n \times n$ matrices. As was first shown by Valiant [78], computing the permanent of a matrix is a complete problem for the class of counting problems associated with NP problems⁹, if an exact efficient classical algorithm existed to solve a problem of this class, this would imply P=NP. He coined the term $\# \mathbf{P}$ to describe this class of problems. The key contribution of Aaronson and Arkhipov is to show that there exists a connection between the ability of classical computers to solve the approximate **BOSONSAMPLING** problem (drawing a sample from a distribution that is close to the actual bosonic distribution) and their ability to approximate the permanent of a random complex matrix¹⁰. If there existed an efficient classical algorithm that solved approximate **BOSONSAMPLING**, this would imply that $P^{\#P} = BPP^{NP}$ and hence the polynomial hierarchy would collapse to the third level.

It is an interesting fact that while **BOSONSAMPLING** is a hard problem and the ability to solve it efficiently with a classical computer would have serious complexity consequences, the same problem formulated for fermions lies in P. While the amplitudes of n -boson processes are given by permanents of $n \times n$ matrices, in the case of fermions they are given by determinants. Despite the similarity of the definitions of permanents and determinant, they are dramatically different in their computational difficulty; the permanent is $\# \mathbf{P}$ -complete while the determinant can be calculated in $O(n^3)$ operations (and is therefore in P)¹¹. However, what is even more remarkable is that the computational difficulty of simulating bosonic and fermionic systems is

⁷See appendix A for a description of all complexity classes mentioned here.

⁸More details of what this would mean can be found in appendix A.

⁹The solution of an NP problem is *whether a solution exists*, whereas the solution to a $\# \mathbf{P}$ problem is *how many solutions exist*.

¹⁰Although polynomial-time algorithms for certain classes of matrix exist [79], an efficient algorithm for general matrices has not been found.

¹¹It should be noted that this does not imply that quantum computers built from bosonic systems have more computational power than those built from fermionic systems. The **BOSONSAMPLING** proposal never uses bosons as qubits, instead it exploits the coherence advantages of a bosonic non-interacting system to build a model which is very computationally expensive to simulate classically.

reversed when using the Monte Carlo method for approximating the ground state of a many-body system. In a bosonic system, the ground state is easy to approximate, while in fermionic systems, the cancellations between positive and negative terms (what is known as “the sign problem”) make the ground state very hard to approximate.

2.3 Integrated optics

Politi *et al.* [56] were the first to implement quantum linear optics on chip. Prior to this work, quantum experiments in linear optics were made using bulk optics: large scale components such as beam-splitters and mirrors attached to optical benches, where photons were transmitted through air (or occasionally via fibre) across the optical network. Bulk optics is an inherently non-scalable scheme and not sufficiently reliable on a large scale. The development of photonic waveguide technology [56] permitted the development of optical circuits in which the stability and control over the optical path length is highly amplified with respect to what can be obtained in bulk optics, with the added advantage that the circuit size is dramatically reduced. In these circuits, quantum information is predominantly encoded in the path degree of freedom of the photon, although polarisation encoding has also been demonstrated [80].

Beam-splitters are implemented in integrated photonics via directional couplers, which are realised by bringing together the different waveguides close enough so that the evanescent field of one of them can couple into the other one. By controlling the separation between the waveguides or the length of the coupling region, different split ratios can be obtained and therefore this implementation is entirely equivalent to a beam-splitter in bulk optics. Figure 2.6 shows the implementation of a directional coupler, its schematic diagram in path encoding and a simulation of the coupling of the path via the evanescent field.

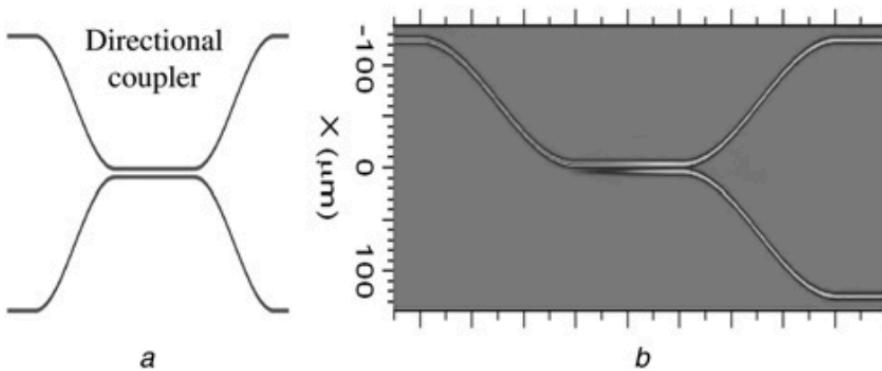


Figure 2.6: Directional coupler, used in integrated photonics to implement the beam-splitting operation: a. Schematic diagram of the directional coupler. b. Optical field propagation simulation with vacuum in the bottom input port. Figure from [81], copyright (2011) by the IET.

Variable phase-shifters can be implemented in integrated circuits, providing complete control over the phase of the qubit. For the silica-on-silicon materials used by the Bristol Photonics group [81], the easiest way to implement reconfigurable phase-shifters is by using thermo-optical switches. The thermo-optical switches use resistive elements fabricated on the surface of the

chip, that via the thermo-optical effect can provide a change of the refractive index n by raising the temperature of the waveguide structure by an amount of the order of $10^{-5}K$. The main drawback of the thermo-optical switches is their slow reconfiguration times of milliseconds. New fast-switches have been proposed in different material systems for integrated optics, such as lithium-niobate [82, 83] which supports an electro-optic effect. Such switches have reconfiguration times of the order of nanoseconds. In appendix G we give a table comparing the switching speed and switching loss of different state-of-the-art switches compatible with integrated optics.

Beam-splitters with variable reflectivity can be achieved by combining directional couplers and variable phase-shifters in a MZI configuration [65] (see figure 2.7). Complete control over the relative phase and amplitude of the two paths in dual rail encoding can be achieved. This MZI will act as a beam-splitter with variable reflectivity and is the basic building block to construct more complicated networks on chip. The recent experimental realisation of the Reck scheme [43] uses an array of these elements to perform a general unitary on N modes.

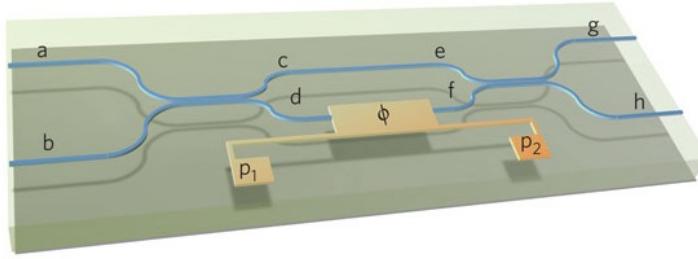


Figure 2.7: MZI constructed in integrated optics with two directional couplers and a variable phase-shifter. Figure from [81], copyright (2011) by the IET.

Quantum interference can be studied with this device by looking at the visibility of the Hong-Ou-Mandel (HOM) [69] dip in a two-photon experiment. The indistinguishability and purity of two single-photon states can be assessed by making them interfere at a beam-splitter. By changing the relative delay of the photons in arriving at the beam-splitter, a dip in the rate of detecting one photon at each output of the beam-splitter can be observed in the case of indistinguishable photons when the delay is near zero, i.e. when both photons arrive simultaneously at the beam-splitter. This study was done by Matthews *et al.* [57], where they changed the relative delay by continuously varying the relative phase ϕ between the two paths, their results can be seen in figure 2.8. The visibility of the HOM dip is plotted against the phase shift, showing both the theoretical fit (solid line) and experimental data. There are two insets that show the HOM dip with a high visibility of $98.2 \pm 0.9\%$ for a reflectivity of $48.4 \pm 0.5\%$ (left) and a low visibility of $12.9 \pm 0.9\%$ for a reflectivity of $94.1 \pm 0.2\%$ (right) visibility.

Integrated photonics significantly reduces the difficulty of realising optical experiments. The complicated process of designing and implementing sophisticated interferometers in bulk optics becomes much easier as the optical networks can be etched onto a chip that needs no further alignment once made. Integrated implementations of the key components of photonic circuits have been realised, obtaining very high fidelity results [56] that show the potential of this type of implementation for linear optics. The miniaturisation and scaling of the optical circuits allows for the creation of a modular architecture in which to perform quantum computation.

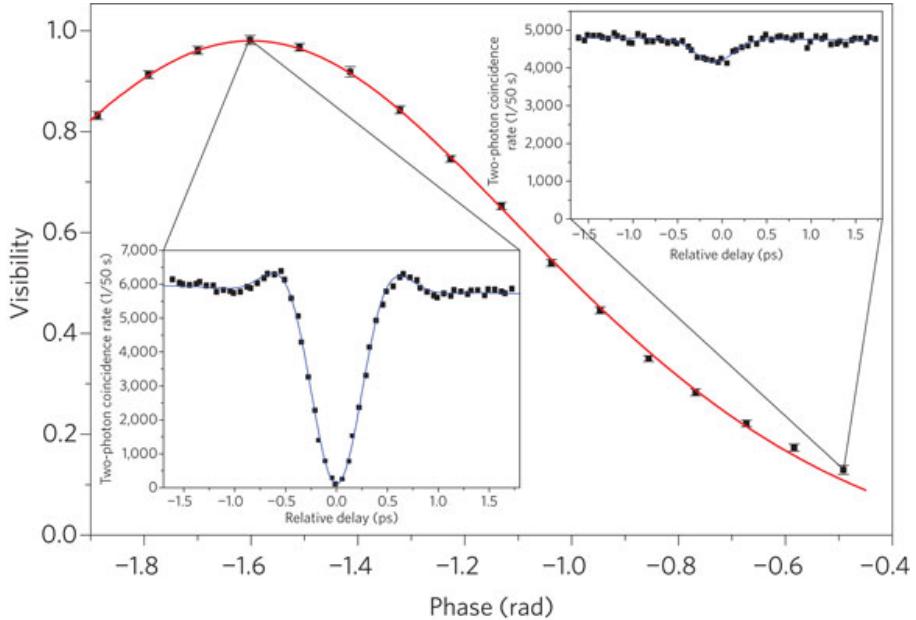


Figure 2.8: Visibility of the Hong-Ou-Mandel experiment. We can observe how coincidences disappear as the photons are made temporally indistinguishable. Figure from [57], copyright (2009) by Macmillan Publishers Limited.

A modular technology for the quantum computer will be key for boosting its performance and allow for hardware reconfiguration. In chapter 5 we described a linear optical architecture proposal which has been created with integrated photonics in mind.

2.4 First proposals for Linear Optical Quantum Computing

In previous sections we have reviewed the basic operations of a linear optical system and showed how it is not possible to create a deterministic entangling gate strictly using these operations. In fact, prior to the work of Knill, Laflamme and Milburn (KLM), it was believed that in order to achieve a two qubit gate, it was necessary to use a non-linear material for any practical implementation of optical quantum computing. A CZ entangling gate could be achieved by using a cross-Kerr non-linear material [63]. Interaction of the photons with this material for long enough time, would implement a phase-shift dependent on the state of the photons. However, it was found [84] physically impossible to create cross-Kerr non-linearities large enough while keeping the noise levels low enough for performing quantum information tasks.

The alternative, proposed in the KLM [2] paper, was to artificially create the necessary non-linearity by using measurement. This measurement-induced non-linearity [85] yields non-deterministic entangling operations, which would in principle not seem sufficient to be able to perform large computations. However, a series of proposals have shown how a full linear-optical quantum computer can be constructed with a polynomial number of resources.

2.4.1 The Knill, Laflamme & Milburn protocol

In their seminal paper of 2001, Knill, Laflamme and Milburn (KLM) showed [2] that optical quantum computation was possible using only linear optical elements (beam-splitters, phase sifters, single photon sources and photo-detectors with feedforward) if one allowed for probabilistic gates. The probabilistic two-qubit gate uses the quantum interference of ancillary photons at a beam-splitter and single-photon detection to induce interactions non-deterministically, and then, after the measurement of the ancillary photons has been performed, one can know if the probabilistic gate has succeeded according to the detection pattern obtained. To achieve a deterministic gate the success probability of the non-deterministic gate can be boosted by using teleportation.

KLM's scheme to perform a 2-qubit entangling gate in a near deterministic fashion is a combination of 3 ideas:

- Using elements of linear optics, perform a non-deterministic entangling gate on two logical qubits, with high probability of failure. This gate, shown in figure 2.10 implements a conditional sign flip (CS) by combining two Non-linear phase Shifts (NS), shown in figure 2.9, one in each mode, with overall success probability of 6.25%.

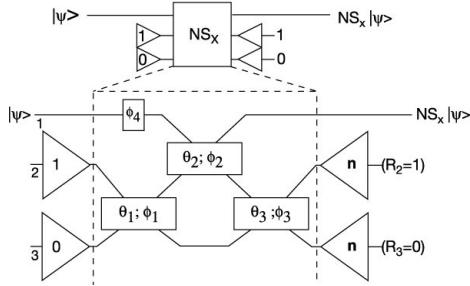


Figure 2.9: Non linear phase shift (NS) on a single mode. Figure from [2], copyright (2001) by Macmillan Publishers Limited.

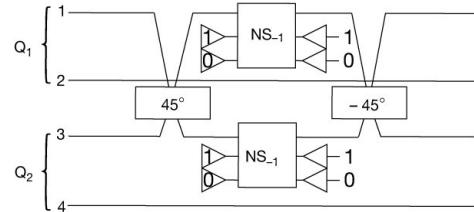


Figure 2.10: Conditional sign flip (CS). Figure from [2], copyright (2001) by Macmillan Publishers Limited.

- Improve the performance of this non-deterministic entangling gate by combining it with quantum teleportation, obtaining a entangling gate that succeeds with probability $p = \frac{n^2}{(n+1)^2}$, where n is an integer corresponding to the number of photons in a large ancillary entangled state, created and stored “offline”. Increasing values of n correspond to increasingly complicated teleportation circuits. In figure 2.11 a CS gate with boosted probability is shown.
- Boost the success probability even further by using quantum error correcting codes, until the gate is near deterministic, allowing for scalable quantum computation. When the teleportation gate fails, it has the effect of acting as a Z measurement with a known outcome on the qubit which was going to be teleported. Therefore, using error correcting codes that protect qubits against the effects of Z measurements improves the probability of a successful teleportation. An example of such an encoding is shown in figure 2.12. The logical qubit can always be recovered as long as the measurement result and the qubit that was measured are both known.

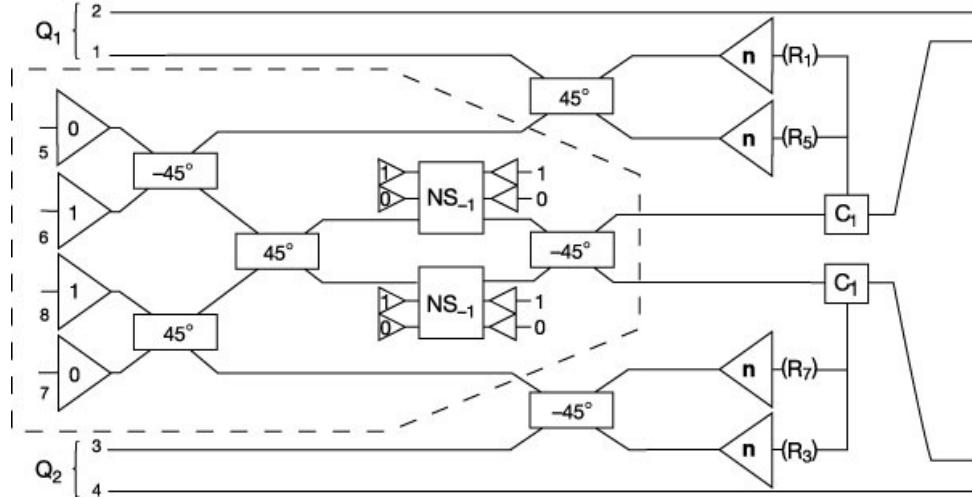


Figure 2.11: Conditional phase shift with probability boosted to 25%. Figure from [2], copyright (2001) by Macmillan Publishers Limited.

$$\begin{aligned} |0\rangle &\rightarrow \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \\ |1\rangle &\rightarrow \frac{1}{\sqrt{2}} (|01\rangle + |10\rangle) \end{aligned}$$

Figure 2.12: Encoding to protect against Z measurements.

This scheme, however, requires tens of thousands of optical elements per entangling gate to achieve a high probability of success. And furthermore, the teleportation circuits require the use of entangled states, the higher the success probability of the teleportation circuit (which is a function of n), the bigger the entangled state required (the number of photons required in the entangled state scales as $4n$). For example, the gate shown in figure 2.11 has $n = 1$ and therefore requires a 4 photon GHZ state. In the estimates quoted in their proposal [2], in order to be able to perform a gate with 95% or higher success rate, they require “300 successful $CZ_{9/16}$ gates per logical two qubit gate”. Translating this estimate into the number of Bell pairs¹² required to perform a single successful logical gate, this scheme requires $6.014 \cdot 10^6$ Bell pairs per single logical two qubit operation. It is clear that, despite being theoretically scalable, further improvements were required to make this scheme experimentally feasible.

Improvements on KLM

It is necessary to mention that following the KLM proposal [2] for a full linear-optical quantum computer, several protocols were proposed with slight variations and improved resource requirements:

- Franson *et al.* [86] : New teleportation circuits were proposed, which boosted the prob-

¹²To estimate the Bell Pair equivalent of the multi-photon entangled states required, we have used the most efficient theoretical linear optical circuits, shown in chapter 4. Full resource calculations for all schemes in this chapter are shown in appendix B.

ability of successfully teleporting a qubit to $1 - \frac{1}{n^2}$. However this was at the expense of having a failure mode that effectively changed the balance of the coefficients of the state, instead of just applying a Pauli gate, which made the error correction much harder.

- Spedalieri *et al.* [87] : This scheme is based on the redefinition of the teleported state, which they take from single-rail to a dual-rail qubit. This small change allows for simple error detection in the teleported gates.

2.4.2 The Yoran & Reznik protocol

Building on ideas from the KLM proposal, in 2003 Yoran and Reznik proposed a new scheme [88] that reduced the resources required per logical gate. Their idea, although unrelated with the MBQC model [33], shares the same concept of pre-preparing states of multiple entangled photons. The structure of these states is dictated by the form of the quantum circuit that one wishes to implement (similar to how the cluster state is shaped using Z measurements prior to performing the computation). They introduce the chain state, shown in figure 2.13, which manifests maximal pairwise entanglement.

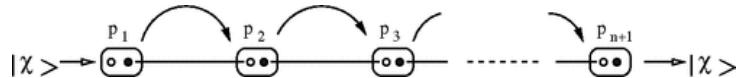


Figure 2.13: Schematic description of a chain state of $n + 1$ photons. Figure from [88], copyright (2003) by the APS.

Each rectangle in figure 2.13 represents a single photon with its polarisation and path degrees of freedom represented respectively by the empty and full circles¹³. The path degree of freedom of each photon in the chain is maximally entangled with the polarisation degree of freedom of the next photon in the chain. Each one of these chain states, represents the world line of a single photon, meaning that all the operations that are sequentially applied on a quantum circuit, in this representation would each be applied to a different photon of the chain. A state can be fully teleported from one end of the chain to the other, and any single qubit operation can be similarly applied using one of these chains. In order to perform a computation, the relevant gates are first applied to different chains, and the input states are subsequently teleported in. In figure 2.14 and 2.15 we can see how a three qubit computation can be translated into a linked state. Once this state has been built, the three inputs are teleported through the state.

The construction of the states is done step-by-step and they require that the gates used to build the linked state have a probability larger than $1/2$ for the combined process of link/gate generation. This restricts them to using the teleported CZ gates proposed in [2] with $n \geq 3$.

One last trick they propose is to add inert links to each chain. The inert links will be photons onto which no gate is applied. For each qubit (a chain in figure 2.15) that takes part in n two-qubit gates (denoted by G_{ij} in figure 2.15), a chain of $2n$ links is constructed and the

¹³It is worth noting that this entire scheme can be equally understood in terms of path degree of freedom alone, with four possible modes per photon.

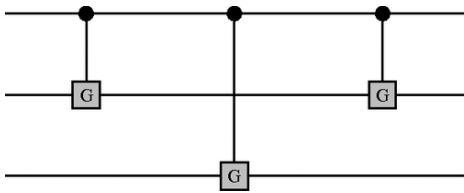


Figure 2.14: Three qubit circuit implemented by the photonic states in figure 2.15. Figure from [88], copyright (2003) by the APS.

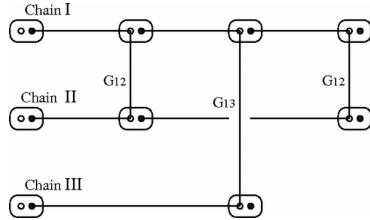


Figure 2.15: Linked photonic state needed for generating the circuit in figure 2.14. Figure from [88], copyright (2003) by the APS.

gates are applied to every second link. The purpose of the added inert links is to prevent the backwards spread of the failure of gates, avoiding the destruction of gates and links that had previously been constructed. This allows them to reduce the number of $CZ_{9/16}$ gates by an order of magnitude (from about 220 to 23).

The reduction in the number of elementary operations is dramatic. KLM [2] required 300 successful applications of their $CZ_{9/16}$ per logical gate (that had an intrinsic 5% error), whereas this proposal only requires five successful applications of $CZ_{9/16}$ KLM gate per logical two qubit gate. They estimate that for the successful application of these gates, they require “ $\sim 23 CZ_{9/16}$ applications on average for every logic gate”. However, to use the trick of having additional inert links, we must also include the application of three $CZ_{4/9}$ per gate. Translating the number of gates into number of Bell required to perform a single logical gate we have that this scheme requires $2.9 \cdot 10^5$ Bell pairs on average per logic gate. This is an improvement of an order of magnitude with respect to KLM, but it still consumes too many Bell pairs per logic gate to have a feasible implementation.

2.5 Optical Quantum Computation with Cluster States

As photons don't interact, entangling operations are very challenging in a full linear optical scheme, and only probabilistic gates are possible¹⁴. The challenge would be diminished if these gates were allowed to fail, and the procedure could be repeated until all the necessary entanglement had been created. The cluster state model (presented in chapter 1) poses an alternative to the circuit model for systems that have unreliable or non-deterministic two-qubit operations. All the entanglement needed is created when preparing the resource state, offline. The actual quantum computation is done by performing a series of reliable single qubit measurements. Therefore this model for quantum computation is particularly well suited to linear optics. It is then understandable that great improvements in resource requirements were achieved by tailoring the protocols to this model.

¹⁴It was recently shown that the success probability can be taken to unity at the expense of consuming infinite multi-photon entangled states, see chapter 4.

2.5.1 The Nielsen protocol

In 2004, Nielsen [89] proposed a scheme for LOQC combining optical quantum computation with linear optics and cluster states. He used the same ideas as KLM for the entangling gates (both schemes use CZ gates with probability $\frac{n^2}{(n+1)^2}$), except that he didn't require the use of error correcting codes to overcome non-determinism. His scheme results in logical gates that work deterministically as opposed to the 5% error experimented by KLM's entangling gates.

The idea he proposes is to build up the cluster state by non-deterministically adding extra qubits to the cluster state using $CZ_{4/9}$ and $CZ_{1/4}$ gates and, once this is achieved, perform the rest of the operations according to the KLM scheme. The cluster is built by attempting to add a site connected by a single bond (with success probability $p = \frac{2}{3}$ after teleportation) or attempting to add a site connected by a double bond (with success probability of $p = \frac{4}{9}$, as two single successful bonds are required). Figure 2.16 shows the procedure of growing a cluster state by adding a new qubit, S , to it. If this qubit is only linked to qubit B (single bond), we will have a success probability of $p = \frac{2}{3}$, while if this qubit has bonds to both qubits A and B (double bond), we will have a success probability of $p = \frac{4}{9}$. It can be calculated that for every two attempts of adding a site to the cluster, the average number of sites added is $\frac{2}{9}$. The key idea is that failure of a gate does not destroy the cluster state, it only removes a single qubit from it.

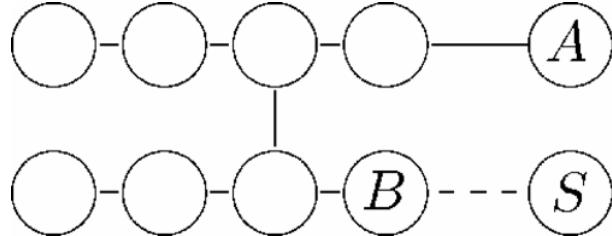


Figure 2.16: Cluster state construction. A new site S is added, if it only has a bond to qubit B the process succeeds with probability $p = 2/3$, but if it has two bonds, one to qubits B and one to qubit A , the process succeeds with probability $p = 4/9$. Figure from [89], copyright (2004) by the APS.

In terms of the resources required to simulate a standard quantum circuit, Nielsen concludes that the fairest estimate needs 24 successful $CZ_{4/9}$ gates per entangling gate. Each of these gates is composed of 70 beam-splitters, 30 photo-detectors and 12 single-photon preparations, which means that the number of elements required per single entangling gate is $O(10^2)$, which is an improvement of 2 orders of magnitude over the KLM proposal. However, Nielsen's proposal is still very expensive in terms of the number of Bell pairs required per single logical entangling gate. Requiring 24 successful $CZ_{4/9}$ per logical entangling gate means a requirement of 54 8-photon entangled states, which on average need $1.075 \cdot 10^4$ Bell pairs for their preparation. It is, again, an improvement of 2 orders of magnitude from KLM's requirements, but it is still a very expensive logical entangling gate.

2.5.2 The Browne & Rudolph protocol

In 2005 Browne and Rudolph [90] followed up on Nielsen's ideas by introducing two so-called fusion mechanisms that allowed the construction of entangled photonic states (cluster states). Their proposal has the advantage that they don't require photon-number discriminating detectors (for one of the gates) or elaborate interferometers with multiple beam-splitters in series. Also, there is a key difference in the type of interference used for the entangling gates. While KLM, Yoran and Reznik's and Nielsen's approach rely on Mach-Zehnder-type interference [91], Browne and Rudolph only make use of the HOM coincidence form [69], therefore only requiring stability over the coherence length of the photons and not needing to maintain phase stability of the interferometer.

The main resources used are two-photon polarisation-entangled Bell states that can be obtained via linear optics and photodetection with success probability $p = \frac{3}{16}$ from four single photons¹⁵ [92]. They introduce two fusion gates, Type-I and Type-II, which are shown in figure 2.17.

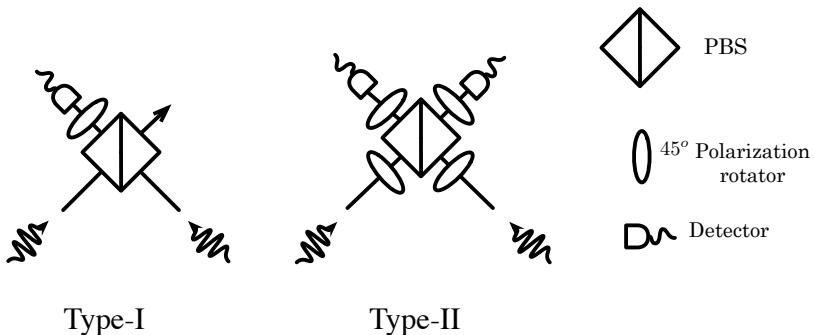


Figure 2.17: Type-I and Type-II fusion gates. Their action of cluster states is depicted in figures 2.18 and figure 2.19. Figure (modified) from [90], copyright (2005) by the APS.

The Type-I fusion gate takes two spatial modes and mixes them at a PBS, then it rotates the polarisation degree of freedom of one of the output modes by 45° and measures it with a polarisation discriminating photon counter. The gate succeeds when only one polarised (either H or V) photon is detected (which happens 50% of the time) and fails if zero or two photons are detected. When the gate succeeds, the two separate qubits become fused in a single qubit that inherits all the bonds from the input qubits, but when it fails, the gate has the effect of measuring both qubits in the computational basis (see figure 2.18). We can map the success and failure outcomes of this gate to two different evolutions of the qubits in the cluster state. When this gate succeeds, the unitary operation is a CNOT gate performed between both qubits followed by a measurement of the target qubit in the computational basis, while if the gate fails, the evolution would be measurements in the computational basis. The failure outcome would split any cluster we are trying to build, and therefore this is not an optimal gate. In addition, this gate is not protected against photon loss. As we only measure one of the modes, it could be

¹⁵In fact this probability can be boosted to $p = \frac{1}{4}$ by using an extra switch and a correction linear optical circuit [92].

the case that the gate fails but one of the qubits is lost and leading to the incorrect assumption that the gate has succeeded. This can introduce a Pauli error in our computation [93].

Browne and Rudolph introduce the use of “redundant encoding”, whereby a single qubit on the cluster is represented by multiple photons: $|\phi_0\rangle|0\rangle + |\phi_1\rangle|1\rangle \rightarrow |\phi_0\rangle|0\rangle^{\otimes n} + |\phi_1\rangle|1\rangle^{\otimes n}$. A Pauli X measurement on the redundant photons would not split the cluster but would rather remove the photon measured from the redundant encoding and combine the adjacent qubits into one single qubit that inherits the bonds of the input qubits, maybe adding a phase. Type-II fusion gates are an evolution of Type-I gates that make use of this redundant encoding to generate two-dimensional clusters. They take two input modes and rotate each of them by 45° before mixing them in a PBS. Then, they measure them in the rotated basis. This gate is successfully applied to a single photon of each of a pair of logical qubits when a single photon is detected at each detector, its effect is to project the pair of logical qubits into a maximally entangled state ($|\phi^\pm\rangle$). When the gate fails (as heralded by zero or two photons in one of the modes), it performs a measurement on the X basis on each of the photons, removing them from the redundant encoding (but not destroying the logical qubit).

Cluster states can now be efficiently constructed using these two gates, as can be seen in figures 2.18 and 2.19. First, linear clusters can be constructed by using Type-I fusion gates (see figure 2.18), and higher dimensionality can be achieved by fusing linear clusters with Type-II gates (see figure 2.19). In fact, by using redundant encoding, all gate operations could be made with the Type-II fusion gate. This has the advantage of not needing photon-number-discriminating detectors and naturally detecting loss errors. However, as two photons are measured in this gate, Bell states are not a sufficient resource for building a cluster (a Type-II fusion gate applied to a pair of Bell states would produce another Bell state if successful) and one would have to use three-photon clusters instead, which increases the resource requirements but still keeps them below previous proposals.

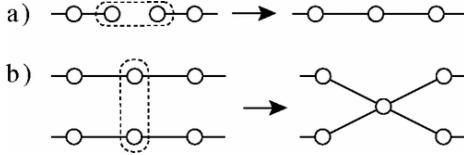


Figure 2.18: Successful action of Type-I fusion gate. Figure from [90], copyright (2005) by the APS.

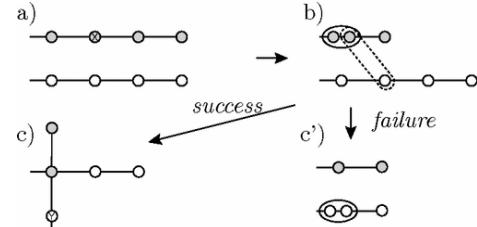


Figure 2.19: Action of Type-II fusion gate. Figure from [90], copyright (2005) by the APS.

This proposal significantly reduces the number of linear optical elements required to perform a single successful entangling gate. It moves away from the complexity of the KLM CZ gates with multiple teleportation procedures and replaces them with a maximum of 4 polarisation rotators and one PBS, plus two detectors, which are not always required to be number-detecting. But the savings are not only in the number of linear optical elements. This scheme requires on average only 52 Bell pairs to implement a two-qubit logical gate. This is a two orders of magnitude saving with respect to Nielsen’s proposal, and therefore substantially decreases the complexity of realising LOQC in practice.

2.6 Parity-encoded optical quantum computing

A crucial feature of the KLM teleported gates [2] is that their failure mode implements a Z measurement on the qubit that was to be teleported. Protecting against such measurements, as proposed in [2] can significantly improve the success probability of the computation. In the parity encoded schemes proposed in [94, 95], an incremental approach to the encoding is proposed, simplifying the process of gate attempts and recovery, and outlining procedures to performing a universal set of gates on the encoded qubits.

2.6.1 The Hayes, Gilchrist, Myers & Ralph protocol

This proposal [94] is a variation of the KLM protocol, in which they make use of an incremental approach to error correction in order to boost probability of success. They use the teleported gates introduced in [2] with qubits encoded to protect against teleportation failures and computational basis measurements. Their logical unencoded qubits correspond to the polarisation states of a single photon. The proposed encoding follows the original KLM encoding against teleporter failures (see figure 2.12), where the logical $|0\rangle$ corresponds to a state with even parity and the logical $|1\rangle$ corresponds to an odd parity state. Hence the name of *parity encoding*.

$$\begin{aligned} |0\rangle^{(n)} &\equiv \frac{|+\rangle^{\otimes n} + |-\rangle^{\otimes n}}{\sqrt{2}} \equiv |\text{even}\rangle^{(n)} \\ |1\rangle^{(n)} &\equiv \frac{|+\rangle^{\otimes n} - |-\rangle^{\otimes n}}{\sqrt{2}} \equiv |\text{odd}\rangle^{(n)} \end{aligned}$$

Figure 2.20: Parity encoding. The index n represents the number of component qubits on which the logical state is encoded.

This encoding of the logical state will protect it against computational basis measurements on any of the physical qubits, as any such measurement will only reduce by one the level of encoding. This can be easily explained by noting that $\langle 0|\psi^{(n)}\rangle = |\psi^{(n-1)}\rangle$ and $X\langle 1|\psi^{(n)}\rangle = |\psi^{(n-1)}\rangle$, where X can be acting on any single qubit. Therefore, a Z -measurement followed by the conditional application of an X gate leaves the logical qubit in the correct encoded states, but encoded in one less physical qubit than before.

In this paper they propose an incremental encoding scheme instead of using the concatenated approach presented in [2]. Component qubits are added to the encoded state incrementally, and whenever a Z -measurement occurs, the component qubit that has been removed from the state can be replaced by using a non-deterministic encoding circuit. The procedure for applying the logical gate follows a repeat-until-success strategy, and the success of the computation reduces to maintaining an appropriate level of encoding throughout the computation.

To be able to perform universal quantum computation, they need to be able to implement gates from the set $\{X_\theta, Z, CNOT, Z_{\pi/2}\}$, where $X_\theta = \cos \frac{\theta}{2} \mathbf{1} + i \sin \cos \frac{\theta}{2} X$. Gates from the set $\{X_\theta, Z\}$ can be performed easily on the logical qubit. The logical Z can be performed by applying a Z gate to all physical qubits, while the X_θ can be performed by applying the rotation

to any one of the physical qubits. The gates $\{CNOT, Z_{\pi/2}\}$ can be performed more efficiently by applying re-encoding circuits to a subset of component qubits that have had the desired operation performed on them. A CNOT is performed by first applying a CNOT between a pair of component qubits corresponding to different logical qubits and then encoding the state until it has reached the size of the original state. In the event of teleportation failures, the entire subset of component qubits will be lost and the CNOT should be reattempted again on another qubit. Once the re-encoding has been achieved successfully, the unaltered qubits can be measured out. Similarly, the $Z_{\pi/2}$ gate is performed by suitably rotating a component qubit and encoding from the rotated qubit until the subset again reaches the size of the original state.

The authors calculate that in order to obtain an encoded CNOT with 95% success probability, they require on average 90 physical CS gates and 32 elimination circuits. This is a significant improvement over the 1000 elimination circuits and less than 2250 CS circuits required to obtain the same encoded gate using the original KLM proposal. In terms of the entanglement consumed, we can calculate that they would require $1.92 \cdot 10^3$ Bell pairs per single encoded logical gate. This is an improvement of one order of magnitude over Nielsen's scheme [89], which was the best proposed scheme at the time (2004).

2.6.2 The Gilchrist, Hayes & Ralph protocol

An improvement on the parity-encoded scheme [94], can be achieved by using the fusion gates proposed in [90]. The new scheme presented in [95] makes use of some features of cluster state computation, but essentially retains the KLM circuit-based approach.

The linear optical gates used for encoding and building the resource state for teleportation are the fusion gates Type-II and Type-I. A rotated version of the Type-II gate (without the input wave-plates) can be used to add n qubits to an encoded state $|\psi\rangle^{(m)}$ by fusing it to a resource state $|0\rangle^{(n+2)}$. If the gate succeeds, the logical state will be encoded in $m + n$ physical qubits, whereas if the gate fails the outcome will be the product state $|\psi\rangle^{(m-1)}|0\rangle^{(n+1)}$ on which the gate might be attempted again. The Type-I gate, combined with a Hadamard operation, can also be used to build up the resource state. This approach has the advantage over using the Type-II, that it only loses one qubit for both input states. However it has the big disadvantage that whenever it fails, it completely destroys all entanglement in both input states. A combination of both these approaches is deemed the most resource efficient [95].

Gates from the set $\{X_\theta, Z\}$ can be performed deterministically on a logical qubit in the same manner as in the previous version of the parity encoded scheme [94]. The $Z_{\pi/2}$ and CNOT gates are performed non-deterministically on the encoded qubits. It is important to note the difference with the previous version of this scheme, where gates are performed on a subset of component qubits that are then re-encoded. The $Z_{\pi/2}$ gate is performed by first applying the $Z_{\pi/2}$ gate to one of the physical qubits and then fusing the logical state to a pre-prepared resource. The CNOT gate is performed by first entangling the control and target states to a resource state (using one Type-I and one Type-II fusion gates) and subsequently measuring the remaining qubits of the control qubit in the computational basis. Depending on the parity of the result, a bit flip might need to be applied. A circuit diagram of both these operations can be found in figure 2.21.

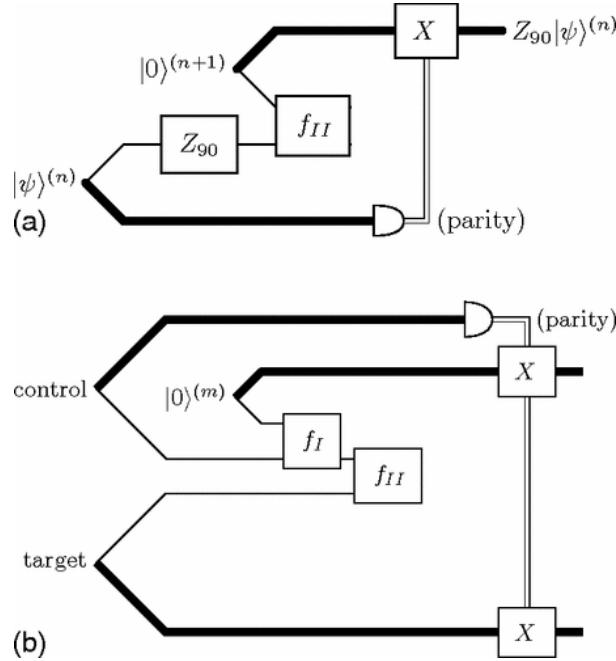


Figure 2.21: a) Implementation of a Z_{90} ($= Z_{\pi/2}$) gate on the encoded parity state. b) Implementation of a CNOT gate on the encoded parity state. The Type-I and Type-II fusion gates are denoted by f_I and f_{II} respectively. The thickness of the lines represent the encoding level. Figure from [95], copyright (2007) by the APS.

The resource consumption of this scheme depends heavily on the level of encoding chosen. It is necessary to estimate the optimal level of encoding, as if the level is too low there is a significant probability that the logical state might be destroyed after a series of failures. However, increasing the level of encoding is costly and might incur in unnecessary resource waste. The authors compute the level of encoding and resources required to achieve an encoded CNOT gate with different success probabilities. In order to have the fairest comparison possible with previous schemes, we chose to compare the resources consumed by the CNOT gate with 96.4% success probability, as it is the closest to the 95% CNOT used in other schemes. For this gate, logical qubits require 6 levels of encoding, for which $1.84 \cdot 10^3$ Bell pairs are required. We can see that this is a slight improvement on the resources needed in the previous parity encoded scheme, with the advantage that higher success probabilities can be achieved for the entangling gates.

2.7 Percolation-based Linear Optical Quantum Computing

All the schemes mentioned so far have one thing in common: they all rely on a repeat-until-success strategy. This means that each step in the computation is going to be repeated until the successful gate is achieved. This strategy, however, implies that the procedure does not have a fixed physical depth. Fixed physical depth is *necessary*, as in a circuit with unbounded depth photons will have to be constantly re-routed, requiring a large network of switches. For current technologies photon loss remains a significant challenge, and it is the switching networks that mostly contribute to photon loss, with about one order of magnitude more loss than any other

component [96, 97, 98]. Furthermore, each photon will have a different error rate, which will depend on the number of linear optical elements and switches it has travelled through. This inhomogeneity in the error rates has an impact on the error and loss correcting codes that can be applied, making it potentially extremely challenging. The fusion gates have a 50% probability of failure (which is heralded by detection patterns in the photon detectors) and to keep track of that, the design would have to implement large amounts of active-switching type of feedforward, that would allow for the quantum system to be routed into different configurations depending on the success or failure of the entangling gates in order to build the intended photonic state. This obstacle of active feedforward can be overcome by using results of percolation theory, as proposed in [99].

The proposed way of constructing a cluster state in percolation based models [99] is by integrating together, via the fusion gates, smaller clusters of entangled photons (micro-clusters), that are more easily produced. What makes percolation models so distinct from previous strategies is that they propose a *ballistic* construction of the cluster: the physical-layer entangling gates are attempted once and only once. With the exception of the very last reconfigurable measurement needed for MBQC, the operations that one photon will go through are completely mapped before hand and no control needs to be applied. Percolation-based cluster state schemes are applicable to any physical system with a probabilistic component either in the presence of qubits or in the implementation of entangling gates. These would respectively correspond to *site* or *bond* percolations models. Examples representative of *site* percolation are Mott hole effects and optical lattices [100], while *bond* percolation instances are found in atoms in optical cavities [101] and photonic systems [102]. These percolation models, however, assume the ability to create small entangled states (GHZ states) on demand. This is a stringent demand, and for linear optics it is still work in progress.

2.7.1 Percolation theory

The phenomenon of percolation is studied in classical statistical mechanics and concerns the behaviour of connected clusters in graphs that have lost some of their bonds (and/or sites) due to a randomised process occurring with a probability p . In figures 2.22, 2.23 and 2.24 we can see an example of such randomised process and the different regimes that exist depending on the value of p . In this example we demonstrate site percolation, where in a regular rectangular lattice, each site is coloured with probability p and left blank with probability $1 - p$. We can see that for low values of p the lattice is almost all fully occupied by the empty sites, whereas for high values of p , almost all sites are occupied and connected in a giant cluster that spans the entire lattice.

At the centre of percolation theory is the idea of a percolation threshold p_c which dictates the global properties of the lattice. The size of the connected components in the lattice will depend almost exclusively on the value of p with respect to the critical value p_c . We can differentiate two distinct phases in the percolation, sub-critical $p < p_c$ and super-critical $p > p_c$, shown in figures 2.22 and 2.24 respectively. In the case of small lattices such as the one presented as an example, the transition between phases is smooth due to boundary effects, however, for infinite lattices, there are two very distinct phases of the system differentiated by a clear phase

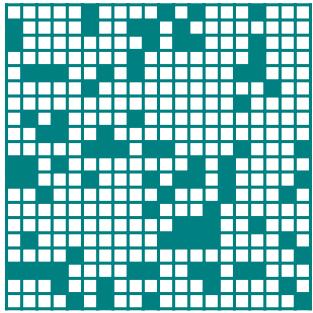


Figure 2.22: $p = 0.20$. Sub-critical regime.

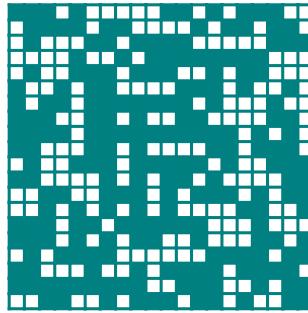


Figure 2.23: $p = 0.55$. Critical regime.

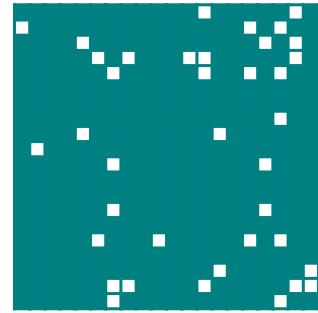


Figure 2.24: $p = 0.90$. Super-critical regime.

transition.

The main result in percolation theory is that above the percolation threshold, there always exists an infinite crossing cluster that spans the entire lattice. For $p < p_c$, all clusters in the lattice are finite and the biggest connected component has a size that scales as $O(\log N)$ [103], N being the linear dimension of the lattice. For $p > p_c$, the size of the biggest connected component scales as $O(N)$. This fact can be used to easily assess if a particular instance of a percolation problem is in the sub-critical or super-critical regime when the percolation threshold is unknown. Another important result which is closely related is that the correlation length of the lattice is finite in the subcritical regime, so there is an exponential decay of correlations, whereas in the supercritical regime the correlation length is infinite. In figure 2.25 we present the two most significant signatures of percolation.

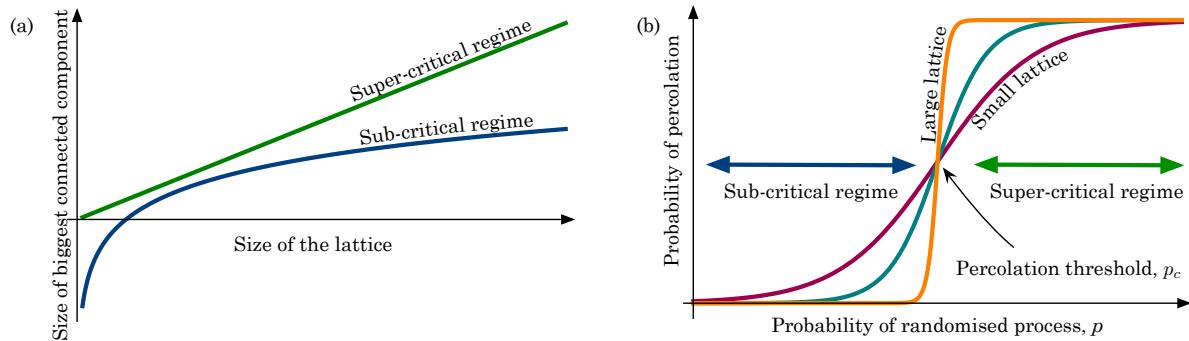


Figure 2.25: Most significant percolation signatures. (a) Scaling of the biggest connected component as a function of the percolated lattice size. This biggest component scales as $\log N$ where N is the size of the lattice when we are in the sub-critical regime, but it scales as $O(N)$ in the supercritical regime, where it becomes a spanning cluster that covers most of the lattice. (b) Probability of percolation as a function of p . The percolation threshold marks a phase transition between the sub-critical and super-critical phases.

In the context of one-way LOQC, the regular lattice corresponds to a highly-connected cluster state, whose bonds/sites are effectively removed due to failure of probabilistic entangling gates together with photon loss. The percolation threshold marks a phase transition in the computational power of the resource state generated [104, 105], which, assuming $\text{BQP} \neq \text{BPP}$, distinguishes the states that can be used for universal quantum computation from those which

cannot. It has been proven [105] that the entanglement width [37, 38] of the cluster state can be used as an order parameter in the phase transition in computational power. Only lattices in the super-critical regime have an infinite correlation length, which therefore implies that the requirement on the scaling of entanglement [37, 38] mentioned earlier is satisfied in this regime. The sub-critical regime cannot yield a universal resource and, in fact, can be simulated classically [104, 105].

A key point is that in the case of these engineered systems, the randomised process that creates a percolated lattice does not relate to the classical probabilistic parameters of a physical model, but rather it arises from the statistical character of measurement in quantum theory.

It is also important to note that the statistical nature of the gates applied might not only affect the existence of a bond between the site qubits, but also the existence of the sites themselves, and therefore the model might not be pure *site/bond* percolation model¹⁶. There exist mixed *site-bond* percolation models which are a natural generalisation from the pure percolation models. In these generalisation, sites and bonds are allowed to be randomly occupied with different probabilities (p_s and p_b respectively), and it reduces to the pure model when $p_s = p_b$. The percolation threshold is now a critical curve in the plane (p_s, p_b) . The general shape of this curve is not yet known. Different proposals have been put forward [106, 107], which match very well some lattices, however the extensive numerical results shown in [108] show that a relationship describing all lattices has not yet been found.

2.7.2 The Kieling, Rudolph & Eisert protocol

The protocol that Kieling, Rudolph and Eisert [99] put forward proposed a change of paradigm for LOQC. All previous proposals had assumed large amounts of active switching, where the quantum systems were re-routed at will into different possible coherent interactions with other systems in order to cope with the probabilistic nature of the linear optical gates. In this proposal [99], it was shown that it was possible to eliminate all active feedforward once initial pieces of cluster state had been obtained. These pieces were used as the building blocks of the cluster state, using fusion gates [90] in order to obtain correlations between them.

The building procedure is *ballistic*, meaning that it is only attempted once. Small clusters of photonic states are fused together and the resulting lattice is processed forward without any re-routing of the quantum states. As the fusion operations have a success probability of 50%, the cluster that is obtained by this procedure has missing links, it is an instance of bond percolation in the chosen lattice. It is worth remembering that the probability of missing links is directly related to the efficiency of performing Bell measurements in linear optics¹⁷.

The proposed way to deal with the randomness of the lattice is to coarse-grain the underlying percolated lattice into a logical lattice where logical qubits correspond to blocks of the percolated lattice, as shown in figure 2.27. Some classical computation is needed to make use of the renormalised blocks, as inside each of these blocks we need to identify a series of crossing paths that connect through the boundaries of the lattice to adjoining renormalised blocks. From

¹⁶A similar effect occurs when we consider models with randomised processes for the occupancy of both sites and bonds. For example, probabilistic entangling gates in combination with photon loss.

¹⁷See chapter 4 for a more in depth explanation of the maximum efficiency on Bell state discrimination.

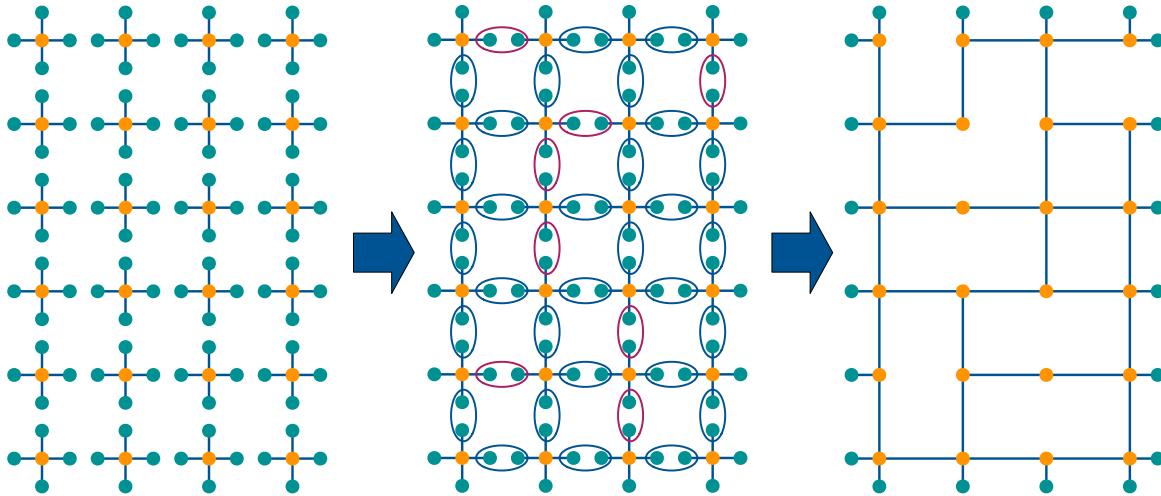


Figure 2.26: Building a cluster state from small photonic clusters. Successful fusion gates are denoted by blue ovals and failed fusion gates by red ovals. Note that the fusion gates represented here are on a rotated basis, which is described in detail in chapter 5, figure 5.4.

percolation theory [109, 103] we know that all these paths are within the spanning cluster.

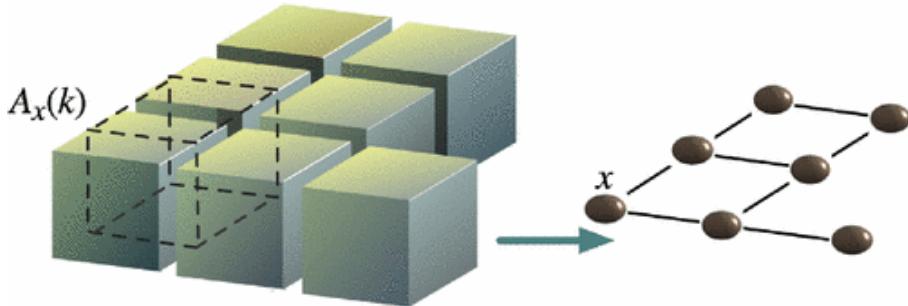


Figure 2.27: Renormalisation procedure: blocks of the percolated lattice give rise to sites in the renormalised lattice. Figure from [99], copyright (2007) by the APS.

The spanning cluster (infinite-crossing cluster in the thermodynamic limit of the supercritical regime) can be identified using polynomial time classical algorithms, which is an efficient process in the system size. Algorithms such as the *Hoshen-Kopelman* algorithm [110] can identify the crossing clusters in each block using $O(k^3)$ steps, where k is the block size.

To reduce the spanning cluster to a regular lattice, it suffices to find “T-junctions” (or three-way connections) and use these to build an hexagonal lattice [104, 105] (see figure 2.28). A hexagonal lattice can be used to perform MBQC, although it might be preferable to reduce it to a square lattice. All unused qubits can be cut out using X and Z measurements appropriately.

Once we have our percolated lattice, the sequence of single-qubit measurements required for MBQC can be determined by an offline classical computation. This scheme reduces considerably the amount of feedforward needed and has, at most, a sub-linear overhead per qubit in comparison to deterministic gates. For this logical lattice to be the resource required for MBQC, it is necessary that it is in the supercritical regime in order to have the appropriate scaling of the entanglement width.

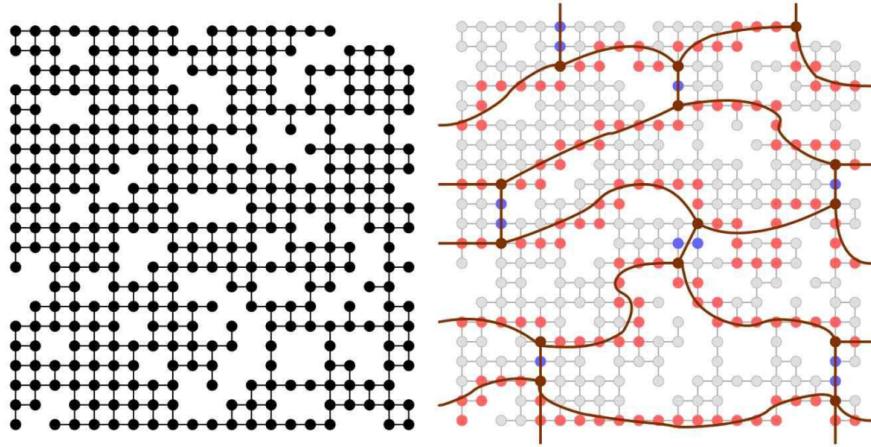


Figure 2.28: Example of a renormalisation of a percolated rectangular lattice into an hexagonal lattice. The left lattice shows the percolated lattice. In the lattice on the right, light grey qubits have been measured in the Z basis and red and blue qubits in the σ_X basis. Figure from [104], copyright (2007) by Springer.

In their proposal [99], Kieling, Rudolph & Eisert first show how it is possible to build a cubic 2-dimensional lattice using 7-qubit clusters as the initial resource. They then show how this resource can ultimately be reduced down to a 4-qubit cluster. Techniques both to increase the probability of occupancy of a bond p and to decrease the critical probability p_c (also known as percolation threshold) are used.

From the beginning, they consider 3-dimensional lattices. The advantage of 3-D lattices is that they have more favourable percolation properties, i.e. a lower percolation threshold, for lattices of the same correlation number (vertex degree). In the first instance they choose to build a cubic lattice, which has a critical probability $p_c = 0.249$ and coordination number 6. As the fusion probability is 50%, starting from 7-qubit star cluster and aligning each arm along one axis yields a lattice in the super-critical regime. They show that the overall resource requirements of $O(L^{2+3\epsilon})$ 7-qubit cluster states (where L is the size of the renormalised block and ϵ can be chosen arbitrarily small), is only a sub-linear overhead with respect to deterministic gates ($O(L^2)$).

To further reduce the size of the initial resource, they turn to the diamond lattice as it is the 3-D lattice with lowest coordination number (vertex degree 4) and a bond percolation threshold of $p_b = 0.389$. Percolating the diamond lattice would require 5-qubit star clusters; the success probability of the fusion gates used to build up the cluster is 50%, which is well above the percolation threshold, ensuring the existence of the spanning cluster. To further reduce the initial resource needed, they turn to the concept of the covering lattice, which is a lattice that has a site localised on each bond of the original lattice, and each such site is connected to all its closest neighbours. From percolation theory, it is known that the bond percolation threshold of a lattice is equivalent to the site percolation threshold of its covering lattice [111, 109]. Thus, they can use 4-qubit clusters (which have the connectivity of a complete graph, i.e. GHZ states) to build the covering lattice of diamond, the *pyrochlore lattice*, by fusing neighbouring corner qubits. This lattice, despite not being two-colourable, can be reduced to a universal resource

state [99].

Finally, they also address other imperfections in the lattice such as the loss of photons once the cluster has been created. To correct such imperfections, they use the standard technique in MBQC of measuring all the surrounding qubits in the Z basis, effectively removing the lost qubit from the cluster. They perform numerical simulations showing that they can tolerate losses up to 10% using a heralded loss model¹⁸. They suggest two strategies to more efficiently cope with loss. The first is fixing the block size, which would have an effect on the site occupancy of the logical lattice. Then fault-tolerant schemes [55] could be used above the respective fault-tolerant threshold [112]. The second strategy is to use loss-tolerant encodings such as the tree encodings introduced in [113] in the initial states to suppress photon loss.

It is important to compare the resource requirements of this percolation scheme with the other repeat-until-success strategies. The authors show [99] that the overhead in resources is only sub-logarithmic in comparison to having deterministic gates. When calculating resources for previous schemes, we calculated the number of Bell pairs consumed per single successful CZ gate. In the percolation scheme, individual CZ gates are not realised, and the analogous operation to a successful CZ gate is the formation of a bond between two neighbouring logical qubits (renormalised blocks).

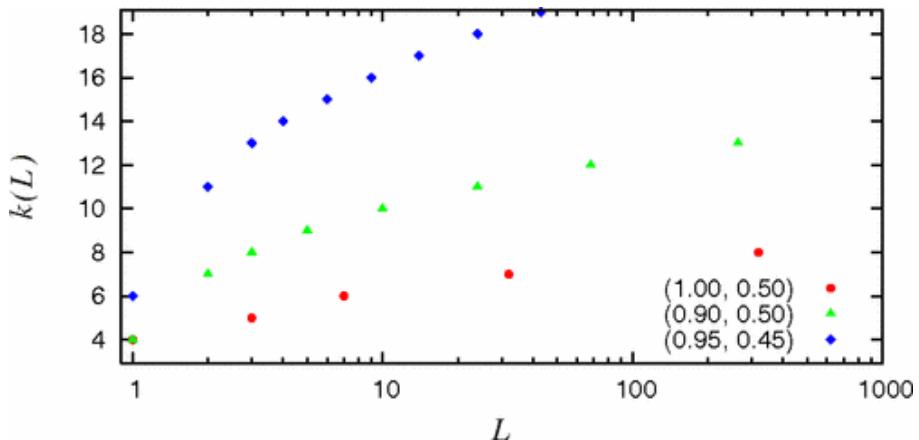


Figure 2.29: Dependence of the diamond lattice block size k^3 on the size L of the renormalised lattice. Figure from [99], copyright (2007) by the APS.

The most economic percolation scheme proposed in [99] is building the *pyrochlore lattice* from 4-qubit cluster states. Each of these 4-qubit clusters require on average 12 Bell pairs to be built. When considering a renormalised block, which is made out of k^3 physical qubits and already has all correlations to neighbouring renormalised qubits, we can estimate that the number of Bell Pairs required per single renormalised block is $\sim 12k^3$. The values for the renormalised block k depend on the size of the renormalised square lattice as shown in figure 2.29. It is clear the the resource consumption of this percolation protocol is higher than the most efficient of repeat-until-success strategies. Although more resource-efficient schemes can be proposed (chapter 5), there is a price to be paid for a ballistic strategy that requires no re-routing.

¹⁸Note that a heralded loss model is not experimentally justifiable, as there is no witness of the loss of a photon until a detector fails to measure a photon.

2.8 Discussion

In this chapter we have presented a literature review of LOQC. We have discussed the linear optical operations that can be used for the processing of quantum information and particularly focused on integrated optics as a promising physical implementation. We have seen that entangling two-qubit gates are very difficult to implement in linear optics, as photons don't interact, and deterministic entangling gates cannot be achieved [114] unless an infinite amount of resources is used [111, 115]. One way to get around this problem is to perform prepare small entangled states that can be used as resources in quantum information protocols. In chapter 4 we will present a number of schemes to generate this small entangled states.

In this chapter, we have also reviewed in detail all proposed protocols for LOQC from the original KLM [2] to the most recent percolation-based approaches. These protocols use different techniques and resources, but in order to compare them fairly we have calculated the number of Bell pairs required per single entangling gate in each case. This comparison, which is provided in detail in appendix B, shows that, although the number of resources required has gone down since the first proposals, most of the repeat-until-success schemes consume too many resources to be experimentally viable. The percolation-based approaches are the most promising as they have a fixed physical depth, i.e. the number of optical elements each photon encounters in its optical path is bounded. This is necessary to have bounded loss rates as photons don't have to be constantly re-routed through switching networks and it makes the experimental design much more amenable. In chapter 5 we will present a novel percolation scheme in which we have lowered the initial resource requirements to only 3-GHZ and Bell pairs. This scheme, unlike Kieling *et al's* [99], is built using loss tolerant gates exclusively. On top of these advantages we also show that it is at least an order of magnitude more resource efficient.

CHAPTER 3

SIMULATION OF STABILIZER COMPUTATIONS

3.1 Introduction

The stabilizer formalism was introduced [116] as a description of a certain subclass of quantum computations. It has proven to be extremely useful for analysing and understanding certain processes, most important of all, error-correcting codes. The key idea of the stabilizer formalism is to work in the Heisenberg picture [117], that is to work with operators rather than quantum states. Not only can the states themselves be described by operators, but the quantum operations they undergo are also described by the evolution of the operators rather than by the evolution of the states themselves [117]. Only a subclass of all possible quantum operations (local Clifford operations) can be described using this formalism; this subclass, however, contains some very important processes. In particular, the stabilizer formalism can be used to describe and study some quantum protocols, such as teleportation, the GHZ paradox, linear error-correcting codes and superdense coding [116].

