



SAPIENZA
UNIVERSITÀ DI ROMA

Ingegneria civile ed industriale
Ingegneria dell'informazione, informatica e statistica

Master di II livello interfacoltà
in
OPTICS AND QUANTUM INFORMATION

ELABORATO FINALE

User interface for a reconfigurable linear optical quantum computing circuit

Relatore
Prof. Fabio Antonio Bovino

Candidato
Pietro De Lellis
Matricola 1980413

Anno Accademico 2020-2021

*"The best that most of us can hope
to achieve in physics is simply
to misunderstand at a deeper level."*

Wolfgang Pauli

Abstract

This work aims to propose a user interface for a reconfigurable linear optical quantum computing (LOQC) circuit. In the first chapter, an insight about the features of quantum computing together with advantages and disadvantages of the possible approaches is provided. In the second chapter, an overview of quantum information processing and the main quantum gates are introduced. The following part provides a description of the building blocks of linear optical quantum computing circuits. Then, in the fourth chapter, the practical implementation of a user interface for controlling a LOQC circuit is presented. Finally, conclusions and possible future developments are drawn up.

Keywords:

Quantum optics, Quantum computing, Optical interferometry, Phase shifters, Beam splitters, Linear optical quantum computing.

Contents

List of Figures	iii
List of Tables	iv
1 Introduction	1
2 Quantum computing principles	5
2.1 Qubits	5
2.2 Evolution	8
2.3 Measurement	9
2.4 Quantum gates	9
2.4.1 Comparison with classical gates	10
2.4.2 Notable quantum gates	11
3 Linear optical quantum computing	14
3.1 Linear optics building blocks	15
3.2 Unitary transformation decomposition	17
3.3 Layout of a circuit performing unitary transformation	19
3.4 Implementing notable gates	30
4 User interface development	33
4.1 NI devices	33
4.2 Interface at startup	35
4.3 Interface during execution	36
4.4 Comparison of results	37
5 Conclusions	38
Bibliography	39

List of Figures

1.1	Quantum computer	1
1.2	Examples of linear optical quantum circuits	2
1.3	Information coded in polarization or spatial modes	3
2.1	Bloch sphere	6
2.2	Unitary transformation on the Bloch sphere	8
3.1	Circuit element	15
3.2	Reck's and Clements' decompositions	17
3.3	Clements' algorithm	18
3.4	Optical, electronics and mechanics integration of InP circuit	19
3.5	Circuit layout	20
3.6	Circuit layout: focus on type 1 gate	21
3.7	Type 1 gate output for $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}'$	22
3.8	Type 1 gate output for $\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}'$	23
3.9	Type 1 gate output for $\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}'$	24
3.10	Type 1 gate output for $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}'$	25
3.11	Circuit layout: focus on type 2 gate	26
3.12	Type 2 gate output for $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}'$	27
3.13	Type 2 gate output for $\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}'$	28
3.14	Type 2 gate output for $\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}'$	29
3.15	Type 2 gate output for $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}'$	30
4.1	NI-9264	34
4.2	NI-cDAQ-9184	34
4.3	NI-MAX interface	34
4.4	User interface at startup	35
4.5	User interface while running	36
4.6	Comparison of theoretical and experimental response	37

List of Tables

2.1	Truth table for classical AND and OR gates	10
2.2	Truth table for classical NOT gate	10
2.3	Truth table for classical CNOT gate	11
2.4	Truth table for classical Toffoli gate	11
3.1	Parameter values for identical gate	30
3.2	Parameter values for identical gate - another option	31
3.3	Parameter values for SWAP gate	31
3.4	Parameter values for CNOT gate	31
3.5	Parameter values for $I \otimes H$ gate	32

Chapter 1

Introduction

Quantum computing has attracted much attention over the years, mostly because of its promise of super-fast integer factorization. There are many different architectures for quantum computers based on many different physical systems. These include atom and ion-trap quantum computing, superconducting charge and flux qubits, nuclear magnetic resonance (NMR), spin and charge based quantum dots, nuclear spin quantum computing, and optical quantum computing. All these systems have their own advantages in quantum information processing. However, even though there may now be a few front-runners, no physical implementation seems to have a clear edge over others at this point.

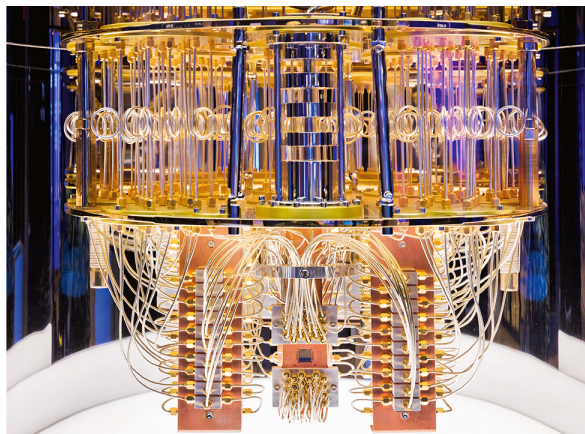


Figure 1.1: Quantum computer

In the early 1980's, Richard Feynman showed that there are fundamental limitations in trying to perform simulations of complex quantum systems on conventional computers, regardless of their size or speed [1]. He noted, however, that these problems could be overcome, at least in principle, by building computers based on quantum mechanics instead of classical physics. One may naturally wonder if these “quantum computers” would be useful for other applications, assuming that they could eventually be built.

The answer was shown to be “yes” when, nearly a decade later, Peter Shor discovered a quantum computing algorithm for efficient factorization of large integers – a problem that has no efficient solution on conventional computers and forms the basis of many secure communications protocols [2]. After that, it was shown that a quantum computer could also be used to search an unstructured database much faster than any conventional computer. Due to these critical theoretical developments, there has been a recent explosion of experimental work aimed towards building a quantum computer. Researchers in many different areas of physics are actively pursuing a variety of methods to accomplish this challenging goal.

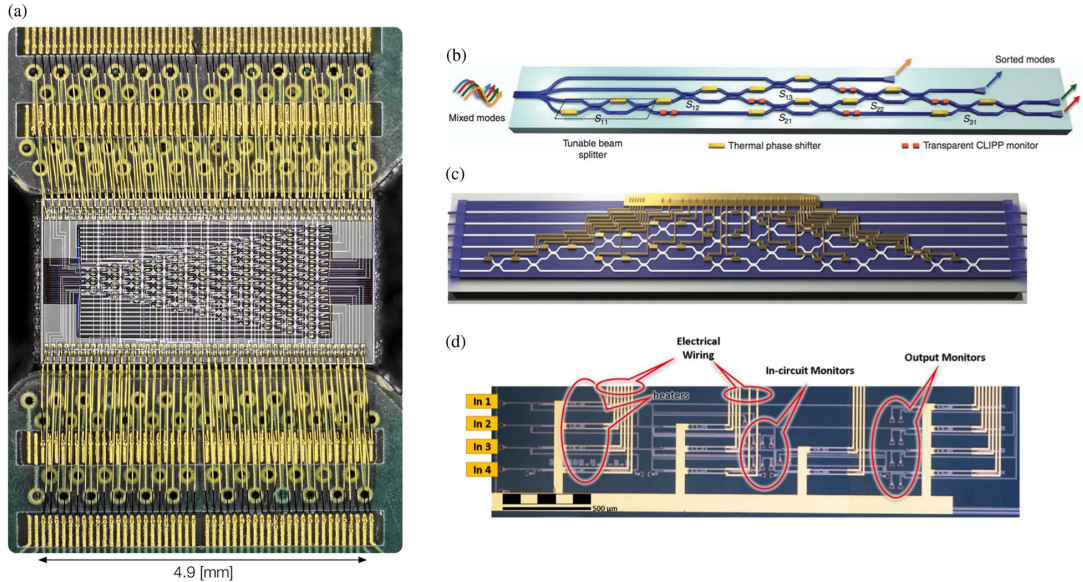


Figure 1.2: Examples of linear optical quantum circuits

Among possible implementations for quantum information processing (QIP) and quantum computation, optical quantum systems are prominent candidates, since they link quantum computation and quantum communication in the same framework.