The focus of this chapter will be the simulation of stabilizer computations, which can be efficiently performed on a classical computer [117]. We also make a brief mention of their main application, quantum error-correcting codes. For the most part we are concerned with simulation efficiency as well as methods that allow the visualisation of the computations. This proves a very useful tool to gain insight into the inner structure of different algorithms. Any process that can be described in the stabilizer formalism can be simulated graphically, which leads to a better understanding of the process itself. The techniques for simulating stabilizer computations are extremely useful when considering protocols to build cluster states for MBQC and understanding quantum error correction procedures, which are two crucial steps for the construction of a fault-tolerant linear optical quantum computer. The implementation of the techniques described in this chapter as a simulator (functions of which are detailed in appendix D) has been essential for obtaining some of the results in chapters 5 and 7.

In this chapter, sections 3.2 and 3.3 are a literature review based on [116, 118, 117, 70, 119, 120, 121, 122, 123, 38]. Section 3.4 contains adaptation of the work in the cited papers, distilled in algorithms to use in a computer simulation, this last section is my own work. Throughout this chapter there will be a running example illustrating the concepts explained. These examples will be shown in boxed spaces for clarity.

3.2 Stabilizer Formalism

A pure quantum state¹ can be described by its vector representation in Hilbert space: $|\psi\rangle \in \mathcal{H}$. However, there is an alternative representation, in which quantum states are described by the operators they are the +1 eigenvalue of, and that is the Stabilizer Formalism [116]. Some examples are:

$$\begin{aligned} |0\rangle &\rightarrow +1 \text{ eigenvalue of } Z, \\ |1\rangle &\rightarrow +1 \text{ eigenvalue of } -Z, \\ |+\rangle &\rightarrow +1 \text{ eigenvalue of } X, \\ |-\rangle &\rightarrow +1 \text{ eigenvalue of } -X \\ \frac{|00\rangle + |11\rangle}{\sqrt{2}} &\rightarrow +1 \text{ eigenvalue of } X_1X_2 \& Z_1Z_2. \end{aligned}$$

More generally, we say that $|\psi\rangle$ is the +1 eigenvalue of some operators and it is stabilized by them. By *stabilize* we mean that if we apply these operators to the state, the state will remain unchanged. The stabilizer formalism is restricted to states that can be stabilized by strings of Pauli operators [118].

The power of the stabilizer formalism comes from the use of group theory. The group that describes all the operators in the formalism is the Pauli group \mathbf{P}_n on n qubits. This group is composed of tensor products of all Pauli matrices (including the identity matrix) over each of the qubits, with the multiplicative factors of ± 1 and $\pm i$. The Pauli matrices form a closed group with very simple commutation relations. For example, the Pauli group for one qubit is

$$\mathbf{P}_1 \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}.$$

The Pauli group on n qubits \mathbf{P}_n is defined by the tensor of the n Pauli groups for one qubit;

$$\mathbf{P}_n \equiv \mathbf{P}_1 \otimes \mathbf{P}_1 \otimes \cdots \otimes \mathbf{P}_1.$$

Using this group, we can define stabilizers more precisely. Let S be an Abelian subgroup of \mathbf{P}_n , we define V_S to be the set of all n qubit states that are stabilized by all the elements of the group. This means that

$$S \text{ stabilizes } V_S \in \mathcal{H} \Leftrightarrow V_S = \{|\psi\rangle \in \mathcal{H} : G_i|\psi\rangle = (+1)|\psi\rangle \forall G_i \in S\}.$$

Therefore we can describe a particular subspace of qubit states V_S by its *stabilizer group* S . The stabilizer group S must be Abelian, since only commuting operators can have simultaneous eigenvectors. We can reduce the representation by describing the group (or subgroup) by its *generators*. The generators of a group are the smallest set of independent elements of the group that can be used to produce all the other elements belonging to that group. This is the most compact way of describing a group [70]. We will use the notation $S = \langle G_1, \dots, G_k \rangle$, where k is the number of generators of the group. It should be noted that to see if a particular vector is

¹There is a version of the stabilizer formalism that includes mixed states [120], but we don't review it here.

stabilized by a group S , it suffices to check that is stabilized by each one of the generators, as if it is stabilized by the generators, it will certainly be stabilized by products of the generators. The number of generators of the group also contains the information of the number of basis elements that span V_S .

The description of both states and processes is done through the evolution of the stabilizer operators. The application of a unitary operator U to a quantum state $\psi \in V_S$ (described by the stabilizer group S) can be described in terms of operators as the conjugation of each generator of S , G_i with U :

$$|\psi\rangle = G_i |\psi\rangle, \quad |\psi\rangle \rightarrow U|\psi\rangle \equiv G_i \rightarrow UG_iU^\dagger \text{ for all } i. \quad (3.1)$$

This transformation could take the Pauli operators in the stabilizer to a large class of unitary operators. However, given that the states that can be described by the stabilizer formalism are eigenstates of the Pauli matrices, the stabilizer formalism only considers a restricted class of unitaries: the gates that leave the Pauli group fixed under conjugation. This set of operators form a group $N(\mathbf{P})$, the normaliser of the Pauli group, which is usually called the *Clifford* group for its relation to the usual Clifford groups [124, 125, 126].

$$N(\mathbf{P}) = \{H \in \mathbf{P}_n : HGH^\dagger = G \ \forall G \in \mathbf{P}\}.$$

The group is generated by the operators Hadamard ($H = |+\rangle\langle 0| + |-\rangle\langle 1|$), Phase ($S = |0\rangle\langle 0| + i|1\rangle\langle 1|$) and CNOT ($CNOT = |0\rangle\langle 0| \otimes I_t + |1\rangle\langle 1| \otimes X_t$). The matrix description of these gates is given in section 3.2.3.

The number of operators needed to describe a stabilizer state is given by the following theorem:

Theorem 1. *Let $S = \langle G_1, \dots, G_k \rangle \in \mathbb{P}_n$ where k is the smallest number of generators for S , then the dimension of the subspace stabilized by S is $\dim V_S = 2^{n-k}$.*

Proof. See [70] for proof.

When $k = n$, the stabilizer group describes a unique state state, whereas if $k < n$ there are extra degrees of freedom. In many error correction codes, these extra degrees of freedom are used to encode a protected logical qubit (see subsection 3.2.2).

State transformation is also described in terms of operators: there are two type of transformations we might want to do on our states, unitary transformations and measurements.

- **Unitary transformations:** The unitary transformations ($|\psi\rangle \rightarrow U|\psi\rangle$) that we can do on the quantum states are restricted to operations on the *Clifford Group*. Therefore if $|\psi\rangle \in V_S$ is stabilized by $S = \langle G_1, \dots, G_k \rangle$, then $|\psi'\rangle = U|\psi\rangle \in V'_S$ is stabilized by $S' = \langle UG_1U^\dagger, \dots, UG_kU^\dagger \rangle$.
- **Measurements:** We consider an observable $A \in \mathbf{P}_n$. The projector associated with this observable is defined as $P_m = \frac{1}{2}(I + (-1)^m A)$ with $m = 0, 1$ being the measurement result.

3. SIMULATION OF STABILIZER COMPUTATIONS

1. *A commutes with $G_i \forall i \in \{1, \dots, k\}$.* There is no randomness in the measurement result as the quantum state is an eigenstate of the observable measured. Therefore if $|\psi\rangle \in V_S$ is stabilized by $S = \langle G_1, \dots, G_k \rangle$, after the measurement $P_m|\psi\rangle = |\psi\rangle \in V_S$ is stabilized by $S = \langle G_1, \dots, G_k \rangle$.
2. *A does not commute with $G_i \forall i \in \{1, \dots, k\}$.* The quantum state is not in an eigenstate of the observable measured and therefore the measurement outcome is random. The observable might anti-commute with more than one generator (G_1, \dots, G_l), but we can choose one of them (G_1) and multiply all the other non-commuting one by the chosen one, so that we end up with a set of generators out of which only one of them anti-commutes with the observable measured ($S = \langle G_1, G_1G_2, \dots, G_1G_l, G_{l+1}, \dots, G_k \rangle$). The observable measured will substitute this non-commuting generator in the stabilizer group multiplied by a phase $(-1)^m$. Therefore if $|\psi\rangle \in V_S$ is stabilized by $S = \langle G_1, \dots, G_k \rangle$, after the measurement $|\psi'\rangle = P_m|\psi\rangle \in V'_S$ is stabilized by $S' = \langle (-1)^m A, G_1G_2, \dots, G_1G_l, G_{l+1}, \dots, G_k \rangle$.

Example:

This is an example to illustrate concepts introduced. We will use the same state throughout all examples in this section, all of which will appear in boxes such as this one.

We define the stabilizer state ψ_E with its representation as a stabilizer subgroup S_E :

$$\psi_E \rightarrow S_E = \langle X_1Z_2, Z_1X_2Z_3, Z_2X_3 \rangle. \quad (3.2)$$

Unitary evolution: An example of a unitary evolution would be the application of a Hadamard on qubit 2:

$$S_E \xrightarrow{H_2} S'_E = \langle X_1X_2, Z_1Z_2Z_3, X_2X_3 \rangle. \quad (3.3)$$

Measurement of observable A:

- A commutes with all generators:

$$A = X_1X_3 \in S_E \Rightarrow S_E \xrightarrow{A} S'_E = S_E. \quad (3.4)$$

- A doesn't commute with all generators:

$$A = X_1 \notin S_E \Rightarrow S_E \xrightarrow{A} S'_E = \langle (-1)^m X_1, X_1Z_2, Z_2X_3 \rangle \neq S_E. \quad (3.5)$$

This formalism has been called [117] a Heisenberg representation of part of Quantum Computation. The operators, rather than the states, evolve in time, and the processing of information is described by some transformation rules that are to be applied to the operators. It is therefore completely equivalent to describe the operators as a set of transformation rules rather than unitary matrices. It suffices to define these operators as transformations of X and Z operators,

as these generate the Pauli group up to phases.

Operator transformation rules:

$$\begin{aligned} H : \quad & X \rightarrow Z, \quad Z \rightarrow X, \\ S : \quad & X \rightarrow Y, \quad Z \rightarrow Z, \\ CNOT : \quad & X \otimes \mathbf{1} \rightarrow X \otimes X, \quad \mathbf{1} \otimes X \rightarrow \mathbf{1} \otimes X, \quad Z \otimes \mathbf{1} \rightarrow Z \otimes \mathbf{1}, \quad \mathbf{1} \otimes Z \rightarrow Z \otimes Z, \\ CZ : \quad & X \otimes \mathbf{1} \rightarrow X \otimes Z, \quad \mathbf{1} \otimes X \rightarrow Z \otimes X, \quad Z \otimes \mathbf{1} \rightarrow Z \otimes \mathbf{1}, \quad \mathbf{1} \otimes Z \rightarrow \mathbf{1} \otimes Z. \end{aligned}$$

Any quantum computation on a state $\psi \in V_S$ involving only Clifford operations and Pauli measurements can therefore be expressed in terms of the transformation of the operators in the stabilizer group S .

3.2.1 Gottesmann-Knill Theorem

Theorem 2. *A quantum computation involving only state preparations in the computational basis, Clifford group transformations and measurements of Pauli observables can be efficiently simulated by a classical computer [117].*

The proof of this theorem is implicit in our description of the stabilizers formalism above, as the method for doing this computation would simply be to keep track of the stabilizer generators, and this would scale polynomially in n . In section 3.3.4 we will give a full account of computational complexity of each simulation step.

It must be noted that the quantum computation described here is not universal, as the set of gates applied does not constitute a universal set. We need to add the $\pi/8$ gate ($T = |0\rangle\langle 0| + e^{i\pi/4}|1\rangle\langle 1|$) to the gates in the Clifford group to make the computation universal (see proofs in [70]).

3.2.2 Error correction with Stabilizers

The stabilizer formalism has been particularly useful for representing a subclass of quantum codes and understanding the structure of said codes. It has been particularly fruitful in the field of quantum error correction, as it can fully describe several important codes such as Shor's nine qubit code [76], CSS codes [127, 128] and topological codes [129, 55]. All these error correcting codes belong to the class of stabilizer codes $C(S)$: an $[n, k]$ stabilizer code on n physical qubits is defined [70] as the vector space V_S , of dimension 2^k , stabilized by the a subgroup S of G_n , where S has $n - k$ independent generators.

The Hilbert space can be partitioned in subspaces via the eigenvalues of the generators in the stabilizer set:

$$\mathcal{H} = \bigoplus_{\bar{s}} C_{\bar{s}}, \quad \bar{s} = (s_1, \dots, s_k), \quad s_i = \pm 1 \text{ for all } i,$$

where \bar{s} is defined as the syndrome vector. Different syndrome vectors are associated with the different subspaces in which the Hilbert space is partitioned. The classification of any state in the Hilbert space into one of these subspaces can be done in the following way

$$|\psi\rangle \in C_{\bar{s}} \Leftrightarrow G_i |\psi\rangle = s_i |\psi\rangle \text{ for all } i = 1, \dots, n - k.$$

3. SIMULATION OF STABILIZER COMPUTATIONS

The codespace of a stabilizer group is defined as

$$C = C_{(+1,+1,\dots,+1)}.$$

The dimension of the codespace is: $\dim C = \dim C_{\bar{s}} = 2^{n-k}$ for any \bar{s} . This follows from theorem 1 and implies the existence of $t = n - k$ virtual/logical qubits.

Logical operators for t encoded qubits: We choose $H_1, \dots, H_t \in \mathbf{P}_n$ so that $H_1, \dots, H_k, G_1, \dots, G_k$ forms a complete and mutually commuting set of independent Pauli elements of \mathbf{P}_n . Our logical Z operators are defined as $\bar{Z}_i = H_i$, and the logical X operators as $\bar{X}_i \in \mathbf{P}_n$ such that $\bar{X}_i \bar{Z}_i \bar{X}_i^\dagger = -\bar{Z}_i$ and $\bar{X}_i \bar{Z}_j \bar{X}_i^\dagger = \bar{Z}_j$ for $j \neq i$. These logical operators map codewords to codewords, i.e. $\bar{X}_i, \bar{Z}_i : C_{\bar{s}} \rightarrow C_{\bar{s}}$.

The uncorrectable errors perform a logical operation on the encoded qubits and they commute with the stabilizer generators, hence they are not detected. They are described by the centraliser of the stabilizer group S in the Pauli group \mathbf{P} , which in this case corresponds with the normaliser of S , $N(S) = \{E \in \mathbf{P}_n : EgE^\dagger \in S \forall g \in S\}$.

Given an encoded state $|\psi\rangle$ in a stabilizer code S , $\{E_i\}$ with $E_i \in \mathbf{P}_n$ is a set of *correctable Pauli errors* if $E_i^\dagger E_j \notin N(S) - S$ (see [70] for proof). The *error detection* is done by measuring $G_1, \dots, G_k \in S$ and obtaining the syndrome vector $\bar{s} = (s_1, \dots, s_k)$. We determine: $F \in \mathbf{P}_n$ such that $FG_i F^\dagger = s_i G_i$ for all i . A decoder algorithm is used to determine what is the error correction that needs to be implemented. The *error correction* is made by applying F^\dagger over the state with errors. This action corrects the noise as given an error $E \in \mathbf{P}_n$ such that $EPE^\dagger = FPF^\dagger$, $F^\dagger E \in S$ therefore the error is corrected.

In chapter 7 we give further details of the error detection and correction procedures, focusing on some topological codes that are particularly relevant for LOQC.

3.2.3 Stabilizer circuits

Stabilizer circuits are quantum circuits whose action can be described using the stabilizer formalism. We re-state the basic components of the stabilizer formalism organised as preparation, processing and readout:

- Initialisation: Qubits are initialised in the computational basis states: $|0\rangle, |1\rangle$.
- Gates:
 - Single qubit Hadamard gate: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.
 - Single qubit Phase gate: $P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$.
 - Two-qubit entangling CNOT gate: $CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.
- Readout: Measurement in the computational basis.

From the Gottesman-Knill theorem [117], we know that these operations can be efficiently simulated in a classical computer. The quantum states that can be prepared using stabilizer circuits lie on the octahedron [130] shown in figure 3.1. It should be noted that the octahedron lies inside the Bloch sphere and they only touch at the points corresponding to the eigenstates of the Pauli matrices.

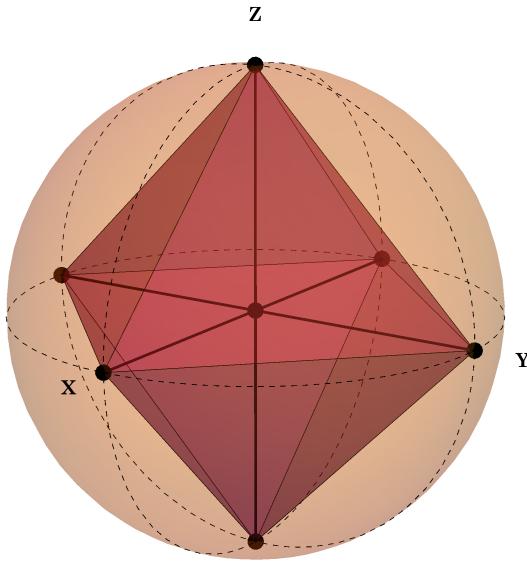


Figure 3.1: Stabilizer octahedron embedded in Bloch sphere. Image concept from [130], copyright (2005) by the APS.

It was shown by Bravy and Kitaev [130], that if we combine stabilizer circuits with some ancilla qubits prepared in a particular *magic state*, even if those ancillas are noisy, we can simulate universal quantum computation. Given that the stabilizer formalism can be efficiently simulated on a classical computer, magic states are then the resource for quantum speed up for this model. Recent remarkable results [131] have linked magic states to contextuality, which provide a fundamental characterisation of uniquely quantum phenomena.

3.2.4 Graph States

Graph states are an important and useful subclass of the stabilizer states. Given an undirected graph defined by a set of vertices and edges $G = (V, E)$, a graph state is the quantum state resulting in putting qubits in the $|+\rangle$ state at each vertex and performing CZ gates where the vertices are. It is easy to realise that cluster states, which were defined in chapter 1 as resource states in the MBQC model [33], are just a subclass of graph states, where the qubits are placed in a rectangular grid.

The stabilizer operators describing a particular graph state are given by

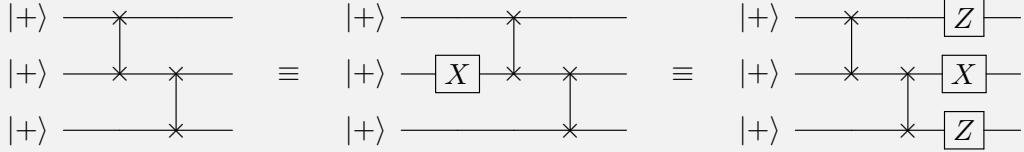
$$K_i = X_i \prod_{j \sim i} Z_j, \quad \forall i \in V, \quad K_i \in \mathbf{P}_n \quad \text{where } j \sim i \text{ means } j \text{ and } i \text{ are adjacent.}$$

It can be easily shown why these are the stabilizers for a graph state (see figure 3.2.4). The cluster is built by applying CZ gates to qubits initially in the state $|+\rangle$. As the stabilizer of the

3. SIMULATION OF STABILIZER COMPUTATIONS

initial state is the X operator on every qubit, its action leaves the state invariant.

Example:



Commuting a stabilizer operator through the graph building process. Image concept from Terry Rudolph's "Introduction to Quantum Information" lecture notes [132].

The commutation of an X operator with a CZ gate generates a correlated Z operator on the other qubit involved in the CZ gate:

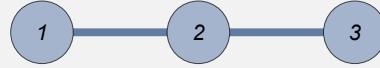
$$CZ(I \otimes X)CZ = Z \otimes X, \quad (3.6)$$

which means that after the action of the CZ gates (which belong to the Clifford group as they can be generated by the multiplication of Hadamards and CNOTs), the stabilizer description of the X_i operator is $\prod X_i Z_j$ where j are the adjacent sites to i .

The stabilizer subgroup is defined as $S = \langle K_i \rangle \quad \forall i \in V$, and it has n independent generators ($[K_i, K_j] = 0, \forall i, j \in V$) for n qubits. This means that the stabilized subspace, according to theorem 1 is 1-dimensional, and therefore the cluster state is uniquely defined up to a global phase. An interesting corollary is that graph states do not encode any logical qubit.

Example:

The graph state that corresponds to the stabilizer subgroup $S_E = \langle X_1 Z_2, Z_1 X_2 Z_3, Z_2 X_3 \rangle$ is



It is not only the case that every graph state has a stabilizer description, but it is also true that every stabilizer state has a graph state description [121]. We will explain this in section 3.3.2, with the help of a binary representation of the stabilizers.

3.3 Simulation

The Gottesman-Knill theorem gives us a recipe for efficiently simulating stabilizer computations on a classical computer: represent quantum states by operators and track the computation by the change of these operators. In this section, we go into further details of this simulation. We specify the representation of the code in binary, we provide algorithms to link the stabilizer and graph representation of a state and introduce a novel representation that improves performance.

3.3.1 Binary representation of a stabilizer code

To implement a code on a classical computer that will allow us to simulate a (non universal) quantum computation via the Gottesmann-Knill theorem, we need to be able to describe the stabilizers and the unitary operations on them efficiently in the language of binary vector spaces. In his thesis [118], Gottesman gives a binary representation for a stabilizer code that allows us to do exactly that. In this approach, we exploit the homomorphism between the Pauli group under matrix multiplication, (P_1, \cdot) , and a two-dimensional binary vector space under modulo 2 addition, $(\mathbb{Z}_2^2, +)$. Each stabilizer generator is represented by a string of bits, where the Pauli operator on each qubit is represented by two classical bits according to the following rules:

$$\begin{aligned} I &\rightarrow 00, \\ X &\rightarrow 10, \\ Y &\rightarrow 11, \\ Z &\rightarrow 01. \end{aligned}$$

In this encoding, information about the overall phases of the Pauli operators is lost. We can recover this information by adding an extra bit string representing these phases.

A set of k stabilizer generators on n qubits is represented by a matrix of $2n$ rows and k columns with entries that are either 0 or 1. With a slight abuse of notation, we will denote the stabilizer matrix that represents the stabilizer subgroup by S . Each column represents a stabilizer generator g_i , the first n rows represent the Z operator on each of the n qubits, and the last n rows represent the X operator on each of the n qubits. For example, here we show how a stabilizer would be mapped to a bit string, which would then be a column on the stabilizer matrix:

$$X_1 X_2 Z_3 I_4 Y_5 \Rightarrow \underbrace{10}_{X_1} \underbrace{10}_{X_2} \underbrace{01}_{Z_3} \underbrace{00}_{I_4} \underbrace{11}_{Y_5} \Rightarrow \underbrace{11001}_{X} \mid \underbrace{00101}_{Z}.$$

The general form of the matrix S is as follows

$$S = \begin{bmatrix} z_{11} & z_{21} & \dots & z_{n1} \\ z_{12} & z_{22} & \dots & z_{n2} \\ \vdots & \vdots & \dots & \vdots \\ z_{1n} & z_{2n} & \dots & z_{nn} \\ \hline x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & \dots & \vdots \\ x_{1n} & x_{2n} & \dots & x_{nn} \end{bmatrix} \leftarrow \begin{array}{l} q_1 \\ q_2 \\ \vdots \\ q_n \\ q_1 \\ q_2 \\ \vdots \\ q_n \end{array} \quad (3.7)$$

↑ ↑ ↑

$g_1 \quad g_2 \quad \dots \quad g_n$

where the columns, g_i , are the stabilizer group generators in the binary picture and the rows

3. SIMULATION OF STABILIZER COMPUTATIONS

correspond to the operators acting on each qubit, q_j .

Therefore, in this binary picture, an operator (such as the generators G_i) acting on a group of qubits is represented by a binary string with $2n$ bits.

Example:

The stabilizer matrix representation of the state ψ_E is

$$S_E = \left[\begin{array}{ccc} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right] \quad (3.8)$$

where the columns of the matrix, g_1, g_2, g_3 correspond to the stabilizer generators in the binary picture.

Two operators A and B commute if and only if their binary string representations a and b , where ($a, b \in \mathbb{Z}_2^{2n}$), are orthogonal with respect to the symplectic inner product:

$$[A, B] = 0 \iff a^T \cdot \mathbb{P} \cdot b = 0, \quad (3.9)$$

where \mathbb{P} is the symplectic inner product on the space \mathbb{Z}_2^{2n} . Its matrix representation is given by

$$\mathbb{P} = \left[\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right]. \quad (3.10)$$

In this picture, the stabilizer subgroup corresponds to a n -dimensional linear subspace of \mathbb{Z}_2^{2n} (a matrix of n columns with $2n$ bits each). The stabilizer subgroup is abelian, which in the binary picture can be expressed by saying that it is its own orthogonal complement with respect to this symplectic inner product:

$$S^T \cdot \mathbb{P} \cdot S = 0. \quad (3.11)$$

The generator matrix for the stabilizer state, S , is not unique. Multiplying the generators of the stabilizer subgroup together produces a new member of the generator group. In the binary picture this operation corresponds to a change of generator basis. Operationally, a change of basis of the stabilizer generators amounts to multiplying each sub-matrix (S_X and S_Z) of the generator matrix S to the right with an invertible $n \times n$ matrix, R , which performs the basis change in the binary subspace [121]. This is the same kind of operation as changing the coordinates of vectors in linear algebra, except that here, for convenience, we multiply on the right.

$$S \cdot \left[\begin{array}{c} R \\ R \end{array} \right] = \left[\begin{array}{c} S_Z \\ S_X \end{array} \right] \left[\begin{array}{c} R \\ R \end{array} \right] = \left[\begin{array}{c} S'_Z \\ S'_X \end{array} \right] = S'. \quad (3.12)$$

Example:

Change of basis: An example of change of basis in the generators would be defining a stabilizer subgroup $S_F = [g_1 g_2 g'_3]$ where $g'_3 = g_1 \cdot g_3$ (or equivalently addition modulo 2), g_i being the columns of the matrix representation of S_E . As we have only performed a change of basis by multiplying two of the stabilizer generators together, the matrix representations of S_E and S_F are related by the change of basis matrix R

$$R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.13)$$

The matrix representation of the stabilizer state ψ_F is

$$S_F = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

It is worth remembering that although their representation is different, i.e. $S_E \neq S_F$, $\psi_E = \psi_F$ as both sets of generators stabilize the same quantum state.

The operations on the stabilizer state in its binary representation are just the transformations of \mathbb{Z}_2^{2n} which preserve the symplectic product. These are a subgroup of the Clifford group as they don't include phase changes. This ensures that we remain in the valid stabilizer subspace up to overall phases. We will hereafter refer to these operations as Clifford operations, as they are equivalent in this binary representation².

$$S_C = Q \cdot S, \quad S_C^T \cdot \mathbb{P} \cdot S_C = 0 \Rightarrow S^T \cdot Q^T \cdot \mathbb{P} \cdot Q \cdot S = 0, \quad (3.15)$$

but $S^T \cdot \mathbb{P} \cdot S = 0$. Therefore, any Clifford operations on the stabilizer matrix can be described [121] by a $2n \times 2n$ matrix Q that satisfies

$$Q^T \cdot \mathbb{P} \cdot Q = \mathbb{P}. \quad (3.16)$$

The local Clifford group, which acts individually in each qubit, is generated by the operators X, Y, Z, H, P . If no entangling gates are applied (which would act on the operators of two qubits at a time) the matrix Q , which represents the action of the local Clifford group, has a

²When we implement these operations in the simulation described later in the chapter, we include extra binary bits to account for the phases.

block structure, and each block has diagonal structure.

$$Q = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \Rightarrow Q_i = \begin{bmatrix} A_{ii} & B_{ii} \\ C_{ii} & D_{ii} \end{bmatrix}, \quad (3.17)$$

where the matrices A, B, C, D represent transformations between the S_Z and S_X subspaces:

$$\begin{aligned} A : S_Z &\rightarrow S_Z & B : S_X &\rightarrow S_Z \\ C : S_Z &\rightarrow S_X & D : S_X &\rightarrow S_X. \end{aligned} \quad (3.18)$$

As Q implements Clifford operations, which are invertible, this implies that each sub-matrix Q_i is invertible. With all these considerations, we can write the most general local Clifford transformation of the stabilizer group in the binary formalism as

$$S' = Q \cdot S \cdot R. \quad (3.19)$$

3.3.2 Reduction to graph states

As we have briefly mentioned earlier, a graph is a mathematical structure $G = (V, E)$ formed by a set of vertices V and a set of edges E . Edges connect two vertices, and if there is only one edge between any two vertices and no edge connecting a vertex with itself (self-loop) the graph is considered a simple undirected graph [133]. Any undirected graph can be completely described by an adjacency matrix, θ , which is an $n \times n$ matrix where $\theta_{ij} = 1 \iff \{i, j\} \in E$. This matrix is symmetric and with the elements of the main diagonal all zero.

A graph state is a quantum state defined on a graph, where qubits correspond to vertices of the graph and entangling (CZ) operations between qubits correspond to edges. In the binary picture, the representation of a graph state would be

$$S_G = \begin{bmatrix} \theta \\ \mathbb{1} \end{bmatrix}. \quad (3.24)$$

Theorem 3. *Every stabilizer state is equivalent to a graph state under local Clifford operations [121].*

Proof. See proof in appendix C. ■

There are two corollaries that can be extracted from this theorem:

- The theorem implies that the disregard of overall phases is justified.
- We can restrict our attention to graph states when studying the local equivalence of stabilizer states.

The action of local Clifford transformations on graph states can be described in terms of pure graph operations [121]. Let us first introduce the so-called *local complementation* graph operation c_i : Given a graph $G = (V, E)$, performing a local complementation on a vertex $i \in V$ consists on first finding the neighbourhood of that vertex $N(i) \in V$, i.e. the vertices $j \in V$

Example:

Local Clifford operation:

We apply the local Clifford operation H_1P_3 on the state ψ_E , obtaining the state ψ_G :

$$\psi_E \xrightarrow{H_1P_3} \psi_G \Rightarrow S_E \xrightarrow{H_1P_3} S_G = \langle Z_1Z_2, X_1X_2Z_3, Z_2Y_3 \rangle \quad (3.20)$$

The binary matrix representation of this state is

$$S_G = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ \hline 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right] \quad (3.21)$$

The matrix representation of the local Clifford operation is

$$Q = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right) = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \quad (3.22)$$

We can verify from this matrix the local Clifford operations applied to each qubit:

$$Q_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \equiv H_1 \quad Q_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \equiv \mathbf{1}_2 \quad Q_3 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \equiv P_3 \quad (3.23)$$

such that $(j, i) \in E$, and then replacing $N(i)$ by its inverse, which is equivalent to putting edges between vertices that are not connected and disconnecting the vertices that were originally connected. An example of such local complementation on a graph can be seen in figure 3.2. It was shown in [121] that the operations c_i can be realised as local Clifford operations. Reversely, any Clifford operation can be performed by a sequence of local complementations.

3.3.3 Local Clifford equivalence

Recall that in the previous section, we deduced the operational form in the binary picture for Local Clifford (LC) operations, a block matrix Q with diagonal blocks, in equation (3.17). Any local Clifford operation is given by the tensor factor of n Q_i matrices, one for each qubit, each of the invertible. Therefore, given two stabilizer states S_1 and S_2 , they are LC-equivalent iff there exist an operator Q such that $Q \cdot S_1 = S_2$ up to a basis change [122]. See proof in appendix C.

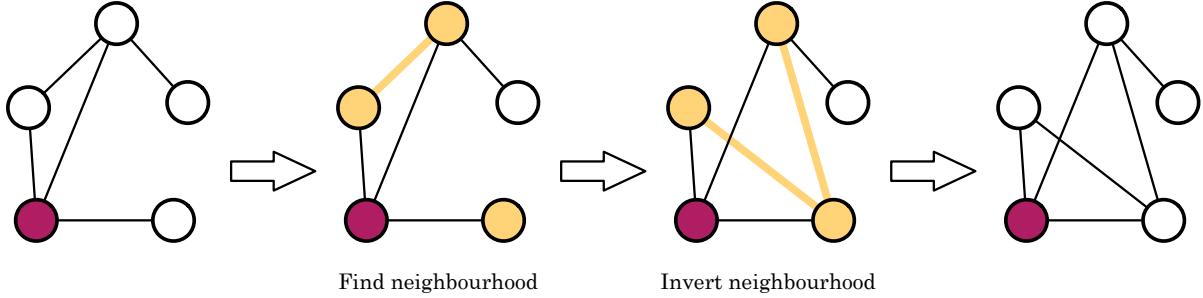


Figure 3.2: Local complementation on the red vertex. The neighbourhood and its inversion are marked in the intermediate steps in yellow.

LU-LC conjecture

The LU-LC conjecture concerns the equivalence under local operations of two graph states: *Given two graph states that are equivalent under local unitary (LU) operations, does this imply that they are LC-equivalent?* It has been shown that two stabilizer states are equivalent under Stochastic Local Operations and Classical Communication if they are LU-equivalent [134]. This implies that entanglement is conserved under LU operations and hence the equivalence classes under local unitaries can be used to classify states according to their entanglement. As we have shown above, algorithms exist to efficiently determine if two stabilizer states are LC-equivalent. If the LU-LC conjecture were true, we could efficiently compute entanglement classes of stabilizer states.

We have seen that any stabilizer state can be represented by a graph, and therefore a local Clifford operation can always be described in graph-theoretic terms (as the transformation from the graph state equivalent to the stabilizer state to the graph state equivalent to the stabilizer states *after* the Clifford operation has been performed). If the conjecture were true, this would imply that any question regarding stabilizer states could be stated in graph-theoretic terms, allowing us to use tools from combinatorics and graph theory to study entanglement properties of the stabilizer states.

This conjecture was an open problem for a number of years, with all the evidence pointing towards it being true: it was proven that LU equivalence implied LC equivalence for certain classes of states [123, 135], numerical evidence was put forward that confirmed the equivalence for stabilizer states of up to 7 qubits [136] and it was shown that the conjecture would be true given that a certain statement about quadratic forms was true [137]. However, the conjecture was disproved [138] by finding a counter example to said statement about quadratic forms in a stabilizer state of 27 qubits, found by systematic computer search.

3.3.4 Efficiency of the classical simulation

From group theory we know that any finite group G has a generating set of size at most $\log_2 |G|$, so if $|\psi\rangle$ is a stabilizer state over n qubits, the stabilizer group S has a generating set of $n = \log_2 2^n$. Using an extended version of the standard form described above (including a row column for the phase), each generator takes $2n + 1$ bits to be specified: n bits to specify the

X operators, n bits to specify the Z operators and a bit to specify the phase (which is always +1 or -1). As there are n generators of the stabilizer group, the total number of bits needed to specify $|\psi\rangle$ is $n(2n + 1)$.

The updates on the stabilizer state can be done in polynomial time if we keep track of these $n(2n + 1)$ bits. The updates corresponding to the unitary gates are very efficient, requiring $O(n)$ time for each gate. Each unitary operation on one qubit changes at most one Pauli operator (the one corresponding to the qubit to which the operation is applied) in each stabilizer, each Pauli operator is represented by 2 classical bits and we have n stabilizers, therefore we perform $2n$ operations. If the unitary operation is performed on two qubits, the same arguments holds but in this case we are updating 2 Pauli operators per stabilizer, performing $4n$ operations.

The updates corresponding to measurements are not as efficient. First, we check whether the measurement outcome will be determinate or random (depending if the measurement operator commutes with all the stabilizers or not) takes $O(n)$ steps (we have to check one bit in each stabilizer, n in total). If the outcome is random, updating the state takes $O(n^2)$. As explained in a previous section, when the measurement operator doesn't commute with at least one stabilizer, we choose only one of the stabilizers that do not commute, we multiply any other stabilizer that doesn't commute by this first chosen one and substitute the chosen stabilizer by the measurement operator. We multiply the chosen stabilizer by at most $n - 1$ others, each stabilizer multiplication involving n bit modulo 2 multiplications, therefore explaining the bound $O(n^2)$ on the random outcome measurement.

However, in the case where the outcome is determinate the decision on the measurement result takes $O(n^3)$ time in practice due to the Gaussian elimination process needed to update the stabilizers in order to isolate the measurement operator from the stabilizer operators to find the measurement result. There exists a faster algorithm for Gaussian elimination of $O(n^{2.3727})$ [139], however the software used for our simulation, Mathematica, and indeed most mathematical softwares, have their in-built Gaussian elimination procedure using Bareiss algorithm [140, 141].

3.3.5 Improved simulation using Destabilizers

Aaronson and Gottesman presented [120] a novel idea for improving the classical simulation of stabilizer circuits, in which both deterministic and random measurements can be performed in $O(n^2)$ time. This improvement in the time of the computation comes with a cost in the number of bits needed to specify the state, which is multiplied by a factor of 2. The main idea of the algorithm is to, in addition to the n stabilizers, store n “destabilizers”, which are a set of Pauli operators that together with the stabilizers generate the full Pauli group \mathbf{P}_n . The number of bits required to store now is $2n(2n + 1)$.

The algorithm performs the operations in an extended version of the binary matrix we used before to represent the stabilizers, and which Aaronson and Gottesman call *tableau*. Its form

3. SIMULATION OF STABILIZER COMPUTATIONS

for an n qubit state is:

$$\left(\begin{array}{ccc|ccc|c} x_{11} & \cdots & x_{1n} & z_{11} & \cdots & z_{1n} & r_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & \cdots & x_{nn} & z_{n1} & \cdots & z_{nn} & r_n \\ \hline x_{(n+1)1} & \cdots & x_{(n+1)n} & z_{(n+1)1} & \cdots & z_{(n+1)n} & r_{(n+1)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{(2n)n} & \cdots & x_{(2n)n} & z_{(2n)n} & \cdots & z_{(2n)n} & r_{(2n)} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) = \left(\begin{array}{c|c|c} Dx & Dz & r_i \\ \hline Sx & Sz & r_{n+i} \\ \hline 0 & 0 & 0 \end{array} \right). \quad (3.25)$$

Rows 1 to n of the tableau represent the destabilizer generators (R_1, \dots, R_n) and rows R_{n+1}, \dots, R_{2n} represents the stabilizer generators ³. The last column of the tableau represents the phase of a particular stabilizer or destabilizer, $r_i = 0$ means a positive phase and $r_i = 1$ a negative phase. The last row is added to the tableau as scratch space.

Example:

The binary matrix representation of the stabilizer state S_E using the destabilizer algorithm is

$$S_E = \left(\begin{array}{ccc|ccc|c} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right). \quad (3.26)$$

The algorithm as described in [120] proceeds through the gates in order and performs some operations on the entries of the tableau according to the type of gate implemented. One important subroutine is the group operation on \mathbf{P}_n , defined as the sum modulo 2 of two rows.

The tableau algorithm has some invariants under its action [120]:

1. R_{n+1}, \dots, R_2 generate $S(|\psi\rangle)$, and R_1, \dots, R_{2n} generate \mathbf{P}_n .
2. R_1, \dots, R_n commute.
3. For all $h \in \{1, \dots, n\}$, R_h anticommutes with R_{h+n} (every destabilizer anticommutes with its corresponding stabilizer).
4. For all $i, h \in \{1, \dots, n\}$ such that $i \neq h$, R_i commutes with R_{h+n} (every destabilizer commutes with the rest of stabilizers).

³Note that now the stabilizer operators are the rows of the matrices instead of the columns as in the previous sections. In each section we follow the notation of the relevant papers.

In this formalism, the commutation or anti-commutation of the operators is given by the symplectic inner product:

$$R_i \cdot R_j = x_{i1}z_{j1} \oplus \dots x_{in}z_{jn} \oplus x_{j1}z_{i1} \oplus \dots x_{jn}z_{in}. \quad (3.27)$$

This tableau procedure obtains deterministic measurement outcomes in only $O(n^2)$ steps. For a deterministic outcome, we need that the measurement operator Z_a must commute with the stabilizers, so

$$\sum_{h=1}^n c_h S_h = \pm Z_a \quad (3.28)$$

for a unique choice of $c_1, \dots, c_n \in \{0, 1\}$. If we can determine the coefficients c_i 's, then by summing the corresponding S_h 's we can learn the measurement outcome (sign of Z_a). The coefficients are given by:

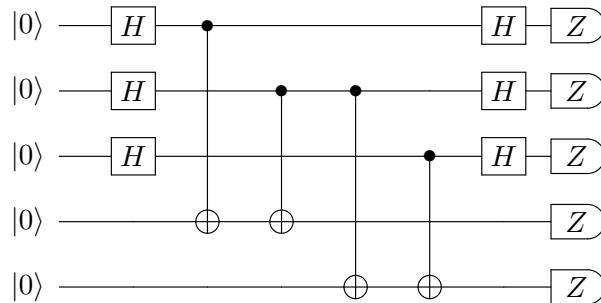
$$c_i \equiv \sum_{h=1}^n c_h (D_i \cdot S_h) \equiv D_i \cdot \sum_{h=1}^n c_h S_h \equiv D_i \cdot Z_a \pmod{2}. \quad (3.29)$$

So we just have to check if D_i commutes with Z_a , or equivalently if $x_{ia} = 1$.

3.3.6 Performance comparison

Mathematica is a programming language that stands out for its ability to manage lists, with a wide range of internal functions that make the manipulation easier. We have reproduced the tableau algorithm on Mathematica, not only the basics functions as described in [120]: CNOT, Hadamard, Phase and Measurement in the standard basis; but also other routines that allow visualisation of cluster states as graph states. The full list of operations that have been so far implemented in the Mathematica code with their descriptions are given in appendix D.

We can make a quick comparison of the efficiency of this improved stabilizer algorithm [120] with the usual matrix representation of a circuit. For this comparison we will be using a circuit with only the allowed operations in. The chosen circuit implements an instance of Simon's algorithm, described by the following circuit:



This procedure uses Simon's algorithm [142] to learn about a “hidden shift”, s of a function f , defined as the string that satisfies $f(x) = f(y)$ where $y = x \oplus s$. The function is the following linear map from 5 bits to 4 bits: $f(a, b, c, d, e) = (a + b, b + c, c + d, d + e)$ [120]. The presence of multiple Hadamard operations in this circuit makes it impossible to use techniques for sparse

matrices that are commonly used to reduce classical computation time by eliminating operations on zero elements.

We simulate the action of this circuit for 5 qubits (3+2), 9 qubits (5+4) and 13 qubits (7+6) to compare the scaling. Table 3.1 shows the Mathematica CPU runtime in ms (all calculations performed on the same computer). We can clearly see the advantage, while the runtime of the stabilizer algorithm scales as $O(n^2)$ as expected, the runtime for the matrix calculation scales exponentially in the number of qubits.

	5 qubits	9 qubits	13 qubits
Stabilizer	6.2	18.3	43.5
Matrix	6.2	322.9	443542

Table 3.1: Runtime(ms) in Mathematica

3.4 Algorithms for the visualisation of stabilizer codes

In this section, we present the algorithms derived from the work in [121, 122] and some extensions. These algorithms are described here so that they can be reproduced in any programming language, and they have been adapted for the most efficient implementation of Gottesman-Knill theorem, i.e. the CHP code [120]. The visualisation of these algorithms is built on Mathematica’s graph functionality to represent the cluster states. The full list of functions of this Mathematica package and their documentation can be found in the appendix D.

There exists in the literature a similar simulator that performs all operations on graph states [143], which can have a more compact description of the stabilizers (if and only if the graphs have low vertex degree with respect to the number of qubits) but the same time complexity as the *tableau* formalism [120]. In this algorithm, the stabilizer state is represented by the adjacency matrix of the graph and a series of vertex operators that act on individual qubits. It might be surprising at first that they use a unique graph representation to a stabilizer state whereas, as can be seen following the proof of theorem 3 in appendix C, one stabilizer state can have many valid graph representations (under the application of local Hadamards). The key is that one stabilizer state can have different graph and vertex operator representations, but they completely and without ambiguity define it. In the *tableau* representation, we do not use these extra vertex operators (which would account for the extra Hadamard operations) and therefore there isn’t a one to one correspondence between graph states and stabilizer states. In the simulator proposed in [143], one stabilizer state could be represented as a graph with vertex operators in different ways, but any one of those representations uniquely defines the stabilizer state.

The basic functions for the *tableau* simulator such as $H, P, CNOT$ and measurements in different basis are described in detail in the original paper [120]. There are many functions in the Mathematica package that have the purpose of dealing with the in-built functionality and will therefore not be described here. The following functions are of interest for the implementation of a full cluster state simulator, with the option to visualise the transformation of the stabilizer states as graph states.

Note that all operations described in the following algorithms are performed modulo 2.

Find basis change between two stabilizer states

Input: Two stabilizer states in their *tableau* form.

Procedure:

- Check that both stabilizer states have the same number of qubits, n .
- Define the $n \times n$ basis change matrix, R .
- Define Sx_1 and Sx_2 as the Sx sub-matrix of the *tableaus* for the two stabilizer states, following the notation from equation (3.25). Do similarly for Sz_1 and Sz_2 .
- Solve the system of equations:

$$\begin{aligned} R \cdot Sx_1 &= Sx_2, \\ R \cdot Sz_1 &= Sz_2, \\ \text{Det}[R] &\neq 0. \end{aligned}$$

- If a solution exists, identify the corresponding transformation of the stabilizers: $S'_i \rightarrow S_i \cdot S_j$.

Output: If the system of equations has a solution, output it in the form of the matrix R and as a transformation of the stabilizer, $S'_i \rightarrow S_i \cdot S_j$. If there is no solution to the system of equations, output a message conveying so.

Apply basis change

Input: A stabilizer in its *tableau* form and the basis change R as an $n \times n$ matrix.

Procedure:

- Identify from the *tableau* the number of qubits in the stabilizer, n . Check it corresponds with the dimensions of matrix R .
- Redefine the sub-matrices of the tableau as

$$\left(\begin{array}{c|c|c} \boxed{R \cdot Dx} & \boxed{R \cdot Dz} & R \cdot r_i \\ \hline \boxed{R \cdot Sx} & \boxed{R \cdot Sz} & R \cdot r_{n+i} \\ \hline 0 & 0 & 0 \end{array} \right). \quad (3.30)$$

Output: The stabilizer *tableau* after the basis change.

Find local Clifford equivalence between two stabilizer states

Input: Two stabilizer states in their *tableau* form.

Procedure:

3. SIMULATION OF STABILIZER COMPUTATIONS

- Identify from the *tableaus* the number of qubits in each stabilizer, n_1, n_2 . The number of qubits should be the same, $n_1 = n_2 = n$.
- Define Sx_1 and Sx_2 as the Sx sub-matrix of the *tableaus* for the two stabilizer states, following the notation from equation (3.25). Do similarly for Sz_1 and Sz_2 .
- Define the matrix Q that describes the local Clifford operation. It is formed of four diagonal matrices

$$Q = \begin{bmatrix} A & B \\ C & D \end{bmatrix}. \quad (3.31)$$

The elements of the diagonal matrices are labelled a_i, b_i, c_i, d_i respectively.

- Solve the set of equations given by:

$$\left(\begin{array}{c|c} Sz_1 & Sx_1 \end{array} \right) \cdot \begin{pmatrix} A & C \\ B & D \end{pmatrix} \cdot \begin{pmatrix} Sx_2 \\ Sz_2 \end{pmatrix} = 0, \quad (3.32)$$

$$a_i \cdot d_i + c_i \cdot b_i = 1. \quad (3.33)$$

The last set of equations ensures the unitarity of each local Clifford operation, which is given by $Q_i = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$.

- This set of equations can have many different solutions, each corresponding to a set of local operations that can be applied to the qubits in order to transform the state. Each solution is therefore a set of Q_i matrices (representing single qubit Clifford operations), one for each qubit. There are only 6 single qubit Clifford operations: $\{\mathbf{1}, H, P, PH, HP, HPH\}$. From the Q_i matrices the algorithm can identify the correct single qubit Clifford operation for each individual qubit.

Output: If the two stabilizer states are local Clifford equivalent, the algorithm outputs the list of all the possible local Clifford operations Q , both in matrix form and as a list of operations.

Apply local Clifford operation

Input: A stabilizer in its *tableau* form and the basis change Q as an $2n \times 2n$ matrix.

Procedure:

- Identify from the *tableau* the number of qubits in the stabilizer, n . Check the dimensions of matrix Q are correct.
- Define Sx, Dx, Sz and Dz from the *tableaus* for the stabilizer state, following the notation from equation (3.25).
- From matrix Q , identify the matrix blocks A, B, C, D .

- Redefine the sub-matrices of the tableau as

$$\left(\begin{array}{c|c|c} \boxed{Dz \cdot C + Dx \cdot D} & \boxed{Dz \cdot A + Dx \cdot B} & r_i \\ \hline \boxed{Sz \cdot C + Sx \cdot D} & \boxed{Sz \cdot A + Sx \cdot B} & r_{n+i} \\ \hline 0 & 0 & 0 \end{array} \right). \quad (3.34)$$

Output: The stabilizer tableau after the operation has been applied.

Trace out product state qubits

Input: A stabilizer state in its *tableau* form.

Procedure:

- Identify from the tableau the number of qubits in the stabilizer, n .
- Define the S matrix as the Sx and Sz sub-matrices of the *tableau*, following the notation from equation (3.25).
- Count the number of non-zero elements in each row of S . If there is only one this corresponds to a qubit in a product in the eigenvector of X or Z . If there are two non-zero elements, but they are on the same column of the sub-matrices Sx and Sz , the qubit is on an eigenstate of the Y operator. Make a list, l , with all qubits that are on an eigenstate of one of the Pauli operators. These are the qubits we will remove from the *tableau*.
- Remove the columns of the *tableau* that correspond to the qubits in list l .
- The removal of these qubits causes some rows in S to be all zeros. The number of rows that will be all zeros corresponds to the number of qubits removed. Delete those rows (stabilizer operators) and their corresponding destabilizer operators from the *tableau*. The number of stabilizers should now be the same as the number of qubits that remains in the cluster, as required for a well defined *tableau*.

Output: The stabilizer state containing only the qubits that are not in a product state. A message informs of which qubits from the original state have been removed and what qubits remain indicating the map between the old labelling and new labelling (as the formalism names the qubits $1 \rightarrow n$ according to their arrangement in columns).

Identify possible options to transform stabilizer state into graph state

Input: A stabilizer state in its *tableau* form.

Procedure:

- Identify from the tableau the number of qubits in the stabilizer state, n .
- Identify the rank k of the sub-matrix Sx , defined following the notation on equation (3.25).

3. SIMULATION OF STABILIZER COMPUTATIONS

- If $k = n$, the graph state associated with is uniquely defined and no Hadamard gates need to be applied to any qubit.
- If $k < n$, we run a sub-routine that finds all possible sub-matrices of Sx with rank k . For simplicity this sub-routine should first perform Gaussian elimination on the matrix Sx . If there exists only one sub-matrix with rank k , then the graph state is uniquely defined. If there exist more than one sub-matrix that fulfils this property, the sub-routine should output all the combinations of columns (which correspond to qubits) that form a sub-matrix of rank k together with the first k rows (as we performed Gaussian elimination, full rank sub-matrices will always be in the first k rows).
- If $k < n$, we determine which qubits require the application of a Hadamard gate in order to convert the stabilizer state into a graph state. These qubits correspond to the columns that weren't part of the full-rank sub-matrices obtained above. There are as many transformation options as full-rank sub-matrices.

Output: The algorithm outputs if the graph state associated with the stabilizer state is uniquely defined or not. If $k = n$ there is no further output. If $k < n$, the algorithm outputs a series of lists. Each list contains the qubits on which we need to perform Hadamards in order to convert the stabilizer state into a graph state.

Comments: The sub-routine that finds all possible sub-matrices of Sx with rank k is by far the most computationally complex function in the entire implementation of the code. Finding one full rank sub-matrix is an efficient process if one uses QR decomposition algorithms, such as Gram-Schmidt decomposition [144] or Householder reflections [145], both of which have a computational complexity of $O(n^3)$, where n is the number of columns and rows.

The problem of finding all possible full rank $k \times k$ sub-matrices requires finding all possible combinations of rows and columns that yield an invertible matrix. The search space becomes exponentially large, as it scales with the binomial coefficient $\binom{n}{k}$ which scales exponentially in n .

This is the only algorithm in the entire simulation code for which the scaling is not polynomial.

Transform stabilizer state into graph state

Input: A stabilizer state in its *tableau* form and a list of qubits. This list will be empty if the sub-matrix Sx is invertible (see output from previous function).

Procedure:

- Identify from the tableau the number of qubits in the stabilizer state, n .
- Apply Hadamard to the qubits given in the input list (if there are any). This will make the submatrix Sx full rank.
- Calculate the inverse *Inv* matrix of the Sx submatrix.

- Build a tableau were Dx' is an all zero $n \times n$ matrix, Dz' is an $n \times n$ identity matrix, $Sx' = Inv \cdot Sx$ which should correspond to an identity matrix and $Sz' = Inv \cdot Sz$. Add the phases column set to zero and the extra scratch-space row.

Output: The stabilizer in *tableau* form, such as $Sx = \mathbf{1}_{n \times n}$ and $Sz = \theta_{ij}$ is the adjacency matrix of a simple graph.

Comments: The graph state representation of a stabilizer state has no information of the phase of the stabilizer operators, hence the phase column can be set to zero.

Implement an arbitrary single qubit Clifford gate given in the Heisenberg representation

Input: A stabilizer state in its *tableau* form, the qubit on which the operation will be performed, k , and a list of transformation rules given by : $\{X, Y, Z\} \rightarrow \{\pm\sigma_a, \pm\sigma_b, \pm\sigma_c\}$, where a, b, c are used to indicate the corresponding Pauli operator.

Procedure:

- Identify from the tableau the number of qubits in the stabilizer state, n .
- Transform the list of rules into *if* statements for each of the operators represented in binary.
- Loop over all stabilizers (rows of the *tableau*). For each row, read the operator corresponding to qubit k (given by the combination of columns k and $k+n$) and apply the corresponding rules by executing the binary *if* statements. The overall phase of each operator (given by column $2n+1$) should also be updated accordingly.

Output: The stabilizer in *tableau* form after the arbitrary single-qubit Clifford gate has been applied.

Comments: It can be noted that as the operator $Y = iXZ$, it shouldn't be necessary to specify what is the transformation for the operator Y given the other two. However, the phase of the operator Y posed a problem as X and Z do not commute but we cannot enforce this anti-commutation in the *tableau* formalism, hence the extra condition of requiring transformation rules for Y .

Implement an arbitrary two-qubit Clifford gate given in its Heisenberg representation

Input: A stabilizer state in its *tableau* form, the two qubits on which the operation will be performed, i and j , and two lists of transformation rules given by : $\{\mathbf{1}_i \otimes \sigma_{X_j}, \mathbf{1}_i \otimes \sigma_{Y_j}, \mathbf{1}_i \otimes \sigma_{Z_j}\} \rightarrow \{\pm\sigma_{a_i} \otimes \sigma_{a_j}, \pm\sigma_{b_i} \otimes \sigma_{b_j}, \pm\sigma_{c_i} \otimes \sigma_{c_j}\}$, and $\{\sigma_{X_i} \otimes \mathbf{1}_j, \sigma_{Y_i} \otimes \mathbf{1}_j, \sigma_{Z_i} \otimes \mathbf{1}_j\} \rightarrow \{\pm\sigma_{a_i} \otimes \sigma_{a_j}, \pm\sigma_{b_i} \otimes \sigma_{b_j}, \pm\sigma_{c_i} \otimes \sigma_{c_j}\}$ where a, b, c are used to indicate the corresponding Pauli operator and i, j the corresponding qubit.

Procedure:

- Identify from the tableau the number of qubits in the stabilizer state, n .

- Transform both lists of rules into *if* statements for each of the operators represented in binary.
- Loop over all stabilizers (rows of the *tableau*). For each row:
 - Copy the row onto the scratch space at the bottom of the *tableau*.
 - Read the operator corresponding to qubits i and j (given by the combination of columns $i - i + n$ and $j - j + n$).
 - Find which rules or combination of rules apply: if one of the operators is the identity, we will only need one rule, but if none is the identity, find a rule from the first set that has the same operator as qubit j and one from the second set that has the same operator as qubit i .
 - Apply rule from the second set on the row and the rule from the first set onto the copy of the stabilizer in the scratch space.
 - Determine which operators are represented in qubits i and j of both copies of the operator, from a lookup table in the programme determine any phase adjustments necessary to account for anti-commutation (i.e. $ZX = -XZ = -Y$).
 - Adjust the overall phase if the transformation rules requires it.

Output: The stabilizer in *tableau* form after the arbitrary two-qubit Clifford gate has been applied.

3.5 Discussion and outlook

In this chapter we have reviewed the stabilizer formalism [116], which is a Heisenberg representation of a certain class of quantum operations [117]. We have focused on a binary representation of the algorithm that allows an efficient classical simulation of the formalism [118]. A significant result from the literature [121] is that all stabilizer states can have a graph representation and therefore the quantum operations can be understood as a series of graph transformations. When the stabilizer operations are simulated classically, this graphical representation allows to follow a series of stabilizer operations as the evolution of the connectivity of a graph. This provides a very helpful intuition of the action of different quantum operations and a novel way to do calculations, as we can use not only stabilizer operations but also graph operations to understand equivalences between states and operations. For this purpose, we have presented a series of algorithms that turn the theoretical results from the literature [122, 121, 123, 134] into algorithms for the most efficient classical simulator of the stabilizer formalism, Aaronson's and Gottesman's CHP code [120]. We have implemented this code on Mathematica; the code's main functions can be seen in chapter D.

The most important application of the stabilizer formalism is the description of quantum error correction codes. The simulator based on the algorithms presented in this chapter (and which is described in detail in appendix D) has proved extremely useful to understand some quantum error-correcting codes and to obtain some of the results presented in chapter 7.

CHAPTER 4

GENERATING PHOTONIC STATES

Achilles had overtaken the Tortoise, and had seated himself comfortably on its back.

“So you’ve got to the end of our race-course?” said the Tortoise. “Even though it DOES consist of an infinite series of distances? I thought some wiseacre or other had proved that the thing couldn’t be done?”

What the Tortoise said to Achilles.
LEWIS CARROLL

4.1 Introduction

In chapter 2 we reviewed most of the linear optical protocols for quantum computing, and although quite different, they all require one key ingredient: small entangled states. As we have seen previously, two-qubit entangling operations cannot be performed deterministically on photons and therefore the preparation of these states is extremely challenging. So far only generation of two and three qubit heralded entangled states have been experimentally demonstrated¹ [147, 148]. In this chapter we present a series of theoretical schemes to generate n -photon GHZ states from single photon sources and give a full account of the success probability and resource cost of these schemes.

After explaining briefly the key concepts of parametric down-conversion sources, we focus on the optical implementation of Bell-state measurement. Recently proposed new schemes [111, 115] can achieve higher than 50% probability of success by using non-vacuum ancillary modes. We use adapted versions of these schemes to increase the probability of generating n -GHZ states from single photons, which are heralded from parametric down-conversion sources². We present different schemes, which can use Bell pairs or single photons as input states, and can have different success probabilities depending on the number of resources used and the level of loss tolerance required. We finally compare the cost involved in a near-deterministic generation

¹Here we are referring to schemes where each photon represents a qubit, and not experiments such as [146], where one photon can be the physical support of different qubits in its different degrees of freedom.

²Schemes such as proposed in [149], which use correlated parametric down-conversion processes, are not studied in this chapter. They could provide a new interesting approach and, in conjunction with some of the techniques presented in this chapter, could provide a new avenue for the generation of small entangled states.

when using multiplexed schemes.

4.2 Conventions

Throughout this chapter we will use certain conventions that we introduce here.

- We encode qubits in the polarisation degree of freedom of photons³, the notation used for the creation operators is as introduced in chapter 2: the operators will be written as $h_i^m v_j^n$, where the subscript indicates the spatial optical mode, the letter of the operator (h or v) indicates the logical (polarisation) state and the superscript indicates the number of photons with said polarisation in that optical mode. The eigenstates of the X operator will also have a specific letter associated, as this will simplify many equations in the chapter: $|+\rangle_i = (h_i + v_i)/2 = p_i$ and $|-\rangle_i = (h_i - v_i)/2 = m_i$.
- We will represent states by the action of the creation operators for the different modes acting on the vacuum, where the vacuum should be understood at the end of all state equations. For example, the Bell state $|\phi^+\rangle$ will be represented as

$$|\phi^+\rangle_{1,2} = \frac{1}{\sqrt{2}} (|H_1 H_2\rangle + |V_1 V_2\rangle) \rightarrow \frac{1}{\sqrt{2}} (h_1 h_2 + v_1 v_2). \quad (4.1)$$

- The labelling convention we will use is that optical modes continue in straight lines through the optical circuits. This is compatible with the definition of the action of the PBS in chapter 2, i.e. the spatial mode is transmitted through the PBS while the vertical mode is reflected.
- In order to understand the optical diagrams better, we will colour-code the most used operations: 45°-rotated PBS, Type-I fusion gate and Type-II fusion gate, which can be seen in figure 4.1.

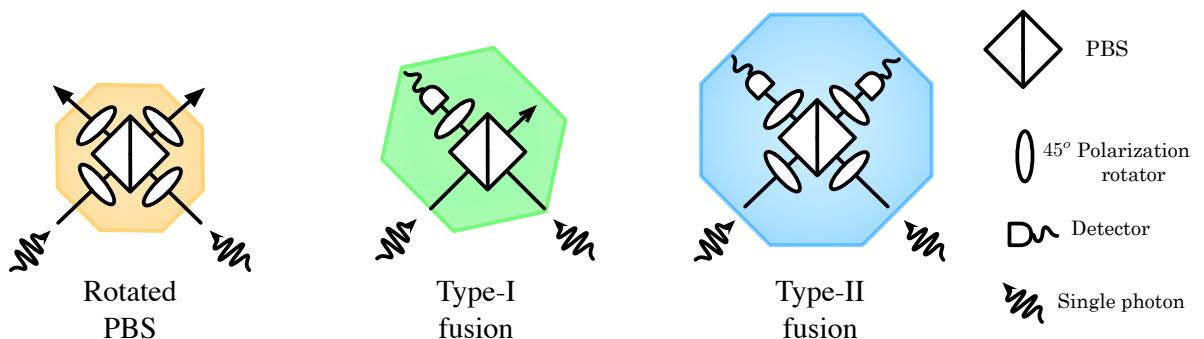


Figure 4.1: Colour convention for the rotated PBS and the fusion gates.

³As highlighted in chapter 2, path encoding and polarisation are equivalent representations that can be easily interconverted. Therefore, all protocols stated in this chapter are equally valid for path encoded systems. We have chosen to work with polarisation as the representation of the linear optical circuits is more compact.

- All detectors represented in the linear optical circuits of this chapter are assumed to be capable of distinguishing polarisation and number of photons. In the case of polarisation, this can easily be performed experimentally by substituting each detector by a PBS that leads on to two detectors, each of which will detect photons of one polarisation. High-efficiency number-resolving detectors [60] are possible, although not yet widely available, but other techniques such as cascading arrays of photon detectors [59] can be used to resolve the number of photons in each mode.
- When studying the effect of different linear optical circuits on an input state, it is useful to expand the wave function as a summation of terms, for example

$$|\psi^-\rangle_{1,2}|\psi^-\rangle_{3,4} = \frac{1}{2}h_2h_4v_1v_3 - \frac{1}{2}h_1h_4v_2v_3 - \frac{1}{2}h_2h_3v_1v_4 + \frac{1}{2}h_1h_3v_2v_4. \quad (4.2)$$

We will refer to the individual terms of the input state wavefunction as “initial terms” and to the individual terms in the output state as “final terms”. This convention is useful for the analysis of many linear optical circuits. In many, it is necessary to understand which initial terms generated certain final terms in order to optimise photonic entangled state generation.