Quantum computing with linear quantum optics, the subject of this work, has the advantage that the smallest unit of quantum information (the photon) is potentially free from decoherence: the quantum information stored in a photon tends to stay there [3]. In this framework, superpositions of quantum states can be easily represented, transmitted and detected using photons.

In the optical approach, quantum bits, or “qubits”, of information are represented by the quantum state of single photons. For example, the logical value 0 can be represented by a horizontally polarized photon, while the logical value 1 can be represented by a vertically polarized photon. Alternatively, 0 and 1 could be represented by the presence of a single photon in one of two optical fibers [4].

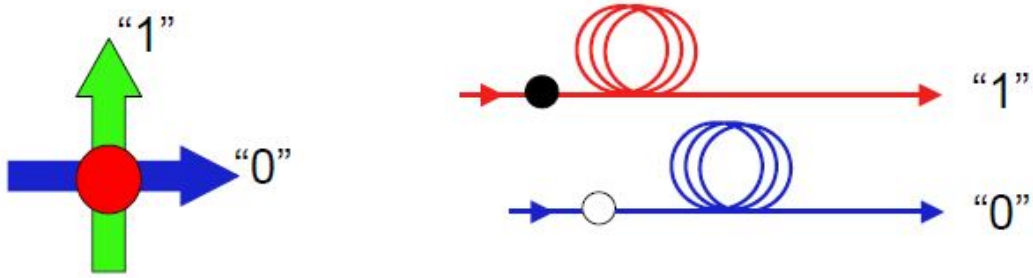


Figure 1.3: Information coded in polarization or spatial modes

Linear elements of optical systems may be the simplest building blocks to realize quantum operations and quantum gates. Each linear optical element applies a unitary transformation on a finite number of qubits and there are systems of finite linear optical elements to constructs linear optics networks, which can realize any diagram based on

the quantum circuit model (figure 1.2).

The primary advantage of an optical approach to quantum computing is that it would allow quantum logic gates and quantum memory devices to be easily connected together using optical fibers or wave-guides in analogy with the wires of a conventional computer. This allows a modularity that is not normally available in other approaches. For example, the transfer of qubits from one location to another in ion-trap or NMR systems is a very complex process. Another great advantage is operational temperature, that can be normal room temperature, while in other approach the systems need to be cooled to very low temperature not be affected by noise.

The main drawback of optical quantum computing is introducing interactions among photons, which may become an issue: this can be circumvented by using an exponentially growing number of optical modes, but this is by definition not scalable.

Chapter 2

Quantum computing principles

The power of quantum computing is related to the fact that quantum mechanics allows the qubits to be in states that do not correspond to specific values of 0 or 1. The basic working principles of this kind of computing will be described hereafter.

2.1 Qubits

The quantum analogue of a conventional bit, a qubit, has rather more freedom. It lives in a two-dimensional Hilbert space, with a state of the following form:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2.1)$$

where α and β , are two complex numbers whose squares correspond to the probability of measuring a value of 0 or 1. In this equation it seems that there are four degrees of freedom (as α and β are complex numbers), but one is removed by the normalization constraint $|\alpha|^2 + |\beta|^2 = 1$. So, the parameters can be changed in:

$$\begin{aligned} \alpha &= e^{i\delta} \cos \frac{\theta}{2} \\ \beta &= e^{i(\delta+\varphi)} \sin \frac{\theta}{2} \end{aligned} \quad (2.2)$$

Additionally, as for a single qubit the global phase factor can be neglected, only two degrees of freedom remain:

$$\begin{aligned}\alpha &= \cos \frac{\theta}{2} \\ \beta &= e^{i\varphi} \sin \frac{\theta}{2}\end{aligned}\tag{2.3}$$

and the state vector becomes:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle\tag{2.4}$$

In contrast to classical bits (which always have a definite value of either 0 or 1) the qubits can, in some sense, behave as if they had the values of 0 and 1 at the same time. There are many physical quantum systems that can be considered for use as qubits. For example, a single two-level atom in its ground state could correspond to a 0, while the same atom in its excited state would correspond to a 1. In quantum optics, photon polarization or presence in different spatial modes (in fiber optics) can be physical representations of qubits.

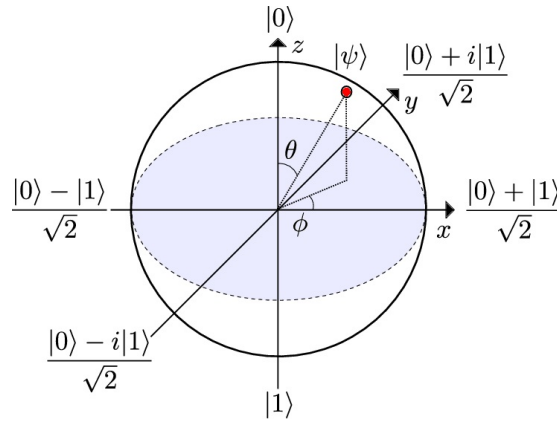


Figure 2.1: Bloch sphere

In the space represented in figure 2.1, which is known as Bloch sphere, a conventional bit can be placed only on the poles, while a qubit can be anywhere on the surface of the sphere (or sometimes inside, if it does not represent a pure state, but a mixed state). States such as in equation 2.1 are superposition states and have amplitudes for $|0\rangle$ and $|1\rangle$ at the same time.

A set, or register, of n conventional bits can be in just one of its 2^n possible different states at any particular time. In sharp contrast, a register of n qubits can be in a superposition of all possible conventional register states at the same time, with 2^n amplitudes. Clearly, if it is possible to operate, or compute, simultaneously with all the amplitudes of a quantum register, there is the possibility of massive parallel computation based on quantum superpositions.

A useful tool for dealing with quantum states are *density matrices*, as state vectors can only represent *pure states*, while matrices can also represent *mixed states*. Mixed states can arise in two different situations: when the preparation of the system is not fully known or when a physical system is entangled with another one. The general definition of a density matrix is:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \quad (2.5)$$

where $0 \leq p_i \leq 1$ and $\sum p_i = 1$.

Considering the Pauli matrices (including the identity I):

$$\begin{aligned} \sigma_0 = I &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \sigma_x &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ \sigma_y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} & \sigma_z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{aligned} \quad (2.6)$$

an arbitrary state for a qubit can be written as a linear combination of these matrices, as:

$$\rho = \frac{1}{2}(I + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z) \quad (2.7)$$

where r_x , r_y and r_z are real numbers, and for pure states $r_x^2 + r_y^2 + r_z^2 = 1$.

2.2 Evolution

Quantum systems evolve according to the Schrödinger equation:

$$H|\psi\rangle = i\hbar \frac{\partial |\psi\rangle}{\partial t} \quad (2.8)$$

where H is the Hamiltonian. The effect of this kind of evolution is a reversible unitary operation U on the system, such that:

$$|\psi(t)\rangle = U|\psi(0)\rangle = \exp\left[-i\frac{Ht}{\hbar}\right] |\psi(0)\rangle \quad (2.9)$$

where $|\psi(0)\rangle$ is the initial state. This operation on a single qubit corresponds to a rotation on the Bloch sphere, as illustrated in figure 2.2. More generally, with sufficient control over the Hamiltonian of a register of qubits, it is possible to apply a chosen unitary transformation U on that register. This is necessary because the centrepiece of quantum computing is the ability to apply the transformation dictated by some quantum algorithm or protocol.

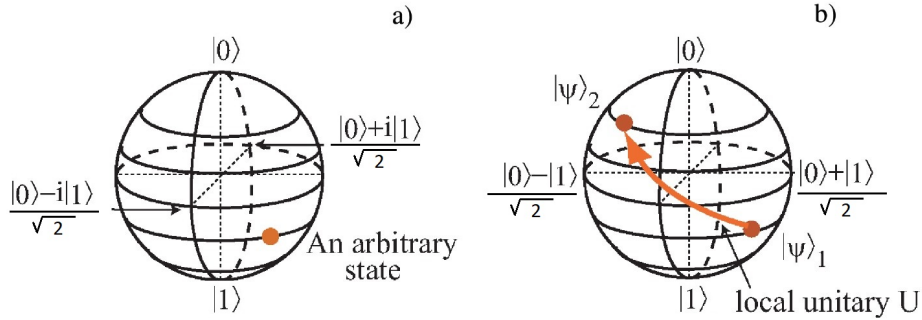


Figure 2.2: Unitary transformation on the Bloch sphere

2.3 Measurement

If the value of a qubit in state 2.4 is measured, the outcome is random, giving 0 with probability $\cos^2(\frac{\theta}{2})$ and 1 with probability $\sin^2(\frac{\theta}{2})$. Quantum measurement is irreversible. At individual level, the system is projected into the quantum state corresponding to the result recorded (if it is not destroyed, such as in the absorption of a photon). Nondestructive measurement can be also considered as the preparation of a new state, as the qubit is left in a known state and can be used for another process. Measurement is the only way to extract information from a quantum system, so the irreversibility dictates that there cannot be information gain by the external observer without disturbance of the quantum system. This is a fundamental feature of quantum mechanics, and not due to the fact that we haven't yet worked out how to improve the process. For a register of n qubits, a measurement on all qubits gives just one bit string out of the 2^n possibilities. So although such a register can in a sense hold (and thus process) an exponential amount of information, quantum measurement only permits a single result to be output. The expectation value (average) of some physical observable L over many measurements (with a repeatedly prepared single system, or an ensemble) is given by

$$\langle L \rangle = \langle \psi | L | \psi \rangle \quad (2.10)$$

If the system is in an eigenstate of L , a measurement of the corresponding observable L always gives the eigenvalue associated with that eigenstate. When the qubit is in a superposition of eigenstates, a single measurement of L cannot reveal the state perfectly and statistical investigation is needed [5].

2.4 Quantum gates

The most widely considered approach to QIP follows the three-stage recipe that has always been followed in quantum experiments. Prepare, evolve, then measure. The evolution section of QIP is where most of the action takes place. The gate model of QIP has a lot of similarity in its description to the abstract gate model of conventional

classical computing. We decompose the unitary operator prescribed by the algorithm into elementary unitary operations on qubits, which is analogous to the decomposition of a conventional computation into elementary logic gates operating on bits. It is important to note that in general one cannot simply make quantum versions of conventional logic circuits because these contain irreversible gates (between pairs of bits), such as AND and OR.

a	b	$a \wedge b$	$a \vee b$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Table 2.1: Truth table for classical AND and OR gates

As quantum evolution is unitary and reversible, these gates do not have quantum counterparts. However, although it is not generally used in practice, it is possible to do conventional computing reversibly. Since the deletion of bits by irreversible gates increases entropy, all garbage has to be kept in a reversible computation.

2.4.1 Comparison with classical gates

For one input bit, there are two possible classical reversible gates. One of them is NOT (table 2.2), the other is the identity gate, which maps its input to the output unchanged.

a	NOT a
0	1
1	0

Table 2.2: Truth table for classical NOT gate

For two input bits, the only non-trivial gate is the controlled-NOT (CNOT) gate, which XORs the first bit to the second bit and leaves the first bit unchanged (table 2.3).

Unfortunately, there are functions that cannot be computed using just those gates. In other words, the set consisting of these gates is not universal. To compute an arbitrary function using reversible classical gates, another gate is needed. One possibility is the

a	b	a CNOT b
0	0	00
0	1	01
1	0	11
1	1	10

Table 2.3: Truth table for classical CNOT gate

Toffoli gate: this gate has 3-bit inputs and outputs and its behaviour is to flip the third bit if the first two bits are set. The following is a table of the possible input and output:

a	b	c	OUTPUT
0	0	0	000
0	0	1	001
0	1	0	010
0	1	1	011
1	0	0	100
1	0	1	101
1	1	0	111
1	1	1	110

Table 2.4: Truth table for classical Toffoli gate

This concept straightforwardly introduces the gate model for quantum computation. Gates must have the same number of outputs as inputs, without deleting quantum information. Although practical considerations usually dictate the way a conventional computation is broken up into gates in a real machine, in principle, it is very convenient to know that there exists a universal set of gates. In order to better understand theory and the minimum requirements for hardware, it is very useful to know how to break up a computation into fundamental gates. In fact, the complexity of an algorithm or a process is essentially determined by the number of elementary gates needed to perform it, as a function of the size of the input n .

2.4.2 Notable quantum gates

The gates needed for universal quantum computing are known: it must be possible to perform arbitrary single qubit operations and some form of entangling operation

between pairs of qubits. As with conventional computing, there is no unique set of gates. So, it is useful to see first some notable example of quantum gates that perform unitary transformations.

For quantum gates, obviously there is no such a thing as a truth table as input variables are not just 0 and 1, but they can be represented by the unitary matrices which performs the transformation on the input state.

Starting with single qubit operations, useful gates are Pauli matrices (2.6). Among these, there is σ_x , which, if applied to classical bit would result in a NOT gate, but does not represent a universal NOT for quantum gates, as it only maps correctly the north pole and the south pole of the Bloch sphere.

The 2×2 Hadamard transform is another gate which is extensively used in quantum computing. It performs a one-qubit rotation, mapping the qubit basis states $|0\rangle$ and $|1\rangle$ to two superposition states with equal weight. This corresponds to the following transformation matrix:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \quad (2.11)$$

Passing on to two-qubits gates, notable example are SWAP and CNOT gates, which are characterized by the following matrices:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.13)$$

The controlled-NOT (CNOT) gate, in particular, is an important gate because the set of all one-qubit gates and the CNOT is the most common set of gates used to build quantum circuits. Other sets of universal gates with fixed phase rotation are known and can be used. However, these sets are not very efficient as they have to repeat fixed angles rotations to approximate the desired precision of a single qubit gate. Anyway, even if a given set of gates is universal, and therefore in principle it is sufficient for any type of computation, sometimes it is convenient to use a super-complete set to build more efficient circuits.

Chapter 3

Linear optical quantum computing

The basic building blocks of linear optics are beam splitters, half- and quarter-wave plates, phase shifters. In this chapter we will describe these devices mathematically and establish the relations that are used throughout the rest of the work.

Operations via linear optical elements (beam splitters and phase shifters, in this case) preserve the photon statistics of input light. For example, a coherent (classical) light input produces a coherent light output; a superposition of quantum states input yields a quantum light state output. Due to this reason, single photon source are used to analyze the effect of linear optical elements and operators.

An intrinsic problem in using photons as information carriers is that photons hardly interact with each other. This potentially causes a scalability problem for LOQC, since nonlinear operations are hard to implement, which can increase the complexity of operators and hence can increase the resources required to realize a given computational function.

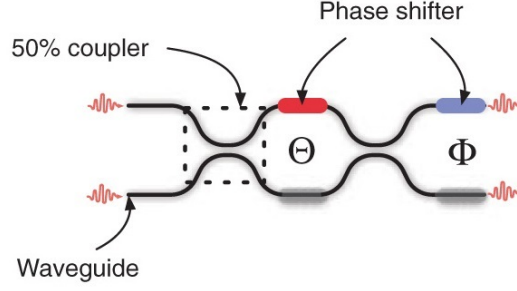


Figure 3.1: Circuit element

3.1 Linear optics building blocks

Although we have seen that CNOT gate together with the rotation of individual qubits constitutes a gate for a universal transform, a common method for constructing a reconfigurable mode transformation from N input modes into N output modes is to divide the problem into a network of transformations of 2×2 modes consisting of Mach–Zehnder interferometers (MZI) [6], which therefore constitute the building blocks of the circuit.