4.3 Photon sources and entanglement operations

Entanglement is a crucial resource for quantum computation, albeit a difficult one to attain for linear optical systems. We want to achieve entanglement between qubit modes, however as mentioned in chapter 2 it is not possible to achieve this deterministically. It is important to distinguish the type of entanglement we require for LOQC protocols, which is at the level of the mode operators used to describe qubits [63], from other types of entanglement such as NOON states or squeezed states. Details on these other types of entangled photonic states can be found in [150, 63]. In this section we will briefly introduce heralded photon sources and Bell-state measurements, which will be used in subsequent sections to generate entangled photonic states.

4.3.1 Spontaneous parametric down-conversion sources

Spontaneous parametric down conversion (SPDC) is a non-deterministic process that is widely used to produce high-quality photon sources⁴. In the simplest terms, a nonlinear crystal is stimulated with a pump beam photon, which leads to the spontaneous appearance of two correlated photons in the output modes, which are historically called signal and idler [150]. There are two types of SPDC processes: type-I in which the polarisation of the signal and idler photons is the same and orthogonal to that of the pump, and type-II where the signal and idler photons have orthogonal polarisations. The Hamiltonian describing the down-conversion process is that of a two-mode squeezer [63] and in the case of a type-II SPDC source, the state

⁴A thorough review of other mechanisms to generate single photons can be found in [58].

created is given by

$$S(\lambda)|0\rangle = \sqrt{1 - \lambda^2} \sum_{n=0}^{\infty} \lambda^n |\phi_n\rangle \quad (4.3)$$

where

$$|\phi_n\rangle = \frac{1}{\sqrt{n+1}} \sum_{m=0}^n (-1)^m h_S^m v_S^{n-m} h_I^{n-m} v_I^m. \quad (4.4)$$

The parameter λ of the down-conversion depends on the intensity of the incident laser but usually is very small, thus the lower order terms have the most contribution. In particular, when only the first and second order terms are present the state generated is

$$S(\lambda)|0\rangle \simeq \sqrt{1 - \lambda^2}|0\rangle + \frac{\lambda}{2} (h_I v_S + v_I h_S) = |\psi^+\rangle_{I,S}, \quad (4.5)$$

which is a superposition of vacuum and a maximally entangled state. One of the most common uses of this type of sources is to produce heralded single photons, as putting a photon detector on the signal mode will herald the presence of a single photon on the idler mode. Bell states can also be heralded if higher order terms of the expansion are used [151]. This is done by combining two beam-splitters with a two-fold detection of the third order term ($O(\lambda^3)$).

In this brief introduction we have aimed to summarise the key concepts which are required to understand the content in upcoming sections. A detailed explanation of the non-linear physical processes in SPDC sources can be found in [150].

4.3.2 Bell measurements in Linear Optics

A Bell state measurement (BSM) is the projection of two qubits onto maximally entangled states (Bell states). It is a crucial feature of many quantum protocols such as quantum computation [32], quantum teleportation [35] and quantum communication [152]. However, as we have seen in chapter 2, it is not possible to perform a deterministic two-qubit entangling gate in linear optics and hence a deterministic BSM is not possible either. In this section we focus on how probabilistic BSM can be performed and what success probability can be attained.

Optical Bell state measurement

Braunstein and Mann first proposed [153] a scheme to measure the optical version of the Bell operator by generalising the HOM interferometer [69] to allow for states with arbitrary polarisations. The set up requires a beam-splitter that implements the mode transformations:

$$h_1 \rightarrow \frac{1}{\sqrt{2}} (h_1 + i h_2), \quad v_1 \rightarrow \frac{1}{\sqrt{2}} (v_1 + i v_2), \quad (4.6)$$

$$h_2 \rightarrow \frac{1}{\sqrt{2}} (h_2 + i h_1), \quad v_2 \rightarrow \frac{1}{\sqrt{2}} (v_2 + i v_1), \quad (4.7)$$

and two detectors that distinguish polarisation and photon number, see figure 4.2. This linear optical device allows to unambiguously distinguish two of the four Bell states. To understand why this is the case, we look at how the four Bell states transform under the action of the

beam-splitter:

$$|\psi^+\rangle = \frac{1}{\sqrt{2}} (h_1 v_2 + v_1 h_2) \xrightarrow{BS} \frac{i}{\sqrt{2}} (h_1 v_1 + h_2 v_2) \quad (4.8)$$

$$|\psi^-\rangle = \frac{1}{\sqrt{2}} (h_1 v_2 - v_1 h_2) \xrightarrow{BS} \frac{1}{\sqrt{2}} (h_1 v_2 - h_2 v_1) \quad (4.9)$$

$$|\phi^\pm\rangle = \frac{1}{\sqrt{2}} (h_1 h_2 \pm v_1 v_2) \xrightarrow{BS} \frac{1}{\sqrt{2}} (h_1^2 + h_2^2 \pm v_1^2 \pm v_2^2) \quad (4.10)$$

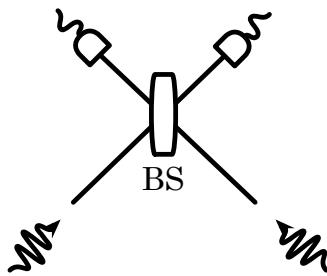


Figure 4.2: Optical scheme to measure in the Bell basis. Two of the four Bell states can be unambiguously discriminated using this setup.

We can immediately see why only two states are distinguishable. The state $|\psi^-\rangle$ is invariant under the action of the beam-splitter and retains one photon in each spatial mode, being the only one of the four Bell states which does so. The beam-splitter transformation on the states $|\psi^+\rangle, |\phi^\pm\rangle$ bunches both photons in the same spatial mode, however in the case of $|\psi^+\rangle$ the photons have different polarisations while in the case of $|\phi^\pm\rangle$ they have the same polarisation. Hence, when two photons of the same polarisation are detected in the same spatial mode, it is not possible to determine which of the two $|\phi^\pm\rangle$ was in the input.

It is possible to tailor which two of the four Bell pairs are distinguishable, as linear optical operations allow the transformation of every Bell state into every other. Therefore, *any two* of the four Bell states can be unambiguously discriminated using a beam-splitter, but *only two* of them. Considering equiprobable Bell states, this means that the success probability of the BSM is 50%.

A complete BSM using only linear optical elements and vacuum ancillary states has been shown impossible [114]. In fact, for setups that use linear optical elements, classical feed-forward, perfect number-resolving detectors and vacuum ancillary modes, the maximum efficiency of the BSM is 50% [154]. The key point to realise is that these proofs are only valid when we don't allow any ancillary states. A way to improve the success probability has already been presented in chapter 2, the KLM scheme [2] proves that the success probability can be increased up to unity by using entangled states in ancillary modes, in combination with a rather complicated protocol. However, recently two schemes have been proposed which implement the Bell measurement operator with a probability higher than 50% using much simpler interferometers than KLM.

Boosting success probability with ancilla states

The no-go theorem for performing BSMs in linear optics with higher than 50% probability [114, 68] assumes that only vacuum ancillary modes are used. It was recently shown that introducing entangled ancilla pairs or single photons improves the success rate to 75%. Moreover, the introduction of 2^m occupied ancillary modes yields a Bell-state measurement with a success rate $1 - 2^{-(m+1)}$. It must be noted however that this does not mean the introduction of 2^{m-1} entangled pairs, but the introduction of large m -photon entangled states.

Grice first showed [111] that using interference with ancillary photons, a Bell measurement can be boosted to be arbitrarily complete. This boosting happens in stages, starting with non-boosted BSM which succeeds with probability 50%, which we will call stage $m = 0$, to a series of boosted BSM with success probability $1 - \frac{1}{2^{m+1}}$. The way interference improves the success probability can be understood from studying the simplest case, i.e. interference with a Bell pair, which is shown in figure 4.3.

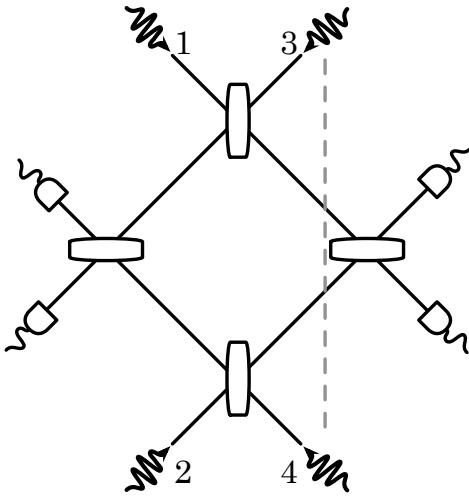


Figure 4.3: Improved scheme to measure in the Bell basis, using a Bell pair ancilla (photons 3 and 4). Two Bell states can be unambiguously discriminated with 100% success probability, and the two other with 50% success probability, yielding a scheme with overall success probability of 75%, when considering equiprobable Bell states. Figure adapted from [111], copyright (2011) by the APS.

In the case of the BSM boosted with a Bell pair in figure 4.3, the degeneracy of the detection pattern for $|\phi^\pm\rangle_{1,2}$ is reduced due to the interference with a Bell pair $|\phi^+\rangle_{3,4}$ (using $|\phi^-\rangle_{3,4}$ would yield similar results). We define n_H and n_V as the total number of photons with horizontal and vertical polarisation respectively and n_i as the number of photons in spatial mode i . The

detection patterns that yield distinguishable states are:

$$n_H \& n_V \text{ odd}, n_1 + n_3 \text{ even} \rightarrow |\psi^+\rangle_{1,2} |\phi^+\rangle_{3,4}, \quad (4.11)$$

$$n_H \& n_V \text{ odd}, n_1 + n_3 \text{ odd} \rightarrow |\psi^-\rangle_{1,2} |\phi^+\rangle_{3,4}, \quad (4.12)$$

$$n_H \& n_V \text{ even}, n_H = 2 \& n_1 + n_2 \text{ even} \rightarrow |\phi^+\rangle_{1,2} |\phi^+\rangle_{3,4}, \quad (4.13)$$

$$n_H \& n_V \text{ even}, n_H = 2 \& n_1 + n_2 \text{ odd} \rightarrow |\phi^-\rangle_{1,2} |\phi^+\rangle_{3,4}. \quad (4.14)$$

Due to the interference with the Bell pair in modes 3 and 4, the measurement outcomes for $|\phi^\pm\rangle_{1,2}$ are of two kinds: either all photons have the same polarisation, i.e. $n_H = 4$ or $n_V = 4$, or half the photons are horizontally polarised and half vertically polarised, i.e. $n_H = n_V = 2$. In this latter case, $n_1 + n_2$ is even for $|\phi^+\rangle_{1,2}$ and odd for $|\phi^-\rangle_{1,2}$. As the probability of obtaining a detection pattern with $n_H = 2$ for $|\phi^\pm\rangle_{1,2}$ is 50%, it follows that $|\phi^\pm\rangle_{1,2}$ can be unambiguously distinguished with 50% probability. Thus, the success rate in distinguishing equiprobable Bell states goes from 50% to 75%.

This process can be repeated in stages using increasingly complicated interferometers. Each stage also requires an increasing number of resources, where the resources are increasingly bigger GHZ states, for boosting to a probability $1 - \frac{1}{2^{m+1}}$ we need resources

$$\{2^{m-1} \times \frac{|0\rangle^{\otimes 2} + |1\rangle^{\otimes 2}}{\sqrt{2}}, 2^{m-2} \times \frac{|0\rangle^{\otimes 4} + |1\rangle^{\otimes 4}}{\sqrt{2}}, \dots, \frac{|0\rangle^{\otimes 2^m} + |1\rangle^{\otimes 2^m}}{\sqrt{2}}\}. \quad (4.15)$$

As was shown in [111], this process can be iterated, with each iteration making half of the remaining indistinguishable initial terms distinguishable, hence the probability is increased by $\frac{1}{2^m}$ in the m^{th} iteration⁵. It can be noted that the size of the resources increases exponentially, and it must also be noted that the resources required to prepare these ancilla states also increase exponentially (this will be shown later in the chapter). For example, to achieve a success probability of $\sim 97\%$, the scheme would require the use of 30 entangled photons, making the scheme highly impractical. However, the first stage of this improvement process, which only requires a Bell pair, has been shown to be very useful (see chapter 5).

It was realised by Ewert and van Loock [115] that in fact, at least for the first stage of the boosting process, it is not necessary to require a Bell pair, and single photons are enough to boost the success probability. In this case, instead of inputting one photon per mode however, there are two photons, one in vertical and one in horizontal polarisations, in modes 3 and 4. This BSM scheme can be seen in figure 4.4. As there are more photons, the measurement patterns are slightly different, but they are in the same spirit as Grice's: at each stage the indistinguishability of $|\phi^\pm\rangle$ is reduced by half. Just as in Grice's scheme, more complex interferometers and larger entangled states⁶ are required to push the success probability closer to unity. Although this improved BSM achieves 75% success probability in the first stage, it was shown [115] that using only single photons the success probability could be further boosted to 78.125% by using the

⁵This process of attempting to reach 100% success probability by taking smaller and smaller steps at each stage is reminiscent of Zeno's paradox of Achilles and the tortoise.

⁶To achieve the same scaling as Grice's scheme, Ewert and van Loock also require entangled states for stages with $m > 1$.

single photons in the interferometer for the $m = 2$ boosting stage. This was found by numerical simulation and is not an extension of the analytical proof, hinting that although these schemes improve BSM, they are not optimal strategies.

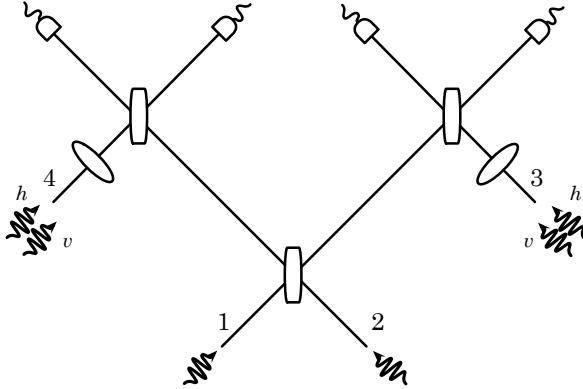


Figure 4.4: Improved scheme to measure in the Bell basis, using 4 single ancillary photons. Two Bell states can be unambiguously discriminated with 100% success probability, and the two other with 50% success probability, yielding a scheme with overall success probability of 75% when considering equiprobable Bell states.

The proof of scaling in this new strategy [115] follows the same steps as [111] so it can be easily understood that these two proposals are intimately related. What is not so obvious is that both schemes are performing the same unitary operation on the input state, and they only differ in the ancilla state used. On first inspection, the interferometers are obviously different. Grice's scheme for the stage $m = 1$ implements the series of linear optical operations: $BS_{1,3} \rightarrow BS_{2,4} \rightarrow BS_{1,2} \rightarrow BS_{3,4}$ where $BS_{i,j}$ indicates a beam-splitter operation in modes i, j and the arrows show the order of the operations. It can be checked that the operation performed on the optical modes is equivalent to doing the same beam-splitter operations in a different order, i.e. $BS_{3,4} \rightarrow BS_{1,2} \rightarrow BS_{1,3} \rightarrow BS_{2,4}$. Looking at the operations in this order, we can see that $BS_{3,4}$ acts on the ancilla modes before any interference takes place with photons 1 & 2, and can therefore be absorbed in the preparation of the ancilla state. The other three beam-splitters now implement exactly the operation of Ewert and van Loock's interferometer.

The fact that both these schemes are actually the same but using different ancilla states seems to hint that there is an *equivalence* between these states. We have found a task for which one Bell pair and 4 single photons are equivalent resources. Understanding this equivalence better could lead to the design of even more efficient BSM schemes which would benefit LOQC enormously.

4.3.3 Fusion gates

In chapter 2 we introduced the fusion gates [90], which are crucial to the resource efficiency improvement of the Browne-Rudolph protocol for LOQC. We also explained their action on the formation of linear and two-dimensional cluster states. In this section, we review these gates in more detail, focusing on the mode transformation they implement on input photons. In particular, we justify the mapping of the different detection patterns to the projections on

the qubit subspace. This mapping was briefly provided in [90], we reproduce it here in detail to prove the validity of the procedure used to determine the mapping. This same procedure is used later in the chapter to determine the mapping of rotated and boosted fusion gates. In figure 4.5 we reproduce the fusion gates as linear optical circuits in the polarisation basis.

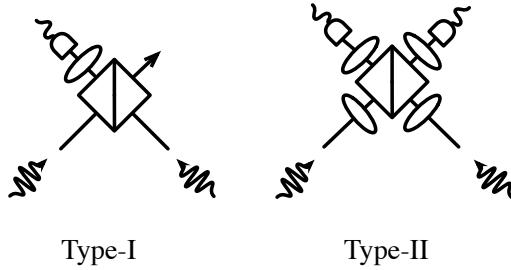


Figure 4.5: Type-I and Type-II fusion gates in polarisation basis. Figure adapted from [90], copyright (2005) by the APS.

The Type-II fusion gate [90] performs a successful fusion when one only one photon is measured at each detector and fails when both photons are detected in the same mode. A successful fusion on two modes, 1 and 2, is equivalent to a projection of the measured photons in the subspace:

$$\left\{ \frac{|++\rangle + |--\rangle}{\sqrt{2}}, \frac{|++\rangle - |--\rangle}{\sqrt{2}} \right\}_{1,2} \quad (4.16)$$

and the exact state in the subspace can be known from the measurement outcomes:

$$h_1 h_2 \text{ and } v_1 v_2 \rightarrow \frac{|++\rangle + |--\rangle}{\sqrt{2}}, \quad (4.17)$$

$$h_1 v_2 \text{ and } v_1 h_2 \rightarrow \frac{|++\rangle - |--\rangle}{\sqrt{2}}. \quad (4.18)$$

A failed fusion is equivalent to a projection of the measured qubits on the subspace:

$$\{|+-\rangle, |-+\rangle\}_{1,2} \quad (4.19)$$

with

$$h_1^2 \text{ and } v_1^2 \rightarrow |+-\rangle, \quad (4.20)$$

$$h_2^2 \text{ and } v_2^2 \rightarrow |-+\rangle. \quad (4.21)$$

We can justify these mappings by studying the evolution of the detected outcomes (final terms) through the Type-II fusion gate “in reverse”, i.e. using the detected outcomes as inputs on the time-reversed Type-II fusion gate to see what are the initial terms that have generated

them:

$$h_1 h_2 \rightarrow \frac{1}{2} (p_1 p_2 + m_2 p_2 + m_1 p_1 + m_1 m_2), \quad h_1^2 \rightarrow \frac{p_1^2}{2} + m_2 p_1 + \frac{m_2^2}{2}, \quad (4.22)$$

$$h_1 v_2 \rightarrow \frac{1}{2} (p_1 p_2 + m_2 p_2 + m_1 p_1 + m_1 m_2), \quad h_2^2 \rightarrow \frac{p_2^2}{2} + m_1 p_2 + \frac{m_1^2}{2}, \quad (4.23)$$

$$v_1 h_2 \rightarrow \frac{1}{2} (p_1 p_2 - m_1 p_1 + m_2 p_2 - m_1 m_2), \quad v_1^2 \rightarrow \frac{p_1^2}{2} - m_2 p_1 + \frac{m_2^2}{2}, \quad (4.24)$$

$$v_1 v_2 \rightarrow \frac{1}{2} (p_1 p_2 + m_1 p_1 - m_2 p_2 - m_1 m_2), \quad v_2^2 \rightarrow \frac{p_2^2}{2} - m_1 p_2 + \frac{m_1^2}{2}. \quad (4.25)$$

Whenever we apply a fusion gate to a pair of photons it is *assumed* that there is only one photon per mode, thus the terms that have more than one photon in any input mode (such as $p_a m_a$) cannot possibly have triggered the detectors, therefore we have:

$$h_1 h_2 \rightarrow \frac{1}{2} (p_1 p_2 + m_1 m_2), \quad h_1^2 \rightarrow m_2 p_1, \quad (4.26)$$

$$h_1 v_2 \rightarrow \frac{1}{2} (p_1 p_2 + m_1 m_2), \quad h_2^2 \rightarrow m_1 p_2, \quad (4.27)$$

$$v_1 h_2 \rightarrow \frac{1}{2} (p_1 p_2 - m_1 m_2), \quad v_1^2 \rightarrow -m_2 p_1, \quad (4.28)$$

$$v_1 v_2 \rightarrow \frac{1}{2} (p_1 p_2 - m_1 m_2), \quad v_2^2 \rightarrow -m_1 p_2, \quad (4.29)$$

which shows the correspondence shown previously.

To do a similar procedure for the Type-I gate we have to consider the final terms including the mode that is not measured (mode 2 in this calculation). This gate succeeds when one and only one photon is detected in mode 1 and fails when two photons are detected.

$$h_1 \rightarrow |0\rangle\langle 00| + |1\rangle\langle 11|, \quad (4.30)$$

$$v_1 \rightarrow |0\rangle\langle 00| - |1\rangle\langle 11|, \quad (4.31)$$

$$h_1^2 \rightarrow |01\rangle, \quad (4.32)$$

$$v_1^2 \rightarrow -|01\rangle. \quad (4.33)$$

We follow the same procedure as before to justify this mapping. We first input the final terms into the reversed Type-I gate:

$$h_1 h_2 \rightarrow \frac{1}{\sqrt{2}} (h_1 h_2 + h_2 v_2), \quad h_1^2 \rightarrow h_1 v_2 + \frac{h_1^2}{2} + \frac{v_2^2}{2}, \quad (4.34)$$

$$h_1 v_2 \rightarrow \frac{1}{\sqrt{2}} (v_1 v_2 + h_1 v_1), \quad h_2^2 \rightarrow h_2^2, \quad (4.35)$$

$$v_1 h_2 \rightarrow \frac{1}{\sqrt{2}} (h_1 h_2 - h_2 v_2), \quad v_1^2 \rightarrow -h_1 v_2 + \frac{h_1^2}{2} + \frac{v_2^2}{2}, \quad (4.36)$$

$$v_1 v_2 \rightarrow \frac{1}{\sqrt{2}} (h_1 v_1 - v_1 v_2), \quad v_2^2 \rightarrow v_1^2. \quad (4.37)$$

Restricting the initial terms to those that only have one photon per mode we have:

$$h_1 h_2 \rightarrow \frac{h_1 h_2}{\sqrt{2}}, \quad h_1^2 \rightarrow h_1 v_2, \quad (4.38)$$

$$h_1 v_2 \rightarrow \frac{v_1 v_2}{\sqrt{2}}, \quad h_2^2 \rightarrow 0, \quad (4.39)$$

$$v_1 h_2 \rightarrow \frac{h_1 h_2}{\sqrt{2}}, \quad v_1^2 \rightarrow -h_1 v_2, \quad (4.40)$$

$$v_1 v_2 \rightarrow \frac{-v_2 v_2}{\sqrt{2}}, \quad v_2^2 \rightarrow 0. \quad (4.41)$$

We can see that detecting h_1 heralds the projection $h_1 h_2 + v_1 v_2$ and detecting v_1 heralds $h_1 h_2 - v_1 v_2$. Therefore, the successful detection of one photon in mode 1 implements the Kraus operators $|0\rangle\langle 00| \pm |1\rangle\langle 11|$, where the sign is determined by the polarisation of the mode detected. When two photons are detected in mode 1, this is equivalent to a measurement in the computational basis of the input modes.

The fusion gates perform an entangling operation with 50% probability, and when they fail they measure the qubits involved in the X (Type-II) or Z (Type-I) bases. The success probability of these gates can be improved by using the boosted BSM schemes presented earlier, we present new boosted fusion gates in the following section. Moreover, the measurement basis in which photons are measured in the success and failure cases can be modified to adapt it for different scenarios where the original gates might not be performing the optimal entangling gate. An example is the generation of a large cluster state from micro-clusters, as proposed in Kieling *et al*'s LOQC protocol. Their scheme suggests using Type-I fusion gates, which are not loss tolerant, however Type-II does not perform the required entangling operation. Variations on the Type-II gate are studied in section 4.6.

Boosted Fusion gates

We have presented results which have shown how the probability of successfully implementing BSMs in linear optics can be improved to different extents by using resources such as Bell pairs [111] or single photons [115]. As the Type-II fusion gate is in essence a rotated BSM, they too can be improved. We present two boosted fusion gates that perform the exact same projection as the original fusion gates presented in the previous section, but with a boosted success probability of 75%. These gates can be seen in figure 4.6.

In figure 4.6 we present two versions of the Type-II boosted gate. These proposals are based on [111, 115] and therefore require one Bell pair and four single photons respectively to boost the success probability to 75%. All the photons are measured (the two input photons and the ancillary Bell state) hence the boosted gate is loss tolerant in the same was as the original Type-II, and the success or failure of the gate is given by the detection pattern. These detection patterns for the boosted version are the same as the ones in the BSM schemes they are based on and we will omit them here. It is worth mentioning that both these proposals inherit the loss-tolerance of the original Type-II fusion gate.

We have mentioned boosting the Type-II fusion gate but not Type-I. Type-I is not a full BSM as not all photons that enter the gate are measured, therefore it is not possible to use the

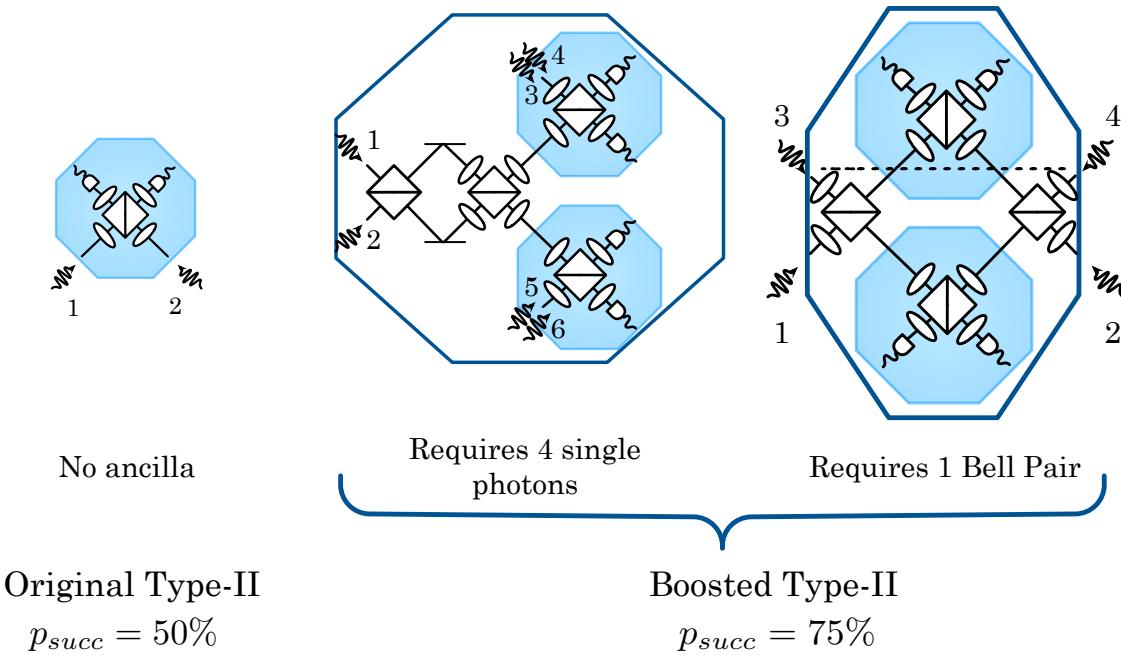


Figure 4.6: Boosted Type-II fusion gate. Photons 1 and 2 represent the photons on which the gate is applied, while photons 3 and 4 are the ancillary states. As they are adaptations of the boosted BSM schemes by Grice and Ewert-van Loock, they use a Bell pair and 4 single photons as resource states. Both these gates have the exact same success and failure outcomes as the original Type-II fusion gate and only their success probabilities differ. Here and in future figures, the navy octagon represents the boosted Type-II fusion gate.

same boosting technique as in Type-II. It is possible that improvements can also be made to improve the success probability of the Type-I fusion gate, but they have not been found yet.

4.4 Ballistic circuits for generation of small entangled states

In this section we present a series of linear optical circuits that project an initial state into a GHZ state by using a series of Type-I and Type-II gates. The original results in this section are based on the Bell pair generator and 3-GHZ generator proposed in [92, 93, 147]. We propose generalised circuits to generate n -GHZ states and improve their success probability by using the boosted fusion gates introduced in the previous section.

4.4.1 Bell pair generation from single photons

The scheme for generating “event-ready” entangled pairs was first proposed by Zhang *et al.* [147]. The scheme, shown in figure 4.7 (a), produced Bell pairs with probability $p_s = 3/16$ upon detection of two photons in different modes (this includes spatial and polarisation modes). The

detection patterns that herald a Bell pair in this circuit are:

$$\begin{aligned} \text{Measuring } h_1h_3 \text{ heralds } & \frac{h_2h_4 + v_2v_4}{8}, & h_1v_3 \text{ heralds } & \frac{v_2h_4 + h_2v_4}{8}, \\ v_1h_3 \text{ heralds } & \frac{v_2h_4 + h_2v_4}{8}, & v_1v_3 \text{ heralds } & \frac{h_2h_4 + v_2v_4}{8}, \\ h_1v_1 \text{ heralds } & \frac{h_2v_2 + h_4v_4}{8}, & h_3v_3 \text{ heralds } & \frac{h_2v_2 + h_4v_4}{8}. \end{aligned}$$

The probability of obtaining a Bell pairs is therefore 6 “success” detection outcomes⁷ times the probability of each of those outcomes, i.e. 1/32. Therefore the probability of producing an “event-ready” Bell pair is $p_s = 3/16$.

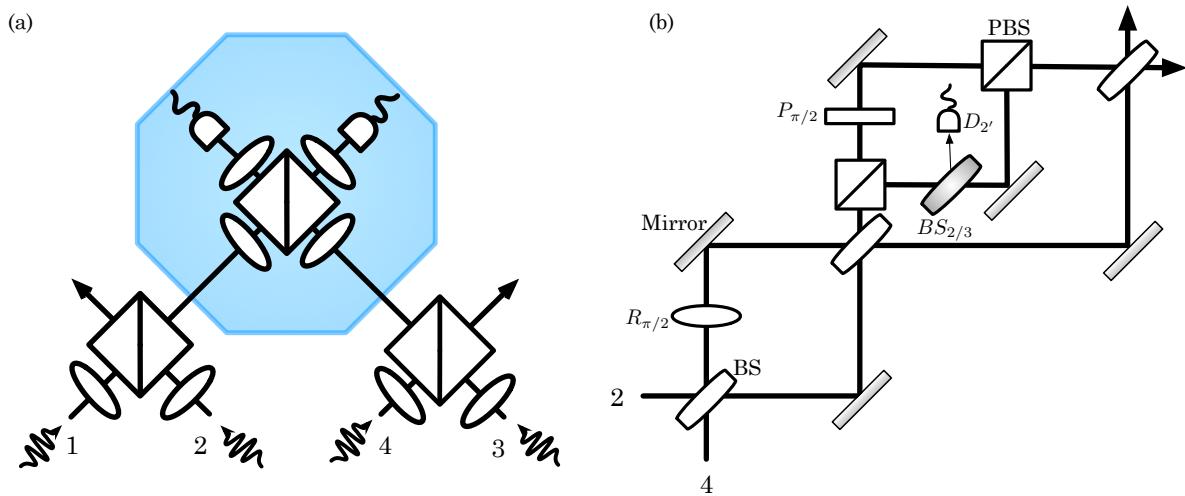


Figure 4.7: Scheme for generating a Bell pair from single horizontally polarised photons. (a) Main linear optical circuit. Four single photons prepared in the horizontal polarisation are put through the circuit. Upon measurement of two photons in different modes, a Bell pair in modes 2 and 4 is heralded, which occurs with probability 3/16. (b) Correction circuit. A “failure” case where two photons of the same polarisation have been measured in the same detector can be turned into a correctly formed GHZ state by applying the correction circuit. Upon measurement of vacuum in detector D'_2 , the outcome of the circuit will be a Bell pair, which brings the success probability up to 1/4. Note however that a switch is required to apply the correction circuit. Figure adapted from [92], copyright (2007) by the APS.

This Bell pair generation scheme was improved to have a success probability of $p_s = 1/4$ by Joo *et al.* [92], who showed that when adding an extra correction circuit, some failure outcomes could be transformed into a Bell pair. This correction circuit is shown in figure 4.7 (b). Previously considered “failure” detection patterns such as having two photons in the same mode (e.g. h_2^2) produced an unbalanced entangled state, in which the coefficients of the terms in the superposition were not equal. The correction circuit executes *procrustean distillation* [132] upon the detection of vacuum on mode $2'$. This circuit therefore probabilistically balances the terms in the superposition, generating a Bell pair with probability 1/16, and therefore making the final success probability for generating a Bell pair from single photons equal to $p_s = 25\%$.

⁷Note that the outcomes heralded by detection outcomes h_1v_1 and h_3v_3 are not Bell pairs *per se* but can be deterministically converted into one by applying a PBS to modes 1 and 3.

A key intuition that can be drawn from this scheme to generate Bell pairs (and helps understand how upcoming schemes work) is that the rotated PBS is turning the single photons into probabilistic Bell pairs. These probabilistic Bell pairs can be fused in different configurations to generate larger post-selected entangled states. It is assumed that the single photons are all horizontally polarised, and therefore the action of the rotated PBS is given by

$$h_1 h_2 \xrightarrow{\text{rPBS}_{1,2}} \frac{1}{2} (h_1 h_2 + v_1 v_2) + \frac{1}{4} (h_1^2 - v_1^2 + h_2^2 - v_2^2). \quad (4.42)$$

This state is a mixture of a Bell pair with a state that has two photons on the same spatial mode. The terms from this latter part will always be measured at the same detector, therefore triggering an erroneous outcome. Measuring always n out of the $2n$ modes ensures the presence of the correct number of photons in the output modes.

4.4.2 3-GHZ states from single photons

In [93], Varnava *et al.* propose a linear-optical circuit to generate 3-GHZ states from single photons. Its design serves as a basis for the n -GHZ generators presented in the following sections. In this generation scheme, six single horizontally polarised photons are introduced in the linear optical circuit and whenever three photons are detected at any three detectors, the remaining three photons are projected onto a 3-GHZ state. This scheme is different from the Bell pair state generation circuit presented earlier, as in this case two photons in the same spatial mode but different polarisation never constitute a valid detection pattern, whereas this was in the case of the Bell pair generator. The single photons are considered deterministic, and it was shown in [93] that this protocol is robust to loss. Not only photons lost in the circuit would herald an incorrect detection pattern, but it was found that in this circuit, loss at the input could be understood as independent and identically distributed (iid) loss on the final state.

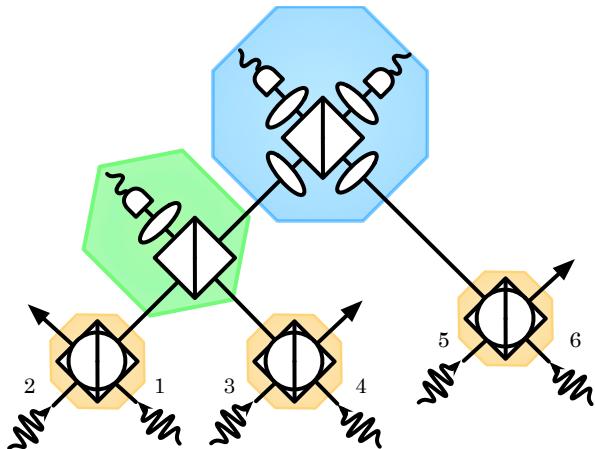


Figure 4.8: Linear optical circuit that generates a 3-GHZ state from single horizontally polarised photons with $1/32$ success probability. The successful generation is heralded by one (and only one) photon being detected in each detector. Figure adapted from [93], copyright (2008) by the APS.

The photons are first put in pairs through rotated PBSs. After the interference at the rotated PBSs, half the modes (even-labelled) enter a Type-I and Type-II gates, while the other half of the modes (odd-labelled), which are considered the output modes, support the photons into which the GHZ state will be projected. The detection of three photons in the even-labelled modes, one in each spatial mode, heralds the successful generation of a 3-photon GHZ state.

To calculate the probability of outputting the right state, we can simulate the evolution of the initial terms through the optical elements and then calculate the probability of the final terms that generate a GHZ state. This is the same technique that is used in [93], but it is cumbersome and lengthy, particularly in the case of the generation of n -GHZ states. We can shorten this calculation by realising that at each PBS, a similar effect to the one shown in equation (4.42) occurs: the state becomes a 50:50 mixture of terms that have one photon per spatial mode and terms that have more than one photon per spatial mode. As the successful GHZ generations are contingent on the detection of one and only one photon per spatial mode, only half of the terms in the state will lead to a successful detection. Therefore, the success probability of a GHZ generating scheme can be estimated by counting the number of PBSs involved. If a scheme has n (rotated or not rotated) PBSs, the success probability is

$$p_s = \frac{1}{2^n}. \quad (4.43)$$

The probability of the GHZ generating protocols presented in following section can be calculated in this way⁸. However, whenever a PBS is involved in a boosted fusion, the entire boosted fusion will count as a $(\frac{3}{4})$ factor.

4.4.3 Generation of larger GHZ states from single photons

The 3-GHZ generation scheme can be extended to a 4-GHZ generation scheme (presented in figure 4.9) in an obvious manner, this new circuit projected 8 pairs of photons into a 4-GHZ state with probability $p_s = \frac{1}{128}$. After pairs of single photons first pass through rotated beam-splitters, the interferometer can be considered as divided in two branches, photons 2 and 4 interfere at a central beam-splitter before interfering along each branch with other photons (2 with 6 and 4 with 8). Upon a successful detection of one and only one photon per mode, a 4-photon GHZ state is heralded on the odd-labelled modes.

Considering the interferometer as divided in two branches allows to easily generalise it for bigger GHZ states. More photons can be added to the two branches by adding extra Type-I gates along each branch. As it was the case before, only half of the photons interfere in the Type-I and Type-II gates, and they do so after interfering in pairs with the photons that will support the GHZ in rotated PBSs. The generalised n -photon GHZ state generator can be seen in figure 4.10.

Assuming deterministic single photons sources, the success probability of these scheme is given by

$$p_{succ} = \frac{1}{2^{n-1}}, \quad (4.44)$$

⁸Note that this technique is not applicable to the Bell pair generation circuit as the valid detection patterns are not only those with one photon per spatial mode.

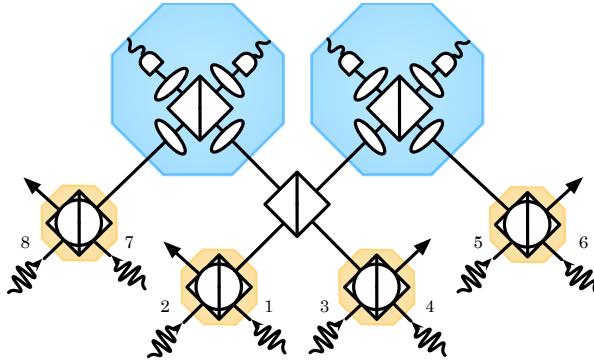


Figure 4.9: Linear optical circuit for the generation of a 4-GHZ state from single photons with $1/128$ success probability. The successful generation is heralded by one (and only one) photons measured at each detector.

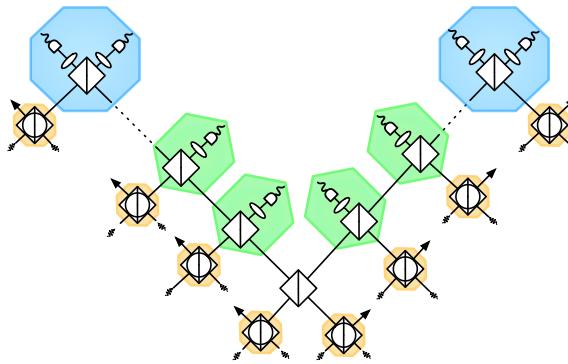


Figure 4.10: Linear optical circuit to generate an n -GHZ from single photons. All input photons are assumed to be horizontally polarised. The success probability of this procedure is $p_{succ} = \frac{1}{2^{n-1}}$.

which can be easily verified using the strategy explained in section 4.4.2. It is worth noting that these circuits benefit from the same kind of loss tolerance as the 3-GHZ generation scheme, as can be shown using the techniques in [93].

Even though the circuits presented in this section involve Type-II fusion gates, there is no advantage in boosting these. The boosted fusion gates don't require the successful detection of one photon per mode and rather they require that the detected patterns satisfy some statistical requirements (number of horizontally polarised photons, number of photons between two spatial modes, etc). Due to the effect of the rotated PBSs, it is possible for the correct statistics in the detection patterns to be achieved when there is the wrong number of photons in the output modes, and therefore the circuit does not herald the correct output state.

4.4.4 Adaptation to use Bell pairs

The linear optical circuit presented in previous sections can be easily adapted to use Bell pairs instead of single photons, as the purpose of rotated PBSs is to create a probabilistic Bell pair from the input single photons. Removing these rotated PBSs and inputting one photon out of each Bell pair increases the probability of generation of a n -GHZ state by 2^n .

The two-branch structure of the generating scheme proposed has two disadvantages. On one hand, the number of linear optical elements the photons go through is not balanced: out of the $n/2$ photons that interfere in the branches, two photons pass through $\frac{n}{2} + 2$ beam-splitters while the rest only pass through 2 beam-splitters. Therefore there are two photons with a radically increased loss rate due to the number of linear optical elements they encounter in their path. On the other hand, the scheme uses predominantly Type-I fusion gates and only two Type-II gates. This is not ideal as it is only known how to boost Type-II fusion gates.

The n -GHZ generator can be restructured to solve both these issues: the photons that interfere at fusion gates will do so predominantly at Type-II fusion gates, in such a way that each photon only ever goes through a maximum of two beam-splitters (three if we want to implement this scheme with single photons). In this design the majority of measurements (all in case of even n and all-but-one in case of odd n) are part of a Type-II fusion gate, which can be boosted using the schemes presented earlier.

In this configuration 4.11, each photon interacts only with two PBSs and one polarisation rotator. Also, out of $(n - 1)$ PBSs, $\lfloor \frac{n-1}{2} \rfloor$ are involved in a Type-II gate (and the rest are not involved in measurements) so now we are boosting much more efficiently. The number of Bell pairs that we need to generate and n -GHZ state using this configuration is $n + \lfloor \frac{n-1}{2} \rfloor$, n to create the state and $\lfloor \frac{n-1}{2} \rfloor$ to boost the fusion gates. The use of boosted fusion gates further increases the success probability to

$$p_s = \left(\frac{1}{2}\right)^{\lceil \frac{n-1}{2} \rceil} \left(\frac{3}{4}\right)^{\lfloor \frac{n-1}{2} \rfloor}. \quad (4.45)$$

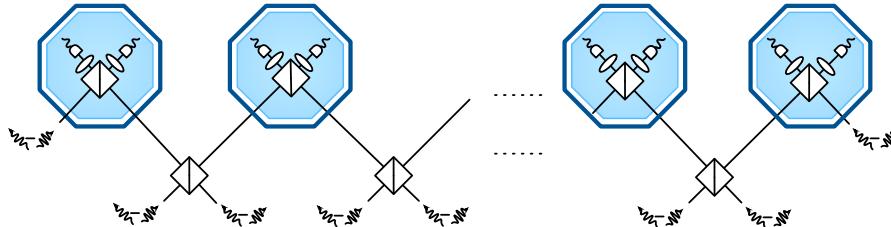


Figure 4.11: Optimised scheme for the generation of n -photon GHZ states from Bell pairs. All the measurements are part of Type-II fusion and therefore we can substitute the original Type-II fusion in this scheme for one of the boosted Type-II gates. Note that this circuit can also be used to produce GHZ states from single photons if each input Bell pair is substituted by a pair of single photons after they have passed a rotated PBS. The success probability when using Bell pairs as inputs and boosted Type-II fusion gates is $p_s = \left(\frac{1}{2}\right)^{\lceil \frac{n-1}{2} \rceil} \left(\frac{3}{4}\right)^{\lfloor \frac{n-1}{2} \rfloor}$.

Introducing the boosted fusion gates in the GHZ generation makes the detection patterns that herald the GHZ state quite complicated. It is also worth noting that when using these gates, some of the heralded outcomes are GHZ states with some rotations. In appendix F we have explained in detail what detection patterns herald the correct outcome in the case of 3-photon and 4-photon GHZ states.

4.4.5 Giving up loss tolerance for higher success probability

Linear optical circuits for the generation of GHZ states with a higher success probability can be obtained at the cost of losing some of the in-built loss tolerance. This improvement comes from an observation on cluster state construction as described in the Browne-Rudolph protocol [90]. In all the circuits proposed so far, we always have $2n$ photons (single or in Bell pairs), out of which n are measured to herald the n -GHZ state on the remaining n photons. This is necessary when generating states from single photons, due to the effect of the rotated PBSs.

However, when using Bell pairs as the starting point, we don't need to measure n qubits necessarily. Browne and Rudolph [90] show how linear clusters can be grown from Bell pairs by using the Type-I gate. Linear clusters and GHZ states are different, except in the case when $n = 3$, in this case they are LC equivalent, i.e. equivalent under some local rotations. Therefore, if a three qubit linear cluster can be built from two Bell pairs using a Type-I gate, so can a 3-photon GHZ state. This observation is also applicable to larger GHZ states, each n -GHZ state can be created using $n - 1$ Bell pairs as resources instead of n as before, and only $n - 1$ photons are measured. In figure 4.12 we present the linear optical circuit corresponding to 4-GHZ state and the generalised circuit.

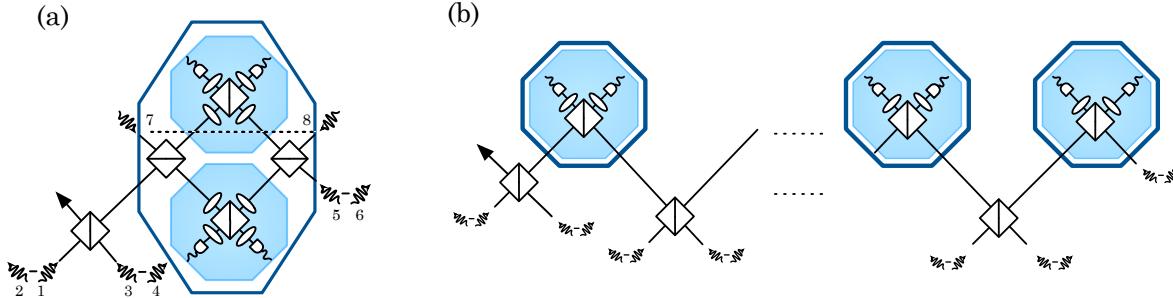


Figure 4.12: Higher success probability schemes for the generation of GHZ states from Bell pairs. (a) Generation of a 4-GHZ state from 4 Bell pairs, with 28.125% success probability. (b) Generalisation to an n -GHZ generator which consumes $(n - 1) + \lfloor \frac{n-1}{2} \rfloor$ Bell pairs and has a success probability of $(\frac{1}{2})^{\lfloor \frac{n-1}{2} \rfloor} (\frac{3}{4})^{\lfloor \frac{n-1}{2} \rfloor}$. Note that these generation schemes are not loss tolerant, in the sense that a failure outcome combined with a loss can be heralded as a successful outcome.

This higher success probability comes at a cost, however. All the circuits presented in previous sections are loss tolerant, in the sense that if the correct number of photons are detected in the appropriate modes we can be *sure* that the output state is a GHZ⁹. Therefore, there is no chance of confusing an unsuccessful generation in conjunction with loss with a successful generation.

In these new circuits with higher success probability, it is possible that a failure outcome combined with the loss of a photon is mistaken for a successful generation. As an example, let's

⁹For the 3-GHZ generation scheme, it was proven in [93] that loss of photons in the input state of these circuits can have two effects, it either affects a mode that is measured, in which case the output state is rejected, or it has the same effect as if the loss had happened *after* the generation procedure, i.e. on the GHZ itself. This type of iid loss has not been proven for general n -GHZ generation schemes.

examine the 4-GHZ generation circuit in figure 4.12 (a). Photons 1, 3, 5, 7 and 8 interfere in the circuit but only photons 1, 5, 7 and 8 are measured. It can be the case that either photon 1 or photon 3 are lost (or in fact never enter the interferometer), but that the detector in mode 1 still measures one photon. This would herald the successful generation of a 4-GHZ state, whereas there will be no entanglement formed between photons 2 (or 4) and 6. This circuit is not loss tolerant in the same way as the Type-I gate is not loss tolerant.

This trick to enhance success probability cannot be used in the case of optical circuits in which we input single photons for the same reason we couldn't use boosted gates with single photons. In the case of single photons, any interference between photons from different pairs is preceded by the action of the rotated PBSs on the single photon pairs. As explained previously, the action of this linear optical element created a Bell pair when the state is post-selected to have one photon in each mode. However, we don't post-select until the very end and therefore we need n detectors in order to ensure that the correct number of photons are on the output modes.

4.5 Removing stochasticity by multiplexing

Stochasticity is fundamentally present in any linear optical setup. There are two main sources of this stochasticity: on one hand, no on-demand deterministic sources of photons currently exist¹⁰ and therefore state preparation is probabilistic; on the other hand, deterministic entangling gates are fundamentally impossible unless infinite resources are consumed. In this section, we review the idea of multiplexing (MUX), which has been proposed as a way to remove stochasticity from linear optical experiments. In particular, we want to use multiplexing to be able to produce on-demand n -GHZ states.

A commonly used approach for the generation of single photons is the use of non-deterministic heralded single-photon sources such as the SPDC sources introduced earlier in the chapter. Such probabilistic sources cannot on their own be a basis for quantum technologies, as the probability of generating p indistinguishable photons decreases exponentially with p . A way of overcoming the scalability problem of these sources is to use a *multiplexed* layout of non-deterministic sources [155, 156, 157, 158, 159, 160], i.e. repeat them in parallel (either spatially or temporally) and integrate all the outcomes via a switching network. Using a switching network and a high enough number of repetitions, the successful event can be located at a spatiotemporal bin of choice making the emission *asymptotically deterministic*. In figure 4.13 we can see examples of spatial and temporal multiplexed sources. The successful event is heralded, which prompts a reconfiguration of the spatial switch or the length of the delay line, to locate the photon in the desired spatiotemporal bin - $m_0 t_0$ in the case presented in figure 4.13.

When creating a deterministic on-demand source using a multiplexing scheme, by *deterministic* we really mean with a probability of emission higher than some desired threshold, p_s . The number of repetitions, k , needed in the multiplexing scheme is determined by the probability of emission of the source, p_η and the desired probability of emission, p_s . We calculate the number

¹⁰There have been several proposals for deterministic single-photon sources based on artificial-atom systems [58], however none of these proposals currently achieves success probabilities necessary to be considered deterministic single-photon sources.

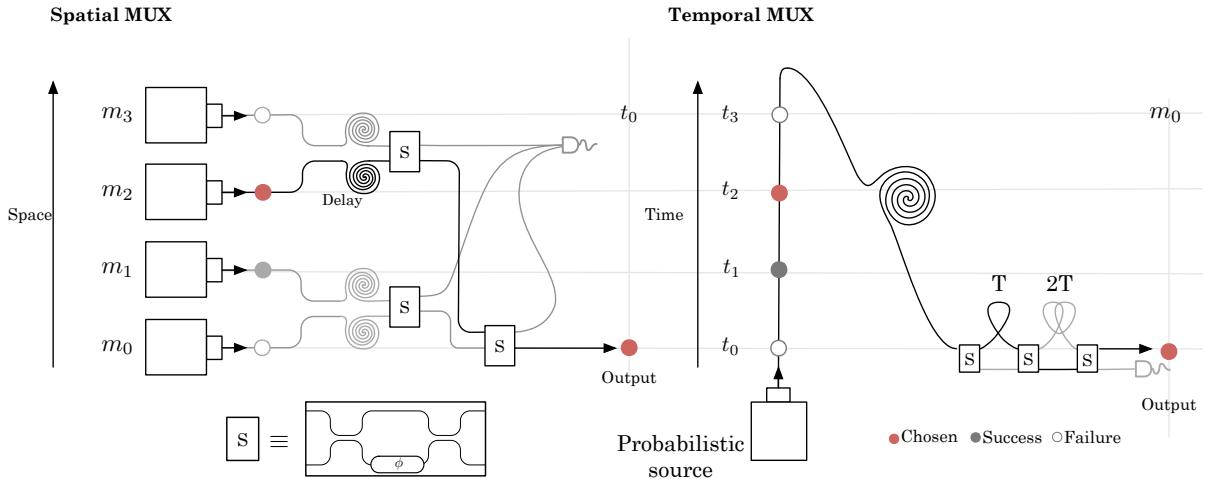


Figure 4.13: Schematic layout for spatial and temporal multiplexing. Note that the axes for space and time have been interchanged in the figures in order to highlight the equivalence of both methods. Both devices emit one photon in the spatiotemporal mode $m_0 t_0$, in spatial MUX an $N \times 1$ switch (realised with a MZI) located the emitted photon in mode m_0 , while in temporal MUX a delay line is used to change the temporal mode of the photon.

of repetitions by solving the equation

$$1 - (1 - p_\eta)^k \geq p_s, \quad (4.46)$$

which gives the minimum number of repetitions needed so that the probability of at least one successful event is higher than p_s . Of course on average many more successful events will have been generated.

This estimation of the number of multiplexing stages needed is a simplified version, as it doesn't take into account the effect of loss. A full analysis of the effect of loss on multiplexed single-photon sources can be found in [161].

4.5.1 Log-tree scheme

Once all the probabilistic events have been generated in the multiplexed source, we require a way of placing one of the successful events in the spatiotemporal bin of choice. Switching networks allow for this re-routing of photons. We require a reconfigurable switch of $N \times 1$ modes, where N is the number of multiplexed events. One way to construct such a switch is to decompose it in a logarithmic tree of 2×2 switches. This yields a required depth of $\log_2 N + 1$ switches. The heralded photons are stored in delay lines while a classical control determines the configuration of the switch and sets it in place.

As the number of switches required scales as the logarithm of the number of multiplexed events, we can optimise the use of switches by having a number of multiplexed events that is

$$N = 2^{\lceil \log_2 k + 1 \rceil}, \quad (4.47)$$

where k is given by the relation (4.46).

Spatial log-tree scheme

In a spatial multiplexing scheme we have N photon sources which are pumped simultaneously to produce N probabilistic single photons, all in the same time bin and different spatial modes. From the successful photon generations, one is chosen to be placed in the output mode. The switch configuration is set for this event to happen while the photons are stored in delay lines. The chosen single photon is then located through the output port while the rest of optical modes are re-routed to a detector. Figure 4.13 shows an example of the spatial log-tree arrangement of switches for $N = 4$ multiplexed events, which places the red qubit in the output port while the remaining modes are routed to a detector or beam dump. The 2×2 switches can be implemented by using a MZI with a variable phase-shifter.

Temporal log-tree scheme

In a temporal multiplexing scheme we have a probabilistic source which is pumped N times to generate a series of photonic events in different time bins and the same spatial mode. Differently to what happens in the spatial case, the switches will have to change their configuration for the different time-bins in order to re-route the chosen photon to the output port while the rest of successful events are measured¹¹. This fast reconfigurability imposes technological restrictions on the switches. Figure 4.13 shows an example of the temporal log-tree scheme, for $N = 4$ multiplexed events. The reconfigurable switches change the length of the delay we subject the photons to, in order for the chosen photon to come out of the output port in the required time bin.

4.5.2 Cost of near-deterministic generation of GHZ states using a multiplexed scheme

The advantages of using GHZ generating schemes that have higher success probabilities becomes more apparent when we consider the multiplexing of such schemes in order to produce deterministic GHZ states. Current technologies have not yet produced deterministic sources of entangled states¹², which are crucial for LOQC protocols such as Kieling *et al.*'s percolation scheme and a novel scheme we will present in chapter 5 of this thesis. In figure 4.14 we compare the Bell pair consumption of both schemes when multiplexed to form a near-deterministic GHZ source.

It is clear that generating GHZ state from Bell pairs has a much higher success probability ($(\frac{1}{2})^{\lceil \frac{n-1}{2} \rceil} (\frac{3}{4})^{\lfloor \frac{n-1}{2} \rfloor}$ compared with $\frac{1}{2^{n-1}}$) and is therefore preferable. However, when we don't have access to Bell pairs on-demand, the question remains whether it is better to use a ballistic circuit to generate GHZ states directly from single photons, or whether it would be more resource efficient to first generate Bell pairs from the single photons and then use the optimised

¹¹Measuring generated photons at this stage is a waste. In chapter 6 we study the advantage of a different multiplexing scheme in order to improve resource efficiency of the LOQC architecture.

¹²Proposals for quantum dot sources [162] and multiplexed probabilistic single photon sources [155, 156, 157, 158, 159, 160] have not yet achieved a near-deterministic regime.

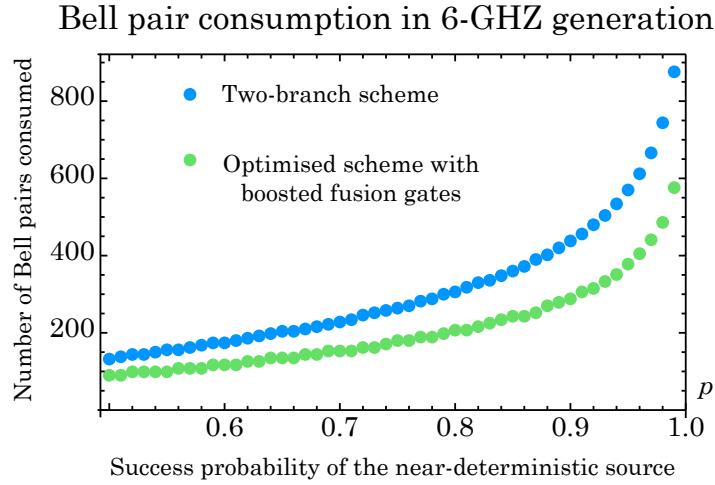


Figure 4.14: Comparison of proposed n -GHZ generation schemes. We calculate the number of Bell pairs consumed when multiplexing the GHZ generation in order to have a near-deterministic GHZ source that emits a GHZ state with probability p . The two-branch scheme consumed more Bell pairs as it produces a GHZ states with lower probability. The difference in Bell pair consumption increases exponentially as the probability of the near-deterministic source approaches unity.

GHZ generating scheme that uses Bell pairs. To assess which strategy is best, we calculate the number of multiplexed photons required to produce a 3-GHZ. We plot the results in figure 4.15.

For the strategy that uses the ballistic 3-GHZ generator from single photons, we first multiplex single photons sources of efficiency η and then multiplex the ballistic 3-GHZ generator. The number of photons consumed is $6 \cdot k_1 \cdot k_2$ where k_1 is the number of multiplexed events required to produce a single photons with probability p_1 and k_2 is the number of multiplexed events required to produce a 3-GHZ state from deterministic photons, the factor 6 portrays that each ballistic 3-GHZ generator requires 6 photons. The probability of generating a 3-GHZ state using this procedure is $p_1^6 \cdot p_2$.

For the strategy that has the intermediate step of generating Bell pairs, there are three stages of multiplexing. First, we multiplex k_1 single photon sources to produce a single photon with probability p_1 . Then we multiplex k_2 Bell pair generators, each using 4 single photons as input, to produce a single Bell pair with probability p_2 . Finally we multiplex k_3 3-GHZ generators, each consuming 4 Bell pairs, to produce a single 3-GHZ with probability p_3 . As in each multiplexing state, the input states are assumed deterministic, the final probability of emission is given by $p_1^4 \cdot p_2^4 \cdot p_3$ and the number of single photons consumed is $16 \cdot k_1 \cdot k_2 \cdot k_3$. Note that we are using the loss-tolerant 3-GHZ generator from Bell pairs that has 37.5% of success. In order for this to be a fair comparison, we use the circuits that have the same loss tolerance. As different values for p_1 and p_2 can give the same final probability we optimise the results to minimise the number of probabilistic source emissions.

In figure 4.15 we present the comparison of the number of bins necessary to produce a 3-GHZ state using each strategy. By ‘‘bins’’ we mean the number of times that a heralded probabilistic single-photon source has to be pumped to produce a single photon. As we can see from figure

4.15, the optimal strategy depends on the source efficiency. For high efficiency sources, the strategy that generates 3-GHZ states directly from single photons is optimal, while for very low efficiency sources, it is better to add an intermediate stage of Bell pair generation in the process.

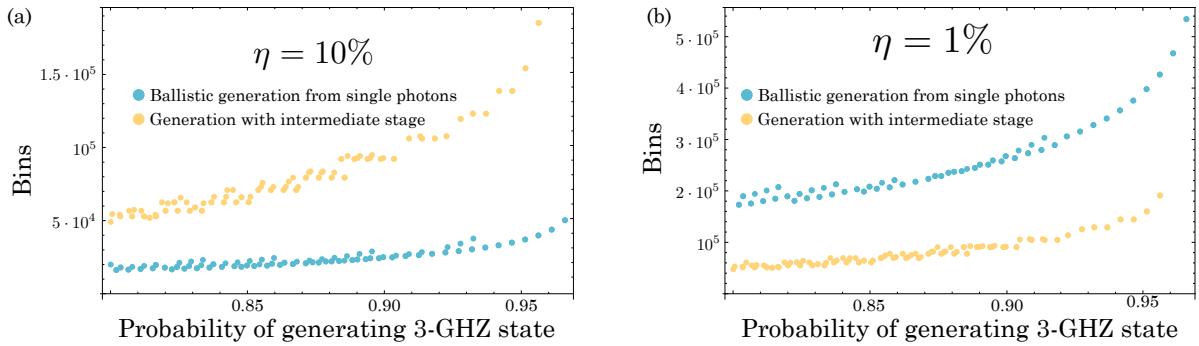


Figure 4.15: Comparison of strategies for the generation of 3-GHZ states, where the ballistic strategy that generates the GHZ state directly from single photons is marked in blue and the strategy that has an intermediate stage where Bell pairs are produced is marked in yellow. (a) Comparison of strategies when using a high efficiency source $\eta = 10\%$. In this case (and for efficiencies higher than this) the ballistic generation is more resource efficient. (b) Comparison of strategies when using a low efficiency source $\eta = 1\%$. In this case (and for efficiencies lower than this) the generation with an intermediate stage is more resource efficient.

4.5.3 Using probabilistic Bell pairs to generate GHZ states

In the calculations we have presented so far referring to the generation of GHZ states, we have always assumed that we multiplex heralded single photon sources. However, the ‘‘heralded’’ property of the source comes from the fact that the output of the source is a Bell pair and we measure one mode in order to herald the presence of a single photon in the other mode. This strategy is extremely expensive, and it seems a waste to use these probabilistic Bell pairs to only herald single photons.