Each MZI consists of two 50:50 beam splitters and two parametrized phase shifters, as shown in figure 3.1. In integrated photonics platforms, beam splitters are commonly realized by directional couplers. Phase shifting, instead, is based on the local variation of the refractive index of a waveguide. This effect can be achieved through different physical mechanisms, depending also on the fabrication platform, but it is often convenient to exploit the thermo-optic effect. By modulating the temperature of the waveguide with a resistive microheater, it is possible to induce a local modification of the refractive index and, in turn, a phase shift on the guided photons. Such a component is usually referred to as thermal shifter.

The interferometer shown, therefore, overall applies the transformation dictated by the product of the following matrices, which represent its individual components and act in sequence:

$$BS_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} i & 1 \\ 1 & i \end{bmatrix} \quad (3.1)$$

$$PS_1 = \begin{bmatrix} e^{i\vartheta_1} & 0 \\ 0 & e^{i\vartheta_2} \end{bmatrix} \quad (3.2)$$

$$BS_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} i & 1 \\ 1 & i \end{bmatrix} \quad (3.3)$$

$$PS_2 = \begin{bmatrix} e^{i\varphi_1} & 0 \\ 0 & e^{i\varphi_2} \end{bmatrix} \quad (3.4)$$

$$U_{MZI} = PS_2 \cdot BS_2 \cdot PS_1 \cdot BS_1 \quad (3.5)$$

Here we assume the unit cell is lossless; accounting for losses requires each MZI to be described by a 4×4 matrix, rather than the 2×2 matrix considered here. Losses can be modeled by “virtual” beam splitters coupling the original mode and a “vacuum” mode. If such virtual beam splitters are included, the overall transformation can still be represented as a unitary transformation of order M , where $M > N$ accounts for the additional loss channels. The $N \times N$ transformation that applies to our input and output waveguide modes in that case is a nonunitary submatrix of $U(M)$.

3.2 Unitary transformation decomposition

In the general case of N input modes and N output modes, the Mach-Zehnder arrangement in order to obtain a generic unitary transformation was first studied by Reck et al. in [7], who identified a triangular arrangement, as shown in figure 3.2(a). After that, the arrangement was optimized by Clements et al. in [8], as shown in figure 3.2(b). This work demonstrated the possibility of realizing any transformation U through a more compact arrangement of these same elements. A further advantage of "Clements" arrangement is given by the symmetry of the number of elements acting on each line.

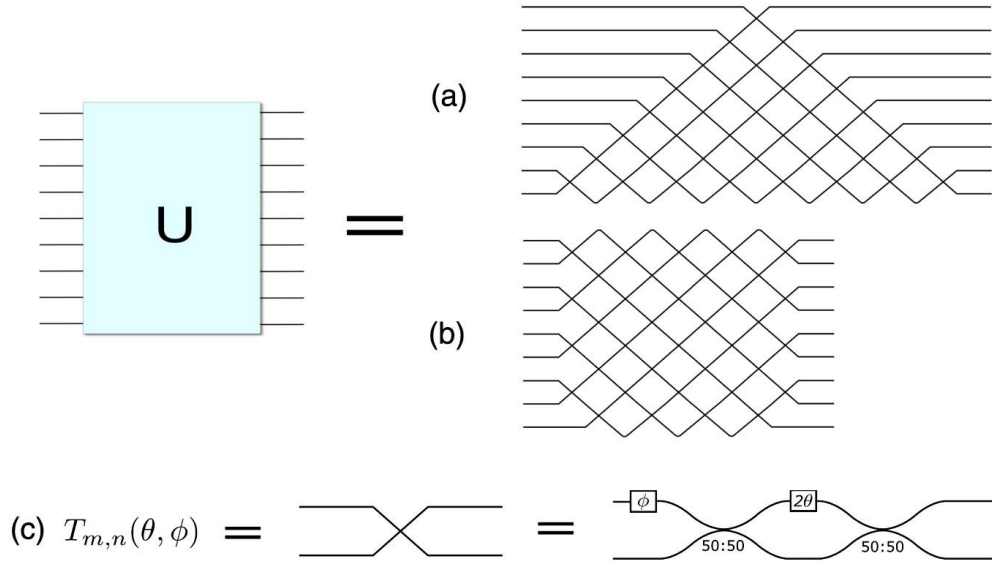


Figure 3.2: Reck's and Clements' decompositions

The algorithm with the steps to achieve "Clements" decomposition is shown in figure 3.3.

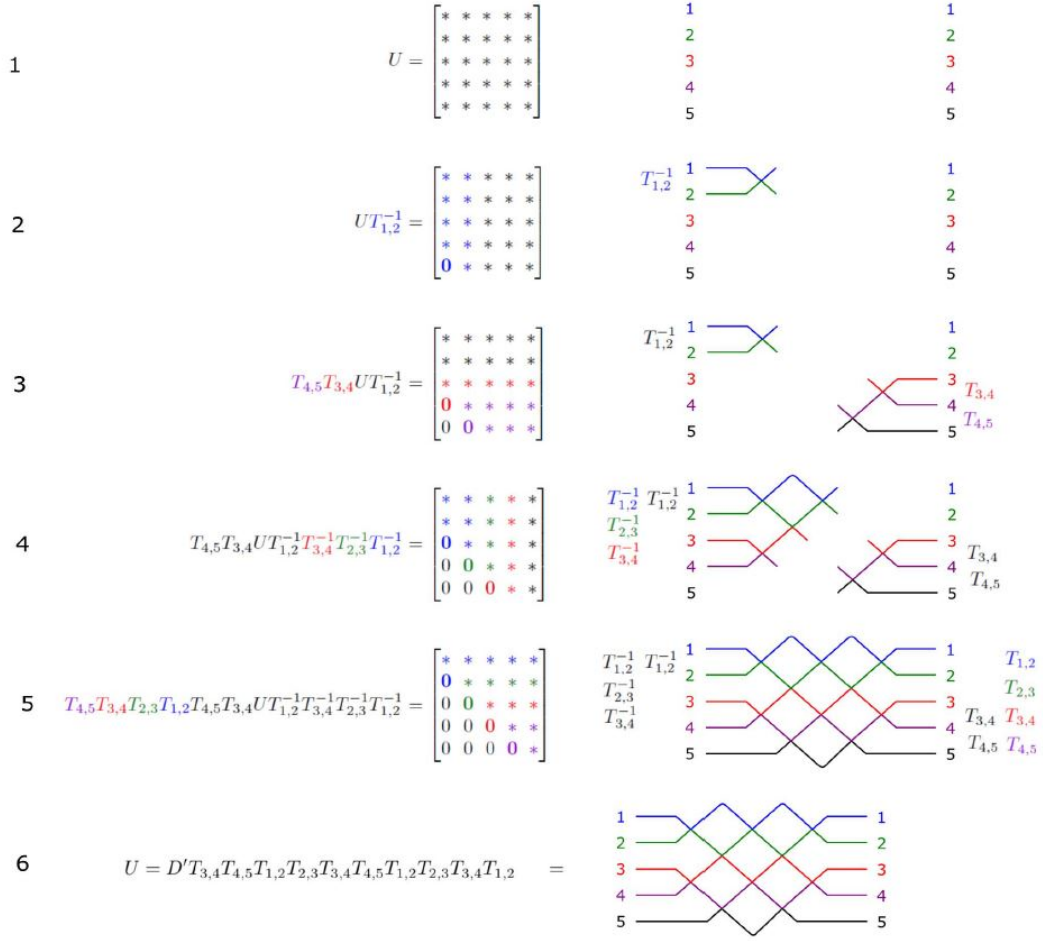


Figure 3.3: Clements' algorithm

Having seen these possible decompositions of a unitary transformation, the following step is to analyze from a theoretical point of view the behaviour of a real circuit with 4 input and 4 output modes, of which a practical implementation was available in order to compare expected response with experimental results and to carry out measurements.

3.3 Layout of a circuit performing unitary transformation

The circuit to be studied was developed by Quantum Technologies Laboratory of *"Scienze di Base e Applicate per l'Ingegneria"* (SBAI) Department at Sapienza University. The prototype has been built within *"COPERNICO"* project of *"Piano Nazionale di Ricerca Militare"* (PNRM). References and details can be found in proceedings [9], [10] and [11] by Bovino.

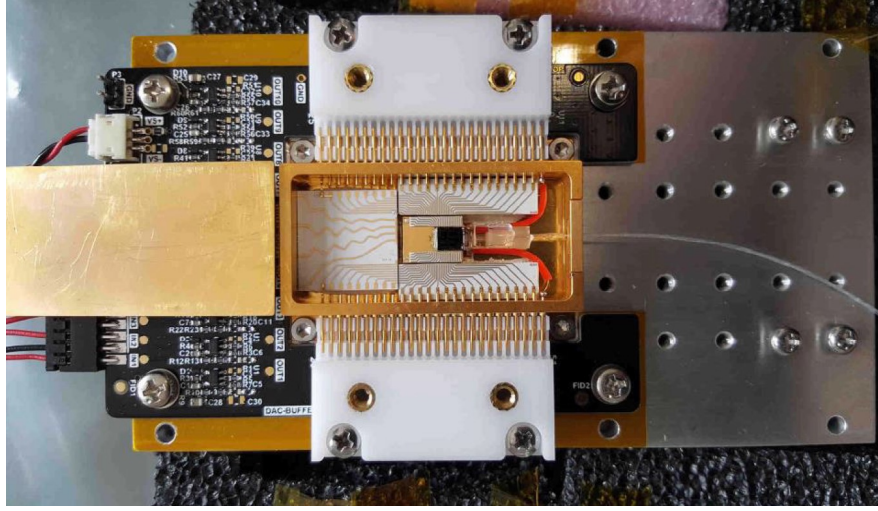


Figure 3.4: Optical, electronics and mechanics integration of InP circuit
Note the gold contacts for MZ voltage setting and control.

A photograph of the circuit is in figure 3.4, while the layout is represented in figure 3.5, and as can be noted, it follows Clements' decomposition scheme.

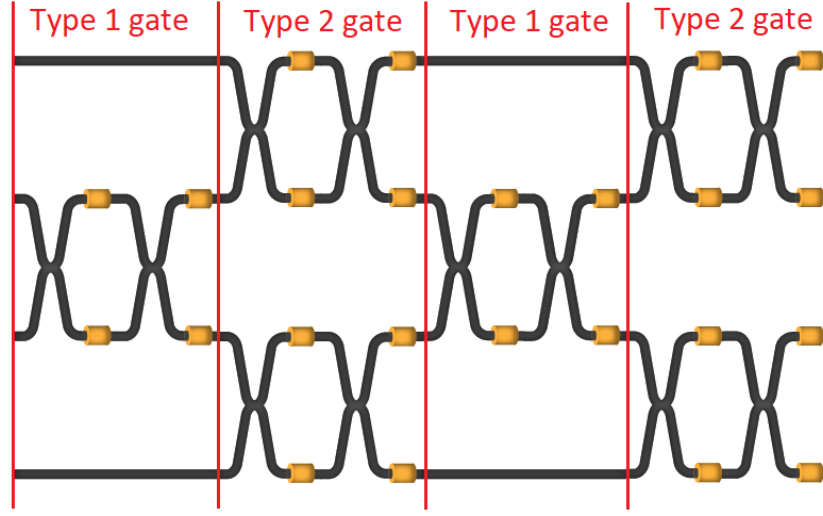


Figure 3.5: Circuit layout

For a deeper analysis it is useful to break it down by observing that it can be seen as the application of four stages in sequence and these four stages are equal in pairs. The theoretical analysis of this circuit therefore can be conducted by modeling the two types of gates, each through a 4×4 matrix, and evaluating the expected theoretical output given an input photon on one of the input modes. The varying parameter is the phase shift induced by the thermal shifters.

It is possible to observe that, for each stage, the first pair of phase shifters affects the output module, while the second pair may have the function of compensating for any unwanted phase shifts.

Let's see now, still from a theoretical point of view, the specific behavior of the two types of gates for some possible input state vectors, that is for an input photon on the first mode or on the second or third or fourth mode.

In figure 3.6, we can see type 1 gate, which constitutes the first and third stage of the circuit.

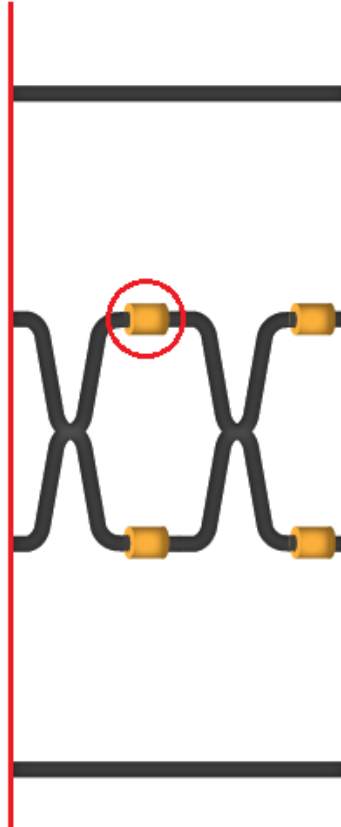


Figure 3.6: Circuit layout: focus on type 1 gate

For input vector $[1 \ 0 \ 0 \ 0]'$, as expected, the gate does not perform any transformation, in fact the photon enters the first mode and leaves from the first mode, also because at this stage the Mach-Zehnder acts only on modes 2 and 3. This expectation is supported from what is reported in the two diagrams in figure 3.7.

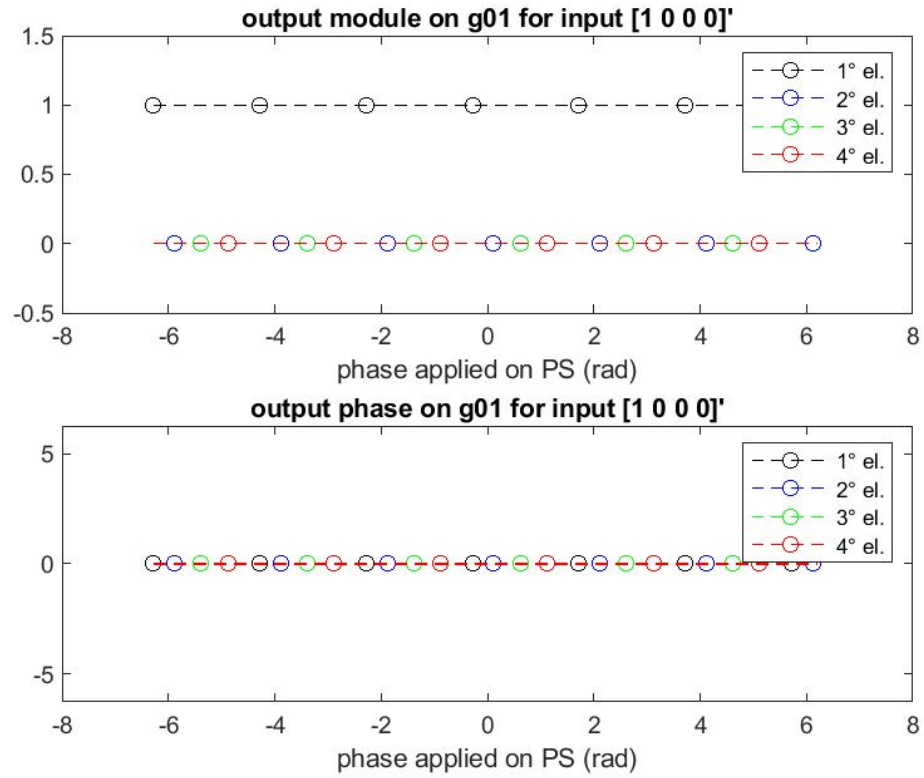


Figure 3.7: Type 1 gate output for $[1 \ 0 \ 0 \ 0]'$

On the other hand, when the input photon is on mode 2, varying the parameter on the phase shifter highlighted with a red circle, we obtain the output represented in figure 3.8. In details, when the phase shifter parameter is set to 0, this stage behaves like a SWAP gate, when instead it is set to π , it behaves like an IDENTITY gate.