The key realisation is that in order to herald n single photons from probabilistic sources we measure n modes in a system with $2n$ modes. This is very similar to some of the circuits we have presented so far, where $2n$ photons are used to create n -GHZ states. These measurements not only herald the correct state, but also herald the presence of photon pairs in the correct modes. Therefore we can remove the multiplexing stage of single photons and directly input the probabilistic pairs into the GHZ generator. It must be noted that in this case, the measurement of n photons is *necessary* as we need to herald the sources, and therefore the schemes presented in section 4.4.5 cannot be used.

In figure 4.16 we present the generation of a 3-GHZ state from probabilistic SPDC sources. This circuit has a success probability of 25% when it has deterministic Bell pairs fed into it, and success probability $\eta^3/4$ when the Bell pairs are probabilistic, where η is the efficiency of

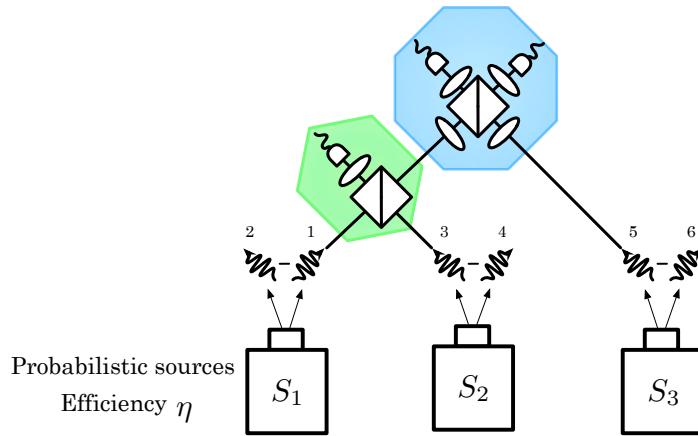


Figure 4.16: 3-GHZ generation using Bell pairs from probabilistic SPDC sources. The success probability of this circuit is $\eta^3/4$ where η is the efficiency of the sources.

the source¹³. In order to compare the efficiency of this approach with generating a 3-GHZ with deterministic single photons (after they have been multiplexed from the source) we will count the number of bins necessary in each case to produce a near-deterministic GHZ state.

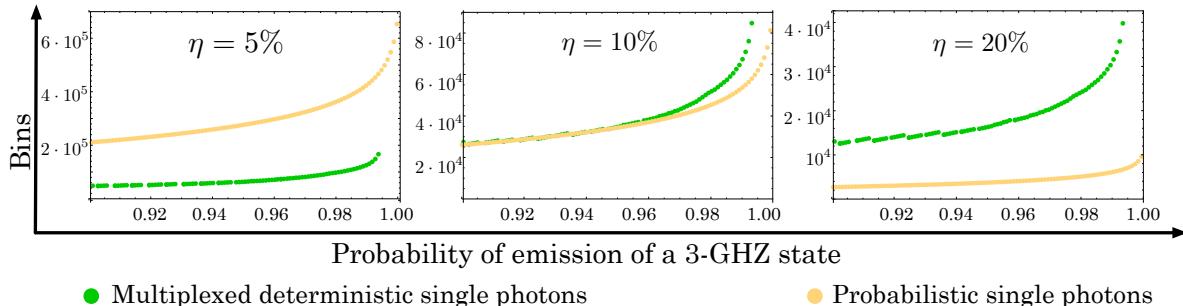


Figure 4.17: Comparison of 3-GHZ multiplexing generation schemes. For three different source efficiencies, we plot the number of probabilistic source emissions as a function of the probability of generating a GHZ state. In green we represent the results for the scheme where we herald single photons from the SPDC sources and multiplex them in order to use them as deterministic in the “3-GHZ from single photons” scheme. In yellow, we represent the results of using the scheme presented in figure 4.16, where the output of SPDC sources is fed directly into a 3-GHZ generator. We can see that the optimality of the strategy depends on the source efficiency, with the ballistic approach (where we don’t multiplex single photons) being more beneficial for higher source emission rates.

In figure 4.17 we present the comparison of both strategies for three different source efficiencies. In the case with the single photon multiplexing stage, as it was explained earlier, we require $6 \cdot k_1 \cdot k_2$ bins and the probability of generating the 3-GHZ is $p_1^6 \cdot p_2$. For the case that simulates the circuit presented in figure 4.16, we only have one stage of multiplexing. The optimality of these strategies depends on the efficiency of the source (in a similar way as we saw in the previous section). For low source efficiency $\eta < 10\%$, it is more favourable to add a

¹³Note that we have not used the boosted version of this scheme. Although using the boosted version would give a success probability 1.5 times larger, the extra η factor due to the source efficiency would effectively give a lower probability.

multiplexing stage at the level of single photons, while for high source efficiency $\eta > 10\%$ it is better to use the probabilistic Bell pairs directly from the source.

This strategy that uses directly the probabilistic Bell pairs emitted by the sources has one further good quality. So far we have been assuming that in the case when one of the sources doesn't fire, we discard the event as a failure. However, certain detection patterns with a lower than expected number of measured photos can herald smaller entangled states. For example, in the case of the 3-GHZ generation circuit presented in figure 4.17, studying the possible measurement outcomes when only two photons are detected (hence implying one of the sources has not fired) we realise that if source S_3 doesn't emit a photon pair, the circuit generates a heralded Bell pair. From the three detectors in the circuit we cannot tell which source did not produce a Bell pair (and it is necessary to know in which mode the generated Bell pair is in order to consider it heralded), but adding an extra detector in mode 6 allows us to determine if it was source 3 that did not produce a Bell pair. In the case where only two photons are detected (and the measurement pattern corresponds to one that we would consider successful), detecting vacuum on mode 6 heralds a Bell pair in modes 2 and 4. A full analysis of the output states showing that this is true can be found in appendix F.

4.6 Rotated Type-II

As we have seen previously, it is helpful to view the fusion gates as two effective projections to understand their effect on the cluster states defined on the photons. The fusion gates presented so far always perform the BSM in the same basis, meaning that the effective projective measurement on the qubits is also in the same basis. By changing the basis of the BSMs, we change the effective projections and can therefore achieve different cluster operations. In this section we will illustrate how a fusion measurement affects the structure of a graph state. We will give the results of other possible rotated fusion gates condensed in figure 4.18. As we will see in this figure, the graph state changes significantly after a success or failure outcome, with differences not only in the graph structure but also in the amount of entanglement in the unmeasured qubits. The ability to choose the success and failure outcomes appropriately will be very useful to optimise the construction of a big cluster state from small entangled states, which we do in chapter 5.

The procedure for obtaining the effective projections is the same as was used to justify the projective measurements in the case of the fusion gates and will therefore be omitted here. The different effective projections of rotated fusion gates can be seen in figure 4.18.

We now show how two star graph cluster states (LC equivalent to a GHZ) can be fused together using the Type-II fusion gate. Expressed in Dirac notation, these states are:

$$|GHZ_4\rangle|GHZ_4\rangle = \frac{1}{2} (|0+++ \rangle + |1--- \rangle) (|0+++ \rangle + |1--- \rangle). \quad (4.48)$$

A successful Type-II fusion would yield

$$\left(\frac{\langle ++| + \langle ---|}{\sqrt{2}} \right)_{2,6} |GHZ_4\rangle|GHZ_4\rangle = \frac{1}{\sqrt{2}} (|0+0++ \rangle + |1-1-- \rangle)_{1,3,4,5,7,8}, \quad (4.49)$$

4. GENERATING PHOTONIC STATES

which is a star graph with the middle qubit redundantly encoded, while the failure of the fusion gate would yield

$$(\langle ++|)_{2,6} |GHZ_4\rangle |GHZ_4\rangle = (|0 + +0 + +\rangle)_{1,3,4,5,7,8}, \quad (4.50)$$

which is a disconnected graph.

Note that doing the projection on the other state of each subspace would just introduce a Z rotation on half of the state. Changing the polarisation rotation of the photons before they interfere at the PBS yields different subspaces onto which the state of the photons is effectively projected to, resulting in different fusion operations on the cluster states. Doing a similar analysis for the different rotated fusion gates, we obtain the results presented in figure 4.18. In this figure, some of the graph states have two qubits without a bond between them, we use this notation to denote redundantly encoded qubits.

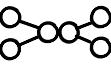
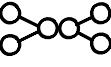
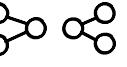
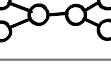
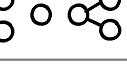
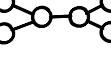
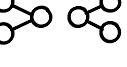
GATE	SUCCESS		FAILURE	
	Effective projection	Outcome	Effective projection	Outcome
	$\frac{ ++\rangle \pm --\rangle}{\sqrt{2}}_{2,6}$	$\frac{ 0++0++\rangle + 1-1-\rangle}{\sqrt{2}}$ 	$ +-\rangle_{2,6}$ or $ +-\rangle_{2,6}$	$ 0++0+\rangle$ 
	$\frac{ 00\rangle \pm 11\rangle}{\sqrt{2}}_{2,6}$	$\frac{ 0++0++\rangle + 1-1-\rangle}{\sqrt{2}}$ 	$ 01\rangle_{2,6}$ or $ 10\rangle_{2,6}$	$(0+\rangle + 1-\rangle) \times (0+\rangle + 1-\rangle)$ 
	$\frac{ +0\rangle \pm 1-\rangle}{\sqrt{2}}_{2,6}$	$\frac{ 0++\rangle + 1-1-\rangle}{2} + \frac{ 1--\rangle + 0++\rangle - 1-1-\rangle}{2}$ 	$ +1\rangle_{2,6}$ or $ 0-\rangle_{2,6}$	$ 0++\rangle \times (0++\rangle + 1-\rangle)$ 
	$\frac{ 0(+i)\rangle \pm 1(-i)\rangle}{\sqrt{2}}_{2,6}$	$\frac{ 0++\rangle + 1-1-\rangle}{2} + i\frac{ 1--\rangle + 0++\rangle - 1-1-\rangle}{2}$ 	$ 0(-i)\rangle_{2,6}$ or $ 1(+i)\rangle_{2,6}$	$(0+\rangle + 1-\rangle) \times (0+\rangle + i 1-\rangle)$ 

Figure 4.18: The different rotated versions of the Type-II gate can be seen in this figure, where the effective projection on the qubits and the final state of the rest of the photons is given, in Dirac notation and as a graph.

We have presented the action of the rotated fusion gates on two star graph states as this operation is crucial for percolation-based LOQC protocols. We can see how the rotated gate at the bottom of figure 4.18 has the optimal success and failure outcomes for the task of generating large clusters from small GHZ states: when it succeeds the two central sites of the star graph are linked by a cluster edge, while when the gate fails, the two measured photons are removed

from the cluster but no more entanglement is destroyed¹⁴. However the optimality of one gate or another is highly dependent on the input states and the intended final graph structure, other protocols with different input states might benefit more from using one of the other rotated fusion gates.

4.7 Discussion and outlook

In this chapter we have provided varied optical circuits that produce GHZ states from different resources and with different probabilities of success. In figures 4.20 and 4.21 we present summaries of these linear optical circuits. Figure 4.20 shows the generation of Bell pairs, 3-photon, 4-photon and n -photon GHZ states from single photons with the correspondent success probabilities and the resources needed for each circuit. Equally, figure 4.21 shows the generation of 4-photon and n -photon GHZ states from Bell pairs with resources required and success probabilities.

It must be noted that the optimal strategy for building GHZ states is not known. The results presented in this chapter improve the efficiency of the generation but they are not proven, or in fact thought of, to be optimal. In figure 4.19 we plot the number of deterministic Bell pairs consumed when generating GHZ states. As we can see the consumption of resources grows exponentially. The new generation schemes proposed have been obtained mainly by generalising previous results and exploiting the intuition gained from considering the fusion gates as cluster building operations. New methodologies are necessary in the search for an optimal GHZ generation circuit.

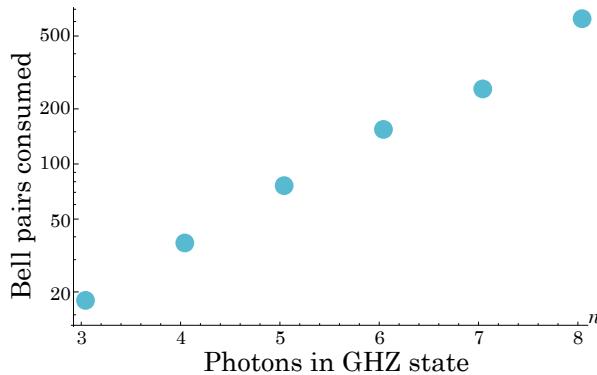


Figure 4.19: Deterministic Bell pairs consumed when multiplexing the GHZ generation scheme in figure 4.11. The number of resources needed grows exponentially with the size of the generated GHZ state.

All these schemes have been designed using fusion gates. What is really significant about the original [90] and modified (boosted and rotated) fusion gates is that their success and failure outcomes can be mapped to projective measurements in the qubit basis. Therefore they can be used to construct other photonic states (in particular those that can be described as graph states) by just considering them as cluster state operations. The use of these gates in

¹⁴Note that this rotated fusion gate introduces some imaginary phases in the description of the cluster state. As these are *known* phases, they can be taken into account when implementing an MBQC protocol.

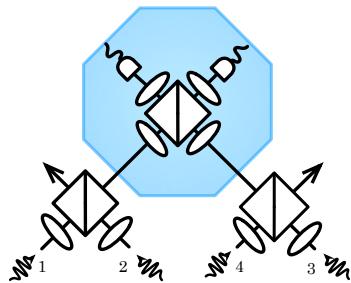
combination with cluster state simulators, such as the one described in chapter 3, provides a novel way to design new photonic generation circuits.

The boosted fusion schemes provide the advantage of increased success probability when performing BSMs, but have not been proven optimal. In fact, numerical results from [115] suggest they are indeed not so. The fact that the Grice and Ewert-van Loock BSMs can be written as the same unitary operation and they only differ in the ancillary states used (Grice using Bell pairs while Ewert-van Loock uses single photons) suggests an equivalence between these resources which appear so different. Investigating this equivalence and formalising the resource which boosts the success probability of BSM is an interesting future line of research.

In order to compare the efficiency of all the schemes presented we have performed multiplexing calculations, to associate a cost in terms of photonic states to the success probability given for each of these schemes. One striking trend that can be observed is that lower source efficiencies favour generation schemes with more intermediate multiplexing stages. As the efficiency of photon sources improves, the most efficient generation schemes will be those with few or no multiplexing stages, which will benefit any LOQC proposal and will reduce the loss rate introduced by active switching.

Throughout this chapter we have provided estimates for the number of resources needed for the different schemes. The numbers are very large and might look discouraging. However, given some reasonable assumptions about future technologies (clock rate $\sim 1\text{GHz}$, 1 micron distance between waveguides in the wafer and 1 foot of propagation distance per ns time bin), it is possible to show that a common-sized silicon wafer such as $12'' \times 12''$ can store up to $3.1 \cdot 10^6$ time bins. Therefore any of the multiplexing schemes we have presented can be comfortably stored in these wafers. We reiterate that the schemes presented in this chapter are the best we know of, but have not been proven optimal. Devising a systematic approach to find the optimal schemes is the next natural step to improve efficiency and reduce the number of resources consumed.

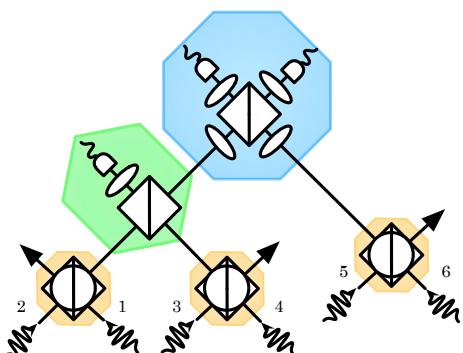
Single photons



Success probability: $\frac{1}{2^3} = 12.5\%$

Resources: 4 single photons

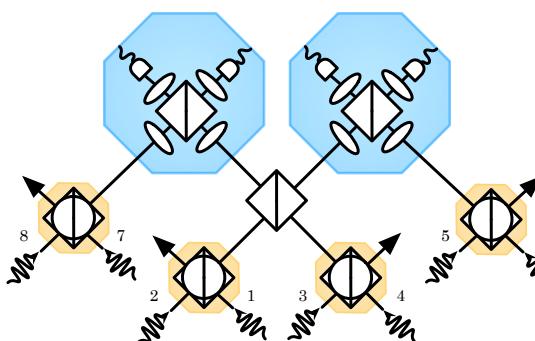
Output: Bell Pair



Success probability: $\frac{1}{2^5} = 3.125\%$

Resources: 6 single photons

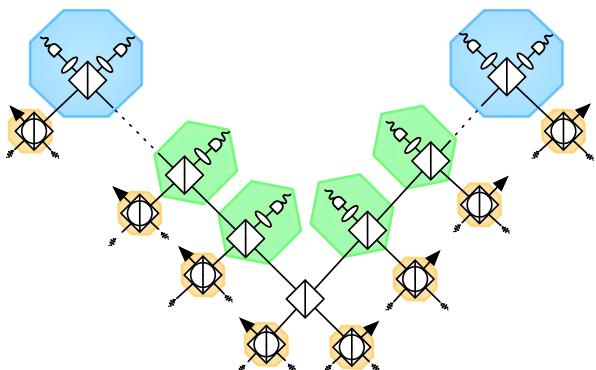
Output: 3-photon GHZ



Success probability: $\frac{1}{2^7} \simeq 0.78\%$

Resources: 8 single photons

Output: 4-photon GHZ



Success probability: $\frac{1}{2^{2n-1}}$

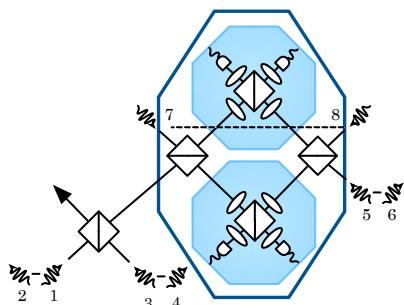
Resources: $2n$ single photons

Output: n -photon GHZ

Figure 4.20: Summary of entangled states generation from single photons.

Bell Pairs

Most efficient

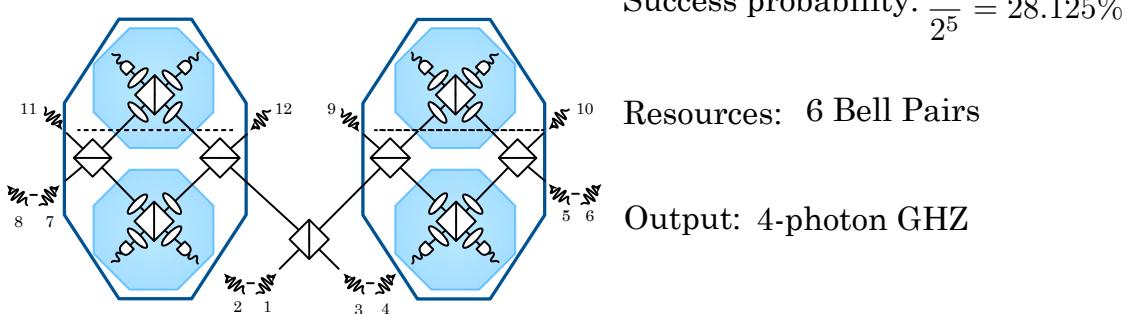


Success probability: $\frac{3}{2^3} = 37.5\%$

Resources: 4 Bell Pairs

Output: 4-photon GHZ

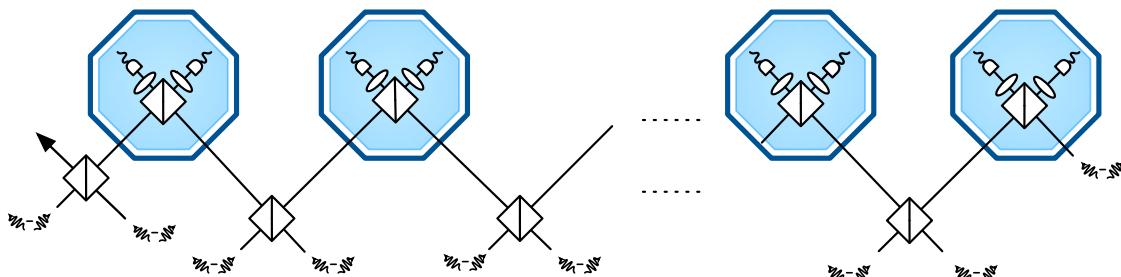
Loss tolerant



Success probability: $\frac{3^2}{2^5} = 28.125\%$

Resources: 6 Bell Pairs

Output: 4-photon GHZ



Success probability: $\left(\frac{1}{2}\right)^{\lfloor \frac{n-1}{2} \rfloor} \left(\frac{3}{4}\right)^{\lfloor \frac{n}{2} - 1\rfloor}$

Output: n -photon GHZ

Resources: $(n-1) + \lfloor \frac{n-1}{2} \rfloor$ Bell pairs

Figure 4.21: Summary of entangled states generation from Bell Pairs

CHAPTER 5

A PERCOLATION-BASED SCHEME FOR LINEAR OPTICAL QUANTUM COMPUTING

5.1 Introduction

In the literature review presented in chapter 2, we presented the most significant proposals for LOQC since KLM’s [2] first proof of principle. Recent demonstrations [58, 56, 43, 59, 60] have made significant progress towards fulfilling the experimental requirements needed in those proposals. In particular, the use of integrated photonics to implement large-scale, complex interferometers on a chip shows great promise. However, active feed-forward remains challenging, it requires fast switching which is a dominant source of photon loss and has not yet been experimentally demonstrated in an integrated device. Fast (GHz) switching, required for optical feed-forward, is expected to be the leading source of heat and power consumption in a large-scale device. It is therefore desirable to avoid fast switching and feed-forward where possible.

Of previous approaches to linear optical quantum computing, only Kieling *et al.’s* proposal [99] is *ballistic* - meaning that active switching is not required for the process of cluster state generation from the small resource states. It is thus the most suitable previous approach to LOQC in an integrated setting. It has a number of shortcomings, however. Firstly, it requires 4 or 5-photon entangled states as input, which are costly and difficult to generate in a (near)-deterministic manner. Secondly, it is not constructed from loss-tolerant components, and therefore photon loss during the process will lead to the generation of an undesired state. In the scheme presented in this chapter, we adapt advances in Bell state measurement [111, 115] to the ballistic cluster state generation scheme, to provide a new approach to scalable ballistic LOQC with significant advances on Kieling *et al.’s* approach. Off-line resources are reduced to 3-photon entangled states (local-Clifford equivalent to GHZ states), while all gates are loss-detecting. The scheme has an in-built robustness to loss and will succeed, without additional loss-encoding, even if $\sim 1\%$ of the photons entering the gates are lost. As seen in chapter 4, deterministic n -qubit entangled state generation becomes experimentally more challenging with increasing n , and the reduction to resource states to only 3 photons is thus a significant improvement. A full resource comparison, demonstrating at least an order of magnitude reduction in resources compared with earlier schemes, is also presented.

It must be noted that the scheme presented here and Kieling *et al.’s* are very different from other LOQC schemes, as in these schemes, once the resource state enters the cluster generation

stage, any photon will only ever interfere with one and only one other photon¹.

5.2 Boosted fusion mechanisms in the context of percolation

Browne and Rudolph proposed [90] two different fusion gates, Type-II and Type-I for combining cluster states, as introduced in earlier chapters. Previous percolation schemes for LOQC [99] have made use of both these gates. Type-I performs a logical fusion operation in the computational basis, as presented in chapter 2, and only consumes one photon. However it is not robust to loss, and lost photons can be translated to logical errors [93]. A success event is heralded by the detection of a single photon, while detection of two photon or the vacuum indicates failure. However, when combined with loss these distinct events can become mixed. A failure that would have been heralded by detection of two photons, when one of the photons is lost, becomes a heralded success, hence the introduction of logical errors in the lattice. In contrast, Type-II fusion detects incident photons separately, and all photons entering the gate are measured. Any loss events can be identified, as the total number of detected photons will be measurably reduced by loss. The generalised Type-II gates, presented in chapter 4, share this loss detecting property. All photons must reach the detectors for the gate to succeed.

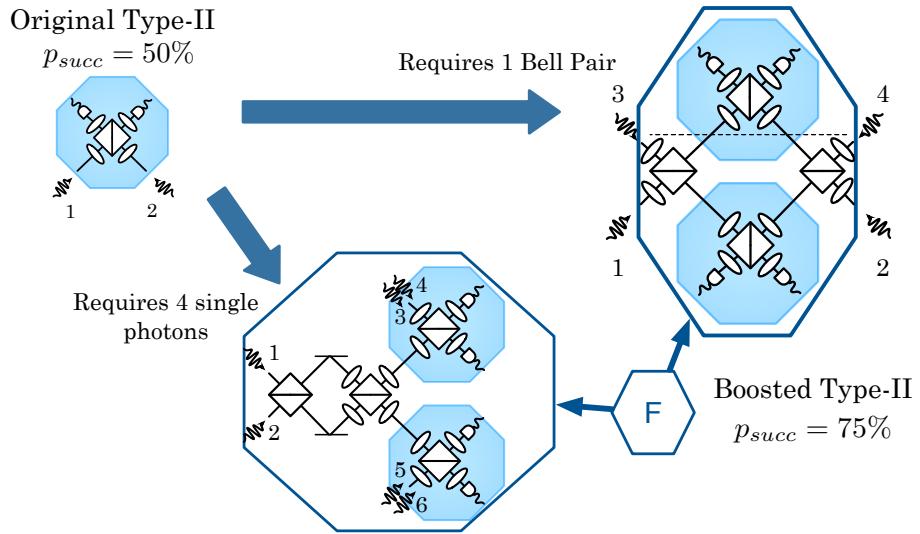


Figure 5.1: Boosted Type-II fusion gate, marked here onwards as an hexagon labelled “F”. Photons 1 and 2 represent the photons on which the gate is applied, the rest are ancillary photons. The implementation based on [111] requires a pair of maximally entangled photons, while the implementation based on [115] requires 4 single photons. The boosted gates have the exact same success and failure outcomes as the original Type-II but with a higher success probability. Note that all photons are measured.

The basic building block of our scheme is Browne and Rudolph’s Type-II fusion gate, which can be used to connect small cluster state fragments into a large cluster state for MBQC. This gate is equivalent to a Bell state measurement in a rotated basis. We use the boosted versions of the Type-II gate as proposed in chapter 4. The advantage of using Type-II fusion instead of

¹It follows that this proposal is different from a boson computer such as the one described in section 2.2.4, which is a non-interacting model for bosons.

Type-I as previous proposals [99], is that this gate detects lost photons and therefore does not introduce logical errors [93]. This enables us to develop a ballistic scheme with a lattice which a 50% Type-II gate would leave below the percolation threshold, but which 75% probability Type-II gates percolate to a universal resource.

5.3 Building the percolated lattice

The phenomenon of percolation has been long studied [109] in classical statistical mechanics as a prototype phase transition on graphs that have lost some of their bonds and/or sites due to a randomised process. As discussed in chapter 2, a ballistic strategy for LOQC can be implemented, where small photonic clusters are probabilistically fused together forming a percolated graph. This graph will define a cluster state, whose bonds/sites are effectively removed due to failure of probabilistic entangling gates together with photon loss. The percolation threshold marks a phase transition in the computational power of the resource state generated [104], which distinguishes the states that can be used for universal quantum computation from those which cannot. Having a universal resource is equivalent to saying that most of the bonds of the lattice are present.

Thus, our aim is to build a cluster state with gates that succeed with a probability higher than the percolation threshold. We want the cluster to be regular mainly because that implies that it can be built with a static linear optical scheme, which makes it simpler to realise experimentally. The resource state used to build the cluster would then be fed into a static linear optical network. The cluster construction is completely ballistic, all operations are independent and can be performed systematically on the input states as they enter the static network. The fusion gates needed to build first the micro-clusters that will go in each site of the lattice, and then the final cluster state, are performed independently of the success or failure of other fusions. In fact, all fusions could be performed in the same time step. In practice however, at any given point of the computation, only part of the cluster state will be formed (see section 5.9.2 for more details); this will reduce the amount of loss and error introduced by delays and/or photon memories. The advantage of this scheme is that it doesn't require multiplexing or feedforward, which speeds up the construction process and lowers the loss rate of each photon.

To be able to successfully perform UQC, only a 2-dimensional cluster state is required, even when using percolation [105]. However, the percolation thresholds for 2D lattices are comparatively high, and given that we have to account for the effect of probabilistic entangling gates and loss of qubits, 2D graphs become quite impractical and 3D lattices show much better prospects. It must be noted that the coordination number of each site in a 3D lattice is on average bigger than on a 2D lattice, and therefore the construction of 3D lattices is more expensive in terms of the number of entangling operations needed per site. As mentioned in chapter 2, the most favourable lattice to implement in a percolated scheme is the diamond lattice, as it is the 3D lattice with the lowest coordination number, namely 4, and yet it shows a low percolation threshold (in fact much lower than the 2D square lattice, which has the same coordination number per site in the lattice [108]).

In this work, we represent the internal structure of a diamond lattice as brickwork in 3

5. A PERCOLATION-BASED SCHEME FOR LINEAR OPTICAL QUANTUM COMPUTING

dimensions (Fig. 5.2). If one takes the diamond lattice and rearranges the directions of the bonds so that they all make right angles with each other, we find that the diamond lattice is isomorphic to brickwork in 3 dimensions, and indeed this description is very useful when arranging the micro-clusters to be fused (both in the computational simulation and the experiment). The diamond lattice is formally isotropic, however its depiction as brickwork is not. It is as if we had stretched the diamond lattice in one direction but not in the others. This does not change the connectivity, but it does change the average number of connections that a site has in each direction. There is a greater average connectivity in the \vec{x} direction and thus a preferred direction for percolation. As will be shown later, we have also optimised the process by which the lattice is generated to take advantage of this anisotropy.

In figure 5.2 we can see how the GHZ states are arranged to create the brickwork structure. For each site in the final lattice, we use three 3-GHZ states to create a five-qubit micro-cluster. Each micro-cluster is created by performing two rotated Type-II fusion gates [90], as described in figure 5.3. The 5-star micro-cluster will be created when both fusions succeed, however in the case of failure the outcomes will still create connectivity in the lattice, contributing still to the percolation of the whole lattice. In the case where we have formed a five qubit star graph state, all the qubits in the exterior are equivalent, however in the cases where failures have happened the way in which we arrange those external qubits affects the connectivity of the lattice. We have shown in figure 5.3 the arrangement that is most suitable for our scheme and that allows us to obtain the lowest percolation threshold.

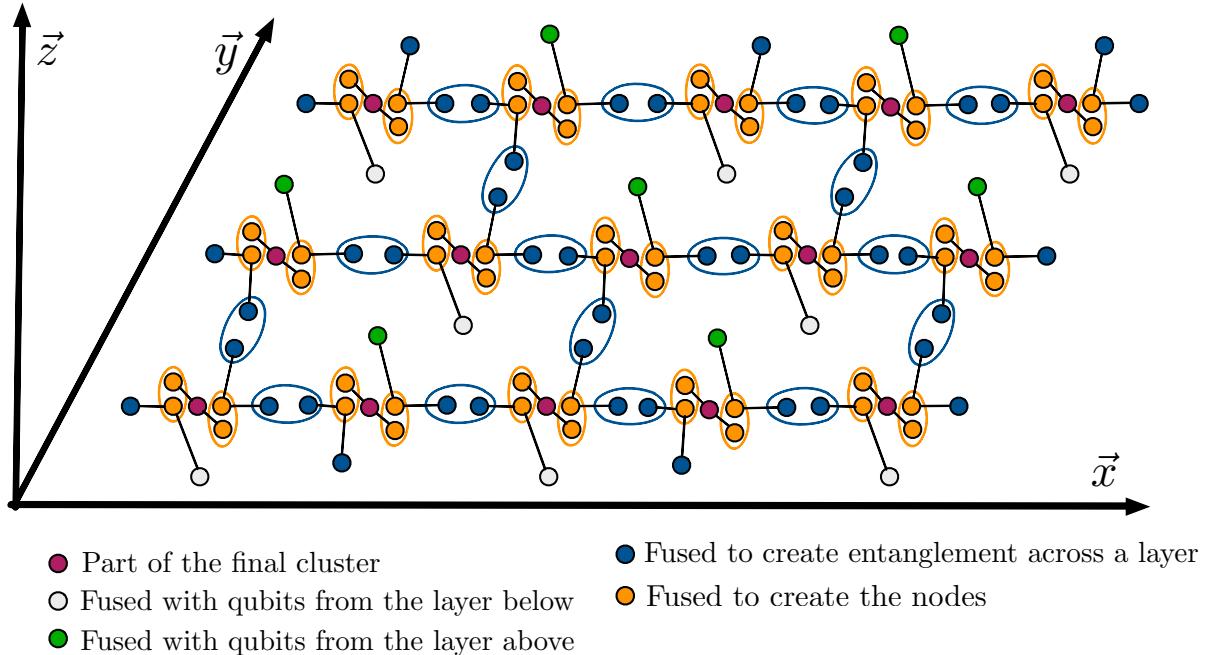


Figure 5.2: Full layout of a layer of the diamond graph using 3-photon GHZ states as input. The legend at the bottom of the figure shows the role of each photon. There are two types of rotated fusion Type-II gate used (marked by orange and blue open circles), their effect on the GHZ states is described in figures 5.3 and 5.4.

It is worth noting that in the case of failure of some of the fusion gates, the cluster will lose

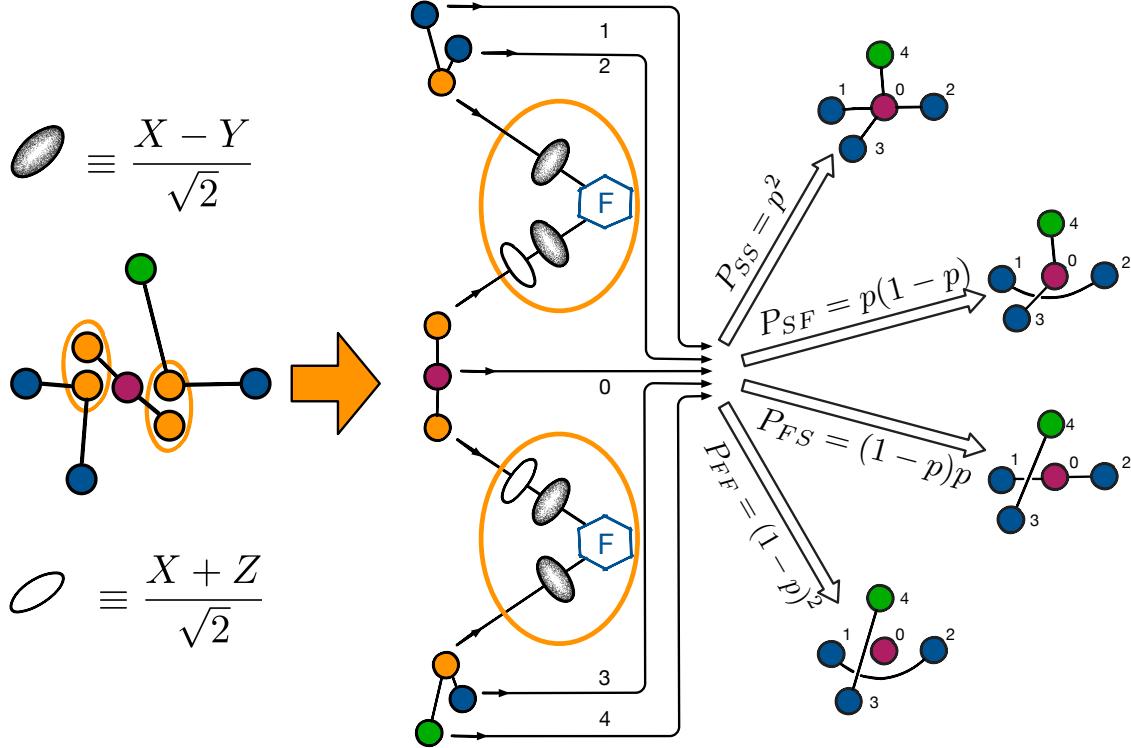


Figure 5.3: Probabilistic creation of star micro-clusters. The four outcomes correspond to both fusions succeeding (with probability P_{SS}), one failing and one succeeding (with probabilities P_{SF} and P_{FS}) and both fusions failing (with probability P_{FF}). The structure of the photonic states in each of these instances is also given.

bonds belonging to the lattice but it will also gain bonds that do not belong to the internal structure of the lattice. Figure 5.6 shows an instance of the percolated lattice, where the diagonal bonds give away the fact that the generated lattice is not a strict subgraph of the brickwork lattice. This occurrence cannot be accounted for in a simple percolation model, however it is beneficial for our scheme as we are mainly interested in increasing the connectivity of the lattice. It has an impact on how the percolation threshold is calculated, which we will address later on.

When attempting to fuse three 3-GHZ states in order to obtain a micro-cluster, there is only one valid success outcome. However, the failure outcomes (when both or either of the gates fail) can contribute to create entanglement across the lattice and the way these failure outcomes are connected can greatly enhance the percolation probability. In figure 5.5 we show two different schemes for micro-cluster formation, which differ in the failure outcomes. It can be seen that scheme B has optimised connectivity along the \vec{x} direction of the cluster. This micro-cluster will connect along the \vec{x} direction to other micro-clusters through qubits 1 & 2, along the \vec{y} direction through qubit 3 and along the \vec{z} direction through qubit 4. In scheme B, there always exists a path from qubit 1 to qubit 2, whereas that is not the case in scheme A or other similarly connected schemes, hence the optimisation. We can check this optimisation numerically and show that the brickwork lattice has a percolation threshold along the length $p_c \sim 63.8\%$ with scheme A and $p_c \sim 62.5\%$ with scheme B. Scheme A and B differ only in the organisation of the

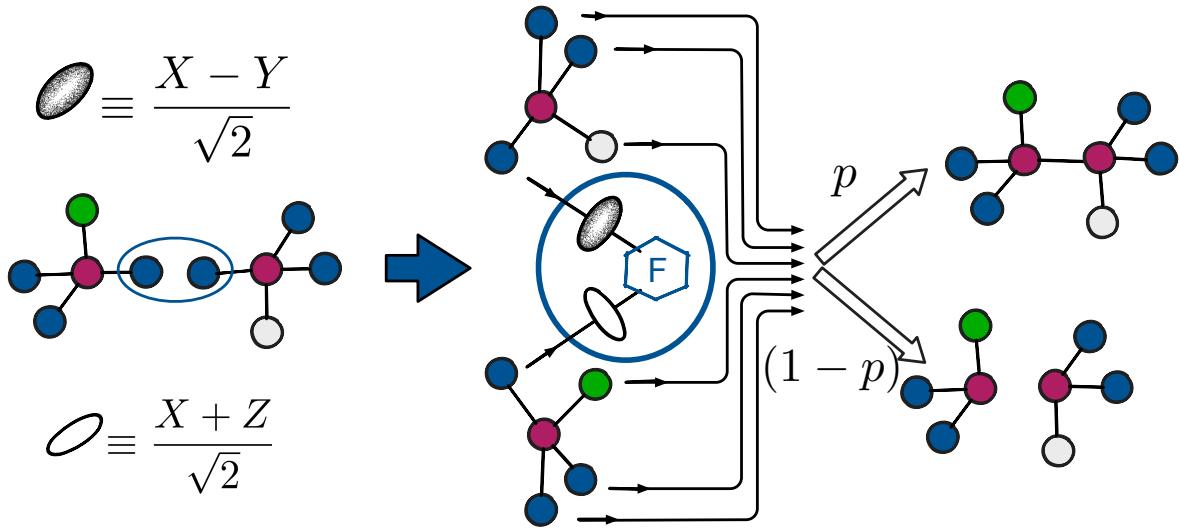


Figure 5.4: Fusion of 5-qubit micro-cluster to form the final lattice. We show the two possible outcomes of the attempted fusion of two 5-qubit micro-clusters, where p is the fusion success probability.

photons in the case of fusion failure. As can be seen from figure 5.3, we are not using Type-II fusion directly, but we are applying a certain $SU(2)$ prior to it. These rotations have the effect of preserving more entanglement in the failure outcomes than the regular Type-II gate (see chapter 4 for a full description of all possible rotations and outcomes). Up to a point, we can adapt Type-II fusion to have the success and failure outcomes most convenient for our scheme. Using Type-II (or a BSM) directly with no adaptation would lead to an overall less connected lattice (such as presented in an alternative scheme [163]), with a higher percolation threshold of $\sim 70\%$. Having a lower percolation threshold is important, not only because it allows for a higher tolerance for loss, but also because the expected size of the final perfect lattice (after the percolated lattice has been renormalised) depends on how far above the percolation threshold p is [105]. The overhead in resources increases as $p \rightarrow p_c$.

5.4 Percolation properties

To assess the percolation properties of the lattice, we use a Monte-Carlo simulation in which we produce many random instances of the lattice and find whether a percolating cluster exists for each instance. In each independent run, our simulation builds the lattice sequentially, modelling the action of the success and failure of the fusion gates and attempts to find a percolation path. In doing so, we achieve a more realistic picture compared to the simpler alternative of deleting nodes from a perfectly formed lattice. This approach also allows us to collect the information which will ultimately be fed to a classical percolation algorithm. For each set of parameters, the simulation is run 10^4 times to ensure that statistical error in the data is $\lesssim 1\%$.

In figure 5.6 we present an instance of the lattice, where we can see why this lattice is not the typical percolated diamond lattice (even when expressed as brickwork). As mentioned

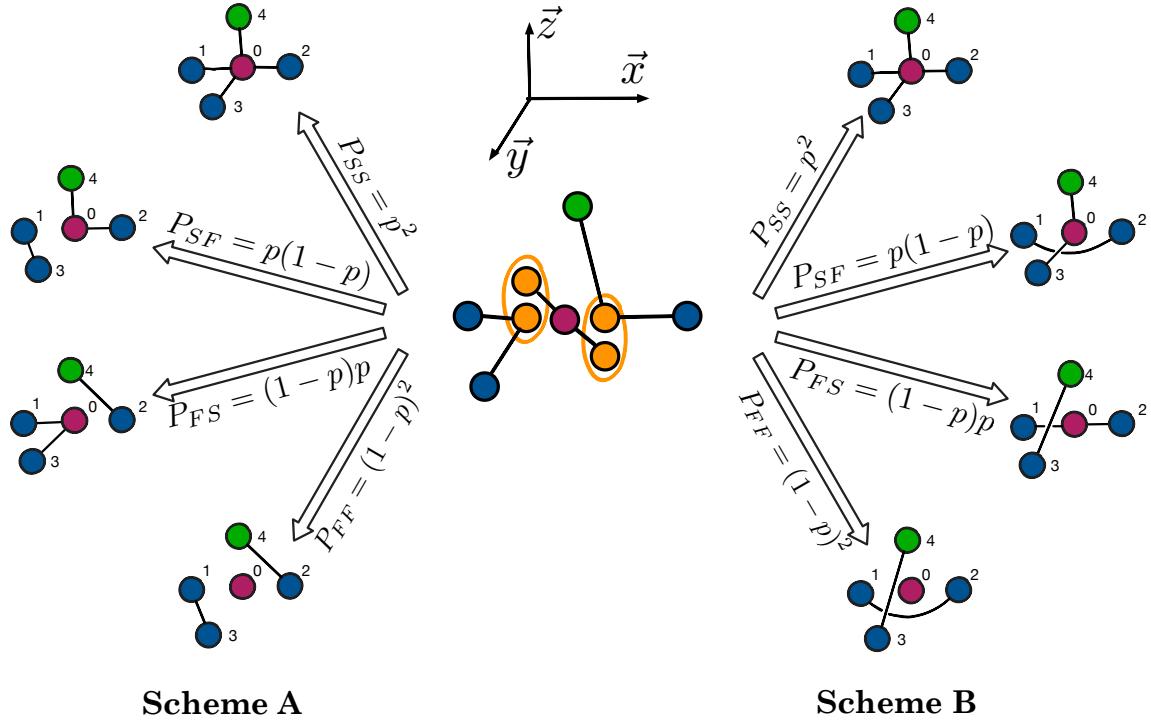


Figure 5.5: Here we present two different ways in which the connectivity of the micro-clusters can vary when the fusions fail. Scheme B has optimised connectivity along the \vec{x} direction, there always exists a path from qubit 1 to qubit 2, which are the qubits that connect to the other micro-clusters along the \vec{x} direction.

earlier, the failures of some of the fusion gates produce correlated bond losses together with the appearance of new diagonal bonds that can be seen in the figure. It must be noted that the presence or absence of the bonds will be known from the pattern of successes and failures of the fusion gates. Thus in any experimental set up, the structure of the percolated lattice could be inferred by a simple classical algorithm.

5.4.1 Calculating the percolation threshold

In percolation theory, percolation properties are defined for infinite dimensional clusters. It is in that limit where plotting the probability of percolation versus the probability of occupancy (in site, bond or mixed site-bond models) yields a step Heaviside function. However, for computational results, simulating an infinite lattice is impossible and we extract results from the analysis of smaller lattices. In fact in most models, Bose-Einstein statistics are assumed as the best approximation for the step function in the finite lattice case. However, this presumes knowledge about the model, in particular about the linearity of the dependence between our figure of merit (i.e. the occupancy) and the connectivity of the lattice. This is a justified assumption in most models, however in the model we are considering, the dependence is not as simple. Consider a single bond between micro-clusters, for example the bond between qubits 1 & 2 in figure 5.7. For that bond to exist, we require three separate fusion events to succeed, however if one of them fails, as is the case for the fusions between qubits 2 & 3 in the same figure, we can end up creating an extra bond, which is not in the regular lattice structure we have been considering.

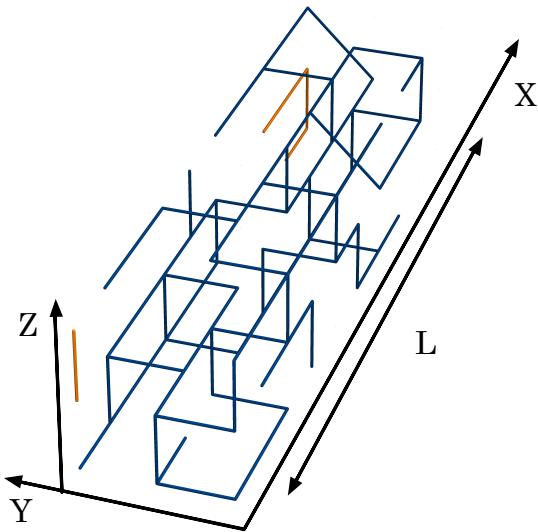


Figure 5.6: Instance of the percolated cluster ($10 \times 3 \times 3$), highlighted in blue is the spanning cluster. In addition to the orthogonal bonds which are expected in the canonical brickwork lattice, we see some diagonal bonds these are the result of failed fusions during the creation of microclusters.

It is obvious then, that we cannot assume a linear dependence between the lattice connectivity and the fusion probability. But we can still find the percolation threshold without making such assumptions.

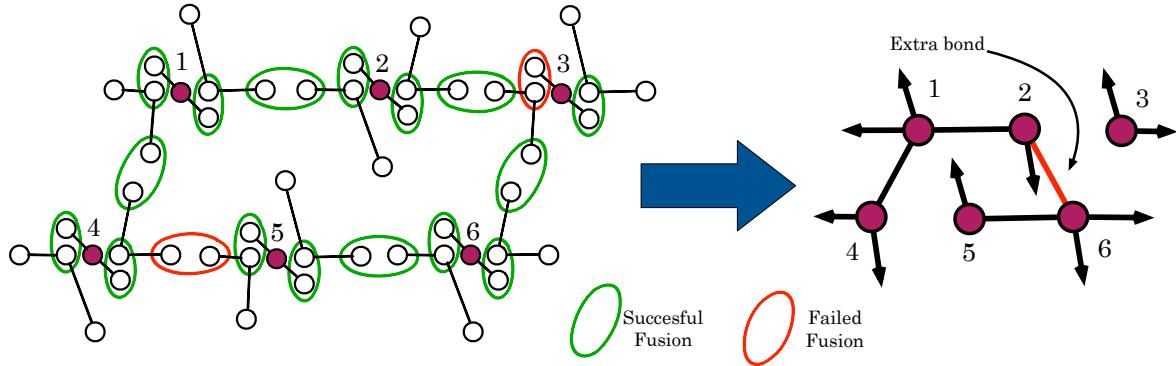


Figure 5.7: Pattern of fusions success and failure. Bonds between qubits only exist if all the fusions between the qubits have succeeded, as it is the case between qubits 1 & 2. If some fusions fail, we may end with no bond at all (as it is the case for qubits 4 & 5, or with correlated bond losses together with extra bonds outside the internal lattice structure, as it is the case for qubits 2, 3 & 6).

Let us define $\Pi(p, L)$ as the probability that a lattice of linear dimension L percolates when built with fusion gates that succeed with probability p . In the case of the infinite lattice we would have that $\Pi = 0$ if $p < p_c$ and $\Pi = 1$ if $p > p_c$, however in the case of a finite lattice $\Pi(p)$ for a set L , it will be a smooth function instead of the Heaviside step function due to finite-size corrections. To find the percolation threshold without making any assumptions

about the functional form of $\Pi(p, L)$ we use known results about the critical point in the context of renormalisation. The basic idea of renormalisation is the self-similarity of the lattice at the critical point (percolation threshold) [109]. The correlation length, ξ , can be defined as the typical cluster diameter, it diverges at the percolation threshold as that is the point where the infinite cluster first appears [103] and it grows monotonically with the occupancy p . What this means is that the size of any cluster at the percolation threshold is much smaller than the correlation length at the percolation threshold (which is infinite) and therefore all the clusters are similar to each other in an average sense.

When performing renormalisation on a lattice, we replace a cell of the lattice (that comprises many sites) by a supersite, provided that the linear dimension of the cell b is much smaller than the correlation length of the lattice ξ . At the percolation threshold, because of the self-similarity of large lattices, the properties of a renormalised lattice will be the same as the original lattice and therefore we will have $\Pi(p_c, L) = \Pi(p_c, L/b)$. That is to say that the value of $\Pi(p_c, L)$ does not depend on the renormalisation parameter and must therefore have the same value for all lattices of different size but with the same shape and dimension. We can thus conclude that to calculate the percolation threshold when we only have access to data in finite lattices, we should obtain values of $\Pi(p, L)$ for different p and L_i and we find the threshold by estimating where the functions of $\Pi(p, L_i)$ intersect.

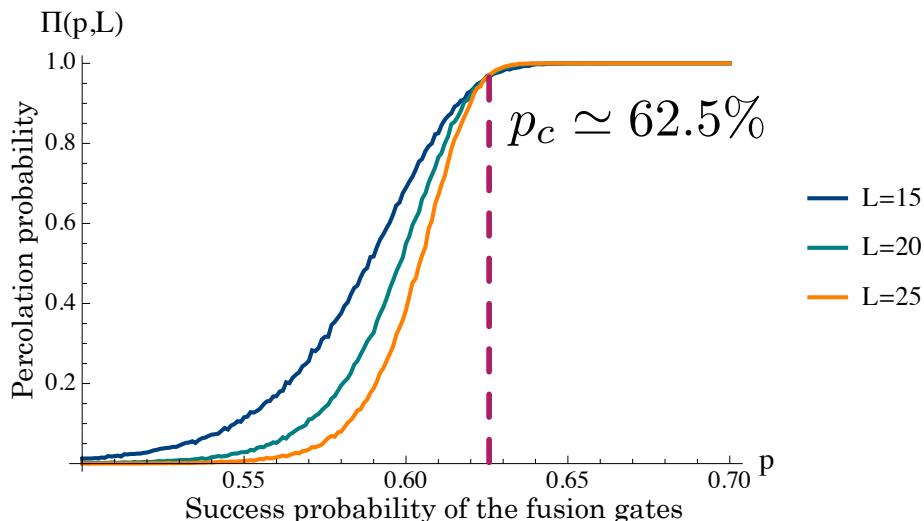


Figure 5.8: Results for simulations on a bulk of cluster of $L = 15, 20, 25$. Each cluster contain L^3 sites and has been generated from $3 \cdot L^3$ GHZ states.

We perform the simulation by generating instances of the lattice with fusion success probability p . In figure 5.8 we have represented the results for lattices of different linear dimension and found the value for the percolation threshold, which is estimated to be $p_c \simeq 0.625$. We conclude that lattices built according to our scheme, using boosted fusion gates with success probability of 75%, are well above the percolation threshold, and are therefore universal for quantum computing.

It must be noted that the boosted gates we have presented cannot achieve the entire range of success probabilities (see chapter 4) that we consider in our numerical simulations. But the

exercise of analysing all the values for p gives us insight into the behaviour of the model and allows us to find the percolation threshold for this scheme. The approach we have taken in this work has the intention of solving a difficulty in the generation of a cluster state in LOQC, but this model could be applied to any other physical system with probabilistic gates.

5.5 A single qubit channel

In traditional MBQC, a single qubit is replaced by a linear cluster (figure 1.2). When two-qubit operations are required, a bond (gate) is created between two linear clusters (qubits). In a paradigm where the creation of entanglement between qubits is probabilistic (such as in MBQC), a three-dimensional piece of cluster state can be used to implement a single functional qubit. If there exists a spanning path through the cluster, information can flow through the channel, allowing the computation to progress. We can then calculate how many operations we can perform on this single qubit.

The cluster channel is parametrised by a fixed cross section (width and height) and variable length, which corresponds to the computational depth. The cross section of this cluster is directly related to its percolation properties: a larger cross section gives a higher percolation probability. Given a desired length, we must choose a cross section in order to have a percolation probability higher than some desired probability of success. In figure 5.9 we show the percolation probability for different cross sections, as a function of the length. We have chosen square cross sections because the brickwork lattice's percolation properties are isotropic in the cross section plane. It was confirmed in preliminary simulations that this geometry performs better than rectangular shaped cross sections.

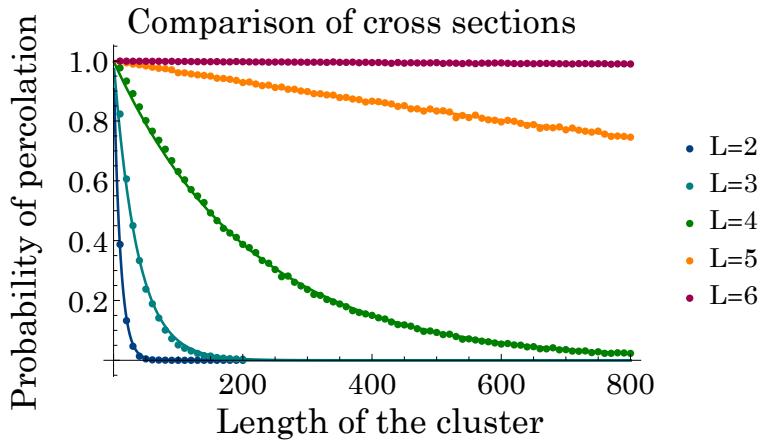


Figure 5.9: Percolation probabilities as a function of the length, for lattices of square cross section L^2 . The length of the cluster correlates to the computational depth of the lattice. The exponential decay shown has a decay constant γ which depends on L , the best fit we have found is $\gamma = e^{0.413L}$.

As we can see from figure 5.9, for a cross section of 6×6 qubits, we can make the cluster very long. Because of computational constraints, simulating large clusters is very challenging. We fit an exponential decay function to the data, obtaining an estimated variance of 10^{-7} .

From this fit we extrapolate that for $L = 6$, a cluster of length 9000 would be obtained with probability greater than $1 - 10^{-3}$.

5.6 Loss tolerance

A question that naturally arises in large-scale schemes for LOQC is tolerance to photon loss. This scheme has been designed with loss robustness from the outset. The Type-II boosted fusion gates can detect all losses that happen in the photons incident in the fusion gates. Our scheme is operating well above the percolation threshold for the lattice, and this headroom leads to a natural loss tolerance. The incoherence induced in the state by a loss error can be fixed by measuring neighbours of lost qubits in the Z basis², thus cutting all bonds from the lost qubit to the cluster state. We have simulated the building of the lattice where each photon has probability p_l of being lost, and when a loss is detected, we measure all neighbours of the lost qubits in the Z basis to cut it out. In figure 5.10, we can see the loss tolerance of a cubic lattice of $L = 25$ in blue, in orange we have highlighted the constant success probability of 90% for comparison. The success probability of the fusion gates used has been taken to be 75%. As we can see, the probability of having a spanning path is larger than 90% for loss rates of up to 1.6%.

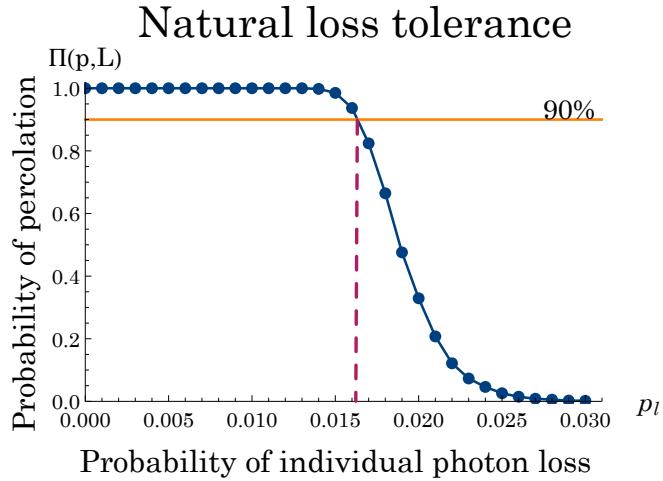


Figure 5.10: Loss tolerance (blue) for a cubic lattice of linear dimension $L = 25$. The individual photon loss is used as a single parameter that encompasses loss errors from the sources, waveguide photon absorption and detector inefficiencies.

We want to stress that this is a natural loss tolerance of the system. Previous proposals [99] have given thresholds for heralded loss, where the location of all loss errors in the final lattice is known. Heralded loss is not experimentally justified in LOQC and only serves as an upper bound for loss tolerance. In order to compare our scheme with previous work we have performed the same kind of heralded loss simulations (shown in figure 5.11) and found that in

²Other strategies, such as measuring stabilizer operators that allow for an indirect [113] Z measurement on the lost qubit, could also be used.

this scenario we could tolerate loss rates up to 15%, which is an improvement of 5% on the numerical results reported in [99].

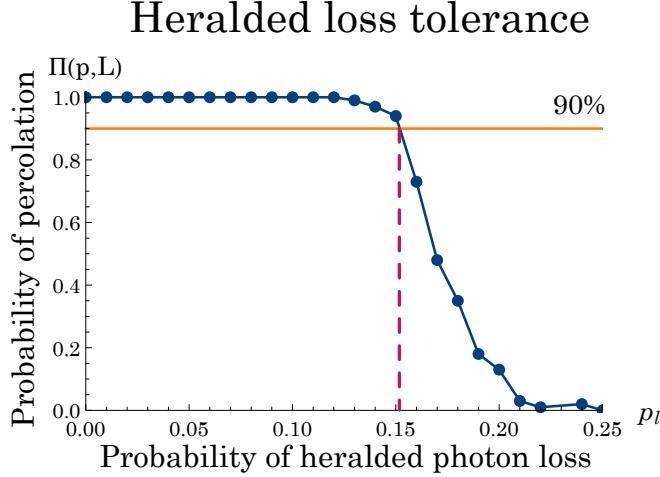


Figure 5.11: Heralded loss tolerance (blue) for a cubic lattice of linear dimension $L = 25$.

5.7 Scaling of resources

In this section we will explain the renormalisation of the lattice (following the procedure introduce by [99]) and assess the scaling, assuming that we have GHZ and Bell pairs on demand³ (for this account of resources we will be using Grice's version of boosted fusion). In figure 5.12 we show the procedure of renormalising the lattice: we take cubic pieces of the lattice and treat each of them as a renormalised qubit. When fusing the qubits that lie on the sides of these cubes we are applying CZ gates in the renormalised lattice. It must be noted that the entire cluster would be created as one big piece in the experiment, as opposed to first building renormalised qubits and then making a cluster out of these qubits.

In figure 5.12 we can see a realistic example of the renormalisation procedure, in which cubic pieces of the lattice become the new renormalised qubits. Within these renormalised qubits the part of the lattice highlighted in blue shows the spanning cluster while the orange highlights the disconnected parts of the cluster. As mentioned in previous sections, the lattice percolates better along the length in comparison with the other directions, which means that sometimes the renormalised qubit will be able to connect to other renormalised qubits along the length but not in other directions. When this is the case we won't be able to perform a CZ gate between the renormalised qubits (shown in figure 5.12 in red). These CZ gates that connect qubits across the width correspond to gates between logical qubits, which are more flexible as we can delay them or reconfigure the circuit slightly, therefore not posing a significant problem for the scheme.

Let's assume that our renormalised qubit is a cubic section of the lattice containing L^3 physical qubits. We renormalise figure 5.9 to show what is the probability of percolation when

³For linear optical circuits that generate this resources from single photons, see chapter 4.

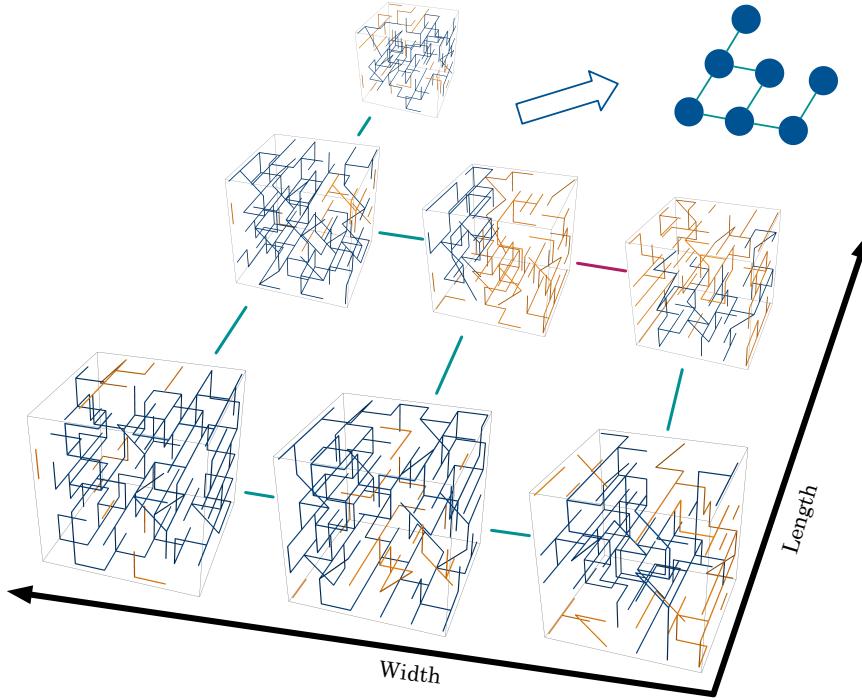


Figure 5.12: View of the lattice in terms of the renormalised qubits, which are cubic pieces of the lattice.

we fuse a certain number of these renormalised qubits, see figure 5.13. When we extrapolate the results for $L = 6$ to see how many we can fuse before the probability drops below 90%, we find that we can have a computational depth of 1500 before this occurs. For $L > 6$ the computational depth that can be achieved will be much higher.