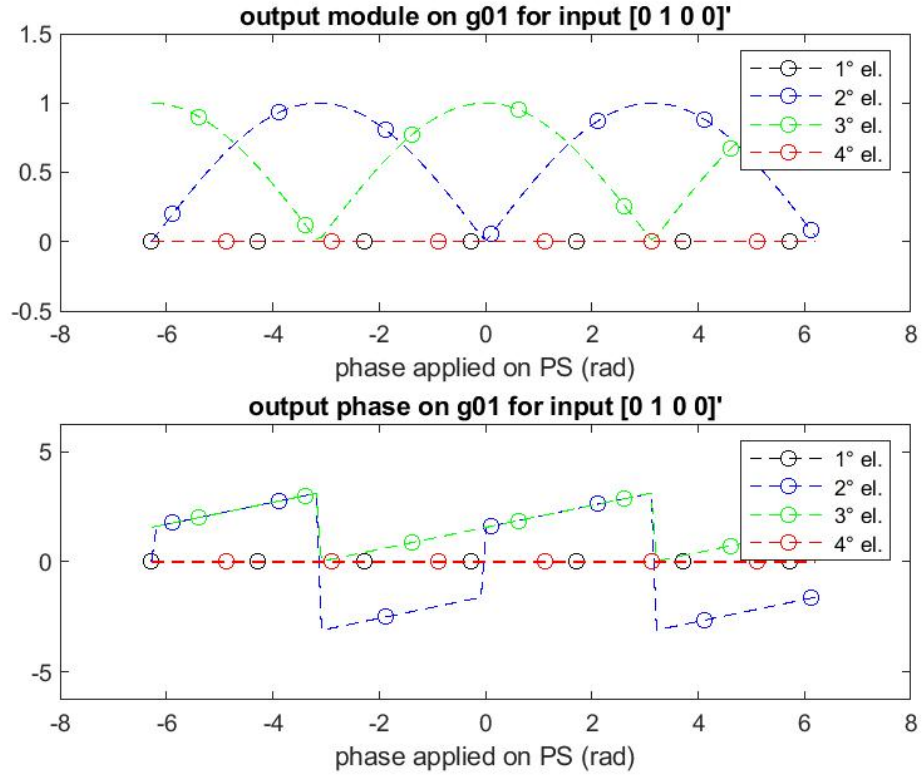


Figure 3.8: Type 1 gate output for $[0 \ 1 \ 0 \ 0]'$

Continuing the analysis of type 1 gate, we observe a similar behavior for the other two possible input vectors, that is $[0 \ 0 \ 1 \ 0]'$ and $[0 \ 0 \ 0 \ 1]'$. In particular, when the photon enters the 4th mode, the behavior can be mirrored to the 1st, and when it enters the 3rd mode, it behaves as a SWAP for phase shifter parameter set to 0 and as IDENTITY if it is set to π .

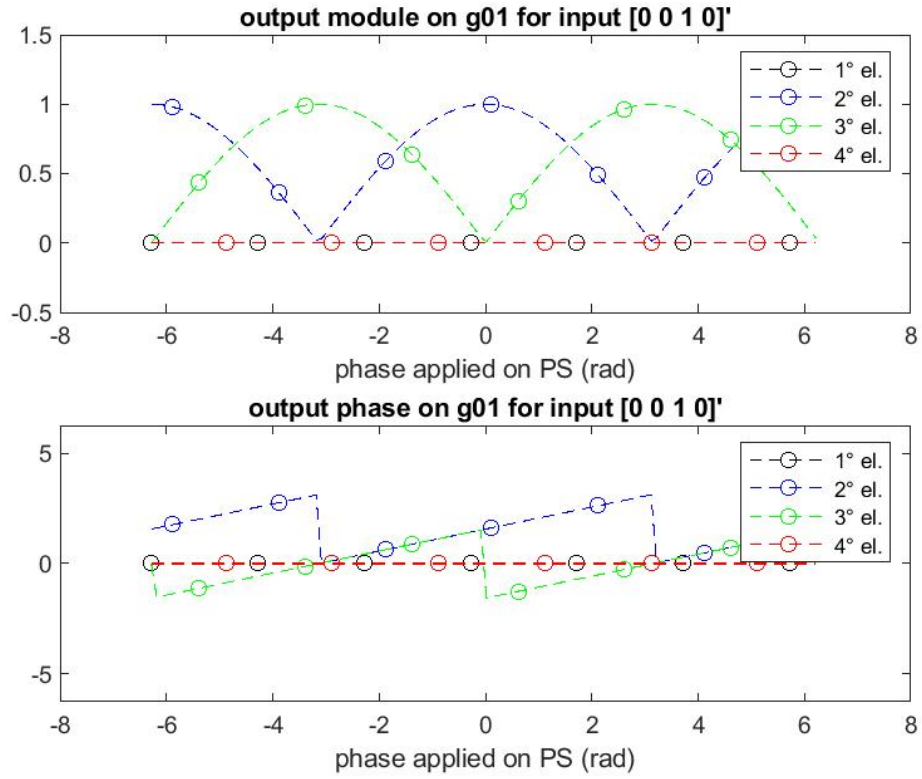


Figure 3.9: Type 1 gate output for $[0 \ 0 \ 1 \ 0]'$

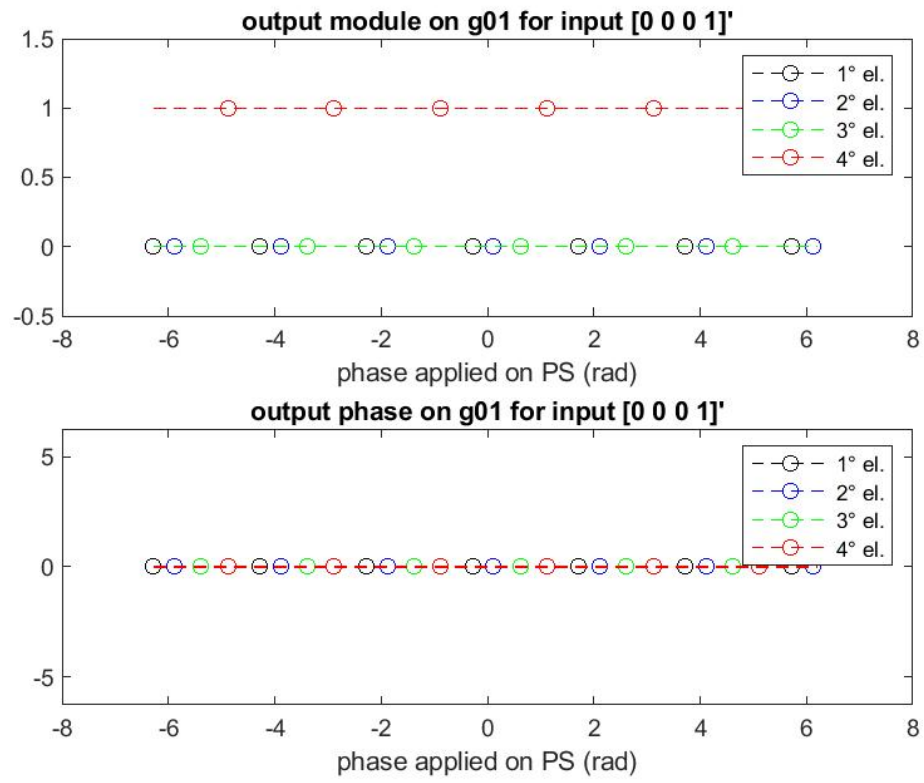


Figure 3.10: Type 1 gate output for $[0 \ 0 \ 0 \ 1]'$

Let's pass to type 2 gate. In this case there is a Mach-Zehnder operating on the first two modes and a second Mach-Zehnder operating on the third and fourth modes. The output follows the same pattern as already shown for modes 2 and 3 of type 1 gate.

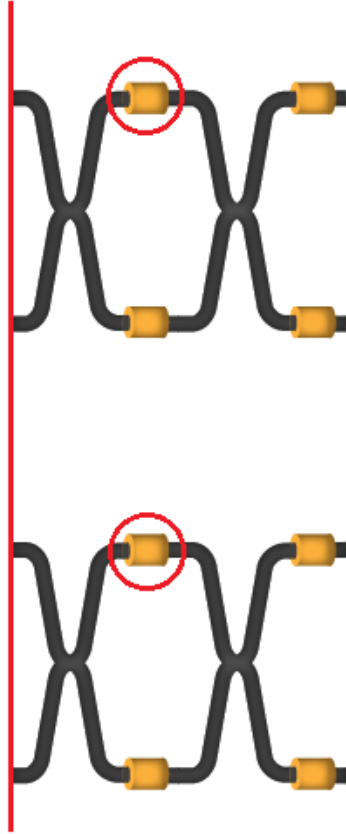


Figure 3.11: Circuit layout: focus on type 2 gate

In figures 3.12 and 3.13, there are output patterns when input vectors are $[1\ 0\ 0\ 0]'$ and $[0\ 1\ 0\ 0]'$, varying the parameters on the circled phase shifters.

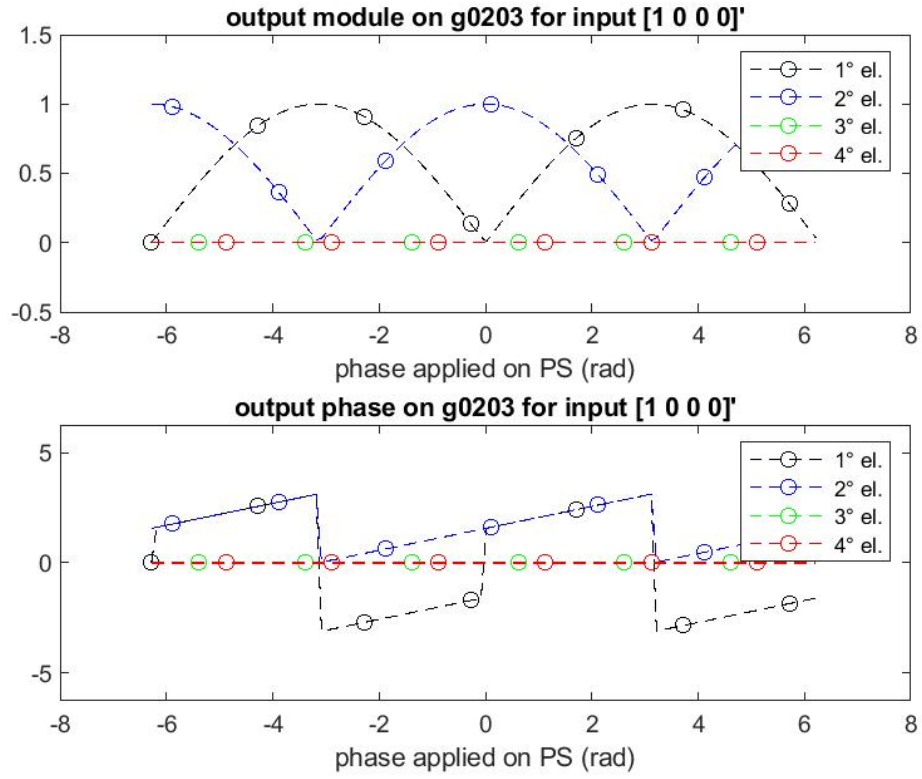


Figure 3.12: Type 2 gate output for $[1\ 0\ 0\ 0]'$

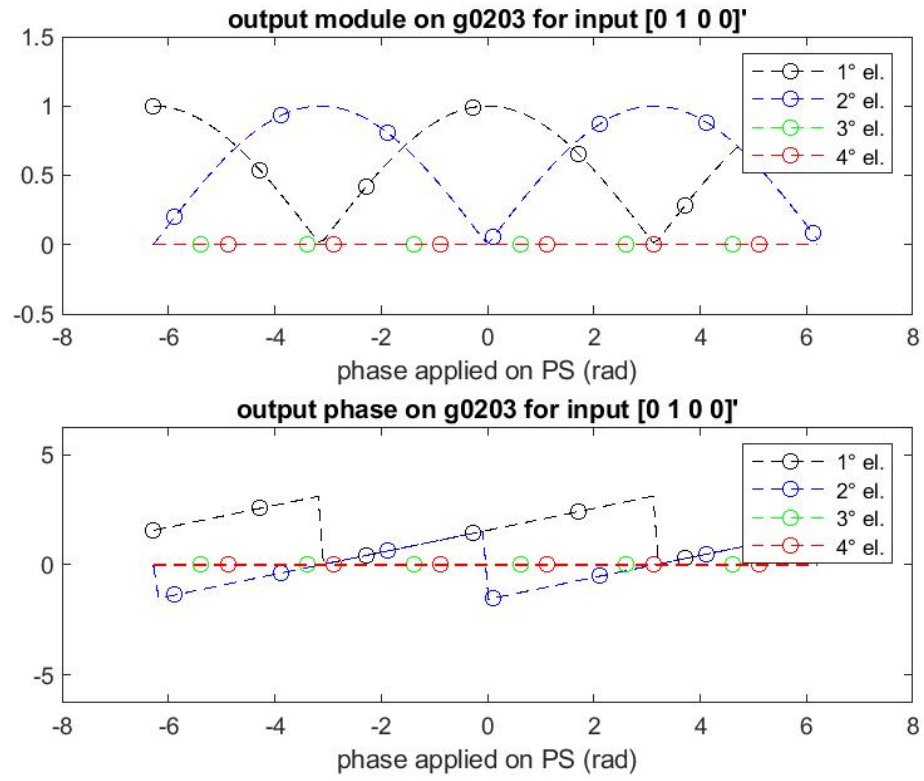


Figure 3.13: Type 2 gate output for $[0 \ 1 \ 0 \ 0]'$

Finally, in figures 3.14 and 3.15 we see the output when there are $[0\ 0\ 1\ 0]'$ and $[0\ 0\ 0\ 1]'$ as input state vectors.