We want a code with computational depth of k (by which we mean the number of measurements we will want to perform in each logical qubit; in the MBQC model, this corresponds to the number of qubits on one line of code). We also want to have n logical qubits (see figure 1.2 in chapter 1 where notation is more clearly indicated). For these variables, the values of the quantities involved and resources needed is:

- Number of renormalised qubits: $(n \cdot k)$.
- Total number of lattice sites: $(n \cdot k) L^3$.
- Number of 3-photon GHZs needed: $3(n \cdot k) L^3$.
- Number of fusions: $4(n \cdot k) L^3$.
- Number of optical elements needed per fusion (success rate of 75%): 15 polarisation rotators and 4 PBSs.

The variables that correlate with the size of the computer/computation are n and k as we don't expect to change the encoding (L). It might be the case that for some small computations we choose a smaller renormalised qubit to save resources, but for big computations, choosing a

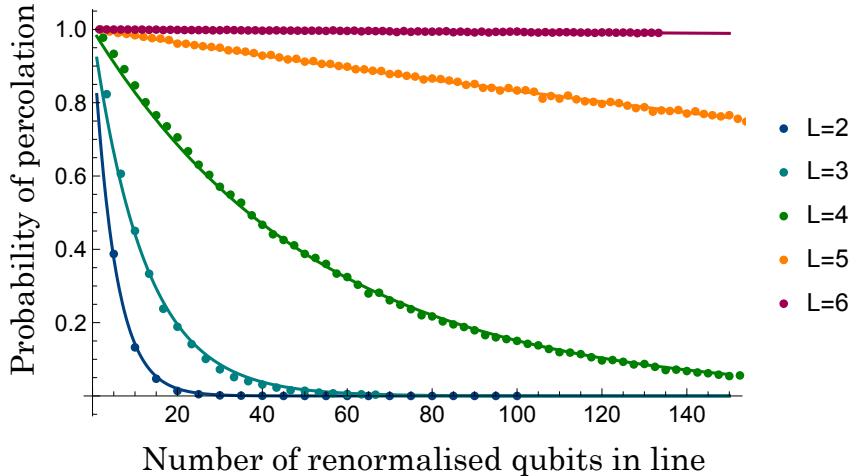


Figure 5.13: Probability of having a percolating path and therefore information flow, after fusing a number of renormalised qubits in line. The probability shows an exponential decay with the number of qubits fused, however for $L \geq 6$ the decay seems negligible when fusing $O(10^2)$ renormalised qubits.

renormalised qubit of $L = 6, 7, 8$ should be good enough for all, and therefore L^3 can be taken as a constant. Lower k values could be used if a loss tolerant code was used for the logical qubits, in which a failed percolation would be treated as a qubit loss. Therefore as the size of the computation only affects variable n and k , we can take it as if $(n \cdot k)$ was the size of the computation and the dependence of the resources on this variable is linear: given on demand 3-photon GHZ and Bell pairs, the scaling is *linear* on the size of the computer.

5.8 Comparison with previous percolation schemes

When designing a feasible architecture for quantum computing, the size of the machine (in terms of number of components and resources required) is one of the biggest concerns. The following comparison shows that our design utilises at least an order of magnitude fewer resources than the design of Kieling *et al.* [99]. To show this we extract data presented in their paper and compared it with ours under the same conditions.

In figure 4 of their paper [99], the authors show the dependence of the diamond lattice block size k^3 on the size L of the renormalised square lattice for three different sets of site bond probabilities (p_{site}, p_{bond}). The overall success probability threshold $P(L)$ was chosen to be $\frac{1}{2}$. In our work, we have performed all the simulations, assuming the GHZ states are provided deterministically. For the scheme comparison to be fair, we choose to compare with the data points that correspond with the data set $(1.00, 0.5)$. From the data used to produce figure 5.9 in our paper, we extract for different ks , what is the maximum value of L we can reach with $\Pi(L) \geq \frac{1}{2}$.

For a cluster size $O(10^2)$, Kieling *et al.*'s scheme requires a renormalised qubit of size $k = 7$

whereas our scheme⁴ can do with a renormalised qubits of size $k = 3$. This is already a significant improvement. But the difference becomes much greater once we consider the number of Bell pairs that are needed to build each renormalised qubit and the entire cluster.

To obtain this comparison, we will first calculate how many Bell pairs are needed to obtain a GHZ with probability greater than 99.9999%.

- The data in [99] is obtained for 4-photon GHZ states. For each 4-photon GHZ state we need 3 Bell pairs and the Linear Optical Network (LON) works with probability $\frac{1}{4}$. In order to have a deterministic 4-photon GHZ we must repeat the generation procedure t times, where t is such that $1 - (1 - \frac{1}{4})^t \geq 1 \Rightarrow t = 51$. In total we consume $3 \times 51 = 153$ Bell pairs in the generation of a deterministic 4-photon GHZ state.
- For this proposal we require deterministic 3-photon GHZ states. For each attempt at generating one, we need 2 Bell pairs and the LON works with probability of success $\frac{1}{2}$. In order to have a deterministic 3-photon GHZ we must repeat the generation procedure t times, where t is such that $1 - (1 - \frac{1}{2})^t \geq 1 \Rightarrow t = 21$. In total we consumer $2 \times 21 = 42$ Bell pairs in the generation of a deterministic 3-photon GHZ state.

With these numbers, we can transform the data of the size of the renormalised qubit for different cluster sizes into the number of consumable resources used. In figure 5.14 we can see the number of Bell pairs required to build an $L \times L$ lattice of renormalised qubits. We can clearly see that the resources required to build a renormalised cluster state are an order of magnitude smaller in our proposal in comparison with the scheme presented in [99]. It must be noted that we do not expect this block renormalisation strategy to be the best use of the percolated lattice. In [99] it is used to obtain an analytic proof of the scaling, and we use it preliminarily to be able to compare this proposal to [99]. In chapter 7 we will consider this approach among others in order to implement fault-tolerant LOQC.

To provide a quantitative resource comparison of the data presented in figure 5.14, we calculate the ratio between the number of Bell pairs needed to build a cluster state of a similar size for both schemes. Comparing points with L of the same order of magnitude, we find that [99] uses at least 14% more Bell pairs than our scheme to build the cluster state. We want to emphasise that our scheme offers further benefits in addition to this reduction in resources. The main differences between this proposal and [99] are the size of the resource states used to build the cluster state and the type of gate that is applied to these resource states in order to built the cluster state. In terms of resources, it is clear from the results presented here that building the cluster out of smaller GHZ states consumes less resources overall, the number of Bell pairs needed to probabilistically create the GHZ is lower and the probability of success is higher. It is also important to note that generation of smaller GHZ states implies a LON with fewer switches and fewer optical components, which reduces the overall loss rate of the photons. In the following section we will give more details about the experimental implementation of this proposal.

The type of gate used in [99], is the Type-I gate introduced in [90], this gate only measures one photon out of the two photons that go into the gate. A failure of the gate combined with

⁴More details can be found in appendix B.

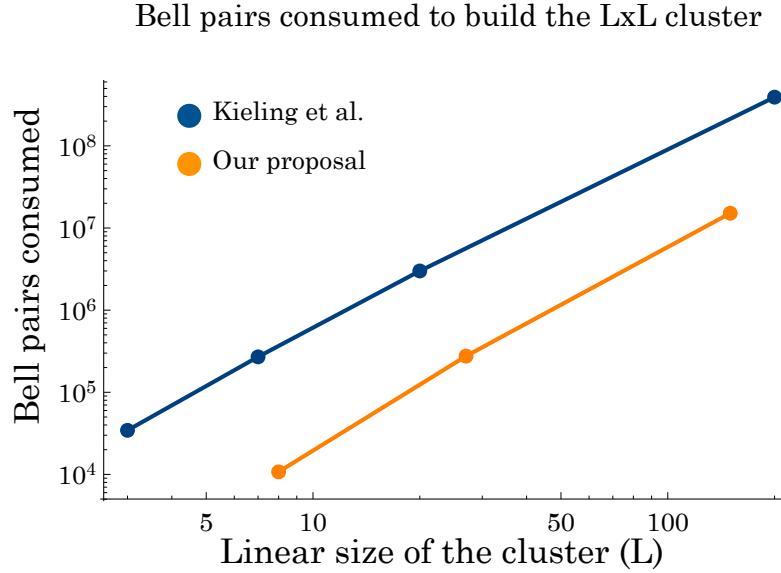


Figure 5.14: Comparison of the number of Bell pairs consumed to build the entire $L \times L$ cluster for different cluster sizes (L)

the loss of a photon would produce a false positive, inducing a logical error on the cluster [93]. In our proposal we only use the Type-II boosted gate which was introduced in chapter 4. This is a loss tolerant gate as it measures all the photons involved, and as such it can never be the case that a loss of a photon transforms into a logical error. Therefore, the improvements of our scheme over the scheme presented by Kieling *et al.* [99] are not only the reduction of the amount of resources needed, but also on the overall robustness of the construction, which is indicated by the 5% improvement on the heralded loss tolerance.

5.9 QNIX: a blueprint for linear optical quantum computing

In previous sections we have proposed a protocol for LOQC that is at least one order of magnitude more efficient than any protocol proposed before. However, when designing quantum computing architectures, not only resource scaling but also experimental feasibility must be addressed. In this section, we present a schematic view of QNIX, an experimental implementation of LOQC. We show all the necessary steps required to achieve UQC, from the generation of the necessary resource states to the kind of feedback control needed to perform MBQC. Current state-of-the-art experimental implementations are still not efficient enough to build such a universal quantum computer, however our architectural proposal only requires moderate improvements on the component specifications and not a completely different technology in order to successfully build a full-scale universal quantum computer.

One of the main advantages of percolation schemes for LOQC is that they yield an architecture with *fixed computational depth*. The number of optical elements in the photon path is fixed and reconfigurability only required for the basis of the final measurement of the qubit.

The architectural design we present in this chapter is *dynamical*, meaning that the cluster state will be built sequentially and there will be a classical control unit that will determine the measurement basis of the qubits based on continuous feedback from the fusion outcomes in the cluster-building layer, and the measurement outcome of previously measured qubits. This process will be repeated for every layer of the cluster state until the computation is finished.

This LOQC proposal is aimed at an implementation in integrated optics, in which miniaturised semiconductor chips are used to manipulate light. The ability to condense the linear optical operations in chips that can be easily integrated together yields a modular and monolithic architecture, with the advantage that chip integration in wafers (thin layers of silicon) provides intrinsic stability of optical phase and mode-matching. A number of experiments in quantum chemistry [164], quantum computing [72] and quantum metrology [165] have been recently successfully performed in this setup, and the ability to “fabricate entire 200 mm wafers populated with hundreds of thousands of working devices” [61] has also been demonstrated, making integrated optics a leading physical system for quantum computing. Multi-purpose reconfigurable chips have also been demonstrated [43], with a wide range of experiments exemplified on the same chip. This reconfigurability at the single chip level means that any large-scale architecture built from these elementary units can be fine-tuned and made multi-purpose.

The material in this section is the result of collaborative work with Pete Shadbolt, Dan Browne, Terry Rudolph and numerous members of the Centre for Quantum Photonics at the University of Bristol, in particular Jeremy O’Brien, Gabriel Mendoza, Jacques Carolan, Nick Russell and Josh Silverstone. The design of the deterministic 3-GHZ generator with minimised switching and the adaptation of the theoretical framework to the physical implementation are my main contributions.

5.9.1 Active switching only in state preparation

Active switching has the highest effect on the photon loss rate, not only because currently available state-of-the-art switches introduce at least one order of magnitude more loss per component than any other linear optical element [96, 97, 98], but also because theoretical simulations have shown that the performance of switches has the highest effect on loss thresholds [62]. It is therefore desirable to reduce the number of active switches to a minimum in the QNIX architecture. A key feature of the percolation scheme presented in this chapter is that it is *ballistic*, meaning that there is no active switching in the protocol, with the exception of the last reconfigurable measurement which cannot be avoided, as it is required by the MBQC and future QEC protocols that may be implemented. However, the protocol assumes that we have access to on-demand 3-GHZ states and ancilla states to boost the fusion gates. This is not a realistic assumption as even the production of deterministic single photons has not yet been achieved experimentally [65]. Nonetheless, adding a single layer of switching for the state generation allows us to use non-deterministic single photon sources to implement this protocol.

In order to be able to produce on-demand 3-GHZ states from non-deterministic single-photon sources, we require a nested multiplexing scheme⁵. In this scheme, on-demand single

⁵Experimental source efficiency [58] is still lower than 10%, therefore according to the results in figure 4.15, a nested multiplexing scheme will be more resource efficient than using probabilistic single photons.

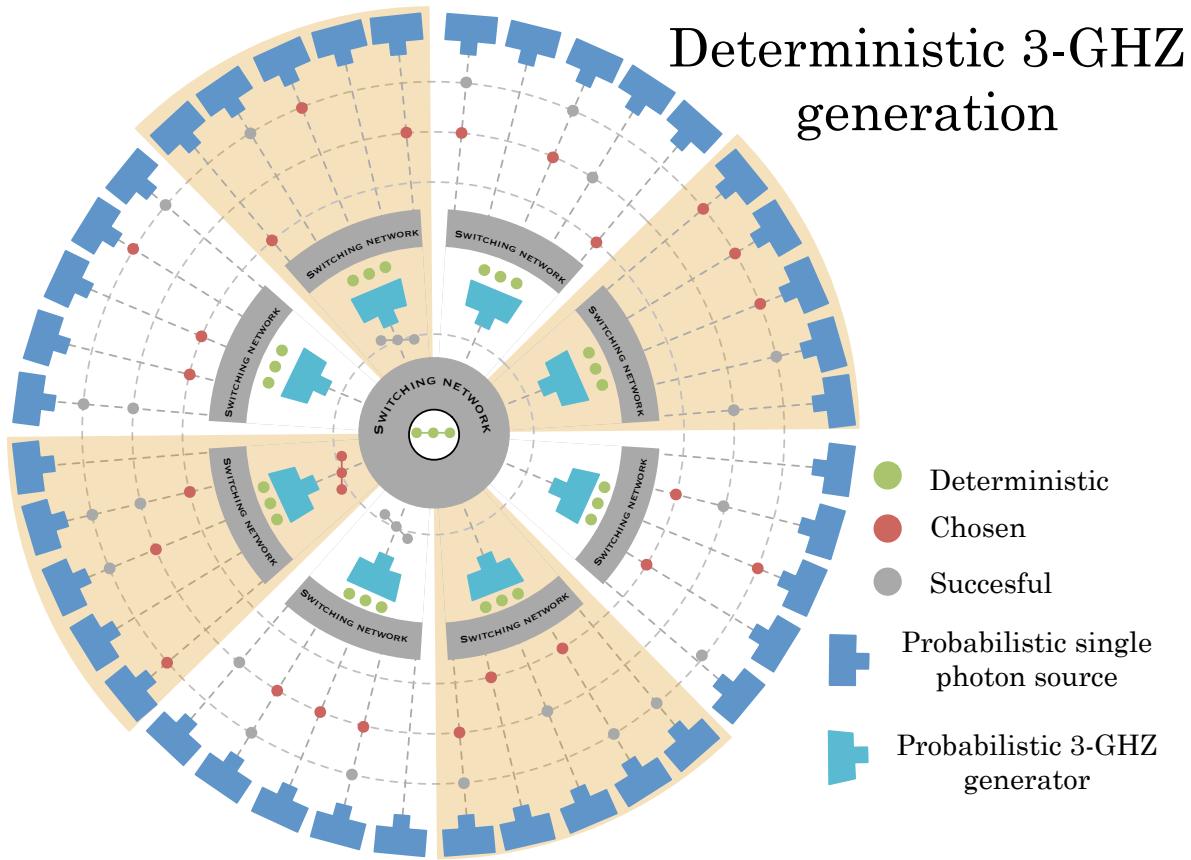


Figure 5.15: Schematic arrangement of probabilistic sources and switching networks that allow the construction of a deterministic 3-GHz generator. In the outer-most layer, we have probabilistic single photon sources that generate photons in a multiplexed scheme. Photons (marked in red and grey) are generated, but only the events marked in red will proceed through the switching network to the next layer. Once these photons pass the switching network they are considered deterministic as they are located in the appropriate bins to enter the probabilistic 3-GHz generators. Again, the successful outputs of the 3-GHz generators are marked in red and grey, but only the red one will proceed through the switching network to become the deterministic 3-GHz produced by this generator. Dashed grey lines indicate the spatial-temporal bins in which photonic events are produced, and the orange and white partitions allow to identify the multiplexing events corresponding to each probabilistic 3-GHz generation. Note that this is a simplified version of the design, as due to its complexity, it would be difficult to condense in one informative figure: the number of spatial and temporal multiplexing layers, as well as the number of photons needed for each stage is reduced from what is actually needed.

photons are first produced from probabilistic single photon sources, and then used to generate 3-GHz states. These in turn are also multiplexed in order to produce on-demand 3-GHz states to feed the percolated lattice generator. A schematic view of this nested multiplexing scheme can be seen in figure 5.15. The purpose of this figure is to give an idea of the general layout of the deterministic generator. It must be noted that this figure is not accurate in the number of multiplexed spatiotemporal bins required, or the number of photons required for the 3-GHz generators. An accurate figure of the deterministic 3-GHz generation would be too complex to be informative. Accurate calculations of the number of multiplexing stages required to build

3-GHZ states from single photons are given in chapter 6.

Probabilistic single photon sources are multiplexed both spatially and temporally. The successful production of single photons is known, as heralded single-photon sources are used, and therefore we know the spatiotemporal bins where the photons are located and can reconfigure a switching network in order to put forward the chosen single photons. After these switching networks, the single photons that come out of the output ports can be considered deterministic, and therefore the probabilistic 3-GHZ generators are fed on-demand single photons as required. The probabilistic 3-GHZ generators, which use the design first proposed in [93], require 6 deterministic single photons (not only three as shown in figure 5.15) and produce one 3-GHZ state with probability 1/32. The success or failure of the 3-GHZ generation is also heralded by the measurement pattern of the detections. This particular generation circuit has been chosen as we showed (see appendix G) that it required less multiplexing stages and could tolerate higher loss rates per active component in the switching network.

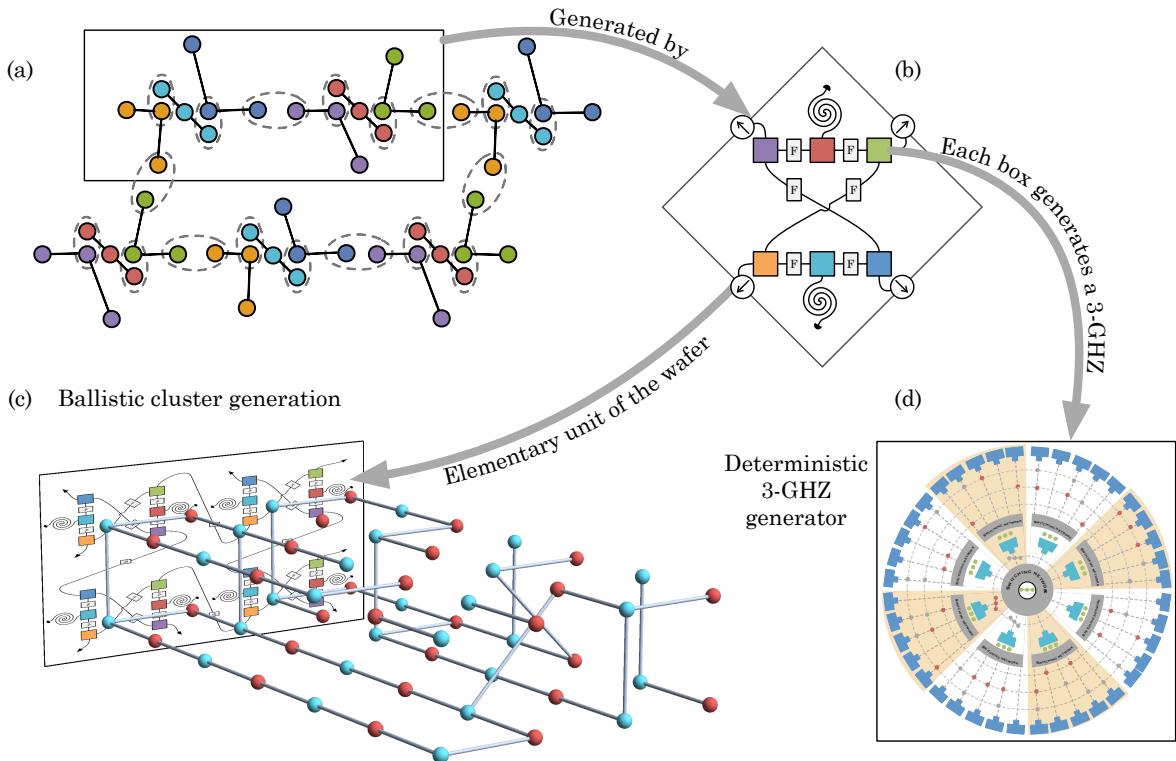


Figure 5.16: (a) All 3-GHZ states that are used to build the diamond percolated lattice can be classified in six types according to the fusion operations they undergo. The enclosed 6 states can be generated by the elementary unit of the wafer, shown in (b). (b) Modular cell which generates the enclosed 6 GHZ states of (a). Note that they have connections to the surrounding cells to created an interconnected cluster. Each of the coloured boxes represents a deterministic 3-GHZ generator, shown in (d) and figure 5.15. (c) Wafer (flat sheet of silicon) onto which modular cells (b) are etched. They interconnect in order to produced the percolated cluster as shown. (d) Deterministic 3-GHZ generator, shown in detail in figure 5.15.

In figure 5.16 we show how the deterministic 3-GHZ generator is integrated in the percolated cluster state generator. The deterministic 3-GHZ states that are used to build the percolated cluster can be classified in 6 classes depending on the fusion operations they undergo. Therefore,

5. A PERCOLATION-BASED SCHEME FOR LINEAR OPTICAL QUANTUM COMPUTING

the elementary unit of the wafer deterministically generates one of each of the 6 classes, this unit is repeatedly etched on a wafer a number of times, corresponding to the cluster size required. There are 6 deterministic 3-GHZ generators per elementary unit, these can be directly located on the wafer where the fusions happen, or can be located in a different wafer and the deterministic 3-GHZ states are forwarded to the fusion wafer. The ancilla states required for the fusions can be generated in a similar way to the 3-GHZ states in the state generation layer, we have not included them in the schematic view of the architecture to simplify the structure while keeping the key stages.

5.9.2 A dynamical architecture with fixed physical depth

The QNIX architecture is dynamical. In standard MBQC the cluster state is assumed to be generated in advance and only after the building process is finished, all the measurements are performed. However, storing a photonic cluster state for as long as the computation needs to run would be catastrophic, as the loss rate would increase dramatically due to the long delays. To avoid this situation, we propose to build the cluster state dynamically, so at any given point of the computation, only part of the cluster is created. The part where the earlier operations were performed is already measured and the part where future operations will take place has not been generated yet. The size of the percolated cluster that needs to be stored in the delay lines (which we refer to as “stored cluster”) at any point mainly depends on the speed of the classical control in calculating the basis of the last reconfigurable measurement. MBQC only imposes the constraint that all the correlations of a qubit must already have been formed before any measurement takes place. However, the success of local percolation algorithms should be taken into account, since if the stored cluster is made too small, the path-finding algorithm might fail even when we are in the super-critical percolation regime⁶.

In figure 5.17 we present a schematic view of the full QNIX architecture. Active switching is only present in the 3-GHZ generation layer and in the unavoidable reconfigurable measurement at the end of the computation. The fixed physical depth can be appreciated as the number of layers each photon goes through is fixed. For the qubits that are used to build the correlations of the cluster there are only two steps, generation of entangled state (which can be estimated to $O(10)$ optical elements) and fusion. For the qubits which are part of the percolated cluster there are three steps: generation of the entangled state, long delay and reconfigurable measurement. The size of the computation only affects the width of the wafers and the running time, as a bigger cluster will need to be produced, but it doesn’t affect the number of operations performed on a single qubit. The fusion operations that create the cluster are set in a static network that generates time layers of the percolated lattice, which in figure 5.17 are represented by the planes perpendicular to the time direction. The fusion outcomes allow us to visualise the inner structure of the percolated cluster and it’s therefore used by the classical control unit to calculate the final measurement basis of the qubits, which is sent to the measurement layer. The

⁶In the super-critical percolation regime a spanning cluster and hence a spanning path always exists. However, when the path-finding algorithm only has access to local information, it will not always choose the correct path, which will lead to diminished percolation probability. Initial simulations have shown that the performance will highly depend on the algorithm used, however the probability of percolation throughout the full lattice will be lower as we make the stored cluster smaller.

outcomes of the final reconfigurable measurement are sent back to the classical control unit in order to reconfigure future measurements to account for the negative outcomes in MBQC, loss and errors. The time that the classical control unit needs to calculate the measurement basis of the qubits is the time that we require the cluster to be stored in long delays. Therefore, not only improvements on the quantum computing protocols and experimental implementations are needed, but also fast algorithms for the classical computation are needed to reduce loss error rates and improve overall efficiency.

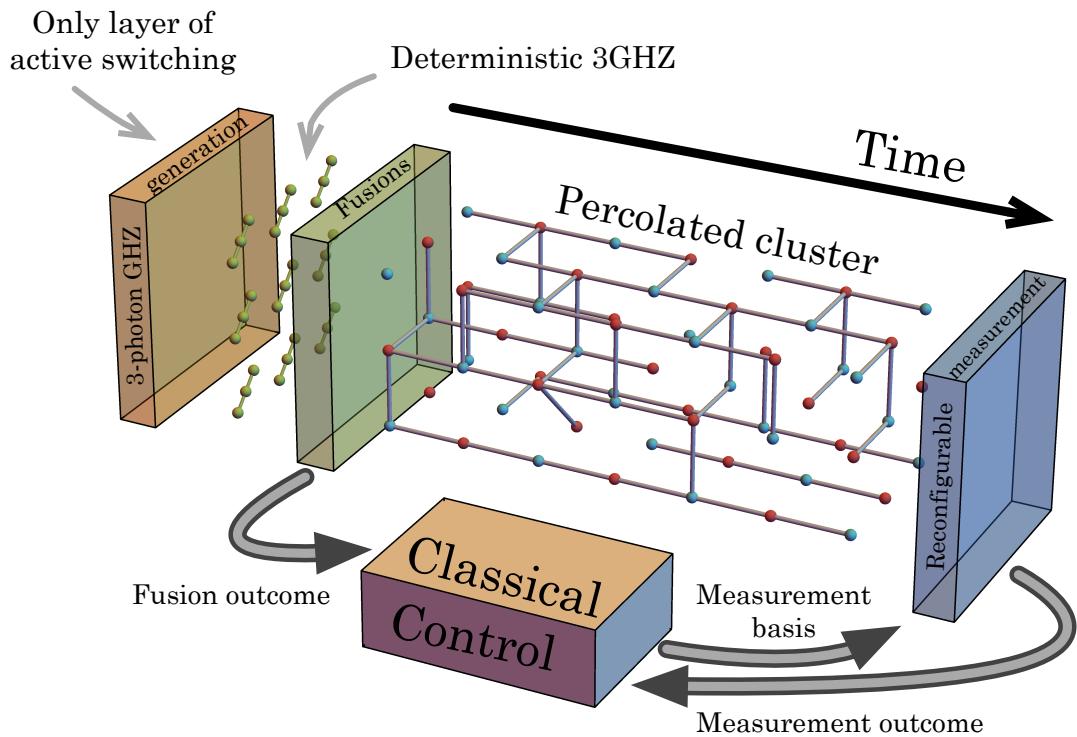


Figure 5.17: Schematic view of the dynamical QNIX architecture. It has only one layer of active switching in the generation of 3-GHZ states, in which waves of deterministic 3-GHZ states are produced. These deterministic GHZ are fused in an array of fusion operations that is static and requires no reconfiguration. The percolated cluster is generated and its exact internal structure can be inferred from the fusion operations outcomes. These outcomes are sent to a classical control processing unit while the percolated cluster is stored in long delay lines. The classical control unit performs path-finding algorithms on the cluster and, combining this information with the measurements necessary to perform MBQC and QEC, decides the measurement basis of the final reconfigurable measurement, which is then instructed and sent to the final layer of detectors. The outcome of these final measurements is fed back to the classical control to perform adjustments in the measurement basis of future layers. The fixed physical depth of this architecture is easily perceived in the figure as the number of computational layers applied to single photons.

5.10 Discussion and outlook

We have presented a ballistic scheme for the construction of a linear optical cluster state that is universal for MBQC. While we have not explicitly included error-correcting codes to provide robustness to loss and errors in the final photonic computational cluster state, the universality of the cluster state implies a number of ways forward, incorporating tree-clusters [113] or the surface code [166, 54] as loss-error and general-error correcting codes. Raussendorf's 3D cluster encoded surface code [55], in particular, seems well suited to ballistic generation. There are a number of approaches which can be followed, which are discussed further in chapter 7.

To implement this scheme with only 3-photon GHZ as resources we have proposed a boosted fusion mechanism based on [111] and [115] that works with 75% probability, which is well above the percolation threshold ($p_c = 62.5\%$) of this lattice. We have shown the robustness of the scheme in the presence of small amounts of photon loss (up to 1.6%) and its favourable resource scaling. Even though this scheme was devised with linear optics in mind, it applies for any physical system with probabilistic gates, and if that probability is higher than 75% it is conceivable that the resources needed could be reduced much further.

For this scheme to be implemented experimentally, it would need a near-deterministic 3-photon GHZ source. It is not yet known what is the optimal way of producing these photonic states, options range from multiplexing a linear optical circuit such as that proposed in [93], using a similar scheme to the multiplexed single photon source such as [161], to producing a 3-photon linear cluster (local Clifford equivalent to a GHZ) with a quantum dot [162]. As any linear optical fully loss detecting gate must necessarily measure all photons incident on it, the 3-photon GHZ is the minimal resource for a loss-detecting BSM-based ballistic scheme. A ballistic LOQC scheme based solely on single-qubit or 2-qubit resources remains desirable, but this would require an approach other than the Bell-state-measurement-based scheme proposed here.

We have explored one way of producing 3-GHZ states and integrated it into a full architectural blueprint of the percolation-based LOQC protocol, where we have used current technologies, such as log-tree switching networks and non-deterministic single photon sources, showing that they can in principle be used to build a photonic quantum computer. Having this architectural view allows a more detailed discussion of technological considerations and brings the theoretical proposal closer to a experimental realisation. Improvements on the architecture can be made by studying it from both the theoretical and experimental point of view. In chapter 6 we will present a new type of multiplexing scheme based on a technological consideration which improves resource efficiency significantly.

Ballistic generation of cluster states for MBQC remains the most attractive approach to scalable linear optical quantum computing. By developing a loss-tolerant and significantly more resource efficient scheme, we have shown that new theoretical ideas continue to ameliorate the technical challenges of building a scalable linear optical quantum computer.

CHAPTER 6

IMPROVING RESOURCE EFFICIENCY

6.1 Introduction

In chapter 5, we showed that an adaptation of recent improvements on Bell-state measurements [111, 115] to the percolation cluster state generation scheme [99] allows for a new approach to ballistic LOQC with significant reductions in resource consumption. Furthermore, we presented a technological blueprint for the implementation of this scheme, considering all the necessary steps to achieve the required states and probabilities.

In this chapter, we present a collection of results that are related to the resource efficiency of the proposed linear optical architecture in chapter 5. Efficiency considerations made in the abstract theoretical model must be tailored to the intended experimental setup, so that adjustments and improvements can be made. In this chapter we show preliminary results on a new type of multiplexing scheme, Relative Multiplexing (RMUX), and its effects when applied at a single level of the architecture. RMUX is an example of an observation inspired by technological considerations that has drastic implications on both the theoretical efficiency of the scheme and the experimental requirements for optical components. We also show how wasteful current state generation is and how improvements can be made by applying RMUX at all levels of the architecture.

These results have been obtained in collaboration with Gabriel Mendoza, Pete Shadbolt, Josh Silverstone and Terry Rudolph. In particular, it was Gabriel Mendoza who came up with the original idea of relative multiplexing, and Terry Rudolph who suggested reusing the redundant states created in the multiplexing process to boost the success probability of the percolated lattice. The calculation and simulation results are my own work.

6.2 Relative Multiplexing

When considering multiplexing of probabilistic operations (as explained in chapter 4), there exists the underlying assumption that the flagged successful event will be always relocated to the same spatial mode or temporal bin. That is, out of all the events that have succeeded, one is chosen and pushed forward in the architecture, using an array of switches to locate it in a predetermined optical mode. This type of multiplexing scheme will be referred to as “Homogeneous MUX”. If we only required the generation of a successful event in *any* bin, we wouldn’t need to use switches at all (thus reducing the loss rate of the photonic state), we

would only need to have the knowledge of the mode in which that photon is located. The only reason we might want to change the optical mode of a photon is because we want to have it synchronised with other events. For the purposes of the architecture presented in chapter 5, the fusion operations where HOM interference takes place require photons to be synchronised. However, there isn't any reason to move the mode of *both* photons that are going to be involved in the fusion, we only require them to be in the same mode (whichever that is), and moving only *one* of them would achieve the same purpose. This relative synchronisation of events is the heart of RMUX.

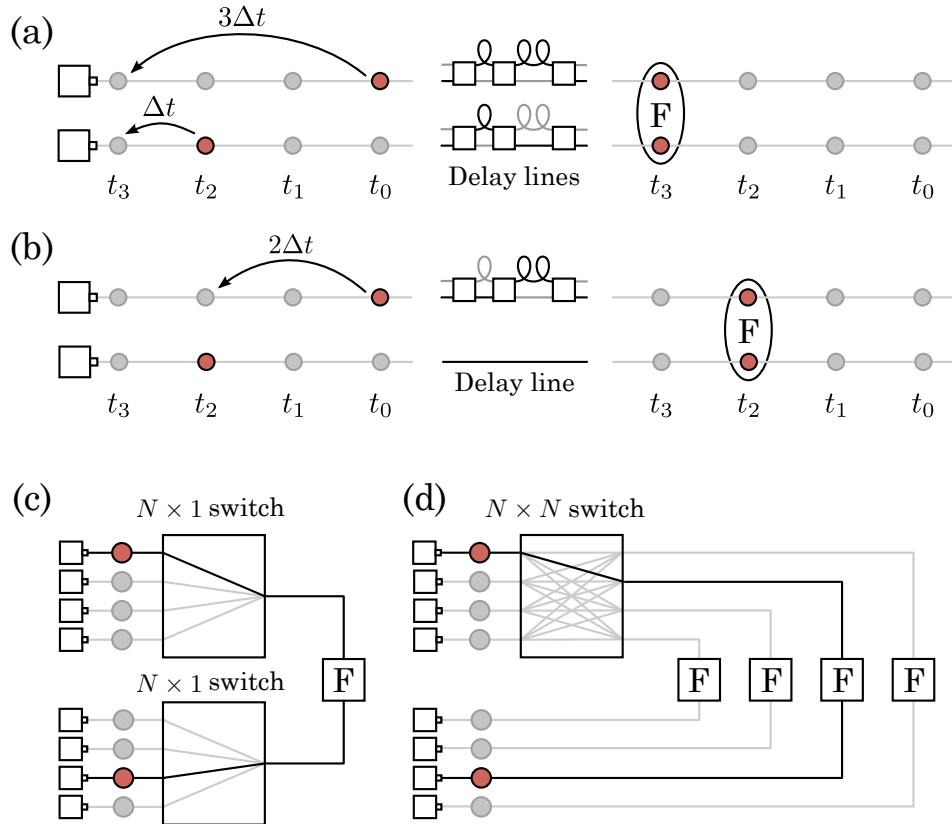


Figure 6.1: Comparison of relative MUX with homogeneous MUX in the task of synchronising two photons for a fusion operation. (a) Temporal homogeneous MUX: two photons are synchronised by delaying both to the same time bin. (b) Temporal relative MUX: two photons are synchronised by delaying the photon ahead in time to the time bin of the second photon. (c) Spatial homogeneous MUX: two photons are synchronised by re-routing both photons, each through an $N \times 1$ switch, to the same spatial mode. (d) Spatial relative MUX: two photons are synchronised by using an $N \times N$ switch on one photon to locate it in the same spatial mode as the second photon. Figure courtesy of Pete Shadbolt.

Using RMUX, events don't have to be synchronised to an overall clock cycle, but instead they are only synchronised with respect to other events. The LOQC framework becomes *asynchronous* as a whole and only photons that need to interfere at beam-splitters are lined up by changing the relative delay between them. This new asynchronous paradigm allows for a better usage of resources and less stringent requirements on optical components, as we will illustrate in following sections. The simplest version of RMUX can be seen in 6.1, where we show the

synchronisation of two photons using homogeneous and RMUX.

As an example, let's look at how temporal synchronisation of two photons in RMUX, where only one of them passes through a switch, can be performed. Each clock cycle, is $2N$ time bins long and photons are probabilistically generated in the first or last N time bins. Out of the two photons involved in a fusion, the one that will go through the switches (referred hereafter as photon 1) is always temporally ahead of the other photon (photon 2), as we can always *delay* photons but not promote them ahead in time. Once both photons are generated and the delay between them is calculated, photon 1 passes through a switching network with delay loops in a binary division configuration (from 0 to $2N - 1$), which locates photon 1 in the same time bin as photon 2.

In this chapter we present results for a basic implementation of RMUX at only one level: half the photons involved in fusion are assumed to not pass through any switches. However it is not known if this is the optimal way of implementing RMUX, or an implementation with variable delays on all photons would have better performance¹. Even so, the results of the simulations show the potential of this new multiplexing scheme.

6.2.1 RMUX in a percolated lattice

Recall the arrangement of fusions to build the 3-dimensional lattice proposed in chapter 5. The GHZ states used to build the percolated lattice can be classified in 6 types according to the fusions they undergo. In figure 6.2 we can see a portion of the lattice, with the different types of GHZ states marked, as well as the fusions that the photons undergo. In figure 6.3 we can see an arrangement of the generation of the GHZ states and the corresponding delays and fusions to implement relative *time* multiplexing.

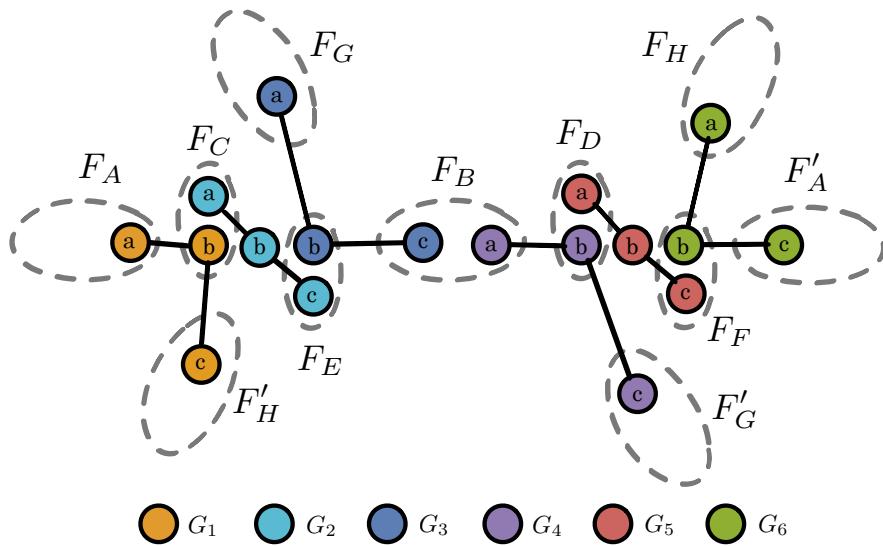


Figure 6.2: Portion of the photonic lattice, in which we indicate the different GHZ states by different colours and the label G_i . The fusions between them are also labelled F_X . This notation allows for a better understanding of figure 6.3, which shows the arrangement on chip of the GHZ generation and the relative multiplexing of the fusion operations.

¹Preliminary simulations suggest that an implementation with variable delays on all photons performs better.

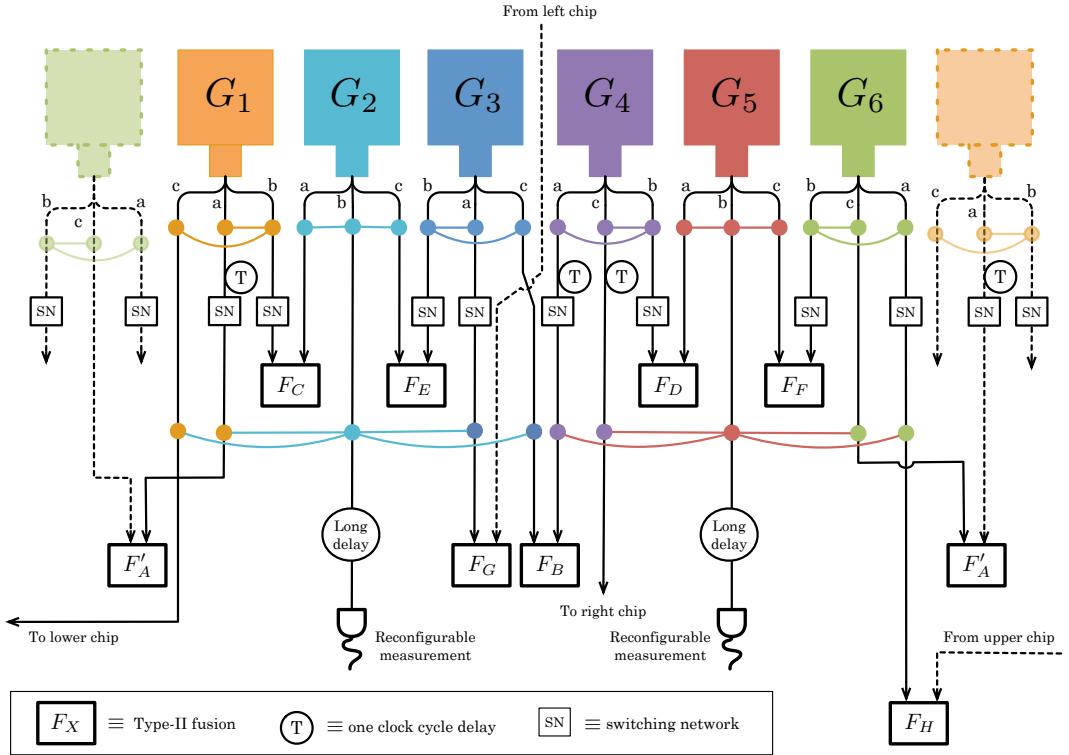


Figure 6.3: Arrangement of the GHZ state generators and fusion operations on chip. It must be noted that this is just a concept figure, and the length of the waveguides has not been arranged appropriately for timings. The notation of this figure follows that of figure 6.2. The arrangements presented here will be repeated in a wafer (such as the concept figure 5.16) where the different chips interconnect as shown. Short delays of one clock cycle in order to synchronise fusions are marked with the letter T, while the switching networks required for the relative multiplexing are marked with SN. The dashed lines represent elements from nearby 6-generator structures. The qubits which undergo the long delays are the *data* qubits and are only subject of one active element, the final measurement must be reconfigurable as required for the MBQC protocol.

The arrangement in figure 6.3 has been chosen for several reasons. First, it implements relative time multiplexing for every fusion, and we multiplex the same number of photons per GHZ state. For the purposes of the numerical simulation, where micro-clusters are generated with the different configurations (see figure 5.3) directly, we required that the photons that were multiplexed in both micro-clusters, formed by GHZ states $\{G_1, G_2, G_3\}$ and $\{G_4, G_5, G_6\}$ respectively, had the same delays. From figure 6.3, it can be noted that the operations and delays of photons in GHZ states G_1 , G_2 and G_3 are the same as the operations and delays on states G_4 , G_5 and G_6 respectively.

Loss from active elements:

We can classify the photons from the GHZ states in 3 classes, depending on the operation performed on them, we will label them Types A, B and C. We consider only loss from active elements and therefore delays and passive elements are considered lossless. This reflects the

$\sim O(10)$ difference in loss between active and passive components, as highlighted in chapter 2.

Type A photons will be part of the final cluster, i.e. the data qubits. To this type belong photons labelled as $G_2(b)$ and $G_5(b)$. We want to minimise the number of active elements they are subject to, and therefore we don't put them through any multiplexing stage, they only have one reconfigurable measurement (which cannot be avoided as it is part of the MBQC scheme). Therefore the loss rate due to the switches for this type of photos is γ_{sw} . Note, that this type of photon will have to go through a long delay line to allow time for the classical processing (percolation, MBQC and QEC) to find the right measurement setting.

Type B photons will be measured in the fusion operations, but will not be actively delayed. To this class belong photons $G_1(c), G_2(a), G_2(c), G_3(c), G_4(c), G_5(a), G_5(c)$ and $G_6(c)$. They go through no active elements and therefore they have no loss due to switches.

Type C photons will be measured in the fusion operations, they are multiplexed in order to put them in the right time bin. To this class belong photons $G_1(a), G_1(b), G_3(a), G_3(b), G_4(a), G_4(b), G_6(a)$ and $G_6(b)$. They go through one stage of active switching, the switch has a log tree depth j (which for this scheme we estimate to be $\sim 7 - 9$). Therefore the loss for this type of photon is given by $1 - (1 - \gamma_{sw})^j$.

In figure 6.4 we show the “world lines” of the different types of photons, i.e. the linear optical elements they encounter from the 3-GHZ source to the detectors.

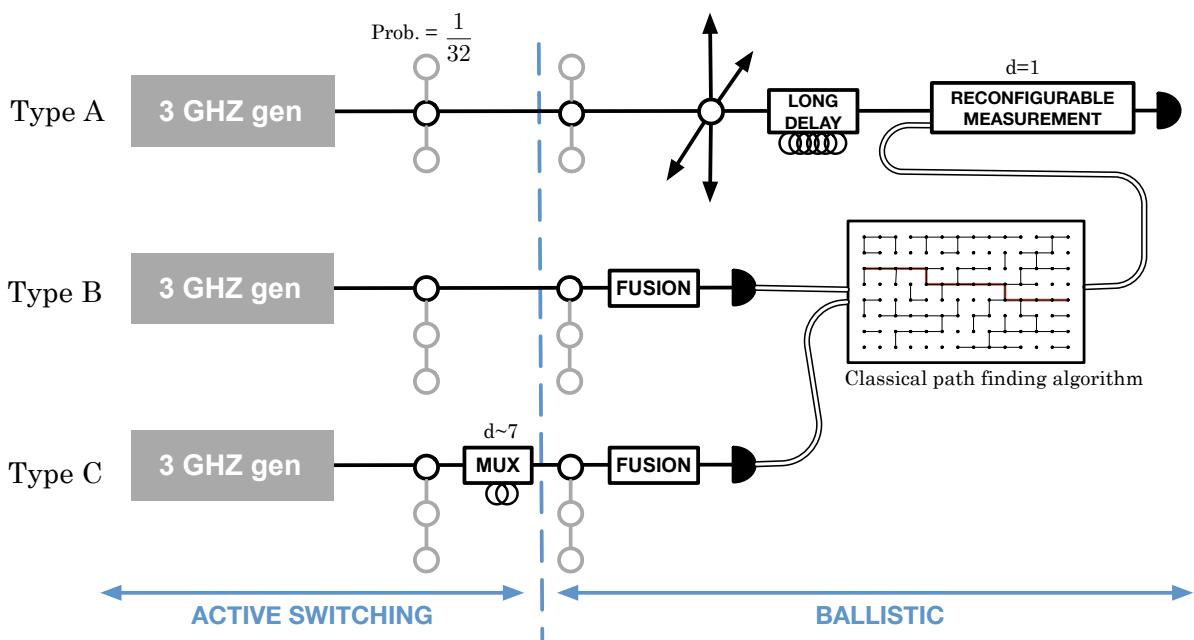


Figure 6.4: World lines of the three types of photons according to this scheme. In this figure we represent a schematic figure of the operations that each type of photon will go through in the architecture and how they relate to the rest of the architecture. We can see that the outcome of the fusion operations is fed as classical information to the path-finding algorithm, which uses the information to decide the basis of the final reconfigurable measurement applied to the data qubits.

6.2.2 Results

The loss tolerance results reported in chapter 5 assumed that all photons involved in a single fusion operation had the same amount of loss. This included both photons being fused, and the ancilla photons used to boost the success probability of fusion. By using the relative time multiplexing scheme presented in this chapter, we can distinguish between photons that go through no switches (or just a reconfigurable measurement), photons that go through a switching network, and ancilla photons. We want to make this distinction as the ancilla photons used to boost fusion will be generated in a different manner and therefore, it is expected that they will have a different associated loss rate. We have previously presented in chapter 4 two boosted fusion mechanisms. One used Bell pairs as ancillas and the other used 4 single photons. To account for loss in this scheme, we apply a simple model: if the number of photons detected is less than the number expected, we count that fusion as failed due to loss. This is the simplest scheme, as it doesn't consider differences in loss tolerance between the two boosted fusion gates or events that can be considered successful despite the loss of a photon², extending our loss model to account for these finer points is ongoing work. When using this model and trying to design a scheme that minimises loss it is more convenient to work with the boosted fusion gate that uses a Bell pair as resource, as the fewer photons that are involved in the fusion gate, the lower the chance of failure due to loss.

In a first instance we will assume that ancilla photons are lossless and compare both the homogeneous and relative time multiplexing to explore the advantages brought in by the latter. The results can be seen in figure 6.5. It can be noticed that for a percolation probability $\geq 90\%$, the relative time MUX scheme can tolerate up to 7% photon loss, while the original homogeneous MUX scheme could only tolerate 2.9%. Note that this result is compatible with the 1.6% tolerable loss rate reported in chapter 5 for this same scheme, as there the loss rate of all photons was considered the same and here we consider lossless ancilla photons.

In order to translate these theoretical loss rates into loss per component, we make the assumption (justified in chapter 2) that loss is only present in the active components of the scheme i.e. switches and reconfigurable measurements. This assumption can be relaxed later on. We assume deterministic on-demand single photon sources and a log tree switching scheme, as detailed in chapter 4.

From the percolation scheme, we can tolerate loss in the individual photons of the 3-GHZ state up to a tolerable loss rate p_l , which is $\sim 2.9\%$ in the case of homogeneous MUX and $\sim 7\%$ in the case of RMUX. From the multiplexing stage we can accept all states that have at least one photon i.e. they are not the vacuum. That means that if the loss rate per switch is γ_{sw} , and the photon goes through m switches, the probability of not having been lost in the switching process is $(1 - \gamma_{sw})^m$. The probability of having been lost at any stage of the switching process is $1 - (1 - \gamma_{sw})^m$ and we want this number to be smaller than the total loss per photon that we can tolerate, $p_l \leq 1 - (1 - \gamma_{sw})^m$. Note that, as in the percolation scheme, the loss is calculated per photon and not per GHZ state; that is the same calculation we do here, we do not require the GHZ state to have a certain loss probability but rather the individual photons. As mentioned above, we only have one stage of multiplexing after the 3-GHZ interferometer.

²There can be enough information from the detected photons to consider that the gate has succeeded or failed.

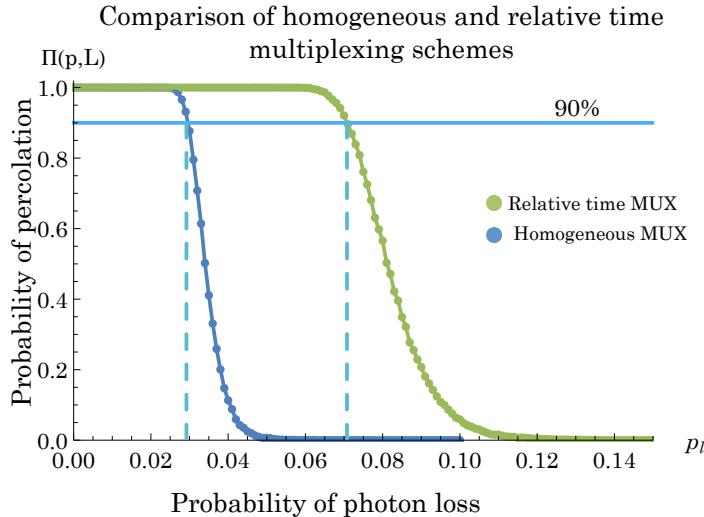


Figure 6.5: Comparison of tolerable loss in the presence of no ancilla loss. The use of the RMUX scheme boosts the tolerable loss to more than twice what could be tolerated with the homogeneous multiplexing scheme.

Single photons (for the purposes of this calculation) are assumed to be obtained on demand from the single photon sources and passed through the 3-GHZ state generator, which is time multiplexed. Each 3-GHZ state generator consumes 6 single photons and produces a 3-GHZ state with probability 1/32. We have chosen this generation procedure from an alternative procedure of a two-step generation, where we first generate Bell pairs from single photons and then generate 3-GHZ from Bell pairs. The reason for this is numerical evidence, provided in appendix G, that the generation of 3-GHZ states directly from single photons is more efficient with deterministic sources. The number of time bins in each clock cycle will depend on the number of multiplexing steps needed to obtain a GHZ state with the required probability.

However, it is not realistic to assume that the ancilla photons used to boost the fusion are lossless. In fact, what we expect is that, unless we have deterministic Bell pair sources (from matter based systems such as NV centres and quantum dots [58]), we will have a combined threshold for percolation that will now depend on two variables, the loss rate of photons that are actively delayed, and the loss rate of ancilla photons. Ancilla photons can be spatially or temporally multiplexed, although spatial multiplexing will reduce the 3-GHZ chip complexity; as we can have on-demand Bell pairs produced in a different chip that act as a deterministic Bell pair source³. We obtained the threshold of photon and ancilla losses by performing percolation simulations, where the probability of a fusion suffering a loss, f_l , is given as a function of the GHZ photon loss p_l and the ancilla photon loss a_l :

$$f_l = 1 - (1 - p_l)(1 - a_l)^2. \quad (6.1)$$

Our ability to perform UQC depends on the percolation properties of the lattice as explained in section 2.7. Mixed percolation thresholds can be obtained numerically to incorporate the

³Note that this simply separates the 3-GHZ and Bell-state generation, and does not reduce the complexity of the overall scheme.

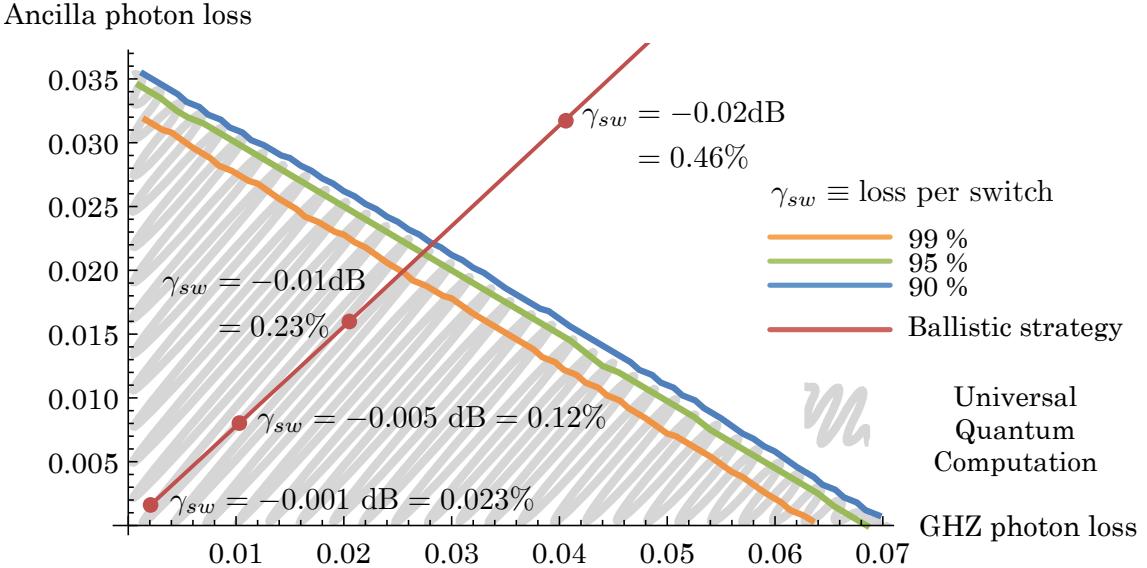


Figure 6.6: Loss threshold trade-off for ancilla and photon loss. The grey shaded area highlights the area of phase space where UQC is possible. We have marked the threshold for different percolation probabilities. We have also indicated the tradeoff of ancilla and GHZ photons loss when they have been generated with a particular strategy and multiplexed using the same switches.

effect of photon loss on the percolation properties of the lattice. In figure 6.6, shaded in grey, we can see the combination of photon loss and ancilla loss that can allow us to perform UQC. We have marked three different thresholds depending on what percolation probability is desired, 90%, 95% or 99%. It might be surprising that, given the non-linear dependence on of the fusion loss rate on p_l and a_l , the threshold is linear. The fusion loss rate f_l is indeed not linear with respect to the individual photon loss rates, however, for the range of photon loss we are dealing with, the leading term in the expansion is the linear term, dependent on $p_l + 2a_l$, and the rest of the terms are negligible, accounting for a maximum of 5% of the total fusion loss rate.

It is important to note that a point in the loss phase space cannot be chosen at will, it depends entirely on how the GHZ states and ancillas have been created. In figure 6.6 we have highlighted the best currently known strategy that assumes perfect photon sources. It uses ballistic generation of the Bell states used as ancillas following the strategy outlined in [92], and ballistic generation the GHZ states using the linear optical circuit outlined in [93]. If we assume these building procedures for the photonic states and the same switches, we achieve the results marked in the graph as “Ballistic strategy”. We have also marked specific points with their corresponding loss rate per switch, γ_{sw} . We can see that UQC would be possible using switches of $\leq -0.01dB$, which translates to a loss rate of $2 \cdot 10^{-3}$ per photon.

It must be noted that the log-tree switch is a conservative approach as there exist new types of switches, such as the recently proposed MEMS switches [167] in which photons only go through one active element overall. Repeating the same analysis to find how much loss per active element we can tolerate (assuming no loss contribution from the passive elements as above), we find that this type of switches gives us an order of magnitude advantage, allowing

us to be in the UQC regime with switches that have $\leq -0.1dB$ loss per switch , or $2 \cdot 10^{-2}$ loss per photon.

6.3 Effective use of resources generated in multiplexing

Multiplexing, although effective, is extremely wasteful. When we calculate the number of repetitions that have to be made in order to achieve an event with probability higher than p , we are conservative and estimate the number of repetitions we need to get *at least* one successful event. However, when we look at the number of successful events *on average* we can see that there is a high waste of photonic states that could have been otherwise used.

6.3.1 Surplus of entangled states

For example, let's look at the generation of a single 3-GHZ state from an array of probabilistic single photon sources. We will assume that this single photon source has an emission rate, η , of 10%. We want to multiplex these sources so that they make an “*almost*” deterministic photon sources, with emission probability $p_1 = 0.99$. This means that we need to have multiplex k_1 times, where k_1 is given by

$$1 - (1 - 0.10)^{k_1} \geq p_1 \Rightarrow k_1 = 44. \quad (6.2)$$

Using a log tree scheme for the switches, we have that the number of switches required for this multiplexed event is

$$m_1 = \lceil \log_2 k_1 \rceil = 6. \quad (6.3)$$

Therefore, the number of time bins per clock cycle are $2^{m_2} = 64$ and the average number of single photons emitted per clock cycle is

$$64 \times 0.10 = 6.4 \text{ photons}. \quad (6.4)$$

From deterministic single photons, we consider the generation of 3-GHZ states using the linear optics circuit (see chapter 4) that succeeds with 1/32 success probability. Again, we require that this near-deterministic 3-GHZ source succeeds with probability $p_2 = 0.99$, so we need to multiplex at least k_2 times, where k_2 is

$$1 - \left(1 - \frac{1}{32}\right)^{k_2} \geq p_2 \Rightarrow k_2 = 146. \quad (6.5)$$

The minimum number of switches in the log tree scheme is

$$m_2 = \lceil \log_2 j \rceil = 8. \quad (6.6)$$

Thus, the number of time bins per clock cycle is $2^{m_2} = 256$ and the average number of 3-GHZ states produced per clock cycle is

$$\frac{1}{32} \cdot 2^{m_2} = 8. \quad (6.7)$$

We require 6 single photons per attempted GHZ generation. We have calculated that we need to repeat the 3-GHZ generation procedure at least 256 times in parallel, therefore we need $256 \times 6 = 1536$ “*deterministic*” single photons. The concatenation of multiplexing schemes means that we need $6 \cdot 2^{m_1} \cdot 2^{m_2} = 98304$ single photon time bins per GHZ clock cycle. These are the necessary time bins to ensure we have a GHZ state with probability higher than $p_1^6 \cdot p_2 = 0.93$. However, on average, many more events will have succeeded. Per GHZ clock cycle we will have 9830.4 single photons produced on average, which are enough to attempt the generations of a 3-GHZ $9830.4/6 = 1638.4$ times. The average number of these attempts that will be successful (if attempted) is $1638.4/32 = 51.2$. So, while attempting to almost deterministically ($p_s = 0.99^7 = 0.93$) produce a single 3-GHZ state, on average we have enough resources to produce 51.

The number of surplus states that is wasted using this strategy depends on the source efficiency and the required probability of success for the 3-GHZ states. We can reproduce these calculations to obtain the number of extra resource states that could have been produced on average when attempting to produce a single 3-GHZ with a certain probability of success p_s . Figure 6.7 shows the results of this simulation, where we present the most economical strategy: the strategy is optimised to waste the least amount of resource states per “*deterministic*” 3-GHZ.

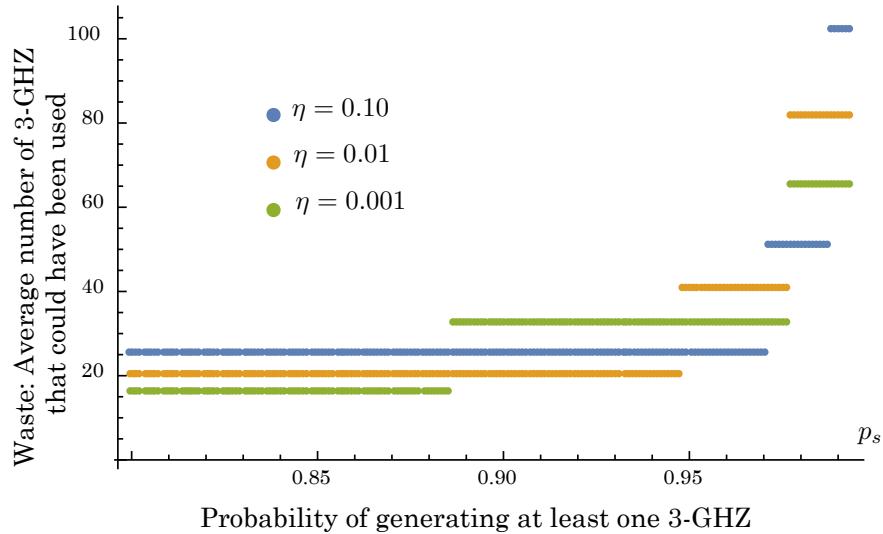


Figure 6.7: Number of 3-GHZ states that could have been produced on average when using a multiplexing scheme that has a probability p_s of at least generating one 3-GHZ. This results are calculated for three different source efficiencies, indicated by the three colours of the graph.

It is interesting to note that for the same probability of generating at least one 3-GHZ state, in some cases a source with 10% efficiency would generate a higher number of unused resources than a source with 1% efficiency but less than the source with 0.1% efficiency (for example in the case $p_s = 0.90$). This is counterintuitive because we would expect a trend, however, the average number of 3-GHZ that could have been produced is the multiplication of the source efficiency (to the power of the number of photons used, i.e. six) times the number of time

bins due to the multiplexing. For a source with very low efficiency, the number of time bins increases dramatically. For example, for the case detailed above ($p_s = 0.93$), we have that a $\eta = 0.1$ efficiency source would have a time bin count of $9.8 \cdot 10^4$, an $\eta = 0.01$ source would have $7.9 \cdot 10^5$ and the $\eta = 0.001$ source would have $1.3 \cdot 10^7$ time bins, which shows an increase of two orders of magnitude of the number of time bins with respect to the $\eta = 0.01$ source, but their efficiency differs only by one order of magnitude.

6.3.2 Impact of an efficient use of the generated resources

We have detailed above a very conservative and wasteful strategy to produce GHZ states, as on average we are creating far more resources than we are using. One key realisation is that, if we could use the resources to their full potential, we could be creating not only the states that we need but also back ups that could boost the percolation probability of the lattice. Instead of having one GHZ per clock cycle, we would have several, and therefore the strategy for creating the micro-clusters could also be repeatedly performed in parallel. Having enough perfectly formed micro-clusters in parallel, they could be considered deterministic (as it will be unlikely that none out of many will succeed), and therefore so could the nodes of the lattice. In this scenario, the percolation properties of the lattice will therefore only depend on our ability to create bonds between the micro-clusters: we have moved to a pure bond percolation rather than a mixed site-bond percolation.

The probability of obtaining a perfectly formed micro-cluster from three 3-GHZ states is $0.75^2 = 0.5625$, we can calculate how many attempts are needed to produce a micro-cluster with a probability high enough to be considered *deterministic*. Figure 6.8 shows the number of attempts needed to obtain a perfectly formed micro-cluster (from deterministic 3-GHZ states) with a certain probability. We also show for comparison, the average number of perfect micro-clusters obtained if all resources from multiplexing are used and $p_s = 0.95$. As we can see in the figure, if all resources are used, the micro-clusters can be considered deterministic.

Looking at the percolation thresholds for the diamond lattice [168, 169], we find that the bond percolation threshold is 0.39, which means that the lattice model percolates if the bond occupancy is > 0.39 . Thus, by using wisely the extra resources generated, we have shown that we can boost the percolation probability in such a way that the non-boosted version of the fusion gates is enough to achieve the supercritical regime. However using boosted fusion would allow us to have an almost deterministic lattice, which would be a great advantage when considering fault tolerant schemes, more details on this can be found in chapter 7.

To efficiently use all the resources generated in the multiplexing process, we need a very well controlled network that allows us to group successful events as needed. But first, a deeper theoretical understanding of the capabilities of RMUX and the limitations of the necessary event re-routing is required. Relative multiplexing allows for *half* of all switching networks to be eliminated, while re-routing requires clever algorithms (such as minimum-cost perfect matching algorithms on bipartite graphs [170]) to group events appropriately (see figure 6.1). More theoretical work studying the implementation of both these ideas is necessary to comprehend the true capabilities of the scheme.

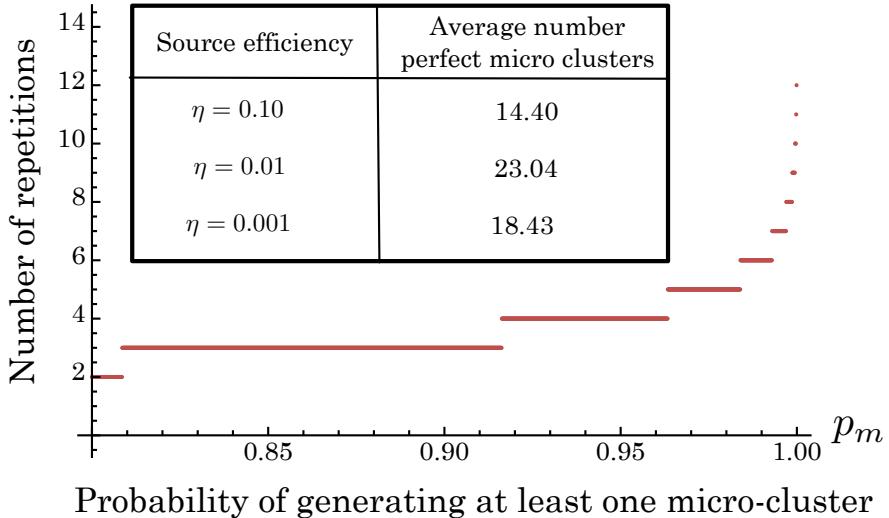


Figure 6.8: In colour red, we have plotted the number of attempts needed to obtain a “deterministic” micro-cluster (with probability higher than p_m). We put as comparison, the number of micro-clusters generated on average by the multiplexing scheme. As examples we have chosen the generation of a 3-GHZ state with probability $p_s = 0.95$ for the three source efficiencies. We can clearly see that if all extra resources generated were used, we could consider micro-clusters to be generated deterministically.

6.4 Discussion and outlook

In this chapter we have presented two main concepts. The first is relative multiplexing, which expands on the insight that in order to achieve determinism in our scheme, we don’t need to have all events synchronised to an overall clock cycle. Secondly we provide evidence of how wasteful an absolute-time multiplexing scheme can be.

There are very immediate extensions to the calculations shown in this chapter. We could include the effect of loss of passive elements and delays, consider the differences between boosted fusion gates and extend the loss model to include contributions from passive elements. But the main conclusion of these results is clear: applying the relative multiplexing idea at every level of the percolation scheme, and using every generated resource can significantly improve the performance of the linear optical quantum computer. We have seen the impact of relative multiplexing when applied at only one level by reducing the loss tolerance requirements of the active components by an order of magnitude. When used extensively throughout the scheme, we expect that it will have a dramatic impact in the resource consumption and efficiency, which in turn will allow us to lift restrictions at the component level. There are still challenges ahead, in particular, to harness the power of relative multiplexing we need to develop a better theoretical understanding of the algorithms required for matching photons and their efficiency bounds.

CHAPTER 7

TOWARDS FAULT-TOLERANCE

*We cannot clone, perforce; instead, we split
Coherence to protect it from that wrong
That would destroy our valued quantum bit
And make our computation take too long.*

Quantum Error Correction Sonnet

DANIEL GOTTESMAN

7.1 Introduction

The crucial requirement for a quantum computer is that the internal computation needs to be done in a “closed system”. Unlike classical digital computers, quantum computers are very sensitive to noise and even the smallest amount of information leakage can cause irreversible damage to the calculation. Therefore, the quantum system needs to be isolated from the rest of the universe so that decoherence does not affect it. However, this is a difficult equilibrium to achieve, as if the system is completely closed to the environment, decoherence would not affect it, but we will not have access to the computation either. No system is truly free of decoherence, but various quantum error correction techniques can help remove some decoherence from a system.

Photonic systems are particularly resilient to the decoherence noise that affects other physical systems [18]. However, in the process of implementing a quantum computation, the control operations performed on the photons will have sometimes unwanted effects that can be characterised as *noise*. For example, inaccurately aligned linear optical elements, partial distinguishability, timing problems or sub-optimal experimental apparatus can all contribute to an intrinsic error rate in photonic qubits, not to mention the far more pressing problem of photon loss. Therefore, any LOQC architecture will need to implement a quantum error-correcting protocol in order to successfully implement any quantum computing protocol. The implementation of quantum error correction in linear optical systems is particularly challenging due to the lack of deterministic gates, which are the key ingredient in standard approaches to fault tolerance. Therefore, an approach different from the usual must be sought after.

In this chapter we present a brief overview of quantum error correction and topological codes. We also give a summary of the first LOQC proposal [62] that takes into account the

implementation of a fault-tolerant code in the optical lattice. This proposal is based on a repeat-until-success strategy and it is therefore not suitable for the QNIX architecture, but it becomes a benchmark for any future LOQC implementation of fault-tolerant codes. In section 7.4 we propose three different ways of implementing error-correction in the LOQC protocol presented in chapter 5: renormalisation, concentration and percolated topological lattices. This last approach is the most promising, however the error model differs from what is usually studied in the literature and we must therefore investigate it in more detail. Preliminary results for this error model are presented in section 7.5. These results have been obtained in collaboration with James Auger (who has performed the numerical simulations in this section), Hussain Anwar, Tom Stace, Dan Browne and Terry Rudolph. My contribution has been in determining the mapping of bond loss errors in the cluster state picture to errors in the surface code and building the simulator presented in chapter 3 which has been used to determine the bond loss error mapping and to perform preliminary calculations for the three-dimensional Raussendorf lattice.

7.2 Quantum Error Correction and topological codes

Quantum information is very sensitive to noise, and in order to perform quantum protocols that compete with classical computation we need to be able to effectively remove the effect of the environment from the qubits used for the computation. In classical error-correction, redundancy is used to preserve information. However, the *no-cloning* theorem [171] forbids the copying of unknown quantum states and therefore more intricate solutions must be devised. Shor [76] proposed the first quantum error-correcting code that protected against all possible errors on a single qubit, by encoding it in 9 physical qubits. The same principle is used in other quantum error-correcting codes to store a small number of logical qubits in a large number of physical qubits. In this section we will briefly review the basic concepts of quantum error correction and will highlight a pair of codes that will be used later on in the chapter.

7.2.1 Foundations of the theory of quantum error correction

The theory of *quantum error correction* (QEC) studies how quantum information can be processed reliably in the presence of noise. QEC codes are used to encode quantum states in a way that is resilient against the effect of noise, and then decode this information when we wish to recover the original state. Quantum states are encoded into a QEC code by performing unitary operations. The effect of noise on the encoded state can be assessed by performing a series of syndrome measurements to diagnose the type of error that has occurred. Once the type of error has been determined, a series of unitary operations are performed on the physical qubits, which recover the original encoded state. For an error to be correctable, it needs to fulfil a simple set of operations, known as the quantum error-correcting conditions [70].

Theorem 4. (*Quantum error-correcting conditions*) *Let C be a quantum code, and let P be the projector onto C that leaves the encoded state unchanged. Suppose \mathcal{E} is a quantum operation with operation elements $\{E_i\}$. A necessary and sufficient condition for the existence*

of an error-correction operation \mathcal{R} correcting \mathcal{E} on C is that

$$PE_i^\dagger E_j P = \alpha_{ij} P, \quad (7.1)$$

for some Hermitian matrix α of complex numbers.

Proof. See [70] for proof.

The elements E_i for the noise \mathcal{E} are called *errors* and if such \mathcal{R} exists, they are considered a *correctable set of errors*. It is usually the case that the exact type of noise affecting the system is not known, and it is therefore useful to protect against an entire class of noise. The quantum error-correcting conditions can be adapted to this scenario [70]:

Theorem 5. Suppose C is a quantum code and \mathcal{R} is the recovery error-correcting operation for a noise process \mathcal{E} with operation elements $\{E_i\}$. Suppose \mathcal{F} is a quantum operation with operation elements $\{F_i\}$ which are linear combinations of the E_i , i.e. $F_j = \sum_i m_{ji} E_i$ for some matrix m_{ji} of complex numbers. Then the error-correction operation \mathcal{R} also corrects the effects of the noise process \mathcal{F} in the code C .

Proof. See [70] for proof.

This theorem allows us to restrict the errors our quantum code needs to correct to a discrete set, as any other error processes that can be described as combinations of this discrete set can also be corrected for using the same recovery operation. For example, for single qubit errors, as any operation elements that describe a single qubit error $\{E_i\}$ can be described as a linear combination of the Pauli matrices, it suffices to use an error-correcting code that protects against Pauli error in order to protect against *arbitrary* single qubit errors.

The pressing question is then to determine how much noise, if any, a QEC code can protect against and how many resources, i.e. physical qubits and correction operations will have to be used in order to protect against such noise. In 1997, Aharonov and Ben-Or proved [53] that noise is no fundamental limit for the performance of large scale quantum computers:

Theorem 6. (Threshold theorem) Provided the noise in individual quantum gates is below a certain threshold it is possible to efficiently perform an arbitrarily large quantum computation with polylogarithmic cost in resources.

Proof. See [53] for proof.

7.2.2 Error model

The sources of error in a quantum system are varied, the most noteworthy are [172]:

- Coherent, systematic *control* errors associated with an incorrect knowledge of the dynamics of the system. This type of errors occur when the apparatus used to implement the quantum logic operations has not been characterised correctly and is implementing a different quantum operation than it was intended. This error is equivalent to the systematic application of an undesired unitary gate operation.

- *Environmental decoherence.* The environment can be modelled as another quantum system which is coupled to the quantum system in which the computation is taking place. This interaction with the environment (which is usually considered stronger the longer the computation) cannot be included in the computation as we don't have access to the degrees of freedom of the environment, and hence can lead to incoherent errors.
- *Qubit initialisation.* This type of error can be modelled as either a coherent or incoherent type of error depending on the physical system and the preparation procedure.
- *Measurement* errors are incoherent errors. An error on the measurement can be modelled as unitary operation acting on the qubit prior to the measurement: $\rho \rightarrow (1-p)\rho + pX\rho X$ where p is the probability of a measurement error, followed by a perfect measurement on the computational basis. This leads to effective measurement operators given by $M_0 = \sqrt{1-p}|0\rangle\langle 0| + p|1\rangle\langle 1|$, $M_1 = \sqrt{1-p}|1\rangle\langle 1| + p|0\rangle\langle 0|$ which implies that after the measurement the qubit will not be in a known state.
- *Qubit leakage*, in which the state of the qubit leaks to states outside the computational space. This is usually the case as most systems utilised for qubits have more than two levels. Recovery from leakage is possible in some cases, when it's not, it is usually considered as qubit *loss*, which is modelled by tracing out the qubit from the state. This means that the qubit cannot be directly measured or coupled to any other qubit. Qubit loss can also happen, as in the case of linear optics, because of absorption of the physical system by the environment, rather than the leakage of information to other levels. This error usually requires additional techniques on top of the standard QEC protocols.

The modelling of error in a quantum system depends on the physical implementation itself, however theorem 5 ensures that as long as we choose a discrete set of errors that span all possible errors, our error-correcting procedure will succeed. Even in the case of incoherent errors, they can be modelled as discrete coherent errors that occur with a certain probability. There are several assumptions that are made about the nature of noise that a QEC code protects against. In order to prove that a particular physical implementation can perform robust quantum computation, these assumptions must be satisfied. Although in some estimates of the computational threshold extra assumptions are made [54], we list here the most commonly used ones:

- *Constant error rate:* The error rate is independent of the size of the QEC code.
- *Weakly correlated errors:* Errors must not be too strongly correlated, either in space or in time.
- *Parallel operation:* We assume the ability to perform many gates in a unit of time, as it is needed to perform the recovery operations.
- *Reusable memory:* Ancilla qubits are used to store the syndrome measurement and must be replaced (or the information erased) quickly.

The *depolarising* channel is an important type of quantum noise, in which a qubit is replaced by the complete mixed state ($I/2$) with probability p :

$$\rho \rightarrow (1-p)\rho + p\frac{I}{2}. \quad (7.2)$$

This type of noise can be used to model most of the incoherent errors that might occur during a quantum computation. A common way of parametrising it is

$$\rho \rightarrow (1-p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z), \quad (7.3)$$

where the state remains unchanged with probability $(1-p)$ and the operators X, Y and Z are applied with probability $p/3$. The depolarising noise model, where each of the Pauli errors occurs independently is the most commonly used error model used in QEC protocols. It can now be fully understood why the stabilizer formalism is such a great tool for QEC: in order to model most errors, we only need to model Pauli errors, which can be efficiently simulated on a classical computer using the stabilizer formalism.

7.2.3 Topological codes

In chapter 3 we introduced the stabilizer formalism and we briefly showed how it could be used for error correction. Its effectiveness lies in the fact that it can easily protect against Pauli errors, and as many errors can be written as linear combinations of Pauli errors, stabilizers is all that is needed to protect against those. Stabilizer codes are usually described as a trio of numbers: $[n, k, d]$, where

- n is the number of *physical* qubits of the code,
- k is the number of *encoded* qubits in the code,
- d is the *distance* of the code, defined as the minimum distance between encoded logical states, i.e. minimum number of physical qubits on which a local error has to occur in order to become a logical error.

The error detection is performed by measuring the $n - k$ stabilizer operators that define the code.

A large class of stabilizer codes can be designed by the so-called Calderbank-Shor-Steane (CSS) code construction [127, 128]. In the CSS code construction two classical linear codes (satisfying certain conditions, omitted here) are used to built a quantum code that corrects against X and Z errors, independently. Rounds of error detection are performed sequentially (possibly with an intermediate step of applying H and $CNOT$ gates). The main characteristic of these codes is that the stabilizer operators used to detect the errors are all products of either X operators or Z operators on individual qubits. A full review of this type of codes can be found in [173].

Topological codes are a particular type of CSS code, where the quantum information is topologically protected in a global degree of freedom, while the stabilizer operators defining the code are strictly local. These codes are defined on lattices with particular topological

properties, hence their name. The syndrome measurements find *defects* in the lattice, which highlight the presence of errors. The process of error correction and recovery can be seen as topological operations on a surface, and topologically protected quantum computation can be performed by considering the defects as *quasi-particles* that implement logical gates on the encoded quantum information when braided [129]. This, however, is far beyond the scope of this thesis, a detailed review can be found in [174]. We will explain two topological codes in more detail: the two-dimensional planar code and the three-dimensional Raussendorf lattice. We will study the implementation of these two codes on a linear-optical lattice built according to the procedure proposed in chapter 5.

Two-dimensions: planar code

The planar code is a type of surface code. Surface codes are a particular class of CSS codes defined on a two-dimensional lattice. In particular, the planar code is defined on a $L \times L$ lattice, where each *edge* of the lattice corresponds to a qubit (notice the difference with the cluster states, where qubits correspond to the lattice nodes). The stabilizer operators that describe the planar code are check operators that contain either only X operators or only Z operators and they are usually referred to as *star* and *plaquette* operators respectively. There is one star operator associated to each vertex of the lattice, which is a tensor product of X in all the edges that meet at the vertex; and one plaquette operator associated to every “tile” of the code, which is a tensor product of Z in all edges that surround the tile¹. In figure 7.1 we can see an example of these check operators. All the check operators commute with each other: operators of the same kind commute trivially, while stars and plaquette operators commute as they always overlap on either zero or two qubits. In this planar topology, all check operators can be constructed with local gates. There exists a difference between check operators in the body of the planar code and those defined on the boundaries, as check operators on the boundaries have support on only 3 qubits rather than 4 (their weight has diminished). It is also important to note that in our definition of the planar code, the lattice will have two types of boundaries (this definition ensures the duality of the primal and dual lattice): a “plaquette boundary” or “rough boundary” on which the plaquette operators have weight 3, while on the “star boundary” or “smooth boundary”, it is the star check operators that have reduced weight.

According to theorem 1 (chapter 3), the number of encoded logical qubits supported by the lattice is given by $k = n - s$ where n is the number of qubits and s is the number of independent check (stabilizer) operators. The planar code of dimension L is obtained from a square lattice that has $L^2 - (L - 1)^2$ edges, where L is the number of edges between rough boundary and rough boundary (or alternatively from smooth boundary to smooth boundary) when following the shortest path between them. There are $(L - 1)^2$ star and $(L - 1)^2$ plaquette operators and they are all independent. Therefore the number of logical qubits encoded in the lattice is $k = L^2 + (L - 1)^2 - 2(L - 1)^2 = 1$.

The logical operators that act on this logical qubit are stabilizer operators that commute with the check operators but are not generated by the check operators. For the planar code, they

¹An alternative view of the dichotomy of the check operators is to define them on the dual lattice, which has a site per each bond of the lattice and vice versa. This however is outside the scope of this thesis, a detailed explanation can be found in [54].

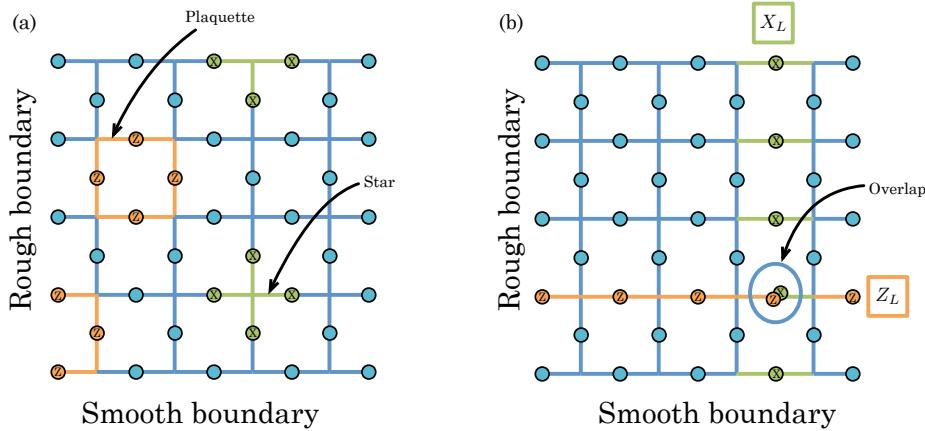


Figure 7.1: Planar code. (a) Check operators that stabilize the code. Plaquette operators are Z operators on the qubits surrounding a tile of the lattice, while star operators are X operators on the qubits next to a vertex of the lattice. Note that on the rough boundary, plaquette operators reduce weight to a 3-body operation, while the same reduction occurs to the star operators on the smooth boundary. (b) Logical operators of the encoded qubit. The logical Z_L operator is defined as a chain of Z operators from rough to rough boundary. The logical X_L is defined as a chain of X operators from smooth to smooth boundary. Note that they overlap on one qubit, which ensures anti-commutation.

are defined as chain operators that cross the lattice. The logical X_L operator acts on the qubits whose edge is parallel to a chain that extends from smooth to smooth edge², while the logical Z_L operator acts on a chain of qubits that extends from rough to rough edge. The two logical operators overlap on one qubit, which means that they satisfy the appropriate commutation relation. An example of such logical operators can be found in figure 7.1 (b).

Because the information is topologically protected in a global degree of freedom, sparse local errors on the qubits don't corrupt the logical qubit. Single qubit errors are detected by the check operators, and many chains of errors will be too. However, chains of errors with a particular topology will not be detected by the check operators and might cause logical errors. In figure 7.2 (a) we show two examples of chains of errors that are not detected by the check operators, the reason being that any check operator overlaps with two errors and therefore commutes with the chain. There are two types of these undetectable error chains (usually called cycles): a trivial cycle, which can be expressed as the product of check operators and therefore is also a stabilizer of the computational subspace; and a non-trivial cycle, which cannot be expressed as a product of check operators and although it commutes with all the check operators, it anti-commutes with at least one logical operator and hence becomes a logical error.

The observed value of the measurement of check operators is called *syndrome*. When there are no errors acting in the planar code, the code subspace corresponds to the +1 eigenvalue of all check operators. Thus, errors can be detected when the measurement values turns negative (such negative measurement outcomes are usually referred to as defects). Plaquette operators will detect X errors and star operators will detect Z errors. As there are always two check operators of each type overlapping in any one qubit, a single error will trigger a negative

²Or equivalently a chain of qubits in the dual lattice.

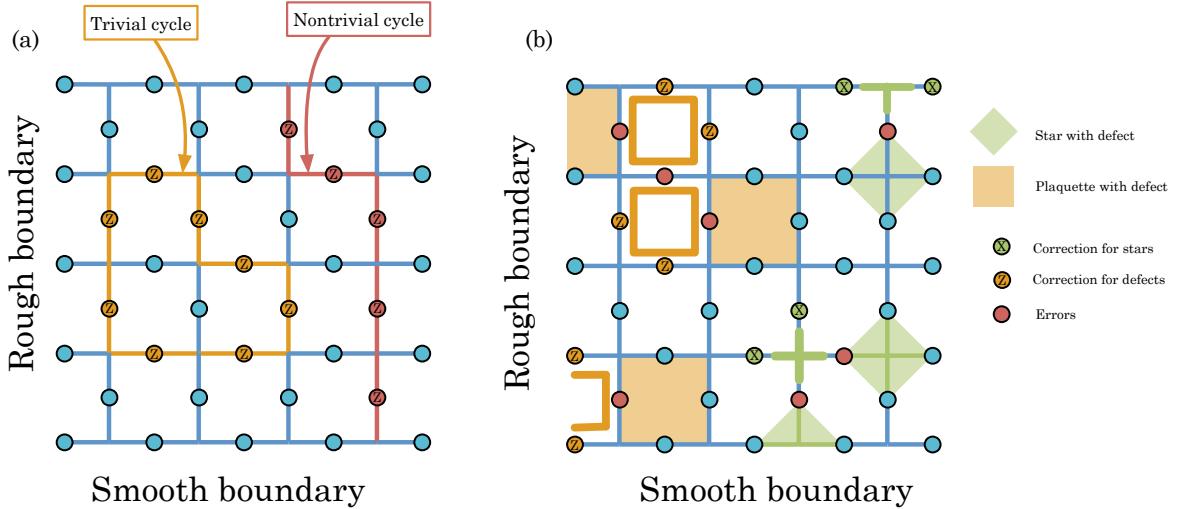


Figure 7.2: Planar code. (a) Cycles of Z operators. The cycle marked in orange is a trivial cycle as it is closed, it is not detected by the stabilizer operators and has a trivial action on the X_L operator. The cycle marked in red is a non-trivial cycle as it is open. It is not detected by the stabilizers and flips the value of the X_L operator. It is therefore a logical operation on the code subspace. Similar cycles can be defined for the star operators. (b) Errors on the red qubits are detected by the stabilizer operations. The plaquette and star check operators that detect the error are highlighted in green and orange. An error detected by a plaquette is an X error and an error detected by a star operator is a Z error. To correct for these defects, single qubit operations (marked in green and orange) have to be applied in order to turn the chains of errors into trivial cycles.

outcome on two check operators. If there is only one error, the adjacent check operators will give a negative measurement outcome, however if there are chains of errors, the defects will be at the end of the chain of errors. It is important to note that it is possible also to have errors highlighted by one single check operator, as the error chains can extend to the boundary of the lattice. In particular, X error chains can end at smooth boundaries and Z error chains can end at rough boundaries. This means that defects can appear either in pairs or singly, and single defects should be paired to the corresponding boundary.

The error recovery consists in turning the detected chains of errors into trivial cycles so that they have no effect on the topologically protected qubit. In figure 7.2 (b) we can see an example of some errors (marked in red) as they are detected by the syndrome measurement of the plaquette and star operators, as well as successful correction that turns the errors in trivial cycles. To determine what is a successful correction, algorithms such as the Minimum-Weight Perfect Matching algorithm [175] are used to pair the defects appropriately. It is usually assumed that pairs of defects (or defects and boundaries) that are close together have a higher probability of having been generated by the same error. However, errors in the recovery or errors in the syndrome measurements can lead to the transformation of these chains of errors into non-trivial cycles, which are logical errors in the code.

The planar code has been shown to have high thresholds for both Pauli errors (11%) and loss (50%) [176], making it an excellent choice for error correction in LOQC. However the effect of bond loss in this code has not been studied, and we do so in section 7.5.

Three dimensions: Raussendorf lattice

The main idea behind the Raussendorf lattice [55] is that “*a three-dimensional cluster state is a fault-tolerant fabric*” that allows a cluster-state computation to be performed fault-tolerantly. The qubits that form this three-dimensional lattice are arranged in the faces and edges of a body-centred-cubic lattice, as shown in figure 7.3. Topologically protected quantum computation can be performed by defining three separate cluster regions: the vacuum, where local X measurements are performed; primal defect and dual defect regions, where local Z measurements are performed; and qubit regions, where qubits are measured on a basis in the $\{X, Y\}$ plane. These regions are then braided to perform logical protected gates. It is beyond the scope of this thesis to describe the details of this procedure, they can be found in [55].

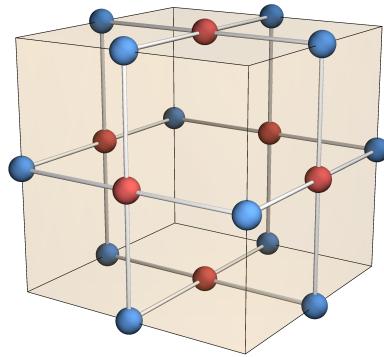


Figure 7.3: Unit cell of the Raussendorf lattice.

Each face of the unit cell of Raussendorf (figure 7.3) has a stabilizer element associated with it, with an X operator acting on the face qubit (red) and Z operators acting on the qubits on edges that form the face (blue). The stabilizer that measures the syndrome bit per cell of the Raussendorf lattice corresponds to a six-body stabilizer with X operators in all the face qubits, which protects the face qubits from Z errors. However, we don’t need to perform a six-body measurement, we only need to make single X measurements on the face qubits and from them infer the value of the six-body cell operator. The Raussendorf lattice is translationally invariant if we move along the diagonal of the unit cell. This translation defines a new *dual* lattice and each face of the primal lattice becomes an edge in the dual (and vice versa) and each site becomes a cube or unit cell (and vice versa). The edge qubits are then protected by measuring the stabilizer on the dual lattice.

The Raussendorf lattice can also be understood as a surface code evolving in time. In the surface code, we performed rounds of measurements of check operators, alternating between plaquettes and stars. This is equivalent to measuring the syndrome measurements of the Raussendorf lattice layer by layer. Errors in the Raussendorf lattice correspond to non-trivial cycles, such as compact regions of error that span a dimension of the lattice. The Raussendorf lattice has been shown [177] to have lower thresholds than the surface code for both Pauli errors (0.6%) and loss (25%). However, it is particularly amenable for LOQC as it doesn’t require many-body measurements to measure the check operators.

7.3 First linear optical proposal of a fault-tolerant quantum computer

A recent proposal by Li *et al.* [62] puts forward the first fault-tolerant protocol for LOQC that includes every step from the initial generation of entanglement to the implementation of a fully fault-tolerant quantum computation. They consider the errors and losses that happen at each stage on the process and determine the resource requirements and per-component tolerable loss rate.

Li *et al.* propose a specific protocol for LOQC based on the planar code, which they simulate in time by creating a 3D lattice (each layer represents one clock cycle of the computation). Each qubit in the surface code is encoded in a snowflake graph state [178, 179], which is their building block for the 3D lattice and contains sufficient redundant encoding to create the planar code correlations despite the probabilistic nature of the BSM used. Each building block has a core qubit which will ultimately be part of the fault-tolerant code and several bridge units which are used to create entanglement between the core qubits via probabilistic entangling operations (PEO). The resources necessary to generate these complex multi-photon states are taken into account and errors are assumed to happen at each individual component (with the exception of single photon sources, which are assumed deterministic). It is also interesting to note that their simulations confirm that the switching networks have the strongest impact on the loss tolerance of the quantum computer, as we have highlighted in previous chapters.

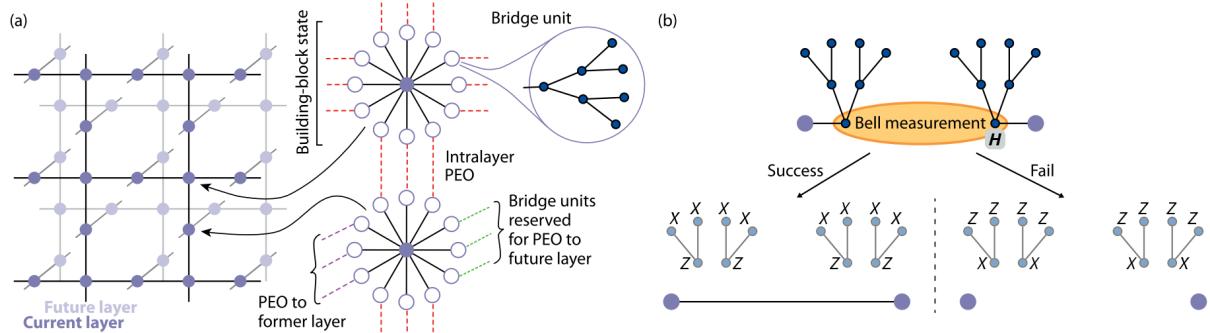


Figure 7.4: Protocol for fault-tolerant LOQC. (a) 3D fault tolerant cluster built in a near-deterministic manner once sufficiently complex building blocks have been created. Probabilistic entangling operations are performed between several bridge units to create entanglement between the core (or data) qubits. (b) Description of the probabilistic entangling operation. Figure from [62], copyright (2015) by the APS.

The authors give an estimate of the overall scale of resources, choosing the number of detectors³ as a metric for the device size, assuming the number of other elements will scale accordingly. The estimate that they require “upwards of $10^5 - 10^6$ ” detectors per physically encoded qubit. Assuming that each logical qubit is encoded in a surface code consisting of ~ 1000 qubits [14], the total number of detectors for a 10^3 fault-tolerant quantum computer is $10^{11} - 10^{12}$. They also calculate that loss rates below $\sim 10^{-3}$ and error rates below 10^{-5} are

³As the authors do not consider temporal multiplexing, the number of detectors is twice the number of single photons used in the scheme.

required.

As the authors of this proposal point out, their result “only represents an upper bound on the physical characteristics that are required of the components in an LOQC system” and further optimisations and improvements will reduce the resource costs. Given the material presented in this thesis, there are two ideas that have not been implemented in this proposal, but that have been proven to successfully reduce the number of resource costs in other protocols: time multiplexing and percolation.

The authors state that this protocol requires a number of physical components that is at least 5 orders of magnitude greater than in comparable matter-based systems. Architectural resource savings such as the ones mentioned in this thesis will reduce the difference in number of physical components required with respect to matter systems. However, it is important to bear in mind that a fault-tolerant quantum computer is expected to require at least $\sim 10^6$ data qubits in order to compete with current state-of-the-art classical computers [14]. Thus, *all* physical implementations of a fault-tolerant quantum computer will need to demonstrate the feasibility of their schemes for 10^9 qubits, counting the physical qubits needed to encode the data qubits. On-chip silicon wafers of millimetre size with hundreds of thousands of working devices have been demonstrated [61, 180] in classical linear optical experiments and therefore it is realistically possible that architectural and technological improvements will make LOQC a competitive candidate for a quantum computer.

The overall strategy used to build the topological lattice in Li *et al.*’s proposal is based on a repeat-until success strategy. As discussed in chapters 1 and 2, not only is this strategy very expensive in terms of resources, but the fact that it does not have a fixed physical depth makes it very unfeasible experimentally. Furthermore, the large amounts of switching required in a repeat-until-success scheme aggravates other important issues such as noise and photon loss. It is therefore necessary to investigate alternatives to this first approach. In the following sections we will focus on implementing QEC on a ballistic architecture which minimises switching and has a fixed physical depth.

7.4 Transforming a percolated lattice into a topological code

The main error in a linear optical system is loss. Specific loss-tolerant codes already exist [113, 181] and have been demonstrated [152], however as topological codes have been shown to have high tolerance to loss errors in addition to Pauli-error tolerance, the possibility of using those codes directly to protect against all possible errors is compelling. It is worth noting that photonic implementations of topological codes have already been proposed for schemes relying on deterministic quantum dot sources [182], although these sources are not yet within our technological capability.

We have so far presented well-known fault-tolerant schemes as well as the first LOQC proposal that considers the implementation of QEC. In this section, we will focus on the implementation of QEC in the LOQC architecture proposed in chapter 5. We propose three possible ways of doing so: renormalising the cluster state into blocks that can become renormalised qubits in the fault-tolerant lattice, concentrating the percolated lattice into the topological lat-

tice and building the topological lattice directly using the percolated scheme. Out of these three approaches, the last one seems the most promising, although extensive analysis needs to be performed to understand the effect of bond losses on the fault-tolerant properties of the code, which has not been studied before.

7.4.1 Renormalisation

To successfully implement UQC in a photonic lattice we need to implement error correction. The simplest way to do so, is to embed a fault-tolerant lattice in the photonic lattice, using the block renormalisation technique mentioned in chapter 5. It consists in picking regular blocks of the percolated lattice and redefining them as one qubit of the topological lattice. Inside each renormalised block, we find a crossing cluster that has the right connections to the neighbouring renormalised qubits, as can be seen in figure 7.5. We believe the Raussendorf lattice [55] is the best fault-tolerant lattice for this task, as it has very high thresholds for both logical errors and, most significantly for LOQC, loss errors [177]. This lattice also has the huge advantage of requiring only local measurements and not CNOT gates to implement error correction⁴.

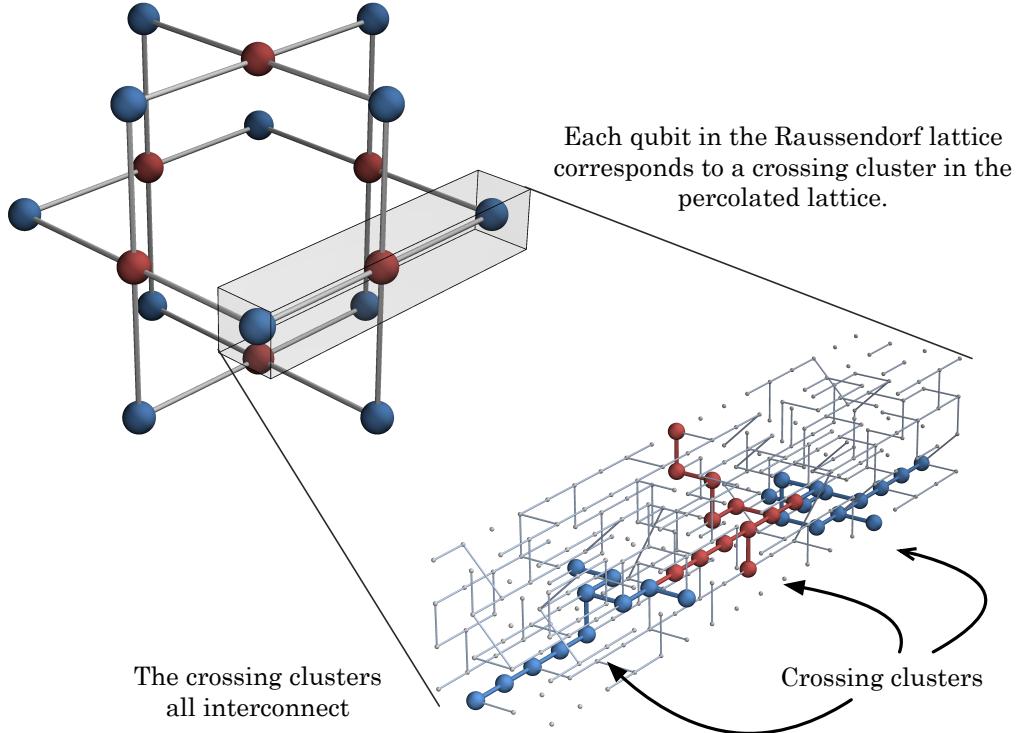


Figure 7.5: Each (renormalised) qubit of the Raussendorf lattice corresponds to a block of the percolated lattice, where a crossing cluster with the necessary connections to adjacent blocks can be found. The crossing clusters in each block of the percolated lattice interconnect, creating a spanning cluster across the lattice which is topologically equivalent to the Raussendorf lattice.

⁴Other implementations of QEC require the ability to perform CNOT gates between the data qubits and some ancilla qubits in order to measure the syndrome. The Raussendorf lattice only requires local measurements to measure the syndrome and it is therefore much more amenable for a system with non-deterministic two-qubit gates such as LOQC.

It must be noted that each block is not created separately, the photonic lattice is created as a whole and it is during the classical post-processing where different photons will belong to different “renormalised” qubits, which will be defined by the blocks. The dimensions of these blocks will depend on the percolation probability that is required, and also the total photon loss present in the fusion, f_l . In the case of the diamond lattice, as it is mentioned in chapter 5, the percolation properties of the brickwork lattice are not isotropic, having one preferred percolation direction. However, if these blocks are to be used to reproduce a structure such as the one of the Raussendorf lattice, isotropic percolation probabilities are required and therefore rectangular cuboid blocks will be optimal renormalised qubits.

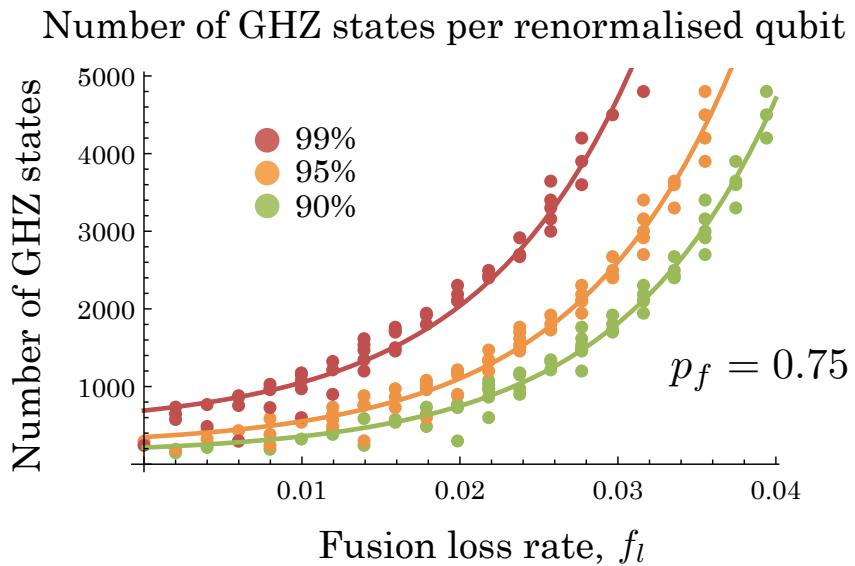


Figure 7.6: Number of GHZ states needed per renormalised block of the Raussendorf lattice as a function of the total loss per fusion for different percolation probability requirements. Note that the simulation results have been obtained using a fusion probability of 75%.

In order to assess how many 3-GHZ states need to be created per singular qubit (renormalised block) we run Monte-Carlo simulations to obtain the percolation threshold for different rectangular cuboids with different aspect ratios as a function of f_l . In figure 7.6 we can see what is the trade-off, where the different points in the graph represent simulations of cuboid renormalised blocks with different aspect ratios. The solid lines mark the exponential growth of the size of the renormalised block as the fusion loss probability increases.

As we can see from figure 7.6, the scaling of this approach is very expensive. The simulation results have been obtained in the case where every renormalised block has the same dimensions. If this restriction is lifted and the block size is variable (determined only by the existence of crossing clusters within a specific region) it is expected that the resource requirements would be lower. However, even with variable block sizes, this strategy would still be making very poor use of the cluster correlations already created, hence other strategies must be sought after. The renormalisation strategy serves as a benchmark for any other strategy.

7.4.2 Concentrating a universal lattice

A different way of using the percolated cluster is transforming the lattice to concentrate a universal cluster state from the qubits in the percolated lattice. The faulty cluster can be transformed into a universal resource for one-way quantum computation by using an algorithm such as the one presented in [105], which transforms an $L \times L$ percolated lattice into an $O(L) \times O(L)$ universal resource for MBQC, which in [105] is an hexagonal lattice. This approach scales optimally, i.e. linearly, in the size of the original lattice. The algorithm that is proposed in [105] can be summarised as follows: First a classical stage in which subsets of qubits with regular hexagonal lattice as a topological minor⁵ are identified. The classical steps are:

- Identify $O(L)$ disjoint paths that percolate through the lattice vertically and horizontally.
- Find “bridges” between horizontal paths, which are pieces of vertical paths that only enter the vicinity of the horizontal path at one point. Discard every other bridge between two horizontal lines to ensure that the global topology is that of an hexagonal lattice.
- Correct local errors by eliminating the superfluous edges in any intersection. Do so by keeping only the shortest horizontal path through an intersection area.

After the subsets of qubits have been identified, the quantum stage of the algorithm applies a series of Z and Y measurements to concentrate an hexagonal lattice from the qubits in the identified subset. The quantum steps of the algorithm are:

- Measure in Z all qubits outside the subset identified by the classical algorithm. This has the effect of disconnecting any measured qubits from the lattice.
- Topologically contract the graph into an hexagonal lattice by measuring Y on all vertices with coordination number 2. A Y measurement on a qubit (j) with coordination number 2 can be described as a graph transformation by connecting the neighbours of qubit j by an edge before disconnecting the vertex j from the lattice. This measurement thus contracts the line and leaves the topology of the subgraph unchanged.

As can be observed in figure 7.7 the only qubits remaining in the lattice have all coordination number 3 and they form a lattice topologically equivalent to a hexagonal lattice.

For the embedding of a fault-tolerant lattice onto the percolated lattice generated by the proposed LOQC scheme, we would require a similar algorithm to find a 3D universal lattice within the percolated photonic lattice. However, as the authors of [105] point out, an extension of their algorithm to three dimensions would require a new approach. The proof of the validity of their algorithm heavily relies on the planarity of the original lattice, which makes the argument not applicable in 3D. There is ongoing work [183] trying to achieve an efficient⁶ algorithm for this problem.

It is clear the improvement on resource efficiency that would be provided by an efficient algorithm of this type. A block k^3 of percolated lattice could be turn into an $O(k)^3$ sized

⁵A topological minor is a graph obtained by removing the maximum subset of vertices from the original graph that leaves the graph topology unchanged.

⁶With only polynomial overhead in time and space.

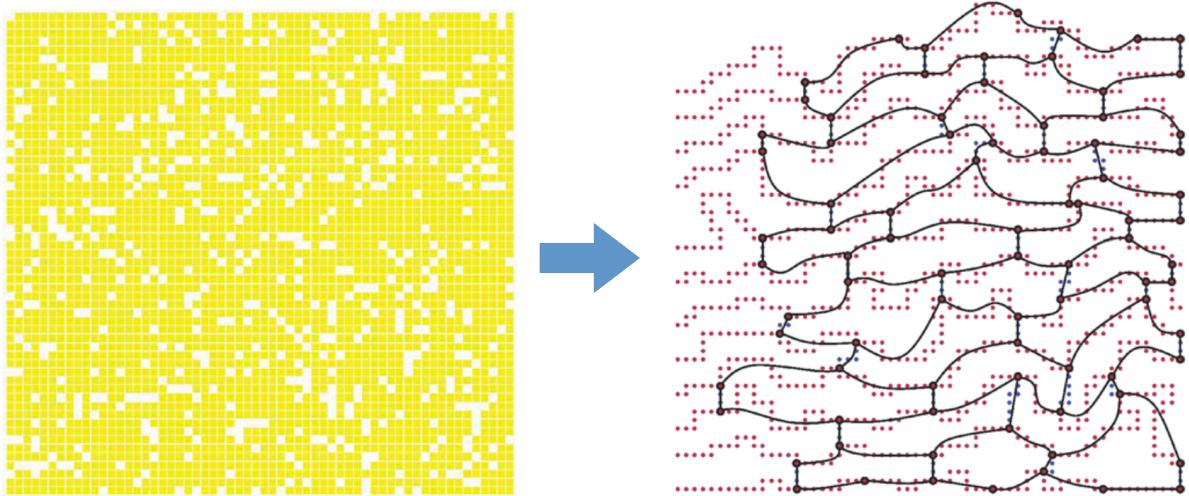


Figure 7.7: Transformation of an $L \times L$ percolated lattice into an $O(L) \times O(L)$ universal lattice for MBQC. Image adapted from [105].

universal lattice, whereas the renormalisation algorithm would use such a k^3 to define a single renormalised qubit in the universal lattice.

7.4.3 Percolated Raussendorf lattice

The Raussendorf lattice is a 3-dimensional lattice with coordination number 4, the exact same characteristics that made the diamond lattice the best choice for percolation. In this section we will focus on the percolation properties of Raussendorf lattice and the possibility of using a percolated Raussendorf lattice directly to perform QEC.

Mixed site bond percolation threshold

As the Raussendorf lattice is not a commonly used lattice in condensed matter systems, to our knowledge no numerical or analytical investigations of its percolation threshold exists in the literature. To obtain the mixed percolation threshold, we performed Monte-Carlo simulations in which we built a perfectly formed Raussendorf lattice and removed sites and bonds with probability $1 - p_s$ and $1 - p_b$ respectively, where p_s and p_b are the probability of occupancy of sites and bonds. We obtained the probability of percolation for different lattice sizes, p_s and p_b and found the threshold following the procedure detailed in section 5.4.1. The resulting numerical threshold is shown in figure 7.8, together with the mixed percolation thresholds of other lattices, obtained from [108]. The data points for the Raussendorf lattice and the diamond lattice are remarkably close, confirming their similar percolation behaviour, as expected given their dimensionality and coordination number.

The mixed site-bond percolation threshold gives valuable information about the lattice and can be used as a guide for designing percolation schemes. For the diamond lattice, we can easily check that the percolation scheme works (before any optimisations) by calculating the effective $p_s = 0.75^2 = 0.5625$ and $p_b = 0.75$ and confirming that they are (just) above the percolation threshold. The fact that both thresholds are so similar means that a small variation of layout

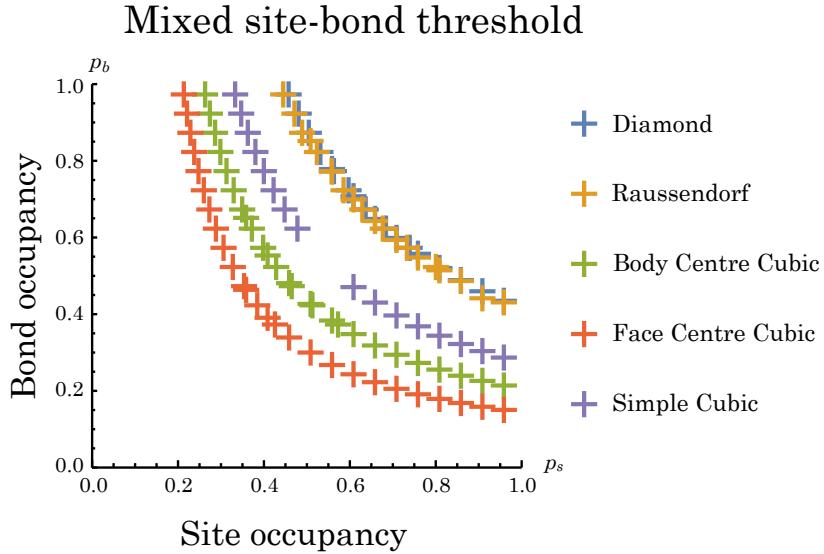


Figure 7.8: Mixed site-bond percolation threshold for various lattices in 3D. The data points of the Raussendorf lattice were obtained by numerical simulations while the data points for the other lattices have been obtained from [108].

on the percolation scheme proposed in chapter 5 will produce percolated Raussendorf lattice instead of percolated diamond.

Using a percolated Raussendorf lattice

In the percolation scheme for the diamond lattice, we were able to lower the percolation threshold by applying rotations prior to the fusion gates. The percolated lattice obtained was not a strict sub-lattice of the diamond, but the extra connectivity was beneficial. The case of the Raussendorf lattice is different, as we are interested in both the percolation and error-correcting properties of the lattice. The correlations created by the connectivity of the lattice is what makes this lattice so useful to protect against errors and loss and thus, extra edges will be detrimental to those correlations. Using the optimisation techniques we used for the diamond lattice would imply that there would be cluster edges between primal and dual qubits of the Raussendorf lattice, which would produce errors in the stabilizer measurements and the error correction. Therefore, for the generation of the Raussendorf lattice we will use a differently rotated Type-II boosted fusion that preserves the geometry of the Raussendorf lattice⁷.

Having this considerations, we build the Raussendorf lattice using an optical scheme similar to the one presented in chapter 5. We run Monte-Carlo simulations to obtain the percolation threshold *as a function of the fusion gate success probability*, which can be seen in figure 7.9. The data presented in 7.9 has been calculated from performing 10^4 repetitions of each data point.

Figure 7.9 shows that the percolation threshold can be estimated to be 69.8%. This threshold, while higher than the threshold for diamond, is still lower than the fusion probability we can achieve with the proposed boosted fusion gates ($p_s = 75\%$), proving that when building

⁷See chapter 4 for a comparison of all success and failure outcomes of the original and rotated fusion gates.

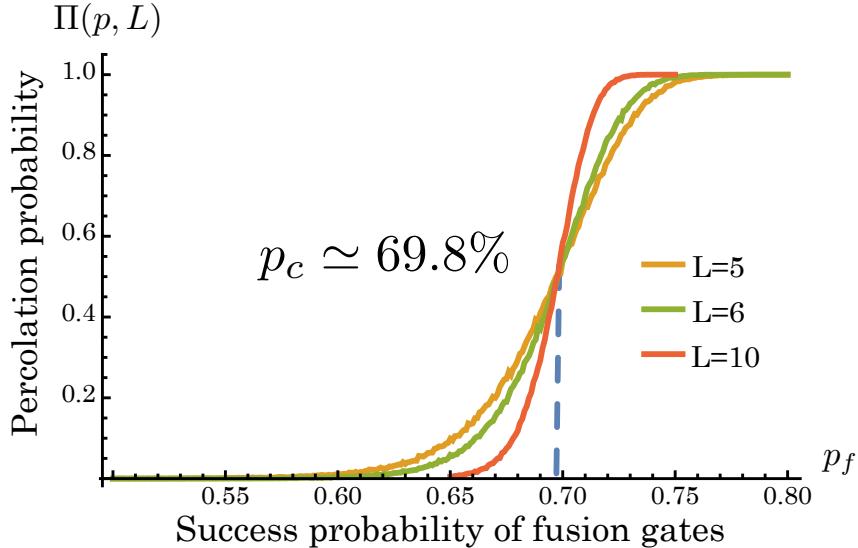


Figure 7.9: Percolation threshold for a Raussendorf lattice built with a similar scheme as the one presented in chapter 5. Note that the value of the percolation threshold, 69.8%, is higher than it was for the diamond lattice. The reason for this is that we are using rotated type-II gates that preserve less entanglement upon failure. These gates are beneficial, however, as they preserve the internal structure of the Raussendorf lattice (hence the percolated lattice *is* a strict sub-lattice of the Raussendorf lattice) which is required to take advantage of its QEC properties.

the cluster with our scheme, we are in the super-critical percolation regime. However, now we are concerned with using not only the percolation but also the fault-tolerant properties of this lattice. This lattice has very favourable properties for QEC codes, but its performance has only been tested when Pauli errors or loss errors affect it [177]. We need to understand how the fault-tolerant properties of the lattice behave when some of the correlations needed for its functioning are never built in.

It is expected that this error model will have a threshold, a bond loss rate above which the error correction fails. To be able to decide whether the lattice built with the linear optical scheme is above or below that threshold we will need to refer to the tradeoff between the fusion success probability and the bond loss rate. This tradeoff is provided in figure 7.10.

7.5 Topological codes under a bond loss error model

Bond loss is not a very well studied error model on topological codes. It can be hardly justified for other physical systems, as in most implementations the many-body stabilizer operators are measured by applying a series of two-qubit gates between the data qubits and some ancilla qubits. Given this implied ability to perform two-qubit operations deterministically, a situation where a missing cluster state bond cannot be replaced is hard to imagine. It is however a situation that easily arises in LOQC as we have shown in the previous section, and we therefore need to understand the effects of this type of error better. No definite results have yet been achieved, with the exception of the incompatibility of a surface code for our LOQC construction, which can be in any case argued unsuitable for other reasons as detailed later. In this section

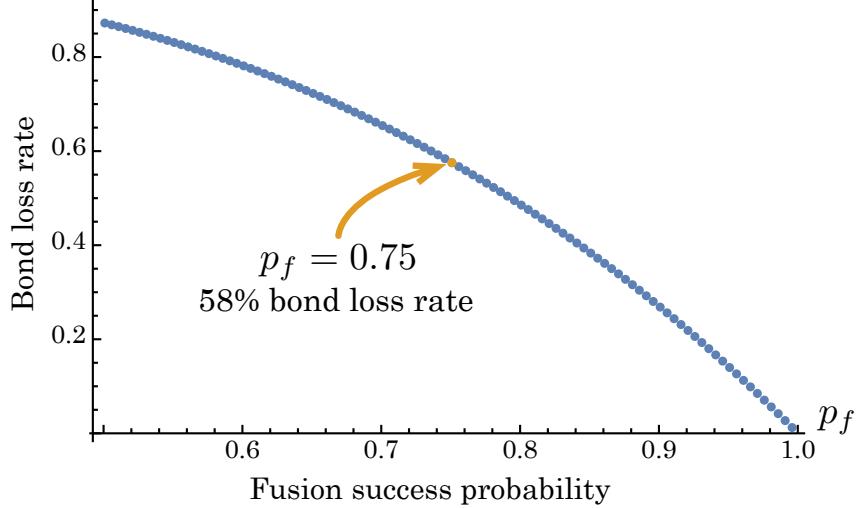


Figure 7.10: Bond loss rate in the Raussendorf lattice created by failed fusion gates. We can see that for 75% fusion probability we have a high bond loss rate of 58%.

we present preliminary results obtained in the study of this error model.

7.5.1 Surface code

The error correcting properties of any topological lattice depend on the correlations between qubits which are generated by the stabilizer operators. When an element of the lattice is missing, the natural error correction of the lattice is therefore impaired. The planar code has been shown to have good tolerance for qubit loss [176], QEC is still possible when the lattice has lost up to 50% of its qubits. The key factor behind this tolerance is the fact that there are always two stabilizers of each kind (plaquettes and stars) overlapping on any qubit. When losing a particular qubit, both operators are affected. However, due to their overlap, we can define a super-operator formed by the two original stabilizers multiplied together, and this allows the code to retain its error-correcting properties.

We analyse the complementary case: the loss of bonds in the planar code lattice. The planar code lattice can be built from a cluster state by performing regular Z and X measurements, or alternatively, it can be built from one layer of the Raussendorf lattice. This last case is the most appropriate in the context of this thesis, as we have shown how to build a Raussendorf lattice using a linear optical scheme in previous sections. Nonetheless, the effect of bond losses is the same in both cases, as the loss of a bond attached to a qubit that is measured in the Z basis has no effect (that bond is always removed from the lattice by the Z measurement). In figure 7.11 we can see a layer of Raussendorf lattice, where the qubits in green are measured in X to build the planar code onto the blue qubits.

The bond loss model is very different to the qubit loss model for the surface code. While two stabilizer operators of the same kind always overlap *on a qubit*, they never overlap *on a bond*

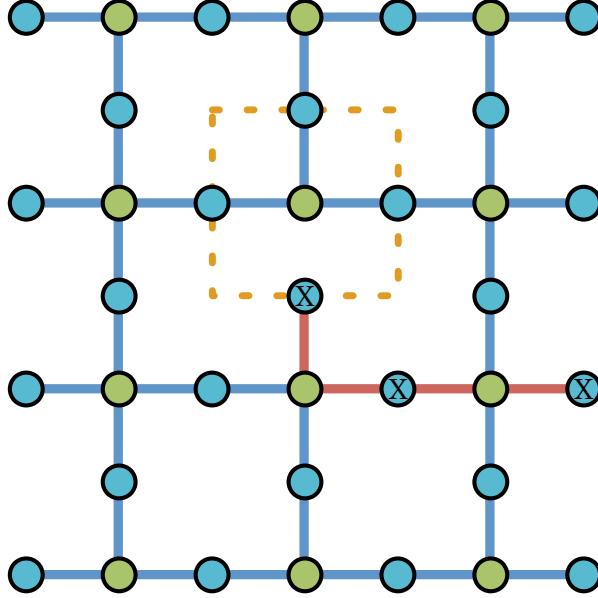


Figure 7.11: Planar code with a bond loss. Note that here the planar code is represented in cluster state form as a layer of Raussendorf lattice: the qubits represented in blue can be recognised as the planar code qubits from figure 7.1 while the green qubits are measured in the X basis in order to form the correlations of the planar code on the blue qubits. The plaquette operator marked in orange is broken, and the error chain marked in red will not be detected by the code stabilizers.

and thus, we cannot redefine a super-plaquette⁸ to avoid the effect of the loss. What is more, missing bonds in the lattice generates internal boundaries in the lattice, error chains between qubits that have lost a bond, or between such qubits and the lattice boundary, commute with all the stabilizers of the broken lattice and are therefore undetectable. One of such error chains can be seen in figure 7.11 marked in red. If the lattice was perfect, that chain would be detected by the plaquette operator marked in orange, but due to the bond loss it goes undetected.

The existence of these undetected error chains effectively reduces the code distance, as they anti-commute with the logical operator of the code (in the case presented in figure 7.11, the code distance has been reduced from 4 to 3). The code distance will therefore be determined by smallest distance between lost bonds. If we have a bond loss rate given by p , the average distance between bonds δ will scale as $\frac{1}{\sqrt{p}}$. As the code distance will be limited by δ , when $\delta < L$, where L is the dimension of the lattice, increasing the size of the lattice will no longer reduce the logical errors. We therefore expect to observe a pseudo-threshold: smaller codes will show the presence of a threshold, which will vanish for lattices with $L > \delta$.

The presence of the lost bonds in the code also breaks the degeneracy of the logical operators. Usually we have a choice on the qubit support of the logical operator and all possibilities are equally reliable to encode the logical qubit. However, if the logical operator is defined crossing a path between two of these bond losses, it will have a much higher probability of being corrupted than if it was defined far from these defects. Therefore, to improve performance when performing

⁸Note that star operators don't suffer from this kind of errors and the duality of the planar code operators is broken by this type of errors

QEC simulations on a lattice with missing bonds, the first step will be to determine the optimal logical operator given the configuration of bond losses present. In figure 7.12 we present the results of a QEC simulation for bond loss rate $p = 0.007$, where first the optimal logical operator is chosen by analysing the distances from each possible logical operator to the nearest lost bonds, and then a usual QEC round of inputting Pauli noise and decoding the errors is performed. We can clearly see how there is a threshold for small lattices, but it disappears as the code size increases beyond δ .

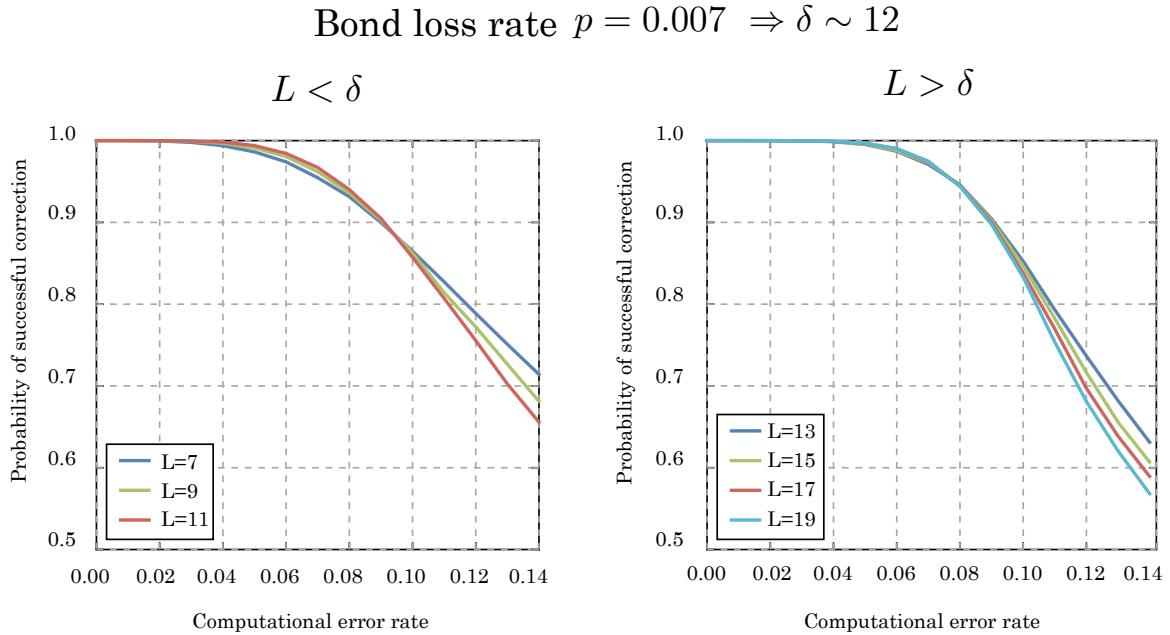


Figure 7.12: Probability of successful correction as a function of the computational error rate in the presence of lost bonds. The results on the left are for lattices with $L < \delta$, where the threshold can be clearly seen. For lattices $L > \delta$, the threshold disappears as we can see on the figure on the right. Simulation results courtesy of James Auger.

In figure 7.10 we provided the bond loss rate of the Raussendorf lattice when built using probabilistic fusion gates. For the fusion success probability of 75%, the bond loss rate was 58%. This would give an average separation between bond losses of $\delta \sim 1.3$, which clearly makes this QEC code unsuitable for our optical scheme. It should be also noted that given the destructive nature of measurements in optics, only one round of error correction would be possible using a surface code, which also makes the use of any surface code hard to justify for a linear optical implementation.

Cluster bond loss as a logical error on the planar code subspace

To understand better the effect of the bond loss on the stabilizers of the planar code, we map its effect to a logical operation *on the qubits of the planar code*. In figure 7.13 we show the cluster state graph with a bond missing. We consider this bond loss prior to the X measurements that would transform this cluster state into the planar code. The main consequence of losing a bond is that we reduce the weight of one of the plaquette operators. The qubit that has lost a bond

(marked in orange in figure 7.13) does not have the appropriate correlation to the qubits in one of the plaquette operators that have support on that qubit. This broken plaquette no longer commutes with the chain logical operator that crossed the lattice from rough edge to rough edge (top to bottom in figure 7.13), and this chain logical operator is reduced to a chain that ends at the broken plaquette. In order to restore the logical operator to its full length, a series of two-qubit operations is needed.

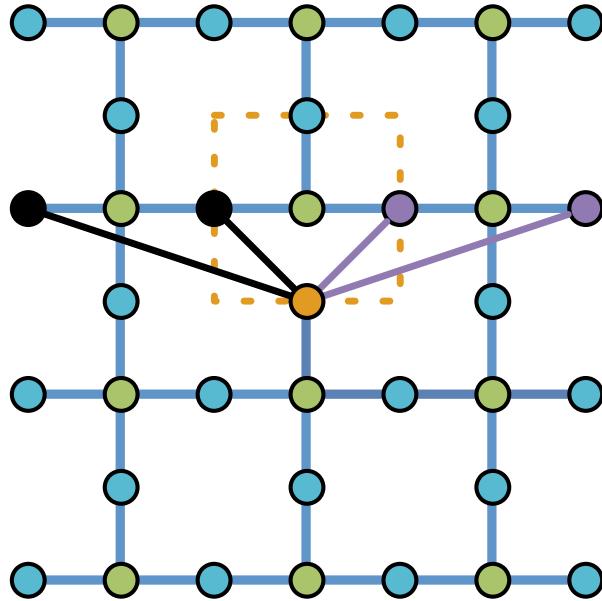


Figure 7.13: Planar code with a bond loss. The plaquette operator marked in orange is broken, the correction operation *on the qubits of the planar code* is a chain of CNOT operations. In the figure, two chains are marked, one in black, one in purple.

By analysing the transformation of the stabilizers⁹ that change when the bond loss happens (the reduced weight plaquette operator and the logical chain operator) we realise that the transformation is equivalent to a chain of CNOT gates applied with control on the qubit with the lost bond (highlighted in orange) and target on a chain of qubits that go from the broken plaquette to the nearest smooth edge. Two examples of such chains of non-local CNOT gates can be seen in figure 7.13. As the CNOT gates are self-inverse, the correction chains can also be used to simulate the occurrence of this type of error on the planar code. It is an unnatural error to happen if the planar code is built directly and not from a cluster state, as non-local correlated errors are unlikely in most physical settings.

In the case of having many lost bonds, this correction (or error) chains can overlap (in figure 7.13 this would mean that one qubit would belong to both correction chains and belong to the sets of black and purple qubits) and therefore more than one CNOT, each corresponding to different bond losses, might be applied to the same qubits.

⁹For this analysis, we used Visual-CHP code presented in appendix D.

7.5.2 Raussendorf lattice

We have shown that the bond loss errors on the planar code can only be corrected by performing a series of two-qubit gates. However, that is not possible for the LOQC percolation scheme, as we cannot perform deterministic two-qubit gates. In three dimensions however, the effect of a lost edge in the lattice is not as detrimental. The stabilizer operators in the (3D) Raussendorf lattice are cubic operators that overlap on the faces. As any lost edge will always be located in a face of the cube, the argument of [176] does apply here: by multiplying together the operators that overlap on the face that contains the lost edge to form a new super-operator, the QEC properties of the lattice should be preserved (at least for bond loss rates lower than some threshold that needs to be determined). In figure 7.14 we show an example of a missing bond, whose effect can be cancelled by multiplying together the logical operators (defined on the red qubits) of the two adjacent qubits.

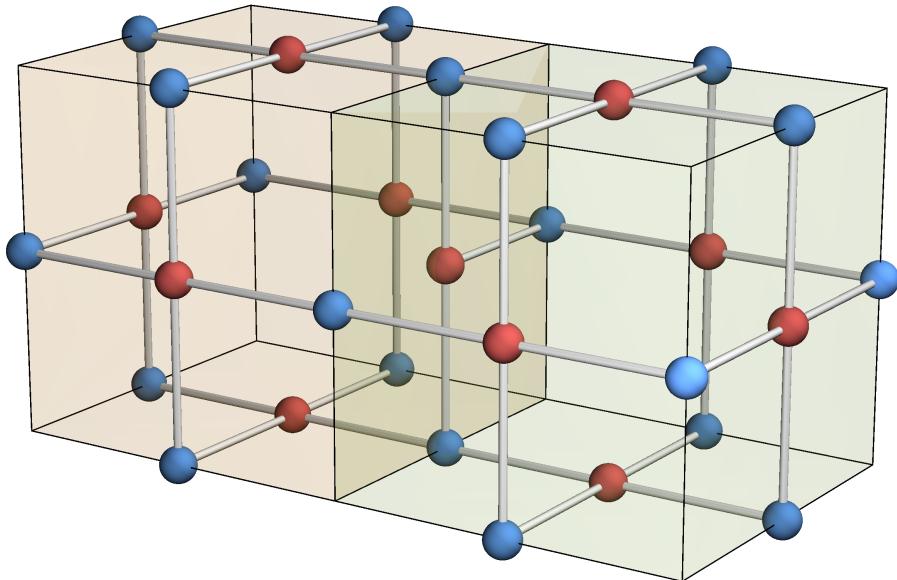


Figure 7.14: A bond is missing in the overlapping face of the two cubes. The damaging effects of the missing edge can be cancelled by multiplying together the two adjacent cubes to form a super-cuboid.

Preliminary results suggest that the tolerance to bond loss of the Raussendorf lattice is much higher than the tolerance of the planar code. However, many more simulations and studies need to be performed in order to determine if using percolated Raussendorf lattice is a viable option for LOQC.

7.6 Discussion and outlook

In this chapter we have presented a number of approaches to integrate error-correcting codes in a linear optical architecture. We have focused on topological codes, as they have been shown [176, 177] to have high thresholds for both Pauli errors and loss, which makes them highly

appropriate for linear optics. We have presented three main approaches for the implementation of QEC: renormalisation, in which blocks of percolated lattice form renormalised qubits of the topological lattice; concentration, in which groups of qubits that are topologically equivalent to the error-correcting lattice are selected and concentrated to form the topological lattice; and finally the possibility of building a percolated topological lattice directly. This last possibility is the most promising as it seems to be the most resource efficient implementation among the rest, however, only preliminary studies have been performed.

Future work will include a topological analysis comparing the differences of 2D and 3D lattices under such bond loss model. It is quite striking that increasing the number of dimensions makes such a difference in the error correcting properties, and although the difference between these codes can be understood from the arguments we have presented, a full mathematical proof of the error-correcting properties of both codes is required. It will also be very interesting to study other types of topological codes such as colour codes [184, 185] and the recently proposed doubled colour codes [186, 187], under qubit loss and bond loss and their implementation on a linear-optical setup.

A pending matter for LOQC is the full characterisation of errors, mapping photon errors such as multi-photon contamination and mode-mismatch to Pauli error rates and loss. Only fully understanding how these physical errors affect the encoded quantum information, we will be able to devise schemes and implement codes that allow us to perform a full fault-tolerant quantum computation.

CHAPTER 8

CONCLUSION

Quantum computers promise computational capabilities far beyond what can be achieved with classical computers, however they have not been built yet. A number of candidate systems have proposed architectures to do so, and in this thesis we explored the suitability of linear optical systems. After reviewing the advantages and challenges of this physical platform, as well as the most important protocols for linear optical quantum computing proposed so far [2, 88, 89, 94, 95, 90, 99], we presented investigations into all the different theoretical stages of the construction of such a quantum computer.

Due to the linearity of the mode transformations in linear optics, entangling two-qubit operations cannot be achieved deterministically [114]. However, entangling operations and entangled states are necessary for quantum computing [70]. In chapter 4 we reviewed schemes for the optical implementation of Bell-state measurement, including some recent results [111, 115] that show how the probability of successfully generating entanglement can be improved when using ancillary photonic states. Using these results we boosted the success probability of Type-II fusion gate [90], which is predominantly used when generating small entangled states, and proposed a series of schemes for generating n -GHZ states. We studied the resource consumption of all these schemes in near-deterministic nested multiplexed setups with probabilistic sources, where the probabilistic generation is repeated multiple times in order to ensure the successful generation of at least one entangled state. An interesting conclusion of the results presented is that setups with higher source efficiency are more resource efficient when using fewer multiplexing steps. This is a promising result for linear optics, because it is expected that the efficiency of single-photon sources [58] will increase as the technology improves, and fewer multiplexing stages means less loss due to active switching in the overall scheme. The results proposed in this chapter are a step forwards, but are by no means optimal. New techniques need to be developed to find optimal circuits in a more efficient manner and a better understanding of some of the processes used is required. In particular, an analysis of the two proposed boosted Bell-state measurements [111, 115] shows that they are not optimal and further research is needed to improve their efficiency.

In chapter 5 we presented a ballistic scheme for the construction of a linear optical cluster state that is universal for MBQC. Only 3-GHZ states and Bell pairs are necessary as resource states, which implies that our schemes consumes one order of magnitude fewer resources than other ballistic schemes proposed. Not only is our scheme more resource efficient but it has a natural loss tolerance which allows to tolerate up to $\sim 1.6\%$ photon loss without the use of

8. CONCLUSION

specific loss tolerant codes. We not only proposed a theoretical scheme, but also integrated this scheme into a full architectural blueprint, which has an important quality: it has fixed physical depth, meaning the number of operations each photon is subject to is fixed. We have used current technologies, such as probabilistic single-photon sources and log-tree switching networks and showed that they are in principle sufficient to build a photonic quantum computer. In chapter 6 we studied in detail the resource efficiency of the scheme proposed and introduce a new multiplexing scheme that aims to optimise the use of resources. Instead of having a quantum computer with a set clock-cycle, we proposed an asynchronous architecture in which events are not synched to an overall clock cycle but only synchronised with other events when interference needs to take place. We have presented the application of this idea on one level of the architecture and showed the resulting resource savings. Understanding better the theoretical aspects of this asynchronous scheme is a future line of research.

The integration of quantum error-correcting codes is fundamental in any quantum computing architecture [70, 47]. In chapter 7 we presented a number of approaches for the integration of fault-tolerant techniques in the linear optical architecture proposed in chapter 5. One of these approaches seems particularly promising, which is the probabilistic construction of a fault-tolerant lattice using the same techniques that were used to build the photonic cluster state in chapter 5. However, the errors caused by the non-deterministic gates used to build the cluster cannot be described by the type of error models commonly used in quantum error correction. This new type of error model is analysed for the two-dimensional case using the stabilizer simulator described in appendix D, but the analysis of a three-dimensional lattice under this error model is ongoing work. Other errors in the linear optical architecture need to be understood better, a full characterisation of physical errors and their mapping to computational errors and loss is an important open question for this architecture.

In this thesis, we have aimed to fully understand the challenges of building a linear optical quantum computer. We looked at every stage of the computation process, from the generation of entangled states to the implementation of error-correction. We proposed schemes that ameliorate the challenges faced, but there's still a long road ahead before the first linear optical quantum computers can be built. However, recent experimental advances [58, 56, 43, 59, 60], in particular the ability to nano-fabricate integrated devices with $O(10^6)$ linear optical elements on a single chip [61], reinforce the suitability of linear optics for quantum computing. The results of this thesis, together with these experimental advances demonstrate that building a linear-optical quantum computer is less challenging than previously thought.

APPENDIX A

COMPLEXITY

This appendix contains a collection of complexity theory results that are mentioned throughout the thesis and are relevant to the topic of this thesis.

A.1 Turing machine

A *Turing machine* is a device containing three main elements: an infinite tape, a read/write head and a control device [188]. The infinite tape is divided into cells, these cells contain a (possibly infinite) sequence of symbols from a finite set called the alphabet. The read/write head moves along the tape and changes the symbols according to the instructions given by a control device, which is a finite-state automaton. At each step of the computation, the control device is in a particular state. The state of the control device and the symbol under the head determine the action performed by the Turing machine: the value of the transition contains the new state of the control device, the new symbol for the cell in the tape and the shift of the read/write head.

A.2 Extended Church-Turing thesis

The Church-Turing thesis, named after Alan Turing [23] and his adviser Alonzo Church [24] states that *any function “naturally to be regarded as computable” is computable by a Turing machine*, i.e. any natural model of computation will give you the same set of computable functions as a Turing machine (or else a subset of them) [28]. The extended version of the Church-Turing thesis states that *any function naturally to be regarded as “efficiently” computable is “efficiently” computable by a Turing machine*.

The significance of the Church-Turing thesis is the statement that the limitations on what can be computed are not imposed by our ingenuity in designing and constructing models of computation or the technology that is used to do so, but the limitations are universal and set by Nature.

A.3 Complexity classes

Problems in complexity theory are generally set as decision problems, with yes and no answers or as counting problems, in which the answer is the number of solutions to said problem. In

this appendix we list a few complexity classes that we mention throughout the thesis.

P: Polynomial-time

Class of decision problems solvable in polynomial time by a Turing machine.

Examples: Multiplication, primality testing.

NP: Nondeterministic polynomial-time

Class of decision problems such that, if the answer is *yes*, there exists a witness which is polynomial in the size of the input and can be verified in polynomial time. If the answer is no, there exists no witness that will fool a verifier that the answer is yes.

Examples: Factoring, graph isomorphism.

Co-NP: Complement of NP

Class of decision problems such that, if the answer is *no*, there exists a witness which is polynomial in the size of the input and can be verified in polynomial time.

Examples: PRIME

NPH: NP-Hard

Class of decision problems such that any *NP* problem can be efficiently (Turing) reduced to it (see notions of reducibility in the following section). They are as hard as any problem in NP.

Examples: Circuit satisfiability.

NPC: NP-Complete

Class of decision problems such that they are in NP and every problem in NP is reducible to them. They are in both the complexity classes *NP*&*NP* – Hard.

Examples: 3-SAT, subset-sum.

BPP: Bounded-error Probabilistic Polynomial-time

Class of decision problems solvable by an NP machine such that:

- If the answer is yes, accept with probability $\frac{2}{3}$.
- If the answer is no, accept with probability $\frac{1}{3}$.

This is the class of feasible problems for a classical computer with access to a genuine random-number generator.

Examples: Monte Carlo simulation of fermionic many-body systems.

PH: Polynomial-Hierarchy

Class of decision problems solvable in polynomial time by a Turing machine with access to infinite number of oracles of the preceding level in the hierarchy. An oracle (also usually called “black box”) is an imaginary device that solves some computational problem immediately. A^B is the set of decision problems solvable by a Turing machine in class A augmented by an oracle for some complete problem in class B . Define $\Delta_0^P = \Sigma_0^P = \Pi_0^P = P$. For $i \geq 0$, define: $\Delta_{i+1}^P = P^{\Sigma_i^P}$, $\Sigma_{i+1}^P = NP^{\Sigma_i^P}$, $\Pi_{i+1}^P = coNP^{\Sigma_i^P}$.

For example, $\Sigma_1^P = NP$, $\Pi_1^P = coNP$, $\Sigma_2^P = P^{NP}$ is the class of problems solvable in polynomial time with an oracle for some NP-complete problem.

#-P: Sharp-P

Class of all problems that can be phrased as *counting* the number of solutions to an NP problem. As this is a counting class and not a decision class, it is difficult to compare it with the other decision classes. A simple solution is to consider the class $P^{\#P}$, which contains all problems decidable by a P machine with access to a $\#P$ oracle. Toda’s theorem [28] says that $P^{\#P}$ contains the entire polynomial hierarchy PH, implying that this counting class is more powerful than the entire PH.

Examples: Calculating the permanent of a matrix.

BQP: Bounded-error Quantum Polynomial-time

Class of decision problems solvable by a quantum Turing machine such that:

- If the answer is yes, accept with probability $\frac{2}{3}$.
- If the answer is no, accept with probability $\frac{1}{3}$

One of the biggest open problem in complexity theory is the relationship between BQP and the polynomial hierarchy. The intuition is that it is unlikely that $BQP \subset PH$, however an oracle relative to which $BQP \not\subset PH$ has not yet been found. It was shown by Bernstein and Vazirani [189] that $BPP \subseteq BQP \subseteq P^{\#P}$. This result implies that quantum computers are at least as powerful as classical probabilistic computers and no more than exponentially faster.

Examples: Factoring, discrete logarithm.

A.4 Notions of reducibility

In computational complexity theory, a polynomial-time reduction is an algorithm for transforming one problem into another problem, which is computable by a deterministic Turing machine in polynomial time.

Cook reducibility

In Cook’s definition of reducibility, an algorithm to solve problem A with access to an oracle for problem B is allowed to make multiple queries to the oracle, see figure A.1.

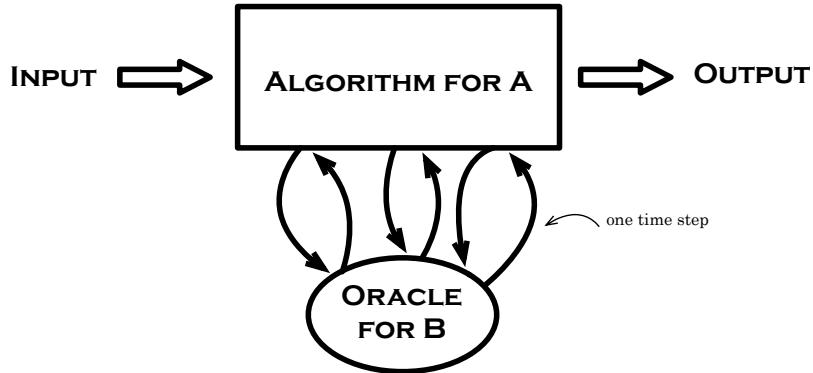


Figure A.1: Cook reducibility

Karp reducibility

In Karp's notion of reducibility, the input for problem A is taken through a function f which transforms instances I of A into instances $f(I)$ of B such that the size $f(I)$ is polynomial in the size of I and I is a yes instance $\iff f(I)$ is a yes instance. This means that only one query is made to the oracle, see figure A.2.

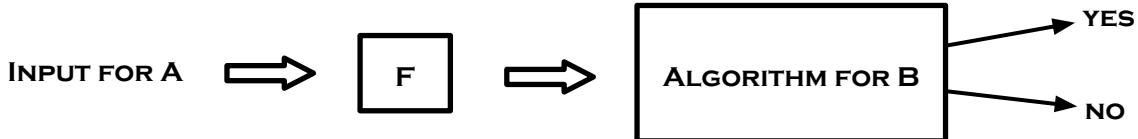


Figure A.2: Karp reducibility

A.5 Collapse of the Polynomial Hierarchy

The Polynomial Hierarchy collapse to the i^{th} level occurs when it can be proven that $\Sigma_i^P = \Delta_i^P$. Examples of this event are the collapse to the 0^{th} level, occurring if it can be proven that $P = NP$, and the collapse to the first level, occurring if $NP = coNP$. In complexity theory, many results (such as the afore mentioned computational complexity of linear optics [30]) have the conclusion that if a polynomial-time solution can be found on a Turing machine to a particular problem, then the polynomial hierarchy would collapse to a certain level.

We can get an intuition to why the polynomial hierarchy should not collapse by looking at the most well known¹ instance of the problem, the collapse to the 0^{th} level, i.e. the P vs NP question. No proof exists to determine if both complexity classes are equal or not, however, it is widely believed they are not equal (and the polynomial hierarchy doesn't collapse). If NP

¹One of the Clay Math Institute Millennium Problems.

problems were solvable in polynomial time, then mathematical creativity could be automated [28]; checking a mathematical proof would be as hard as finding a proof. Furthermore, there exists the general belief that algorithms that solve NP problems in a way that is dramatically better than brute force search don't exist.

Collapse to the 0th level is the most dramatic polynomial hierarchy collapse, and it may be possible that a collapse of the hierarchy to a higher level doesn't have such striking consequences. However, relying on intuition about complexity of problems is tricky, as there are algorithms that have been for decades widely believed to be efficiently intractable, such as primality testing, but have later been proven to have polynomial-time algorithms [29].

APPENDIX B

RESOURCE COUNTING

When considering the resources needed to build a Linear Optical Quantum Computer, one of the best ways to put all schemes on an equal footing is to compare them with respect to a consumable resource (as optical elements can be reused in different ways according to the particular scheme), i.e. photons. However, each scheme uses photonic states of different sizes. To compare them, we will calculate the number of Bell Pairs needed on average to obtain each photonic state. For this task, we will use the Linear Optical Networks (LON) for building GHZ states described in chapter 4, as they are the most efficient we know. Note that we will be using the most resource efficient networks from section 4, but these networks use gates that are not loss tolerant.

There are two types of schemes we will be considering: repeat until success schemes, which require feedforward (KLM [2] , Nielsen [89], Yoran-Reznik [88], parity encoding [94, 95] and Browne-Rudolph [90]) and ballistic schemes that don't require feedforward (the percolating scheme from [99] and the one presented in chapter 5). As the ballistic schemes are based on percolation and not in the individual success of two-qubit gates, we will assess how many resources are needed to build the final cluster state. For schemes with feedforward however, we will first assess how many Bell pairs are needed per logical two-qubit gate.

B.1 Comparison of all proposed schemes

Both KLM and Nielsen's scheme use the $CZ_{n^2/(n+1)^2}$ gate introduced in KLM, which performs a CZ with success probability $p = \frac{n^2}{(n+1)^2}$. Each attempt to implement this gate requires a $4n$ -photon entangled state.

Schemes with feedforward: we estimate average number of Bell Pairs per two-qubit gate:

- **KLM scheme** [2] : In their paper, they use the $CZ_{9/16}$ gate, which requires a 12-photon entangled state per implementation of the gate. They estimate that per single entangling gate they need about 300 successful $CZ_{9/16} \Rightarrow 300 \times \frac{16}{9} \equiv 534$ 12-photon entangled states. To produce a 12-photon GHZ states, we need 11 Bell pairs and obtain the desired state with probability $(\frac{1}{2})^{10} = 0.098\%$. On average we need $11 \times 2^{10} = 11264$ Bell pairs per 12-photon GHZ state. Therefore, to implement a successful two-qubit gate in KLM, we need $534 \times 11264 = \mathbf{6.014 \cdot 10^6}$ Bell pairs on average.
- **Yoran-Reznik scheme** [88]: In their paper, they use the $CZ_{9/16}$ and the $CZ_{4/9}$, which

require a 12-photon and 8-photon entangled states respectively per implementation of the gate. They estimate that per single entangling gate they need about $23 \text{ } CZ_{9/16}$ and $69 \text{ } CZ_{4/16}$ gates. To produce a 12-photon GHZ states, we need 11 Bell pairs and obtained the desired state with probability $(\frac{1}{2})^{10} = 0.098\%$. On average we need $11 \times 2^{10} = 11264$ Bell pairs per 12-photon GHZ state. To produce a 8-photon GHZ state, we need 7 Bell pairs, we obtain the desired state with probability $2^6 = 1.56\%$. On average we need $7 \times 2^6 = 448$ Bell pairs per 8-GHZ state. Therefore, to implement a successful two-qubit gate following the scheme of Yoran and Reznik [88], we need $23 \times 11264 + 69 \times 448 = \mathbf{2.9 \cdot 10^5}$ Bell pairs on average.

- **Nielsen's scheme** [89] : In his paper, Nielsen calculates that to implement a two-qubit logical gate he needs 24 successful $CZ_{4/9}$ gates. Each of these gates requires an 8-photon entangled state, therefore we need $\Rightarrow 24 \times \frac{4}{9} = 54$ 8-photon entangled states. To produce a 8-photon GHZ state, we need 7 Bell pairs, we obtain the desired state with probability $2^6 = 1.56\%$. On average we need $7 \times 2^6 = 448$ Bell pairs per 8-GHZ state. Therefore, to implement Nielsen's scheme we need $24 \times 448 = \mathbf{1.075 \cdot 10^4}$ Bell pairs on average.
- **Browne-Rudolph scheme** [90] : In this scheme, to implement a two-qubit logical gate they add an L-shape to the cluster. On average they calculate that they need **52** Bell pairs on average to do it. This is a very low cost in comparison with the other schemes. The resource efficiency of this scheme is the result of combining the most efficient computational model, MBQC, which already reduced resources in the case of Nielsen's approach, *and* the use of the most resource-efficient gates, the fusion gates. Other schemes either use the MBQC model, or the fusion gates, but it's their combination what makes the resources necessary for the Browne-Rudolph scheme so low.
- **Hayes-Gilchrist-Myers-Ralph scheme** [94]: In this scheme, they estimate that for a 95% probability encoded CNOT, their scheme would require on average 90 physical CS and 32 elimination circuits. This translates into 1300 Bell states and 620 “elimination states”, as shown in [94]. For the elimination states, the authors give a probabilistic preparation procedure where only single photon states are required. However, these elimination states are two-photon states that can be deterministically prepared from a Bell state by applying a linear optical elements. Thus we will consider them in the Bell pair count. Therefore these scheme requires $\mathbf{1.92 \cdot 10^3}$ Bell pairs on average per single entangling gate.
- **Gilchrist-Hayes-Ralph scheme** [95]: To perform an entangling gate, a combination of type-I and type-II fusion gates is used. The authors numerically explore the optimal strategy, and conclude that the best way to obtain a resource of $|0\rangle^5$ states is to first fuse two Bell states ($|0\rangle^2$) with a type-I gate, resulting in $|0\rangle^3$ (with an average cost of $4|0\rangle^2$) and then further fuse two $|0\rangle^3$ states with type-I to form $|0\rangle^5$. This has an average cost of $16|0\rangle^2$ per $|0\rangle^5$. Once there is a supply of $|0\rangle^5$, it is advantageous to proceed by using the type-II gate. The authors give a table of values of the average number of resources consumed to perform an encoded CNOT gate with different success probabilities (the success probability of the gate depends on the level of encoding used in the logical state).

Here we compare the resources consumed by the gate which succeeds with 96.4% as it is the closest to the 95% gate mentioned in the previous parity-encoded scheme. For this gate, they use logical qubits with $n = 6$ levels of encoding, for which they need to prepare 115 $|0\rangle^5$ states. Therefore on average we require $115 \times 16 = 1.84 \cdot 10^3$ Bell pairs per CNOT gate.

Schemes without feedforward: when considering the schemes that are based on percolation, we should first look at the initial micro-clusters needed for each scheme and calculate how many Bell pairs are needed on average to obtain them. In the Kieling-Rudolph-Eisert[99] proposal, they suggest percolation schemes starting with 7-photon GHZ states, 5-photon GHZ states and 4-photon GHZ states. The scheme we propose in chapter 5 requires only 3-photon GHZ states while obtaining the optimal scaling reported in [99].

- 7-photon GHZ state: To obtain a 7-photon GHZ states we need 6 Bell pairs and the LON works with probability $(\frac{1}{2})^5 = 3.125\%$. On average we need $6 \times 2^5 = 192$ Bell pairs to obtain a 7-photon GHZ state.
- 5-photon GHZ state: To obtain a 5-photon GHZ states we need 4 Bell pairs and the LON works with probability $(\frac{1}{2})^3 = 12.5\%$. On average we need $4 \times 2^3 = 32$ Bell pairs to obtain a 5-photon GHZ state.
- 4-photon GHZ state: To obtain a 4-photon GHZ states we need 3 Bell pairs and the LON works with probability $(\frac{1}{2})^2 = 25\%$. On average we need $3 \times 2^2 = 12$ Bell pairs to obtain a 4-photon GHZ state.
- 3-photon GHZ state: To obtain a 3-photon GHZ states we need 2 Bell pairs and the LON works with probability $(\frac{1}{2}) = 50\%$. On average we need $2 \times 2 = 4$ Bell pairs to obtain a 3-photon GHZ state.

We can see that having smaller micro-clusters saves orders of magnitude in terms of the resources, even though the scaling obtained is similar (if not improved) to the one reported with bigger microclusters.

Comparison when building a cluster state of dimensions $L \times L$:

- Nielsen[89] : To build a cluster state, this scheme would require to perform $O(L^2)$ logical two-qubit gates, therefore the number of Bell Pairs required would be $O(10^4)O(L^2)$.
- Browne-Rudolph[90] : This scheme would require $O(10)O(L^2)$ Bell pairs.
- Our percolation scheme: In our scheme, to build a cluster state of renormalised qubits, we would require $k^3O(10)O(L^2)$ Bell pairs, where $k < 10$.

The Browne-Rudolph proposal is more efficient in terms of the number of Bell pairs it consumes than our proposed percolation scheme. Despite this advantage in terms of resources, the percolation scheme has the advantage that it does not require active feed-forward and works on a static LON, making the experimental realisation much more feasible and the loss rate much lower.

B.2 Comparison of percolation schemes

In figure 4 of their paper [99], the authors show the dependance of the diamond lattice block size k^3 on the size L of the renormalised square lattice for three different sets of site bond probabilities (p_{site}, p_{bond}). The overall success probability threshold $P(L)$ was chosen to be $\frac{1}{2}$. In the scheme presented in chapter 5, we have performed all the simulations, assuming the GHZ states are provided deterministically. For the scheme comparison to be fair, we choose to compare with the data points that correspond with the data set (1.00, 0.5). From the data used to produce figure 7 in our paper, we extract for different k s, what is the maximum value of L we can reach with $\Pi(L) \geq \frac{1}{2}$.

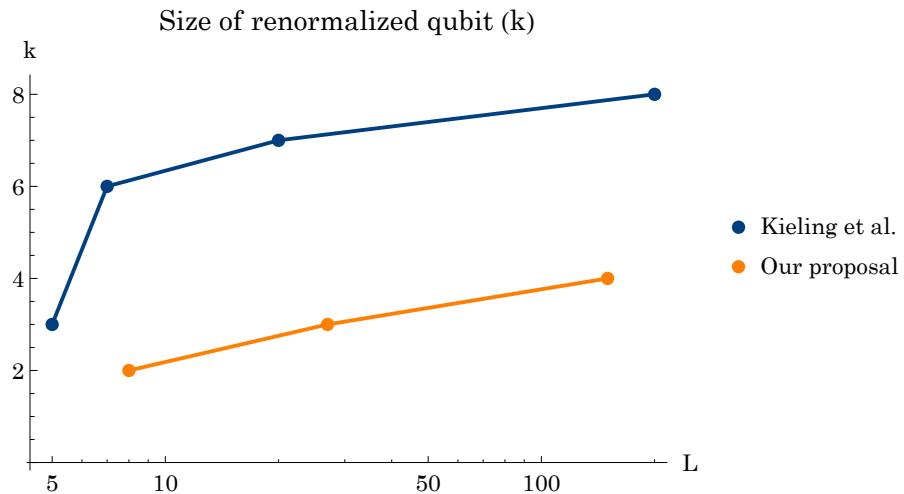


Figure B.1: Comparison of the size of the renormalised qubit (k) for different cluster sizes (L)

We can see that already there is a significant improvement in our scheme, as the size of the renormalised qubit is reduced noticeably in our scheme with respect to Kieling *et al.*'s scheme [99]. But the improvement becomes much greater once we consider the number of Bell pairs that are needed to build each renormalised qubit and the entire cluster.

To obtain this comparison, we will first calculate how many Bell pairs are needed to obtain a GHZ with 100% probability ($\pm 0.0001\%$).

- The data in [99] is obtained for 4-photon GHZ states. For each 4-photon GHZ state we need 3 Bell pairs and the LON works with probability $\frac{1}{4}$. In order to have a deterministic 4-photon GHZ ($p_{succ} = 1 \pm 10^{-6}$) we must repeat the generation procedure t times, where t is $1 - (1 - \frac{1}{4})^t \geq 1 \Rightarrow t = 51$. In total we consume $3 \times 51 = 153$ Bell pairs in the generation of a deterministic 4-photon GHZ state.
- In our proposal we require deterministic 3-photon GHZ states. For each attempt at generating one, we need 2 Bell pairs and the LON works with probability of success $\frac{1}{2}$. In order to have a deterministic 3-photon GHZ ($p_{succ} = 1 \pm 10^{-6}$) we must repeat the generation procedure t times, where t is $1 - (1 - \frac{1}{2})^t \geq 1 \Rightarrow t = 21$. In total we consume $2 \times 21 = 42$ Bell pairs in the generation of a deterministic 3-photon GHZ state.

With these numbers, we can transform the data in figure B.1 into number of consumable resources used for different cluster sizes.

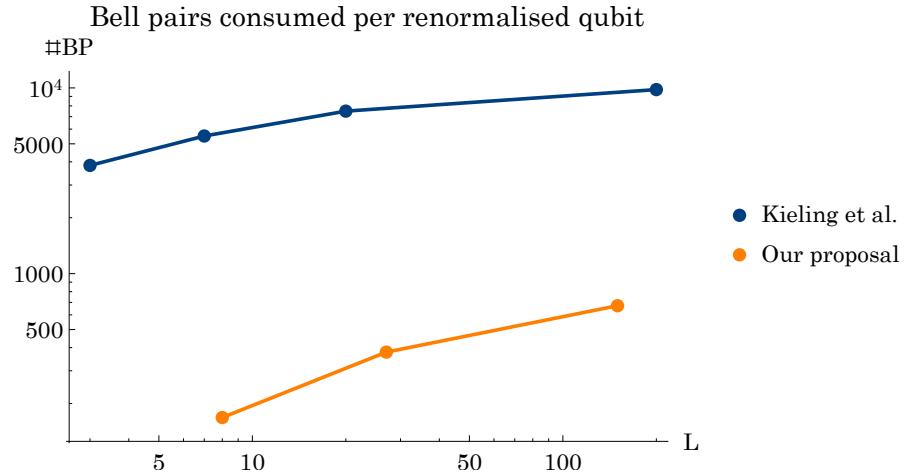


Figure B.2: Comparison of the number of Bell pairs consumed per renormalised qubits for different cluster sizes (L).

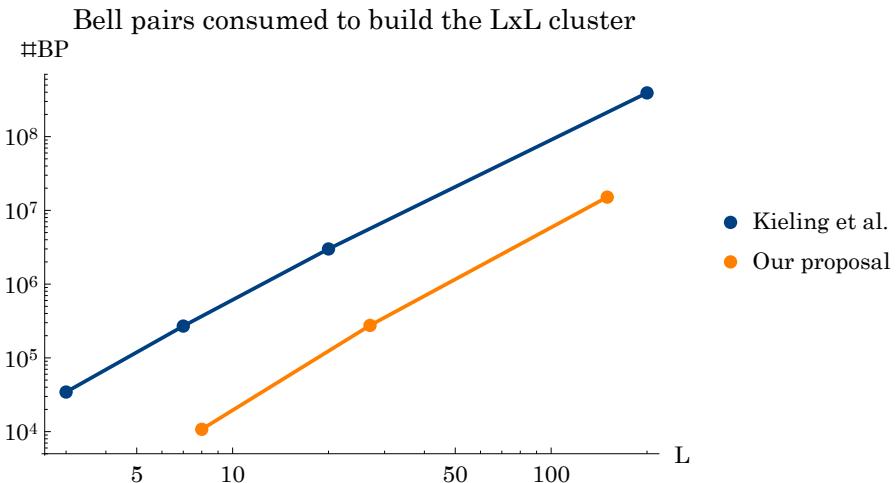


Figure B.3: Comparison of the number of Bell pairs consumed to build the entire $L \times L$ cluster for different cluster sizes (L).

APPENDIX C

CONSTRUCTIVE PROOFS

In this appendix, we provide some formal proofs omitted in chapter 3. The steps in the visualisation algorithms presented in chapter 3 follow the steps in these constructive proofs and therefore these proofs are needed to understand the validity of the algorithms.

C.1 Proof of theorem 3

Theorem. *Every stabilizer state is equivalent to a graph state under local Clifford operations [121].*

Proof. Given an arbitrary stabilizer state $S = \begin{bmatrix} S_Z \\ S_X \end{bmatrix}$, we want to prove the existence of a local Clifford operation such that

$$Q \cdot S = \begin{bmatrix} S'_Z \\ S'_X \end{bmatrix}, \quad (\text{C.1})$$

where S_X is invertible, so that it can act as a basis change for the stabilizer generators. Then,

$$S' = Q \cdot S \cdot S'^{-1}_X = \begin{bmatrix} S'_Z \cdot S'^{-1}_X \\ S'_X \cdot S'^{-1}_X \end{bmatrix} = \begin{bmatrix} S'_Z \cdot S'^{-1}_X \\ \mathbb{1} \end{bmatrix}. \quad (\text{C.2})$$

This new stabilizer state now corresponds to a graph state, S_G as defined in chapter 3. The sub-matrix $S'_Z \cdot S'^{-1}_X$ is symmetric from the property $S'^T \cdot \mathbb{P} \cdot S' = 0$. Not in all cases the submatrix S_X of the stabilizer state S will be invertible. An example of this is the stabilizer describing n qubits in the $|0\rangle$ state, in which case the S_X matrix corresponds to a matrix of all zeros. Therefore there must exist a local Clifford operation that transforms the matrix X into an invertible block.

The first step we take is to perform a basis change (which in this particular case is equivalent to Gaussian elimination) in the original stabilizer to bring it to the form

$$S \rightarrow \begin{bmatrix} R_Z & T_Z \\ R_X & 0 \end{bmatrix}, \quad (\text{C.3})$$

where R_X is a full rank $k \times n$ matrix.

The symplectic self-orthogonality of the stabilizer group implies that $T_Z^T R_X = 0$. We prove

this:

$$S^T = \begin{bmatrix} R_Z^T & R_X^T \\ T_Z^T & 0 \end{bmatrix} \Rightarrow S^T \cdot \mathbb{P} \cdot S = \begin{bmatrix} R_Z^T & R_X^T \\ T_Z^T & 0 \end{bmatrix} \begin{bmatrix} R_X & 0 \\ R_Z & T_Z \end{bmatrix} = \begin{bmatrix} R_Z^T R_X + R_X^T R_Z & R_X^T T_Z \\ T_Z^T R_X & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (\text{C.4})$$

where we have used $R_Z^T R_X = 0$. This can be justified by observing that the sub-matrix $S_R = \begin{bmatrix} R_X \\ R_Z \end{bmatrix}$ is a stabilizer matrix of a bigger vector subspace (in which the stabilizer vector space described by the matrix S is contained) and therefore we have that

$$\begin{bmatrix} S_Z^T & S_X^T \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} S_Z \\ S_X \end{bmatrix} = \begin{bmatrix} S_Z^T S_X \\ S_X^T S_Z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (\text{C.5})$$

Therefore we have that for all stabilizer matrices $S_Z^T S_X = S_X^T S_Z = 0$, which implies $R_Z^T R_X = 0$ and that gives us the relation $T_Z^T R_X = R_Z^T T_X = 0$. Because S has rank n (as it is a stabilizer) and by construction R_X has rank k , this implies that T_Z has full rank and it's therefore the orthogonal complement of R_X . As R_X has rank k , it follows that it is an invertible $k \times k$ sub-matrix. Without loss of generality we can write

$$R_X = \begin{bmatrix} R_X^1 \\ R_X^2 \end{bmatrix}, \quad (\text{C.6})$$

where we assume R_X^1 to be invertible (note that when transforming this constructive proof into an algorithm we cannot make this assumption, see section 3.4 for more details). We can then write the full stabilizer matrix in blocks:

$$S = \begin{bmatrix} R_Z^1 & T_Z^1 \\ R_Z^2 & \circled{T_Z^2} \\ \circled{R_X^1} & 0 \\ R_X^2 & 0 \end{bmatrix}, \quad (\text{C.7})$$

where we have circled the invertible sub-matrices: R_X^1 is invertible by construction, T_Z^2 is also invertible, as a consequence of $T_Z^T R_X = 0$. We can prove this as follows:

Suppose that there exist a vector x such that $(T_Z^2)^T x = 0$. We can define the $n \times n$ vector $v = (0, 0, \dots, x)$ which would satisfy $S_Z^T v = 0$ and therefore $v = R_X y$ for some $y \in \mathbf{Z}_2^k$:

$$\begin{bmatrix} 0 \\ x \end{bmatrix} = \begin{bmatrix} R_X^1 \\ R_X^2 \end{bmatrix} y = \begin{bmatrix} R_X^1 y \\ R_X^2 y \end{bmatrix}. \quad (\text{C.8})$$

Since R_X^1 is invertible by construction, $R_X^1 y = 0$ implies $y = 0 \Rightarrow x = 0$, which proves the invertibility of T_Z^2 . We want to have a lower block (the S_X block invertible). As we can see from the structure of S in equation (C.7), exchanging the lower parts of the S_X and S_Z blocks would accomplish that, as both R_X^1 and T_Z^2 are full rank. Logically, this exchange can be achieved simply by applying the Hadamard operation on the qubits represented by the matrix

rows of T_Z^2 and R_X^2 . ■

C.2 Assessing local Clifford equivalence

Given two stabilizer states $S_1 = \begin{bmatrix} S_{Z^1} \\ S_{X^1} \end{bmatrix}$ and $S_2 = \begin{bmatrix} S_{Z^2} \\ S_{X^2} \end{bmatrix}$, they are LC-equivalent iff there exist an operator Q such that $Q \cdot S_1 = S_2$ up to a basis change [122]. We can use this to write an equation that will allow us to find which LC operations, if any, can be applied to transform one state into another.

The LC equivalence up to a basis change can be written as

$$Q \cdot S_1 \cdot R = S_2 \Rightarrow S_1^T \cdot Q^T \cdot \mathbb{P} \cdot Q \cdot S_1 \cdot R = S_1^T \cdot Q^T \cdot \mathbb{P} \cdot S_2 \Rightarrow S_1^T \cdot Q^T \cdot \mathbb{P} \cdot S_2 = 0. \quad (\text{C.9})$$

Writing this relation explicitly as a function of the matrix elements $z_{ij}^1, z_{ij}^2, x_{ij}^1, x_{ij}^2$ (which correspond to $S_{Z^1}, S_{Z^2}, S_{X^1}, S_{X^2}$ respectively) we have:

$$\sum z_{ij}^1 A_j x_{kj}^2 + x_{ij}^1 B_j x_{kj}^2 + z_{ij}^1 C_j z_{kj}^2 + x_{ij} D_j z_{kj}^2 = 0 \quad (\text{C.10})$$

with the constraint that $A_i D_i + B_i C_i = 1$ to insure invertibility of Q .

In order to establish if the two stabilizer states are LC-equivalent, the system of equations is solved. If there exists a solution, we will obtain the LC operation directly in the form of the Q matrix, otherwise the two states are not LC-equivalent.

APPENDIX D

VISUAL-CHP

The following description of the operations implemented in the Mathematica Visual-CHP code are of the form: **Operation**[*variables*], with the same syntaxes as it should be used in the code, followed by their description.

D.1 Internal functions

Destabilizer[*tableau*, *i*] Extracts Destabilizer *i* from the cluster represented by *tableau*.

Stabilizer[*tableau*, *i*] Extracts Stabilizer *i* from the cluster represented by *tableau*.

StabilizerSupport[*tableau*,*list*] Outputs stabilizers that have support on the set of qubits *list*.

RowSum[*tableau*, *h*, *i*] Internal function equivalent to multiplying two stabilizer (or destabilizer) generators together. Used to do measurements.

InvSubM[*matrix*] Internal function to find a sub-matrix with the same rank as the full *matrix*. Gaussian elimination is assumed to have been applied to the full matrix.

InvSubMAll[*matrix*] Internal function that finds all possible sub-matrices with the same rank as the full *matrix*. Gaussian elimination is assumed to have been applied to the full matrix. The output of this function is just all the possible combinations of columns that form a sub-matrix of rank *k* together with the first *k* rows.

D.2 Cluster building commands

PlusProductState[*n*] Generates a product state of *n* qubits in $|+\rangle$.

ZeroProductState[*n*] Generates a product state of *n* qubits in $|0\rangle$.

LinearC[*n*] Generates a linear Cluster of *n* qubits.

ClosedC[*n*] Generates a linear Cluster of *n* qubits with periodic boundary conditions .

RectangularC[*n*, *m*] Generates a rectangular Cluster of *n* rows and *m* columns.

TriangularC[*n*, *m*] Generates a triangular Cluster of *n* rows and *m* columns.

GHZ[*n*] Creates a GHZ state of *n* qubits .

FromAdj[*matrix*] Builds a cluster state from an adjacency matrix. It eliminates self connecting nodes. If the matrix is not symmetric, it will print a warning and not build the tableau.

FromStabilizers[*input*] Obtains cluster states from a list of stabilizers in the form {“ooooo”, “ooooo”, etc} where o can be x, X, y, Y, z, Z, i, I. It does not immediately write the destabilizers, first the graph should be defined using **ToGraph[]** or **ChooseG[]**.

FromEditedC[*x*] To be used straight after *x*=**PrintGraphInteractive[]**, the function will automatically write the cluster in tableau form from the Graph Editor data.

DrawC[] Opens up a graph editor that allows you to draw any cluster state you want. When the cluster is finished, close the editor and the function will automatically write the cluster in tableau form from the Graph Editor data.

D.3 Quantum Operations

Hadamard[*tableau, qb*] Applies a Hadamard gate to qubits *qb* in cluster *tableau*. *qb* can be a single qubit or a list of qubits.

Phase[*tableau, qb*] Applies a Phase gate to qubits *qb* in cluster *tableau*. *qb* can be a single qubit or a list of qubits.

Cnot[*tableau, {control, target}*] Applies CNOT gate to qubits *control* and *target* in cluster *tableau*. It also accepts a list of control-target pairs.

Cz[*tableau, {qubit1, qubit2}*] Applies CZ gate to qubits *qubit1* and *qubit2* in cluster *tableau*. *qb* can be a single qubit or a list of qubits. It also accepts a list of qubit pairs.

MeasureZ[*tableau, qubit*] Measures Z on qubit *qubit* of cluster *tableau*. States if all the stabilizers commute with the measurement or not (measurement is random or determinate) and gives back the measurement result in print as well as updating the state. The qubit that is measured out remains in the description of the state, there is no renaming of the qubits.

MeasureX[*tableau, qubit*] Measures X on qubit *qubit* of cluster *tableau*. States if all the stabilizers commute with the measurement or not (measurement is random or determinate) and gives back the measurement result in print as well as updating the state. The qubit that is measured out remains in the description of the state, there is no renaming of the qubits.

MeasureY[*tableau, qubit*] Measures Y on qubit *qubit* of cluster *tableau*. States if all the stabilizers commute with the measurement or not (measurement is random or determinate) and gives back the measurement result in print as well as updating the state. The qubit that is measured out remains in the description of the state, there is no renaming of the

qubits. In the case of the Y measurement, the outcome is not relevant as both Y and $-Y$ will give the same outcome (because in the code we can't make the operators X and Z anti-commute, that is taken into account in the update rules for unitary gates but not in this measurement).

BitFlip[*tableau*, *qubit*] Implements a bit flip (X) on *qubit* in the cluster represented by *tableau*.

PhaseFlip[*tableau*, *qubit*] Implements a phase flip (Z) on *qubit* in the cluster represented by *tableau*.

BPFlip[*tableau*, *qubit*] Implements a bit and phase flip ($\bar{Y} = XZ$) on *qubit* in the cluster represented by *tableau*.

GateQ[*tableau*, *qubit*, *rules*] Implements a single qubit gate on *qubit* from cluster represented by *tableau*, given by : $\{X, Y, Z\} \rightarrow \text{rules}$, where *rules* is of the form $\{\pm o, \pm o, \pm o\}$ where *o* can be x, X, y, Y, z, Z, i, I .

GateQQ[*tableau*, *qubit1*, *qubits2*, *rules1*, *rules2*] Implements an arbitrary two qubit gate on *qubit1* and *qubit2* from cluster represented by *tableau*, given by the rule lists: $\{\mathbf{1}_i \otimes \sigma_{X_j}, \mathbf{1}_i \otimes \sigma_{Y_j}, \mathbf{1}_i \otimes \sigma_{Z_j}\} \rightarrow \text{rules1}$ and $\{\sigma_{X_i} \otimes \mathbf{1}_j, \sigma_{Y_i} \otimes \mathbf{1}_j, \sigma_{Z_i} \otimes \mathbf{1}_j\} \rightarrow \text{rules2}$, where *rules1* and *rules2* are of the form $\{\{\pm o, \pm o\}, \{\pm o, \pm o\}, \{\pm o, \pm o\}\}$ where *o* can be x, X, y, Y, z, Z, i, I .

FusionIdS[*tableau*, *qubit1*, *qubit2*] Implements the successful Fusion Gate Type-I in a deterministic way on qubits *qubit1* and *qubit2* of the cluster represented by *tableau*.

FusionIdF[*tableau*, *qubit1*, *qubit2*] Implements the failed Fusion Gate Type-I in a deterministic way on qubits *qubit1* and *qubit2* of the cluster represented by *tableau*.

FusionI[*tableau*, *qubit1*, *qubit2*] Implements the Fusion Gate Type-II with success probability *p* on qubits *qubit1* and *qubit2* of the cluster represented by *tableau*.

FusionIIDs[*tableau*, *qubit1*, *qubit2*] Implements the successful Fusion Gate Type-II in a deterministic way on qubits *qubit1* and *qubit2* of the cluster represented by *tableau*.

FusionIIDF[*tableau*, *qubit1*, *qubit2*] Implements the failed Fusion Gate Type-II in a deterministic way on qubits *qubit1* and *qubit2* of the cluster represented by *tableau*.

FusionII[*tableau*, *qubit1*, *qubit2*] Implements the Fusion Gate Type-II with success probability *p* on qubits *qubit1* and *qubit2* of the cluster represented by *tableau*.

Swap[*tableau*, *qubit1*, *qubit2*] Implements a SWAP gate between the qubits *qubit1* and *qubit2* of the cluster described by *tableau*.

D.4 Cluster Operations

JoinC[*cluster1*, *cluster2*] Joins 2 clusters (represents them as one tableau, there are no links between them, the CZ or fusion gates would have to be applied later). Naming of the qubits starts in cluster 1 and carries on in cluster 2.

EliminateQ[*tableau*] Eliminates unentangled qubits from a cluster (it will only eliminate them if they are in a product state with the rest of the cluster). Useful after measurements to speed up computation.

PosG[*tableau*] Shows the possible groups of qubits on which the Hadamards can be applied in order to put the cluster described by *tableau* in graph form. If the choice is unique, it says so in a message.

ChooseG[*tableau*, *list*] Transforms the cluster state given by *tableau* to a graph state allowing to choose onto which qubits the Hadamard gates are applied. These qubit are inputed as a *list* of the form $\{q1, q2, q3, \dots\}$.

ToGraph[*tableau*] Takes a cluster given by *tableau* and rewrites the stabilizers in graph state form. If the choice of graph is not unique and it matters for subsequent computations **PosG[]** and **ChooseG[]** should be used instead.

LComp[*tableau*, *nodes*] Performs local complementation on a list of nodes $nodes = \{n1, n2, \dots\}$ of cluster *tableau*.

D.5 Output

PrintStabilizers[*tableau*] Obtains the stabilizers in the cluster *tableau* as a string of X_i s and Z_j s.

PrintDestabilizers[*tableau*] Obtains the destabilizers in the cluster *tableau* as a string of X_i s and Z_j s.

GetAdj[*tableau*] Obtains the adjacency matrix that describes the cluster *tableau* in graph form. If the cluster is not in graph form it outputs a warning and the function **ToGraph[]** or **ChooseG[]** should be used.

PrintGraph[*tableau*] Plots the graph corresponding to a particular cluster state given by *tableau*. The cluster state should be in graph form, if not it will print a warning saying so. A warning saying that self loops are not displayed will appear if a stabilizer has the operator Y instead of X.

PrintAllGraph[*tableau*] Prints all graphs that can correspond to the cluster state given by *tableau*, depending on which qubits are the Hadamard gates applied. A warning saying that self loops are not displayed will appear if a stabilizer has the operator Y instead of X.

PrintGraphInteractive[*tableau*] Plots the graph corresponding to a particular cluster state given by *tableau* in an interactive editor. Nodes can be moved around, added and deleted. Edges can be added and deleted as well. If we want to use the edited graph, call **FromEditedC[]** straight afterwards. A warning saying that self loops are not displayed will appear if a stabilizer has the operator Y instead of X.

APPENDIX E

BOSONIC SIMULATOR

We briefly describe the Mathematica-based BOSONICSIMULATOR which has been used to obtain all the results in chapter 4.

- States are represented by the bosonic creation operators following the same notation as in chapter 4: $h_i^n v_j^m$.
- Linear optical elements are described by the transformation rules on the creation operators. The most commonly used are:

- Beam-splitter of transmitivity η acting on modes a and b :

$$\text{BS}[a,b,\eta] = \{h_a \rightarrow \sqrt{\eta}h_a + i\sqrt{1-\eta}h_b, h_b \rightarrow \sqrt{\eta}h_a + i\sqrt{1-\eta}h_b, \quad (\text{E.1})$$

$$v_a \rightarrow \sqrt{\eta}v_a + i\sqrt{1-\eta}v_b, v_b \rightarrow \sqrt{\eta}v_a + i\sqrt{1-\eta}v_b\} \quad (\text{E.2})$$

- 45° Polarisation rotator :

$$\text{PolRot45}[a] = \left\{ h_a \rightarrow \frac{h_a + v_a}{\sqrt{2}}, v_a \rightarrow \frac{h_a - v_a}{\sqrt{2}} \right\} \quad (\text{E.3})$$

- PBS:

$$\text{PBS}[a,b] = \{h_a \rightarrow h_a, h_b \rightarrow h_b, v_a \rightarrow v_b, v_b \rightarrow v_a\} \quad (\text{E.4})$$

- The action of a series of linear optical elements ($\{1, \dots, n\}$) on the optical modes is simulated by applying the rules to the original state (by using Mathematica's rule transformation engine): state $\backslash. op_1 \backslash. op_2 \dots$ etc, where op_i represents the transformation rules of the operator applied in i^{th} position of the series. At the end of the sequence it is usually convenient to expand all the final terms. For example, the application of two polarisation rotators and a PBS on two horizontal photons is simulated as:

$$h_a h_b \backslash. \text{PolRot45}[a] \backslash. \text{PolRot45}[b] \backslash. \text{PBS}[a,b] \rightarrow \frac{h_a h_b}{2} + \frac{v_a v_b}{2} + \frac{h_a v_a}{2} + \frac{h_b v_b}{2} \quad (\text{E.5})$$

- Several functions allow to post-select states based on variable such as number of photons on a mode, number of horizontally or vertically polarised photons in total, number of photons spread over certain modes, particular patterns, etc.

- The probability of each of the terms in the final state (or of a state after post-selection) is calculated by using a *norm* operation that accounts for the multiplicative factors of the creation operators, i.e. each operator h^i has a weight of $\sqrt{i!}$.

APPENDIX F

DETAILS ON 3-GHZ AND 4-GHZ STATE GENERATION

F.1 Boosted 3-GHZ generation from Bell pairs

We take the ballistic circuit to generate a 3-GHZ from single photon and rewrite it so that the input are Bell pairs. This circuit has 3 Bell pairs as input and perform a Type-I fusion gate and a Type-II fusion gate on 3 of the input qubits (one from each Bell pair). This gate succeeds with 25% probability (each fusion gate works with 50% probability). We use a Grice-type scheme to boost the Type-II fusion gate which will now work with 75% probability.

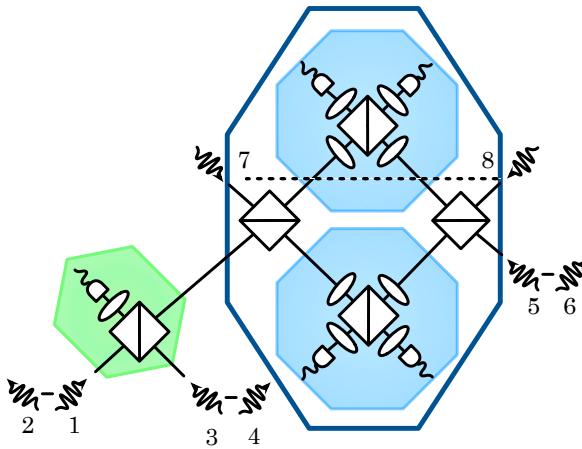


Figure F.1: Optical circuit to generate a 3 qubits GHZ ballistically from Bell pairs with 37.5% probability. The dashed lines between pairs of photons indicate that they belong to the same Bell pair.

The boosted 3-GHZ ballistic circuit is shown in figure F.1. The input for this circuit are 4 Bell pairs (the three original Bell pairs and the pair used to boost the success probability of the Type-II fusion gate). The success probability is boosted to 37.5%. In the non-boosted version, the procedure succeeded when only one photon was detected in each mode. In the boosted version, the postselection patterns that indicate success are increased, but in some of the cases the output state is a GHZ state to which a Pauli rotation has been applied. However, given a particular pattern measured, we can always know which Pauli rotation has been applied to the state, and therefore it can be corrected.

In this circuit, we pair the photons in Bell pairs as 1 & 2, 3 & 4, 5 & 6, 7 & 8, where 7 & 8 is the pair used to boost and 1, 3, 5, 7 and 8 are the photons measured (therefore the GHZ will be located in photons 2, 4 and 6). The notation n_i means the number of photons in that mode or with that polarisation (for example n_V is the total number of photons with vertical polarisation, n_2 is the number of photons in mode 2 and n_{h_1} is the number of photons in mode 1 that are horizontally polarised). The post-selections that indicate a successful procedure are:

- $(n_1 + n_5) = 2 \& n_1 + n_7 + n_{h_3}$ odd $\Rightarrow (h_2 h_4 h_6 + v_2 v_4 v_6)/\sqrt{2}$ with probability 6.25%
- $(n_1 + n_5) = 2 \& n_1 + n_7 + n_{h_3}$ even $\Rightarrow (h_2 h_4 h_6 - v_2 v_4 v_6)/\sqrt{2}$ with probability 6.25%
- $(n_1 + n_5)$ odd & n_H even $\Rightarrow (h_2 h_4 v_6 + v_2 v_4 h_6)/\sqrt{2}$ with probability 12.5%
- $(n_1 + n_5)$ odd & n_H odd $\Rightarrow (h_2 h_4 v_6 - v_2 v_4 h_6)/\sqrt{2}$ with probability 12.5%

Therefore, whenever we obtain a GHZ out of our circuit, $\frac{1}{6}$ of the time we obtain $\frac{|000\rangle+|111\rangle}{\sqrt{2}}$, $\frac{1}{6}$ of the time we obtain $\frac{|000\rangle-|111\rangle}{\sqrt{2}}$, $\frac{1}{3}$ of the time we obtain $\frac{|001\rangle+|110\rangle}{\sqrt{2}}$ and $\frac{1}{3}$ of the time we obtain $\frac{|001\rangle-|110\rangle}{\sqrt{2}}$.

We can compare the boosted and non-boosted versions of this circuit by comparing the number of Bell pairs needed on average to create a single 3-GHZ state.

- Not Boosting: The overall procedure has a success probability of 25%. On average we need to repeat the procedure 4 times to obtain a 3-GHZ state. Each procedure needs 3 Bell pairs to fuse and none to boost. Therefore we need 3 Bell pairs per procedure and 12 Bell pairs in total per 3-GHZ state.
- Boosting: The overall procedure has a success probability of 37.5%. On average we need to repeat the procedure 2.66 times to obtain a 3-GHZ state. Each procedure needs 3 Bell pairs to fuse and 1 Bell pair to boost. Therefore we need 4 Bell pairs per procedure and 10.67 Bell pairs in total per 4-GHZ state.

We can see that the boosted version of this circuit is more resource efficient and we can say that 1 3-GHZ state is equivalent to 11 Bell pairs.

F.2 Boosted 4-GHZ generation from Bell pairs

We take the ballistic circuit to generate a 4-GHZ from single photons and rewrite it so that the input are Bell pairs. This circuit has 4 Bell pairs as input and performs two Type-II fusion gates on 4 of the input qubits (one from each Bell pair). This gate succeeds with 12.5% probability. We use a Grice-type scheme to boost the Type-II fusion gates which will now work with 75% probability rather than 50% as in the non-boosted version. The circuit has an extra beam-splitter that we cannot boost and that multiplies the success probability by $\frac{1}{2}$.

The boosted 4-GHZ ballistic circuit is shown in figure F.2. The input for this circuit are 6 Bell pairs (the four original Bell pairs and the two pairs used to boost the success probability of the Type-II fusion gate). The success probability is boosted to $\sim 28\%$. In the non-boosted version, the procedure succeeded when only one photon was detected in each mode. In the

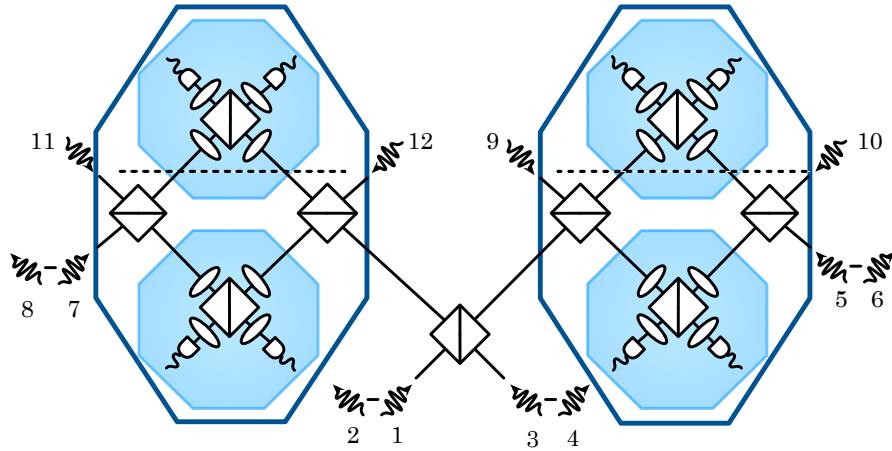


Figure F.2: Optical circuit to generate a 4 qubits GHZ ballistically from Bell pairs with 28.125% probability.

boosted version, the post-selection patterns that indicate success are increased, but in some of the cases the output state is a GHZ state to which a Pauli rotation has been applied. However, given a particular pattern measured, we can always know which Pauli rotation has been applied to the state, and therefore it can be corrected.

In this circuit, we pair the photons in Bell pairs as 1 & 2, 3 & 4, 5 & 6, 7 & 8, 9 & 10, 11 & 12, where 9 & 10 and 11 & 12 is the pair used to boost and 1, 3, 5, 7, 9, 10, 11 and 12 are the photons measured (therefore the GHZ will be located in photons 2, 4 6 and 8). The post-selections that indicate a successful procedure are:

- $(n_1 + n_5) = 2 \& (n_3 + n_7) = 2 \& (n_1 + n_3 + n_9 + n_{12})$ even $\Rightarrow (h_2 h_4 h_6 h_8 + v_2 v_4 v_6 v_8)/\sqrt{2}$ with probability 1.562%
- $(n_1 + n_5) = 2 \& (n_3 + n_7) = 2 \& (n_1 + n_3 + n_9 + n_{12})$ odd $\Rightarrow (h_2 h_4 h_6 h_8 - v_2 v_4 v_6 v_8)/\sqrt{2}$ with probability 1.562%
- $(n_1 + n_5)$ odd & $(n_3 + n_7)$ odd & n_H even $\Rightarrow (h_2 h_4 v_6 v_8 + v_2 v_4 h_6 h_8)/\sqrt{2}$ with probability 6.25%
- $(n_1 + n_5)$ odd & $(n_3 + n_7)$ odd & n_H odd $\Rightarrow (h_2 h_4 v_6 v_8 - v_2 v_4 h_6 h_8)/\sqrt{2}$ with probability 6.25%
- $(n_1 + n_5) = 2 \& (n_3 + n_7)$ odd & $(n_1 + n_9 + n_{h_3} + n_{h_7} + n_{h_{11}} + n_{h_{12}})$ odd $\Rightarrow (h_2 h_4 h_6 v_8 + v_2 v_4 v_6 h_8)/\sqrt{2}$ with probability 3.125%
- $(n_1 + n_5) = 2 \& (n_3 + n_7)$ odd & $(n_1 + n_9 + n_{h_3} + n_{h_7} + n_{h_{11}} + n_{h_{12}})$ even $\Rightarrow (h_2 h_4 h_6 v_8 - v_2 v_4 v_6 h_8)/\sqrt{2}$ with probability 3.125%
- $(n_1 + n_5)$ odd & $(n_3 + n_7) = 2 \& (n_3 + n_{12} + n_{h_1} + n_{h_5} + n_{h_9} + n_{h_{10}})$ odd $\Rightarrow (h_2 h_4 v_6 h_8 + v_2 v_4 h_6 v_8)/\sqrt{2}$ with probability 3.125%
- $(n_1 + n_5)$ odd & $(n_3 + n_7) = 2 \& (n_3 + n_{12} + n_{h_1} + n_{h_5} + n_{h_9} + n_{h_{10}})$ even $\Rightarrow (h_2 h_4 v_6 h_8 - v_2 v_4 h_6 v_8)/\sqrt{2}$ with probability 3.125%

Therefore, whenever we obtain a GHZ out of our circuit, $\frac{1}{18}$ of the time we obtain $\frac{|0000\rangle+|1111\rangle}{\sqrt{2}}$, $\frac{1}{18}$ of the time we obtain $\frac{|0000\rangle-|1111\rangle}{\sqrt{2}}$, $\frac{2}{9}$ of the time we obtain $\frac{|0011\rangle+|1100\rangle}{\sqrt{2}}$, $\frac{2}{9}$ of the time we obtain $\frac{|0011\rangle-|1100\rangle}{\sqrt{2}}$, $\frac{1}{9}$ of the time we obtain $\frac{|0001\rangle+|1110\rangle}{\sqrt{2}}$, $\frac{1}{9}$ of the time we obtain $\frac{|0001\rangle-|1110\rangle}{\sqrt{2}}$, $\frac{1}{9}$ of the time we obtain $\frac{|0010\rangle+|1101\rangle}{\sqrt{2}}$ and $\frac{1}{9}$ of the time we obtain $\frac{|0010\rangle-|1101\rangle}{\sqrt{2}}$.

We can compare the boosted and non-boosted versions of this circuit by comparing the number of Bell pairs needed on average to create a single 4-GHZ state.

- Not Boosting: The overall procedure has a success probability of 12.5%. On average we need to repeat the procedure 8 times to obtain a 4-GHZ state. Each procedure needs 4 Bell pairs to fuse and none to boost. Therefore we need 4 Bell pairs per procedure and 32 Bell pairs in total per 4-GHZ state.
- Boosting: The overall procedure has a success probability of 28.125%. On average we need to repeat the procedure 3.56 times to obtain a 4-GHZ state. Each procedure needs 4 Bell pairs to fuse and 2 Bell pairs to boost. Therefore we need 6 Bell pairs per procedure and 21.33 Bell pairs in total per 4-GHZ state.

We can see that the boosted version of this circuit is more resource efficient and we can say that 1 4-GHZ state is equivalent to 22 Bell pairs.

F.3 3-GHZ generation from probabilistic SPDC sources

In section 4.5.3, chapter 4 we mentioned that one of the advantages of using the probabilistic Bell pairs emitted from the SPDC sources directly to generate 3-GHZ states is that even when some of the sources don't emit a Bell pair, some measurement outcomes yield (upon detection of vacuum on one of the output modes) a smaller entangled state. Here we show all the possible detection patterns that only measure 2 photons (instead of the three that would herald a 3-GHZ state) for the case when one of the sources has not produced a Bell pair. In table F.1 we show the outcome of the circuit when one of the sources didn't fire and a particular pattern was detected. We can see that in the case where source 3 does not fire, the generated state is a Bell pair for many of the detection outcomes. The crucial point is that this Bell pair is on modes 2 and 4, whereas the output state in the cases where source 2 or 1 does not fire (which is never a Bell pair) always has support on mode 6. Therefore, heralding vacuum on mode 6 when only 2 photons have been detected in modes 1, 3 and 5 allows to herald a Bell pair in modes 2 and 4.

Detection pattern	Source 3	Source 2	Source 1
h_1h_3	$\frac{1}{4\sqrt{2}}(h_2h_4 + v_2v_4)$	$\frac{1}{4\sqrt{2}}v_2(h_6 + v_6)$	$\frac{-1}{4\sqrt{2}}h_4(h_6 + v_6)$
h_1h_5	$\frac{1}{4\sqrt{2}}(-h_2h_4 + v_2v_4)$	$\frac{1}{4\sqrt{2}}v_2(-h_6 + v_6)$	$\frac{1}{4\sqrt{2}}h_4(h_6 - v_6)$
h_1v_1	0	0	0
h_1v_3	$\frac{1}{4\sqrt{2}}(h_2h_4 + v_2v_4)$	$\frac{-1}{4\sqrt{2}}v_2(h_6 + v_6)$	$\frac{1}{4\sqrt{2}}h_4(h_6 + v_6)$
h_1v_5	$\frac{1}{4\sqrt{2}}(h_2h_4 + -v_2v_4)$	$\frac{1}{4\sqrt{2}}v_2(h_6 + v_6)$	$\frac{1}{4\sqrt{2}}h_4(h_6 - v_6)$
h_3h_5	0	$\frac{h_2h_6}{4}$	$\frac{v_4v_6}{4}$
h_3v_1	$\frac{1}{4\sqrt{2}}(-h_2h_4 + v_2v_4)$	$\frac{1}{4\sqrt{2}}v_2(h_6 + v_6)$	$\frac{1}{4\sqrt{2}}h_4(h_6 + v_6)$
h_3h_3	$\frac{-h_2v_4}{4}$	0	0
h_3v_5	0	$\frac{-h_2v_6}{4}$	$\frac{-v_4h_6}{4}$
h_5v_1	$\frac{1}{4\sqrt{2}}(h_2h_4 + v_2v_4)$	$\frac{1}{4\sqrt{2}}v_2(-h_6 + v_6)$	$\frac{1}{4\sqrt{2}}h_4(-h_6 + v_6)$
h_5v_3	0	$\frac{-h_2v_6}{4}$	$\frac{-v_4h_6}{4}$
h_5v_5	$\frac{-h_2v_4}{4}$	0	0
v_1v_3	$\frac{1}{4\sqrt{2}}(-h_2h_4 + v_2v_4)$	$\frac{-1}{4\sqrt{2}}v_2(h_6 + v_6)$	$\frac{1}{4\sqrt{2}}h_4(h_6 + v_6)$
v_1v_5	$\frac{-1}{4\sqrt{2}}(h_2h_4 + v_2v_4)$	$\frac{1}{4\sqrt{2}}v_2(-h_6 + v_6)$	$\frac{1}{4\sqrt{2}}h_4(-h_6 + v_6)$
v_1v_5	0	$\frac{h_2h_6}{4}$	$\frac{v_4v_6}{4}$

Table F.1: Output of the 3-GHZ generation circuit portrayed in figure 4.16 when only two photons have been detected in modes 1, 3 and 5. The different outputs are classified according to the detection pattern and which was the source that did not produce a Bell pair.

APPENDIX G

SWITCH LOSS

G.1 Switch loss calculations in multiplexed GHZ generators

We calculate the amount of loss per switch that we can tolerate when generating GHZ states and multiplexing using a log-tree scheme, given the tolerable loss rate in the percolation scheme presented in chapter 5. We assume deterministic on-demand single photon sources and lossless passive elements (as justified in chapter 2). Using the percolation scheme presented in chapter 5, we can tolerate loss in the individual photons of the 3-GHZ state up to $p_l \sim 1.5\%$ using homogeneous MUX and up to $p_l \sim 2.9\%$ using RMUX. Using the multiplexing stage we can accept all states that have at least one photon i.e. they are not the vacuum. That means that if the loss rate per switch is γ_{sw} , and the photon goes through m switches, the probability of not having been lost in the switching process is $(1 - \gamma_{sw})^m$. The probability of having been lost at any stage of the switching process is $1 - (1 - \gamma_{sw})^m$ and we want this number to be smaller than the total loss per photon that we can tolerate, $p_l \leq 1 - (1 - \gamma_{sw})^m$. Note that as in the percolation scheme the loss is calculated per photon and not per GHZ state, that is the same calculation we do here, we do not require the GHZ state to have a certain loss probability but rather the individual photons. The calculations on this appendix are repetitive, but they allow a better understanding of the efficiency of each scheme.

G.1.1 Approach A: Multiplexing 3 photon GHZ generators from single photons

In this first approach there is only one stage of multiplexing after the 3-GHZ interferometer. Single photons are obtained on demand from the single photon sources and passed through the 3-GHZ state generator, which is time or spatially multiplexed. Each 3-GHZ state generator consumes 6 single photons and produces a 3-GHZ state with probability 1/32.

We want to generate a 3-GHZ state from single photons with probability for example $p_s \geq 0.95$, that means we need to multiplex k times where k is given by

$$1 - \left(1 - \frac{1}{32}\right)^k \geq 0.95 \quad \Rightarrow \quad k = 95. \quad (\text{G.1})$$

We assume the switchboard is made out of a log-tree of 2×2 switches (such as described in chapter 4), the number of switches required for a multiplexing of k events is given by $m =$

$\lceil \log_2 k \rceil + 1$. In this case

$$m = \lceil \log_2 95 \rceil + 1 = 8. \quad (\text{G.2})$$

As stated in the preliminaries we require that

$$p_l \leq 1 - (1 - \gamma_{sw})^m \quad (\text{G.3})$$

where $p_{l_1} = 0.015$, $p_{l_2} = 0.029$ and $m = 8$. Therefore

$$0.015 \leq 1 - (1 - \gamma_{sw})^8 \Rightarrow \gamma_{sw} \leq 0.19\%, \quad (\text{G.4})$$

$$0.029 \leq 1 - (1 - \gamma_{sw})^8 \Rightarrow \gamma_{sw} \leq 0.37\%. \quad (\text{G.5})$$

In figure G.1 we plot the results of this same calculation for values of $p \in (0.9, 1)$.

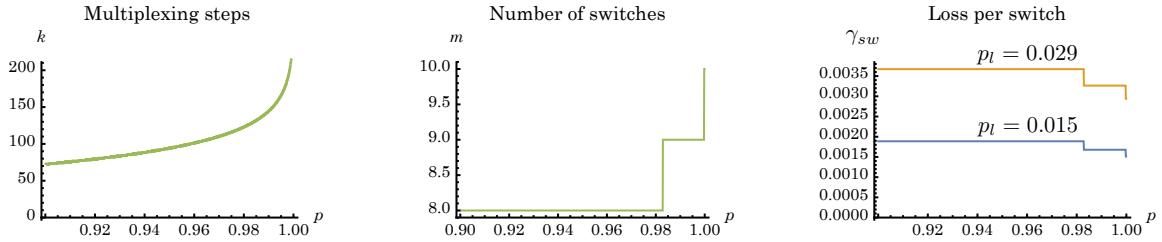


Figure G.1: Approach A: Multiplexing steps, number of switches and loss per switch as a function of the probability of obtaining a GHZ state out of the multiplexed process. The number of multiplexing stages and number of switches is the same for both values of p_l . We can see how the RMUX scheme (yellow), which gives a tolerable loss rate of $p_{l_2} = 0.029$ permits less stringent specifications on the individual switches.

G.1.2 Approach B: Multiplexing 3-GHZ generators from Bell Pairs

In this approach we assume we can obtain Bell pairs on demand from post selecting higher order terms of an SPDC source or from a quantum dot source. There is only one stage of multiplexing after the 3-GHZ interferometer. We use the boosted version of the 3-GHZ generator, thus each 3-GHZ state generator consumes 4 Bell pairs and produces a 3-GHZ state with probability 37.5%.

We want to generate a 3-GHZ state from Bell Pairs with probability for example $p_s \geq 0.95$, that means we need to multiplex k times where k is given by

$$1 - (1 - 0.375)^k \geq 0.95 \Rightarrow k = 7. \quad (\text{G.6})$$

We assume the switchboard is made out of a log-tree of 2×2 switches (such as described in chapter 4), the number of switches required for a multiplexing of k events is given by $m = \lceil \log_2 k \rceil + 1$. In this case

$$m = \lceil \log_2 7 \rceil + 1 = 4. \quad (\text{G.7})$$

As stated in the preliminaries we require that

$$p_l \leq 1 - (1 - \gamma_{sw})^m \quad (\text{G.8})$$

where $p_{l_1} = 0.015$, $p_{l_2} = 0.029$ and $m = 4$. Therefore

$$0.015 \leq 1 - (1 - \gamma_{sw})^4 \Rightarrow \gamma_{sw} \leq 0.38\%, \quad (\text{G.9})$$

$$0.029 \leq 1 - (1 - \gamma_{sw})^4 \Rightarrow \gamma_{sw} \leq 0.73\%. \quad (\text{G.10})$$

In figure G.2 we plot the results of this same calculation for values of $p \in (0.9, 1)$.

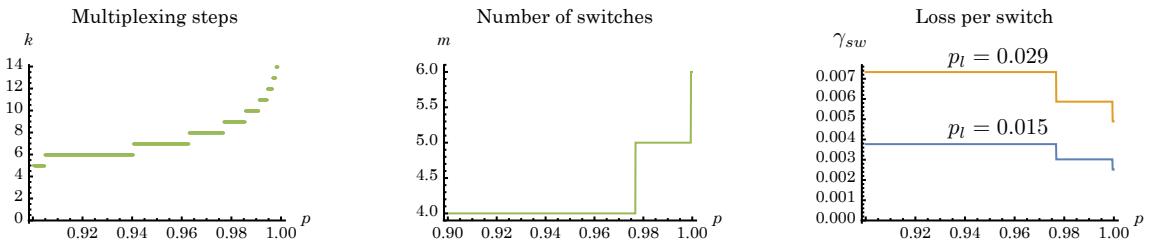


Figure G.2: Approach B: Multiplexing steps, number of switches and loss per switch as a function of the probability of obtaining a GHZ state out of the multiplexed process. The number of multiplexing stages and number of switches is the same for both values of p_l . We can see how the RMUX scheme (yellow), which gives a tolerable loss rate of $p_{l_2} = 0.029$ permits less stringent specifications on the individual switches.

G.1.3 Approach C: Multiplexing Bell pair generators from single photons and 3-GHZ generators from Bell Pairs

In this approach we have two stages of multiplexing. First we generate Bell pairs from single photons, multiplex, then we use the Bell pairs to create 3-GHZ states and multiplex before entering the percolation scheme. Single photons are obtained on demand from the single photon sources and passed through the Bell pair generator, which is time or spatially multiplexed. Each Bell pair generator consumes 4 single photons and produces a Bell pair with probability $\frac{1}{4}$ if we allow for active switching within the Bell pair generator (one switch) or with probability $\frac{1}{8}$ if we don't allow for active switching. We will treat these two strategies separately, labelling them Approach C1 and C2 respectively.

Here the probability of having a successful GHZ out of the multiplexing in the end will be the product of the probability of having successfully multiplexed a 4 Bell pairs out of the Bell Pair generation, p_{BP} , and the probability of having successfully multiplexed the GHZ out of the GHZ generator from Bell Pairs, p_{GHZ} . Thus

$$p = p_{BP}^4 \cdot p_{GHZ}. \quad (\text{G.11})$$

Approach C1

The probability of generating a Bell Pair is $\frac{1}{4}$ and the circuit itself, before multiplexing, requires one switch. Say we want to generate a Bell Pair with probability $p_{BP} \geq 0.95$, that means we need to multiplex k_1 times where k_1 is given by

$$1 - (1 - \frac{1}{4})^{k_1} \geq 0.95 \quad \Rightarrow \quad k_1 = 11. \quad (\text{G.12})$$

We assume the switchboard is made out of a log-tree of 2×2 switches (such as described in chapter 4), the number of switches required for a multiplexing of k events is given by $m = \lceil \log_2 k \rceil + 1$. In this case

$$m_1 = \lceil \log_2 11 \rceil + 1 = 5. \quad (\text{G.13})$$

The next step is to create GHZ states from Bell Pairs, this step is exactly the same as the calculation in Approach B rewritten here for convenience.

We want to generate a 3-GHZ state from deterministic Bell Pairs with probability for example $p_{GHZ} \geq 0.95$, that means we need to multiplex k_2 times where k_2 is given by

$$1 - (1 - 0.375)^{k_2} \geq 0.95 \quad \Rightarrow \quad k_2 = 7. \quad (\text{G.14})$$

The number of switches required for a multiplexing of k_2 events is given by $m_2 = \lceil \log_2 k_2 \rceil + 1$. In this case

$$m_2 = \lceil \log_2 7 \rceil + 1 = 4. \quad (\text{G.15})$$

The total number of switches that a photon has to go through is $m = m_1 + m_2 + 1$, the probability of obtaining a GHZ state at the end of the process is given by $p = p_{BP}^4 \cdot p_{GHZ}$, which is the case of this example is 0.77.

We require that

$$p_l \leq 1 - (1 - \gamma_{sw})^m \quad (\text{G.16})$$

where $p_{l_1} = 0.015$, $p_{l_2} = 0.029$ and $m = 10$. Therefore

$$0.015 \leq 1 - (1 - \gamma_{sw})^{10} \quad \Rightarrow \quad \gamma_{sw} \leq 0.15\%, \quad (\text{G.17})$$

$$0.029 \leq 1 - (1 - \gamma_{sw})^{10} \quad \Rightarrow \quad \gamma_{sw} \leq 0.29\%. \quad (\text{G.18})$$

In figure G.3 we plot the results of this same calculation for values of $p \in (0.8, 1)$. As we can see in the figure, to obtain the same overall probability of creating a GHZ state, there are different choices of p_{GHZ} and p_{BP} that we can take, and it is possible to optimise the choice so as to maximize the amount of loss that we can tolerate per switch (or equivalently, minimising the number of switches that a photon has to pass through). In figure G.4 we show this optimal choice.

Approach C2

The probability of generating a Bell Pair is $\frac{3}{16}$ and the circuit itself, before multiplexing, doesn't require any switch. Say we want to generate a Bell Pair with probability $p_{BP} \geq 0.95$, that means

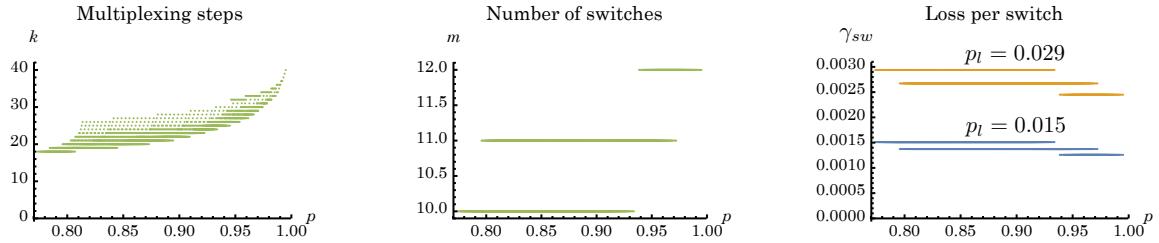


Figure G.3: Approach C1: intermediate stage where Bell pairs are generated with probability $\frac{1}{4}$. Multiplexing steps, number of switches and loss per switch as a function of the probability of obtaining a GHZ state out of the multiplexed process. The number of multiplexing stages and number of switches is the same for both values of p_l . We can see how the RMUX scheme (yellow), which gives a tolerable loss rate of $p_{l_2} = 0.029$ permits less stringent specifications on the individual switches.

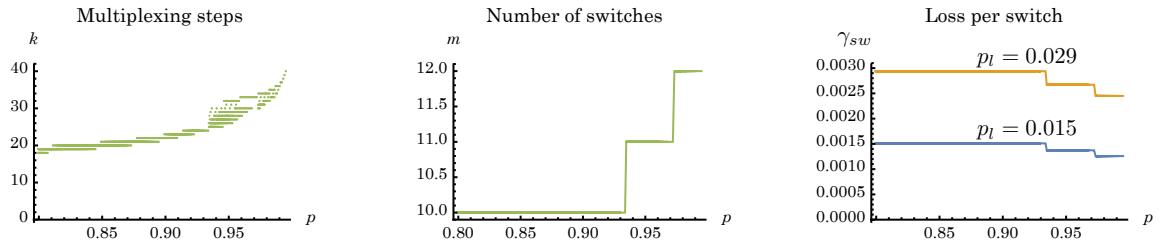


Figure G.4: Approach C1 optimised: Multiplexing steps, number of switches and loss per switch as a function of the probability of obtaining a GHZ state out of the multiplexed process. Optimised to minimise the number of switches. The number of multiplexing stages and number of switches is the same for both values of p_l . We can see how the RMUX scheme (yellow), which gives a tolerable loss rate of $p_{l_2} = 0.029$ permits less stringent specifications on the individual switches.

we need to multiplex k_1 times where k_1 is given by

$$1 - (1 - \frac{1}{8})^{k_1} \geq 0.95 \quad \Rightarrow \quad k_1 = 23. \quad (\text{G.19})$$

We assume the switchboard is made out of a log-tree of 2×2 switches (such as described in chapter 4), the number of switches required for a multiplexing of k events is given by $m = \lceil \log_2 k \rceil + 1$. In this case

$$m_1 = \lceil \log_2 11 \rceil + 1 = 6. \quad (\text{G.20})$$

The next step is to create GHZ states from Bell Pairs, this step is exactly the same as the calculation in Approach B rewritten here for convenience.

We want to generate a 3-GHZ state from deterministic Bell Pairs with probability for example $p_{GHZ} \geq 0.95$, that means we need to multiplex k_2 times where k_2 is given by

$$1 - (1 - 0.375)^{k_2} \geq 0.95 \quad \Rightarrow \quad k_2 = 7. \quad (\text{G.21})$$

The number of switches required for a multiplexing of k_2 events is given by $m_2 = \lceil \log_2 k_2 \rceil + 1$. In this case

$$m = \lceil \log_2 7 \rceil = 4. \quad (\text{G.22})$$

The total number of switches that a photon has to go through is $m = m_1 + m_2$, the probability of obtaining a GHZ state at the end of the process is given by $p = p_{BP} \cdot p_{GHZ}$, which is the case of this example is 0.9.

We require that

$$p_l \leq 1 - (1 - \gamma_{sw})^m \quad (\text{G.23})$$

where $p_{l_1} = 0.015$, $p_{l_2} = 0.029$ and $m = 10$. Therefore

$$0.015 \leq 1 - (1 - \gamma_{sw})^{10} \Rightarrow \gamma_{sw} \leq 0.15\%, \quad (\text{G.24})$$

$$0.029 \leq 1 - (1 - \gamma_{sw})^{10} \Rightarrow \gamma_{sw} \leq 0.29\%. \quad (\text{G.25})$$

In figure G.5 we plot the results of this same calculation for values of $p \in (0.8, 1)$. In figure

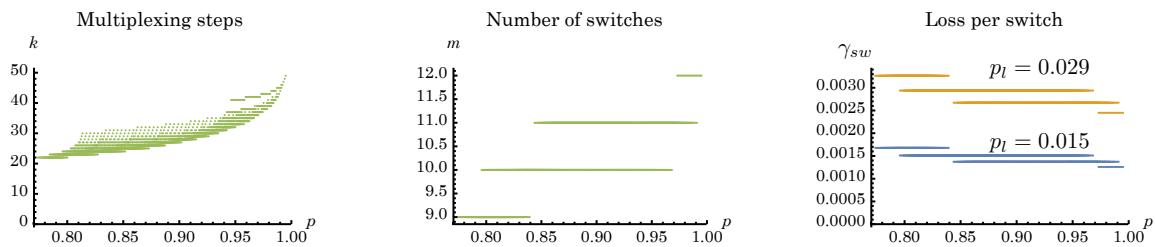


Figure G.5: Approach C2: Multiplexing steps, number of switches and loss per switch as a function of the probability of obtaining a GHZ state out of the multiplexed process. The number of multiplexing stages and number of switches is the same for both values of p_l . We can see how the RMUX scheme (yellow), which gives a tolerable loss rate of $p_{l_2} = 0.029$ permits less stringent specifications on the individual switches.

G.5 we plot the results of this same calculation for values of $p \in (0.8, 1)$. As in the approach C1 we can see that to obtain the same overall probability of creating a GHZ state, there are different choices of p_{GHZ} and p_{BP} that we can take, and it is possible to optimise the choice so as to maximise the amount of loss that we can tolerate per switch (or equivalently, minimising the number of switches that a photon has to pass through). In figure G.6 we show this optimal choice.

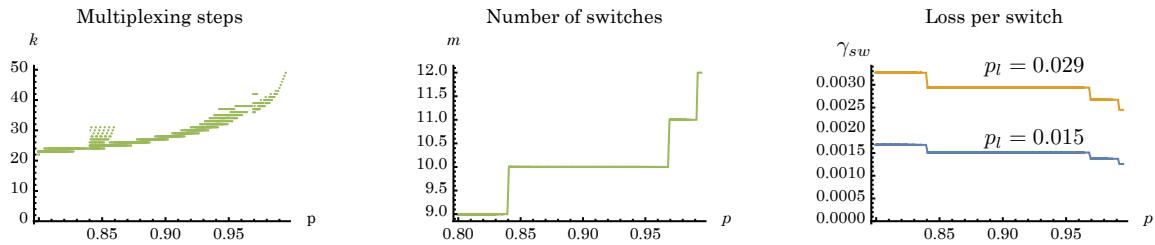


Figure G.6: Approach C2 optimised: Multiplexing steps, number of switches and loss per switch as a function of the probability of obtaining a GHZ state out of the multiplexed process. Optimised to minimise the number of switches. The number of multiplexing stages and number of switches is the same for both values of p_l . We can see how the RMUX scheme (yellow), which gives a tolerable loss rate of $p_{l_2} = 0.029$ permits less stringent specifications on the individual switches.

G.1.4 Comparison

We compare all the schemes studied so far.

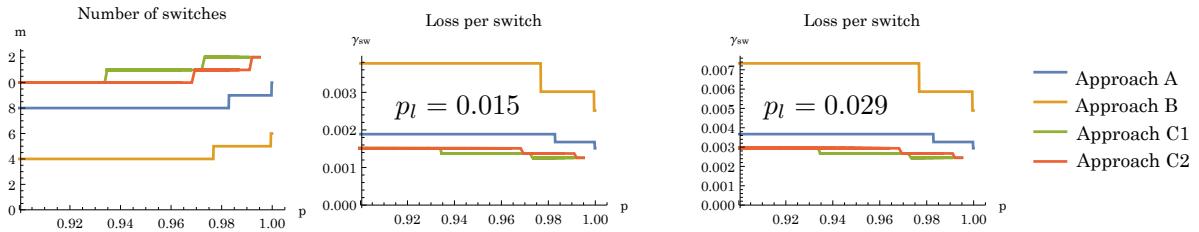


Figure G.7: Comparison of all multiplexing approaches

While it is obvious that starting from Bell Pairs we need less switches and therefore we can tolerate more loss per switch, we can also see that out of the schemes that start with single photons, approach A seems optimal: it is better to multiplex just once instead of optimising at different steps. This result ties in with the results presented in chapter 4, where schemes with less multiplexing stages were more resource efficient when paired with higher efficiency sources. Approach C2 is more efficient than approach C1, which can be understood from the fact that in the Bell generation, approach C1 uses one switch to improve probability from $3/16$ to $1/4$, whereas each switch effectively allows to improve probability by a factor of 2. In other words, that extra switch is not being optimally utilised.

G.2 State of the art

There are different technologies that allow for the realisation of optical switches, the most common are the electro-optical switches [190], carrier-based switches [191], micro-electro-mechanical switches (MEMS) [167, 192], thermo-optical switches [193] and nonlinear optical loop mirror (NOLM) based switches [96]. It is beyond the scope of this thesis to fully explain how each of these switches works, there are many parameters (such as power consumption, heat dissipation, working temperatures, material used, size, etc) which make them suitable for different optical implementations. Here however we are mainly concerned with switching loss and speed, as those are the parameters that mostly affect the theoretical design. In table G.1 we can see a comparison of the switching loss and speed of these optical switches.

Technology	Switching loss	Switching speed
Electro-optic effect	3 dB	18 GHz
Carrier injection	5.5 dB	1MHz - 10 GHz
MEMS	0.77 dB	1 MHz
Thermo-optic	0.23 dB	100 kHz
NOLM	0.6 dB	5 GHZ

Table G.1: State of the art in switches

BIBLIOGRAPHY

- [1] Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. Going Beyond Bells Theorem. In *Bells Theorem, Quantum Theory and Conceptions of the Universe*, number 3, pages 69–72. Springer Netherlands, Dordrecht, 1989.
- [2] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, jan 2001.
- [3] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1982.
- [4] Seth Lloyd. Universal Quantum Simulators. *Science*, 2023(5278):1073–1078, 1993.
- [5] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Scientific and Statistical Computing*, 26(5):1484, 1995.
- [6] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, pages 212–219, New York, New York, USA, 1996. ACM Press.
- [7] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv*, 1307.0411:1–11, 2013.
- [8] David Wecker. Programming a Quantum Computer. In *Microsoft Research Faculty Summit*, 2014.
- [9] Matthew B. Hastings, Dave Wecker, Bela Bauer, and Matthias Troyer. Improving Quantum Algorithms for Quantum Chemistry. *Quantum Information & Computation*, 15(1-2):1–21, 2014.
- [10] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. Chemical basis of Trotter-Suzuki errors in quantum chemistry simulation. *Physical Review A*, 91(2):022311, 2015.
- [11] Sarah Mostame, Joonsuk Huh, Christoph Kreisbeck, Andrew J Kerman, Takatoshi Fujita, Alexander Eisfeld, and Alán Aspuru-Guzik. Towards Outperforming Classical Algorithms with Analog Quantum Simulators. *arXiv*, 1503.01215:1–5, 2015.
- [12] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O’Brien. Quantum computers. *Nature*, 464(7285):45–53, 2010.
- [13] Jay M. Gambetta, Jerry M. Chow, and Matthias Steffen. Building logical qubits in a superconducting quantum computing system. *arXiv*, 1510.04375(October), 2015.
- [14] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012.

BIBLIOGRAPHY

- [15] C. Monroe and J. Kim. Scaling the Ion Trap Quantum Processor. *Science*, 339(6124):1164–1169, mar 2013.
- [16] C. Monroe, R. Raussendorf, a. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Physical Review A*, 89(2):022317, 2014.
- [17] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger. Blueprint for a microwave ion trap quantum computer. *arXiv*, 1508.00420:1–11, 2015.
- [18] J. L. O’Brien. Optical Quantum Computing. *Science*, 318(5856):1567–1570, dec 2007.
- [19] Terry Rudolph and Lov Grover. A 2 rebit gate universal for quantum computing. *arXiv*, 0210187:2, 2002.
- [20] Ingemar Bengtsson. Three Ways to Look at Mutually Unbiased Bases. In *AIP Conference Proceedings*, volume 889, pages 40–51. AIP, 2007.
- [21] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1(3):195–200, 1964.
- [22] David P. DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 51(2):1015–1022, 1995.
- [23] A. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265, 1936.
- [24] Alonzo Church. A Note on the Entscheidungsproblem. *The Journal of Symbolic Logic*, 1(1):40–41, 1936.
- [25] D. Deutsch. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 400(1818):97–117, jul 1985.
- [26] Christof Zalka. Grovers quantum searching algorithm is optimal. *Physical Review A*, 60(4):2746–2751, 1999.
- [27] D. Deutsch and R. Jozsa. Rapid Solution of Problems by Quantum Computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 439(1907):553–558, dec 1992.
- [28] Scott Aaronson. *Quantum Computing Since Democritus*. Cambridge University Press, 2013.
- [29] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, sep 2004.
- [30] Scott Aaronson and Alex Arkhipov. The Computational Complexity of Linear Optics. *arXiv*, 1011.3245, 2010.
- [31] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Proceedings of the 43rd annual ACM symposium on Theory of computing - STOC ’11*, page 333, 2011.
- [32] Daniel Gottesman and Isaac L. Chuang. Demonstrating the viability of universal quantumcomputation using teleportation and single-qubit operations. *Nature*, 402(6760):390–393, nov 1999.

- [33] Robert Raussendorf and Hans J. Briegel. A One-Way Quantum Computer. *Physical Review Letters*, 86(22):5188–5191, may 2001.
- [34] Richard Jozsa. An introduction to measurement based quantum computation. In D.G. Angelakis, M. Christandl, and A. Ekert, editors, *Quantum Information Processing: From Theory to Experiment*, page 22. IOP Press, aug 2006.
- [35] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 70(13):1895–1899, mar 1993.
- [36] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009.
- [37] Maarten Van Den Nest, Akimasa Miyake, Wolfgang Dür, and Hans J. Briegel. Universal resources for measurement-based quantum computation. *Physical Review Letters*, 97(15):1–4, 2006.
- [38] M. Van Den Nest, W. Dür, A. Miyake, and H. J. Briegel. Fundamentals of universality in one-way quantum computation. *New Journal of Physics*, 9(6):204–204, jun 2007.
- [39] W. Dür, G. Vidal, and J. Ignacio Cirac. Three qubits can be entangled in two inequivalent ways. *Physical Review A*, 62:062314, 2000.
- [40] D. Shepherd and M. J. Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105):1413–1439, 2009.
- [41] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, a. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O’Malley, P. Roushan, A. Vainsencher, J. Wenner, a. N. Korotkov, a. N. Cleland, and John M. Martinis. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014.
- [42] J. L. O’Brien, G. J. Pryde, a. Gilchrist, D. F V James, N. K. Langford, T. C. Ralph, and a. G. White. Quantum process tomography of a controlled-NOT gate. *Physical Review Letters*, 93(8):1–4, 2004.
- [43] Jacques Carolan, Christopher Harrold, Chris Sparrow, Enrique Martín-lópez, Nicholas J. Russell, Joshua W Silverstone, Peter J. Shadbolt, Nobuyuki Matsuda, Manabu Oguma, Mikitaka Itoh, Graham D. Marshall, Mark G. Thompson, Jonathan C. F. Matthews, Toshikazu Hashimoto, Jeremy L. O’Brien, and Anthony Laing. Universal linear optics. *Science*, 349(6249):11–16, 2015.
- [44] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum Computation by Adiabatic Evolution. *arXiv*, 0001106, 2000.
- [45] M. Born and V. Fock. Beweis des Adiabatensatzes. *Zeitschrift fur Physik*, 51(3-4):165–180, 1928.
- [46] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation. *SIAM Review*, 50(4):755–787, 2008.
- [47] David P. DiVincenzo. The Physical Implementation of Quantum Computation. *Fortschritte der Physik*, 48(9-11):771–783, sep 2000.

- [48] Carlos A. Pérez-Delgado and Pieter Kok. Quantum computers: Definition and implementations. *Physical Review A*, 83(1):012303, 2011.
- [49] Seth Lloyd. Ultimate physical limits to computation. *Nature*, 406(6799):1047–1054, 2000.
- [50] Gordon E. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.
- [51] Naomi H. Nickerson, Joseph F. Fitzsimons, and Simon C. Benjamin. Freely Scalable Quantum Technologies Using Cells of 5-to-50 Qubits with Very Lossy and Noisy Photonic Links. *Physical Review X*, 4(4):041041, 2014.
- [52] John Clarke and Frank K. Wilhelm. Superconducting quantum bits. *Nature*, 453(7198):1031–1042, jun 2008.
- [53] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing - STOC '97*, volume 14, pages 176–188, New York, New York, USA, 1997. ACM Press.
- [54] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002.
- [55] R. Raussendorf, J. Harrington, and K. Goyal. A fault-tolerant one-way quantum computer. *Annals of Physics*, 321(9):2242–2270, sep 2006.
- [56] Alberto Politi, Martin J Cryan, John G Rarity, Siyuan Yu, and Jeremy L O'Brien. Silica-on-silicon waveguide quantum circuits. *Science (New York, N.Y.)*, 320(5876):646–649, 2008.
- [57] Jonathan C. F. Matthews, Alberto Politi, André Stefanov, and Jeremy L. O'Brien. Manipulation of multiphoton entanglement in waveguide quantum circuits. *Nature Photonics*, 3(6):346–350, jun 2009.
- [58] M. D. Eisaman, J. Fan, A. Migdall, and S. V. Polyakov. Invited Review Article: Single-photon sources and detectors. *Review of Scientific Instruments*, 82(7):071101, 2011.
- [59] Robert H. Hadfield. Single-photon detectors for optical quantum information applications. *Nature Photonics*, 3(12):696–705, 2009.
- [60] Brice Calkins, Paolo L. Mennea, Adriana E. Lita, Benjamin J. Metcalf, W. Steven Kolthammer, Antia Lamas-Linares, Justin B. Spring, Peter C. Humphreys, Richard P. Mirin, James C. Gates, Peter G. R. Smith, Ian a. Walmsley, Thomas Gerrits, and Sae Woo Nam. High quantum-efficiency photon-number-resolving detector for photonic on-chip information processing. *Optics Express*, 21(19):22657, 2013.
- [61] Tom Baehr-Jones, Thierry Pinguet, Patrick Lo Guo-Qiang, Steven Danziger, Dennis Prather, and Michael Hochberg. Myths and rumours of silicon photonics. *Nature Photonics*, 6(4):206–208, 2012.
- [62] Ying Li, Peter C. Humphreys, Gabriel J. Mendoza, and Simon C. Benjamin. Resource costs for fault-tolerant linear optical quantum computing. *Physical Review X*, 5(4):041007, 2015.
- [63] Brendon Lovett and Pieter Kok. *Optical Quantum Information Processing*. Cambridge University Press, 2010.
- [64] Michael Reck, Anton Zeilinger, Herbert Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical Review Letters*, 73(1):58–61, 1994.

- [65] Peter Shadbolt. *Complexity and Control in Quantum Photonics*. Springer International Publishing, 2016.
- [66] Ariel Lipson, Stephen G. Lipson, and Henry Lipson. *Optical Physics*. Cambridge University Press, 2011.
- [67] T. C. Ralph, N. K. Langford, T. B. Bell, and A. G. White. Linear optical controlled-NOT gate in the coincidence basis. *Physical Review A*, 65(6):062324, jun 2002.
- [68] J. Calsamiglia and N. Lütkenhaus. Maximum efficiency of a linear-optical Bell-state analyzer. *Applied Physics B*, 72(1):67–71, jan 2001.
- [69] C. K. Hong, Z. Y. Ou, and L. Mandel. Measurement of subpicosecond time intervals between two photons by interference. *Physical Review Letters*, 59(18):2044–2046, 1987.
- [70] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Books Online, 2010.
- [71] Maarten Van Den Nest. Universal quantum computation with little entanglement. *Physical Review Letters*, 110(February):1–4, 2013.
- [72] Jacques Carolan, Jasmin D. A. Meinecke, Peter J. Shadbolt, Nicholas J. Russell, Nur Ismail, Kerstin Wörhoff, Terry Rudolph, Mark G. Thompson, Jeremy L. O’Brien, Jonathan C. F. Matthews, and Anthony Laing. On the experimental verification of quantum complexity in linear optics. *Nature Photonics*, 8(8):621–626, jul 2014.
- [73] Max Tillmann, Borivoje Dakić, René Heilmann, Stefan Nolte, Alexander Szameit, and Philip Walther. Experimental boson sampling. *Nature Photonics*, 7(7):540–544, 2013.
- [74] Matthew A. Broome, Alessandro Fedrizzi, Saleh Rahimi-Keshari, Justin Dove, Scott Aaronson, Timothy C. Ralph, and Andrew G. White. Photonic Boson Sampling in a Tunable Circuit. *Science*, 339(6121):794–798, feb 2013.
- [75] Nicolò Spagnolo, Chiara Vitelli, Marco Bentivegna, Daniel J. Brod, Andrea Crespi, Fulvio Flaminini, Sandro Giacomin, Giorgio Milani, Roberta Ramponi, Paolo Mataloni, Roberto Osellame, Ernesto F. Galvão, and Fabio Sciarrino. Experimental validation of photonic boson sampling. *Nature Photonics*, 8(June):3–9, 2014.
- [76] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52(4):2493–2496, 1995.
- [77] E. R. Caianiello. On quantum field theory I: explicit solution of Dysons equation in electrodynamics without use of feynman graphs. *Il Nuovo Cimento*, 10(12):1634–1652, 1953.
- [78] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
- [79] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697, 2004.
- [80] Linda Sansoni, Fabio Sciarrino, Giuseppe Vallone, Paolo Mataloni, Andrea Crespi, Roberta Ramponi, and Roberto Osellame. Polarization entangled states measurement on a chip Linda. *Proc. of SPIE*, 8072:80720Q–80720Q–6, 2011.
- [81] M.G. Thompson, A. Politi, J.C.F. Matthews, and J.L. O’Brien. Integrated waveguide circuits for optical quantum computing. *IET Circuits, Devices & Systems*, 5(2):94, 2011.

- [82] Damien Bonneau, Mirko Lobino, Pisu Jiang, Chandra M. Natarajan, Michael G. Tanner, Robert H. Hadfield, Sanders N. Dorenbos, Val Zwillaer, Mark G. Thompson, and Jeremy L. O'Brien. Fast path and polarization manipulation of telecom wavelength single photons in lithium niobate waveguide devices. *Physical Review Letters*, 108(5):1–5, 2012.
- [83] Kevin T. McCusker, Yu-Ping Huang, Abijith S. Kowlgy, and Prem Kumar. Experimental Demonstration of Interaction-Free All-Optical Switching via the Quantum Zeno Effect. *Physical Review Letters*, 110(24):240403, 2013.
- [84] Jeffrey H. Shapiro. Single-photon Kerr nonlinearities do not help quantum computation. *Physical Review A*, 73(6):062305, 2006.
- [85] Stefan Scheel, Kae Nemoto, William Munro, and Peter Knight. Measurement-induced nonlinearity in linear optics. *Physical Review A*, 68(3):032310, sep 2003.
- [86] J. D. Franson, M. M. Donegan, M. J. Fitch, B. C. Jacobs, and T. B. Pittman. High-fidelity quantum logic operations using linear optical elements. page 13, 2002.
- [87] Federico M. Spedalieri, Hwang Lee, and Jonathan P. Dowling. High-fidelity linear optical quantum computing with polarization encoding. *Physical Review A*, 73(1):012334, jan 2006.
- [88] N. Yoran and B. Reznik. Deterministic Linear Optics Quantum Computation with Single Photon Qubits. *Physical Review Letters*, 91(3):037903, jul 2003.
- [89] Michael A. Nielsen. Optical Quantum Computation Using Cluster States. *Physical Review Letters*, 93(4):040503, jul 2004.
- [90] Daniel E. Browne and Terry Rudolph. Resource-efficient linear optical quantum computation. *Physical Review Letters*, 95:2–6, 2005.
- [91] M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 1999.
- [92] Jaewoo Joo, Peter Knight, Jeremy O'Brien, and Terry Rudolph. One-way quantum computation with four-dimensional photonic qudits. *Physical Review A*, 76(5):052326, nov 2007.
- [93] Michael Varnava, Daniel Browne, and Terry Rudolph. How Good Must Single Photon Sources and Detectors Be for Efficient Linear Optical Quantum Computation? *Physical Review Letters*, 100(6):060502, feb 2008.
- [94] A. J. F. Hayes, A. Gilchrist, C. R. Myers, and T. C. Ralph. Utilizing encoding in scalable linear optics quantum computing. *Journal of Optics B: Quantum and Semiclassical Optics*, 6(12):533–541, dec 2004.
- [95] Alexei Gilchrist, A. J. F. Hayes, and T. C. Ralph. Efficient parity-encoded optical quantum computing. *Physical Review A*, 75(5):052328, may 2007.
- [96] Timothy M. Rambo, Kevin McCusker, Yu-Ping Huang, and Prem Kumar. Low-loss all-optical quantum switching. *2013 IEEE Photonics Society Summer Topical Meeting Series*, 3(c):179–180, 2013.
- [97] Yang Zhang, Amir Hosseini, Xiaochuan Xu, David Kwong, and Ray T. Chen. Ultralow-loss silicon waveguide crossing using Bloch modes in index-engineered cascaded multimode-interference couplers. *Optics letters*, 38(18):3608–11, 2013.

- [98] Zhen Sheng, Zhiqi Wang, Chao Qiu, Le Li, Albert Pang, Aimin Wu, Xi Wang, Shichang Zou, and Fuwan Gan. A Compact and Low-Loss MMI Coupler Fabricated With CMOS Technology. *IEEE Photonics Journal*, 4(6):2272–2277, 2012.
- [99] K. Kieling, T. Rudolph, and J. Eisert. Percolation, Renormalization, and Quantum Computing with Nondeterministic Gates. *Physical Review Letters*, 99(13):130501, sep 2007.
- [100] Olaf Mandel, Markus Greiner, Artur Widera, and Tim Rom. Controlled collisions for multi-particle entanglement of optically trapped atoms. *October*, pages 937–940, 2003.
- [101] M. Trupke, J. Metz, A. Beige, and E. A. Hinds. Towards quantum computing with single atoms and optical cavities on atom chips. *Journal of Modern Optics*, 54(11):1639–1655, jul 2007.
- [102] Yuan Liang Lim, Sean D. Barrett, Almut Beige, Pieter Kok, and Leong Chuan Kwek. Repeat-until-success quantum computing using stationary and flying qubits. *Physical Review A*, 73(1):012304, jan 2006.
- [103] Geoffrey Grimmett. *Lectures on Probability Theory and Statistics*, volume 1665 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1997.
- [104] K. Kieling and J. Eisert. Percolation in quantum computation and communication. In *Quantum and Semi-classical Percolation and Breakdown in Disordered Solids, (Springer, Berlin, 2009)*, pages 287–319. dec 2007.
- [105] Daniel E. Browne, Matthew B. Elliott, Steven T. Flammia, Seth T. Merkel, Akimasa Miyake, and Anthony J. Short. Phase transition of computational power in the resource states for one-way quantum computation. *New Journal of Physics*, 10(2):023010, feb 2008.
- [106] M. Yanuka and R. Englman. Bond-site percolation: empirical representation of critical probabilities. *Journal of Physics A: Mathematical and General*, 23(7):L339–L345, 1990.
- [107] J. M. Hammersley. A generalization of McDiarmid’s theorem for mixed Bernoulli percolation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 88(01):167, 1980.
- [108] Yuriy Yu. Tarasevich and Steven C. van der Marck. An investigation of site-bond percolation on many lattices. *International Journal of Modern Physics C*, 10(07):14, jun 1999.
- [109] D. Stauffer and A. Aharony. Introduction to Percolation Theory. *Computer*, 1(4):192, 1994.
- [110] J. Hoshen and R. Kopelman. Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Physical Review B*, 14(8):3438–3445, oct 1976.
- [111] W. P. Grice. Arbitrarily complete Bell-state measurement using only linear optical elements. *Physical Review A*, 84(4):042331, oct 2011.
- [112] Christopher Dawson, Henry Haselgrove, and Michael Nielsen. Noise Thresholds for Optical Quantum Computers. *Physical Review Letters*, 96(2):020501, jan 2006.
- [113] Michael Varnava, Daniel Browne, and Terry Rudolph. Loss Tolerance in One-Way Quantum Computation via Counterfactual Error Correction. *Physical Review Letters*, 97(12):120501, sep 2006.

BIBLIOGRAPHY

- [114] Lev Vaidman and Nadav Yoran. Methods for reliable teleportation. *Physical Review A*, 59(1):116–125, 1999.
- [115] Fabian Ewert and Peter van Loock. 3/4 - Efficient Bell Measurement with Passive Linear Optics and Unentangled Ancillae. *Physical Review Letters*, 113(14):140403, sep 2014.
- [116] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Physical Review A*, 54(3):1862–1868, sep 1996.
- [117] Daniel Gottesman. The Heisenberg Representation of Quantum Computers. *arXiv*, 9807006:20, 1998.
- [118] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, may 1997.
- [119] David Jennings. *Lecture Notes: Quantum Error-Correction, Stabilizers and Measurement-Based Quantum Computation*. 2013.
- [120] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5):052328, nov 2004.
- [121] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Graphical description of the action of local Clifford transformations on graph states. *Physical Review A*, 69(2):022316, 2004.
- [122] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Efficient algorithm to recognize the local Clifford equivalence of graph states. *Physical Review A*, 70(3):034302, 2004.
- [123] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Local unitary versus local Clifford equivalence of stabilizer states. *Physical Review A*, 71(6):062323, 2005.
- [124] Beverley Bolt, T. G. Room, and G. E. Wall. On the Clifford collineation, transform and similarity groups. I. *Journal of the Australian Mathematical Society*, 2(01):60, apr 1961.
- [125] Beverley Bolt, T. G. Room, and G. E. Wall. On the Clifford collineation, transform and similarity groups. II. *Journal of the Australian Mathematical Society*, 2(01):80, apr 1961.
- [126] Beverley Bolt. On the Clifford collineation, transform and similarity groups. (III) Generators and involutions. *Journal of the Australian Mathematical Society*, 2(03):334, feb 1962.
- [127] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098–1105, 1996.
- [128] Andrew Steane. Multiple-Particle Interference and Quantum Error Correction. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 452:2551–2577, 1996.
- [129] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, jan 2003.
- [130] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, feb 2005.
- [131] Mark Howard, Joel Wallman, Victor Veitch, and Joseph Emerson. Contextuality supplies the ‘magic’ for quantum computation. *Nature*, 510(7505):351–355, 2014.

- [132] Terry Rudolph. *Lecture Notes: Introduction to Quantum Information*. 2011.
- [133] John Harris, Jeffry L. Hirst, and Michael Mossinghoff. *Combinatorics and Graph Theory*, volume 51 of *Undergraduate Texts in Mathematics*. Springer New York, New York, NY, 2008.
- [134] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Local Equivalence of Stabilizer States and Codes. *Proceedings of the 16th international symposium on mathematical theory of networks and systems (MTNS)*. KU Leuven, Belgium, page 182, 2004.
- [135] Bei Zeng, Hyeyoun Chung, Andrew W. Cross, and Isaac L. Chuang. Local unitary versus local Clifford equivalence of stabilizer and graph states. *Physical Review A*, 75(3):032325, 2007.
- [136] M. Hein, J. Eisert, and H. J. Briegel. Multiparty entanglement in graph states. *Physical Review A*, 69(6):062311, 2004.
- [137] D. Gross and M. Van Den Nest. The LU-LC conjecture, diagonal local operations and quadratic forms over GF(2). *Quantum Inf. Comput.*, 8(3-4):263, 2007.
- [138] Zhengfeng Ji, Jianxin Chen, Zhaohui Wei, and Mingsheng Ying. The LU-LC conjecture is false. *Quantum Information and Computation*, 10(1):97–108, 2010.
- [139] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. *Proceedings of the 44th symposium on Theory of Computing - STOC '12*, page 887, 2012.
- [140] E.H. Bareiss. Sylvester’s Identity and Multistep Integer-Preserving Gaussian Elimination. *Math. Comp.*, 22(2):565–578, 1968.
- [141] Eric W. Weisstein. “Gaussian Elimination.”. *From MathWorld—A Wolfram Web Resource*.
- [142] D.R. Simon. On the power of quantum computation. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, volume 26, pages 116–123. IEEE Comput. Soc. Press, 1994.
- [143] Simon Anders and Hans J. Briegel. Fast simulation of stabilizer circuits using a graph-state representation. *Physical Review A*, 73(2):022334, feb 2006.
- [144] Peter Petersen. *Linear algebra*. 2012.
- [145] Alston S. Householder. Unitary Triangularization of a Nonsymmetric Matrix. *Journal of the ACM*, 5(4):339–342, oct 1958.
- [146] Xi-Lin Wang, Xin-Dong Cai, Zu-En Su, Ming-Cheng Chen, Dian Wu, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum teleportation of multiple degrees of freedom of a single photon. *Nature*, 518(7540):516–519, feb 2015.
- [147] Qiang Zhang, Xiao-Hui Bao, Chao-Yang Lu, Xiao-Qi Zhou, Tao Yang, Terry Rudolph, and Jian-Wei Pan. Demonstration of a scheme for the generation of event-ready entangled photon pairs from a single-photon source. *Physical Review A*, 77(6):062316, jun 2008.
- [148] Mehul Malik, Manuel Erhard, Marcus Huber, Mario Krenn, Robert Fickler, and Anton Zeilinger. Multi-photon entanglement in high dimensions. *arXiv*, 1509.02561:1–19, 2015.
- [149] Philip Walther, Markus Aspelmeyer, and Anton Zeilinger. Heralded generation of multiphoton entanglement. *Physical Review A - Atomic, Molecular, and Optical Physics*, 75(1):1–5, 2007.

BIBLIOGRAPHY

- [150] Christopher Gerry and Peter Knight. *Introductory Linear Optics*. Cambridge University Press, 2005.
- [151] Cezary Śliwa and Konrad Banaszek. Conditional preparation of maximal polarization entanglement. *Physical Review A*, 67(3):030101, mar 2003.
- [152] Koji Azuma, Kiyoshi Tamaki, and Hoi-Kwong Lo. All-photonic quantum repeaters. *Nature Communications*, 6:6787, apr 2015.
- [153] Samuel L. Braunstein and A. Mann. Measurement of the Bell operator and quantum teleportation. *Physical Review A*, 51(3):1727–1730, 1995.
- [154] N. Lütkenhaus, J. Calsamiglia, and K. A. Suominen. Bell measurements for teleportation. *Physical Review A*, 59(5), 1999.
- [155] Evan Jeffrey, Nicholas A. Peters, and Paul G. Kwiat. Towards a periodic deterministic source of arbitrary single-photon states. *New Journal of Physics*, 6(1):100–100, 2004.
- [156] Kevin T. McCusker and Paul G. Kwiat. Efficient Optical Quantum State Engineering. *Physical Review Letters*, 103(16):163602, oct 2009.
- [157] Thomas Jennewein, Marco Barbieri, and Andrew G. White. Single-photon device requirements for operating linear optics quantum computing outside the post-selection basis. *Journal of Modern Optics*, 58(January 2015):37–41, 2010.
- [158] Xiao-Song Ma, Stefan Zoller, Johannes Kofler, Thomas Jennewein, and Anton Zeilinger. Experimental generation of single photons via active multiplexing. *Physical Review A*, 83(4):043814, apr 2011.
- [159] A. L. Migdall, D. Branning, S. Castelletto, and M. Ware. Tailoring Single and Multiphoton Probabilities of a Single Photon On-Demand Source. *Physical Review A*, 66(5):4, 2002.
- [160] Gabriel J. Mendoza, Raffaele Santagati, Jack Munns, Elizabeth Hemsley, Mateusz Piekarek, Enrique Martin-Lopez, Graham D. Marshall, Damien Bonneau, Mark G. Thompson, and Jeremy L. O’Brien. Active Temporal Multiplexing of Photons. *arXiv*, 1503.01215, 2015.
- [161] Damien Bonneau, Gabriel J. Mendoza, Jeremy L O’Brien, and Mark G. Thompson. Effect of loss on multiplexed single-photon sources. *New Journal of Physics*, 17(4):043057, apr 2015.
- [162] Netanel Lindner and Terry Rudolph. Proposal for Pulsed On-Demand Sources of Photonic Cluster State Strings. *Physical Review Letters*, 103(11):113602, sep 2009.
- [163] Hussain A. Zaidi, Chris Dawson, Peter van Loock, and Terry Rudolph. Near-deterministic creation of universal cluster states with probabilistic Bell measurements and three-qubit resource states. *Physical Review A*, 91(4):1–5, 2015.
- [164] Alberto Peruzzo, Jarrod R. McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(2):1–10, 2014.
- [165] Peter Shadbolt, Tamás Vértesi, Yeong-Cherng Liang, Cyril Branciard, Nicolas Brunner, and Jeremy L. O’Brien. Guaranteed violation of a Bell inequality without aligned reference frames or calibrated devices. *Scientific Reports*, 2:1–7, 2012.
- [166] S. B. Bravyi and A. Yu. Kitaev. Quantum codes on a lattice with boundary. *Quantum Computers and Computing*, 2(1):43–48, 2001.

- [167] Tae Joon Seok, Niels Quack, Sangyo Han, and Ming C. Wu. 50x50 Digital Silicon Photonic Switches with MEMS-Actuated Adiabatic Couplers. In *Optical Fiber Communication Conference*, volume 1, page M2B.4, Washington, D.C., 2015. OSA.
- [168] S. C. van der Marck. Percolation thresholds and universal formulas. *Physical Review E*, 55(2):1514–1517, 1997.
- [169] S. C. van der Marck. Erratum: Percolation thresholds and universal formulas. *Physical Review E*, 56(3):4297, 1997.
- [170] Komei Fukuda and Tomomi Matsui. Finding all minimum-cost perfect matchings in Bipartite graphs. *Networks*, 22(5):461–468, 1992.
- [171] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.
- [172] S. J. Devitt, W. J. Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, page 35, may 2013.
- [173] Barbara M. Terhal. Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2):307–346, apr 2015.
- [174] H. Bombin. An Introduction to Topological Quantum Codes. *Quantum Error Correction*, 2013.
- [175] Jack Edmonds. Paths, trees, and flowers. *Journal canadien de mathématiques*, 17:449–467, 1965.
- [176] Thomas M. Stace, Sean D. Barrett, and Andrew C. Doherty. Thresholds for topological codes in the presence of loss. *Physical Review Letters*, 102(May):1–4, 2009.
- [177] Sean D. Barrett and Thomas M. Stace. Fault Tolerant Quantum Computation with Very High Threshold for Loss Errors. *Physical Review Letters*, 105(20):200502, 2010.
- [178] Yuichiro Matsuzaki, Simon C. Benjamin, and Joseph Fitzsimons. Probabilistic Growth of Large Entangled States with Low Error Accumulation. *Physical Review Letters*, 104(5):050501, feb 2010.
- [179] Ying Li, Sean D. Barrett, Thomas M. Stace, and Simon C. Benjamin. Long range failure-tolerant entanglement distribution. *New Journal of Physics*, 15(2):023012, 2013.
- [180] Jie Sun, Erman Timurdogan, Ami Yaacobi, Ehsan Shah Hosseini, and Michael R. Watts. Large-scale nanophotonic phased array. *Nature*, 493(7431):195–199, 2013.
- [181] Michael Varnava, Daniel E. Browne, and Terry Rudolph. Loss tolerant linear optical quantum memory by measurement-based quantum computing. *New Journal of Physics*, 9(6):203–203, jun 2007.
- [182] David A. Herrera-Martí, Austin G. Fowler, David Jennings, and Terry Rudolph. Photonic implementation for the topological cluster-state quantum computer. *Physical Review A*, 82(3):032332, sep 2010.
- [183] Alexandru Paler and Simon Devitt. Private Communication. 2015.
- [184] H. Bombin and M. A. Martin-Delgado. Topological quantum distillation. *Physical Review Letters*, 97(November):1–4, 2006.

BIBLIOGRAPHY

- [185] H. Bombin and M. A. Martin-Delgado. Topological computation without braiding. *Physical Review Letters*, 98(4):1–4, 2007.
- [186] Sergey Bravyi and Andrew Cross. Doubled Color Codes. *arXiv*, 1509.03239:1–53, 2015.
- [187] Tomas Jochym-O'Connor and Stephen D. Bartlett. Stacked codes: universal fault-tolerant quantum computation in a two-dimensional layout. *arXiv*, 1509.04255:1–15, 2015.
- [188] Alexei Yu. Kitaev, Alexander Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation*. 2002.
- [189] Ethan Bernstein and Umesh Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.
- [190] EOSPACE. Custom High-Speed Lithium Niobate Electro-optic Switches, 2015.
- [191] Fuwan Gan, Felix Jan Grawert, Jan-Malte Schley, Shoji Akiyama, Jürgen Michel, Kazumi Wada, Lionel C. Kimerling, and Franz X. Kärtner. Design of All-Optical Switches Based on Carrier. *Journal of Lightwave Technology*, 24(9):3454–3463, 2006.
- [192] J. Kim, C. J. Nuzman, B. Kumar, D. F. Lieuwen, J. S. Kraus, A. Weiss, C. P. Lichtenwalner, A. R. Papazian, R. E. Frahm, N. R. Basavanhally, D. A. Ramsey, V. A. Aksyuk, F. Pardo, M. E. Simon, V. Lifton, H. B. Chan, M. Haueis, A. Gasparyan, H. R. Shea, S. Arney, C. A. Bolle, P. R. Kolodner, R. Ryf, D. T. Neilson, and J. V. Gates. 1100 port MEMS-based optical crossconnect with 4-dB maximum loss. *IEEE Photonics Technology Letters*, 15(11):1537–1539, 2003.
- [193] Nicholas C. Harris, Yangjin Ma, Jacob Mower, Tom Baehr-Jones, Dirk Englund, Michael Hochberg, and Christophe Galland. Efficient , compact and low loss thermo-optic phase shifter in silicon. *Optics Express*, 22(9):83–85, 2014.