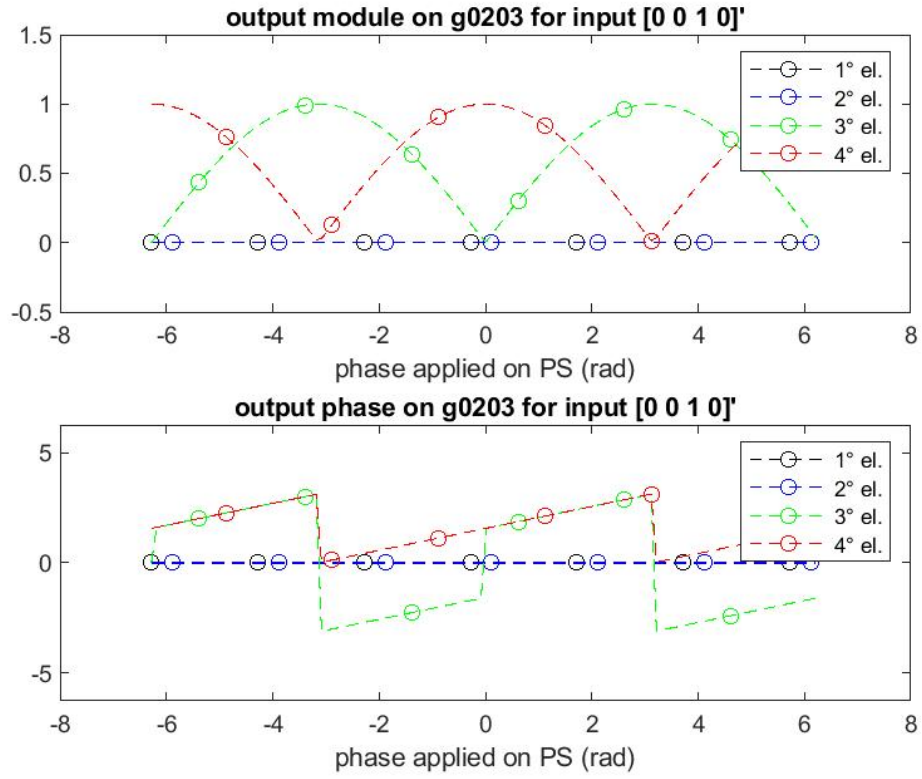


Figure 3.14: Type 2 gate output for $[0\ 0\ 1\ 0]'$

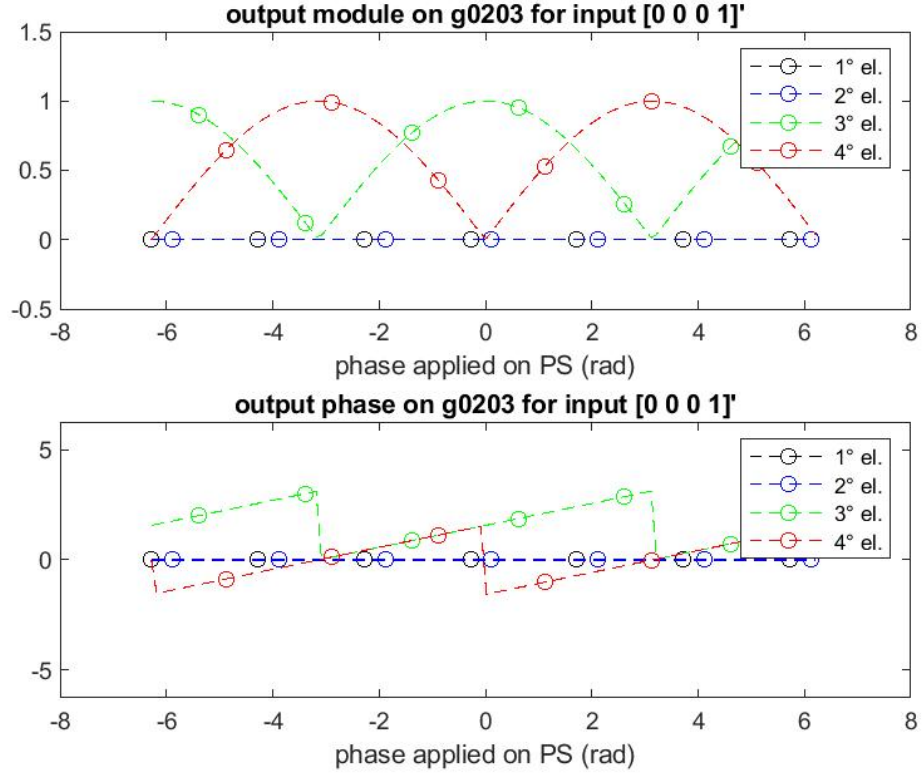


Figure 3.15: Type 2 gate output for $[0 \ 0 \ 0 \ 1]'$

3.4 Implementing notable gates

From the theoretical model, therefore, it is possible to derive the required values for the 24 parameters in order to build well-known gates. For example, if we want to set the circuit to implement an identical transformation (leave the input as it is), the values are shown in the following table:

		π	π			π	π
π	π	0	0	π	π	0	0
0	0	π	π	0	0	π	π
		0	0			0	0

Table 3.1: Parameter values for identical gate

Having more degrees of freedom than needed leaves the possibility to have more than one configuration to implement the same gates, for example an identical transform can be achieved also with the following configuration:

		0	0			0	0
π	0	0	0	π	0	0	0
0	0	0	0	0	0	0	π
		0	0			0	π

Table 3.2: Parameter values for identical gate - another option

Due to the fact that shifters are based on applying voltage to an integrated resistor, among different configurations for the same gate, the preferable is the one that generates less heat.

For a SWAP gate, the values are shown in table 3.3:

		π	π			π	π
π	π	0	0	0	$\pi/2$	0	π
0	0	π	π	0	$\pi/2$	π	0
		0	0			0	0

Table 3.3: Parameter values for SWAP gate

And for a CNOT gate:

		π	π			π	π
π	π	0	0	π	π	0	0
0	0	π	π	0	π	0	$\pi/2$
		0	π			0	$\pi/2$

Table 3.4: Parameter values for CNOT gate

As expected, the described circuit allows to implement all unitary transforms, including those described by the following matrices:

$$I \otimes H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (3.6)$$

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad (3.7)$$

$$H \otimes H = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (3.8)$$

For example, $I \otimes H$ is implemented with:

		$\pi/2$	$\pi/4$			0	0
π	π	π	$\pi/4$	π	π	0	π
0	0	$\pi/2$	$\pi/4$	0	0	0	0
		π	$\pi/4$			0	π

Table 3.5: Parameter values for $I \otimes H$ gate

Chapter 4

User interface development

After the theoretical analysis of the circuit, in this chapter we describe the implementation of the interface to control the circuit. From a practical point of view, the variation of the parameter that characterizes the phase shifters is induced by applying a voltage, experimentally determined in a range between 0 and -10 Volt. With 0 Volt applied, the Mach-Zehnder interferometer behaves like a SWAP gate (with a phase factor to be considered) while with -10 Volt, it behaves like an identity gate.

4.1 NI devices

The developed application controls the phase shifters through National Instrument "analog voltage output modules", specifically NI-9264 (in figure 4.1), connected to a computer through an appropriate chassis, for example NI-cDAQ-9184 (shown in figure 4.2) and a network interface. As every NI-9264 has 16 output lines, in order to control 24 devices, two modules are required. NI devices are shipped with a driver software that allows communication with a computer, as in figure 4.3.



Figure 4.1: NI-9264



Figure 4.2: NI-cDAQ-9184

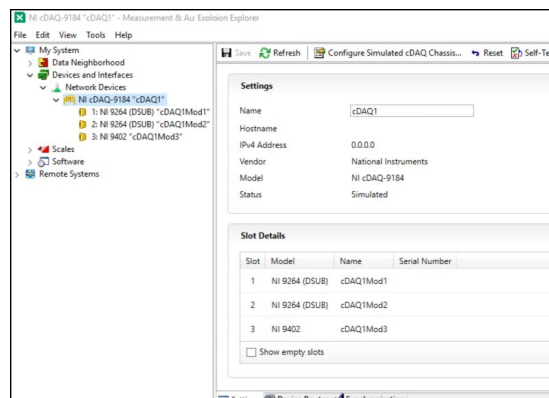


Figure 4.3: NI-MAX interface

4.2 Interface at startup

The user interface was implemented with MATLAB App Designer. After startup, it creates the required channels to control phase shifters, reads a configuration file with a table that links shifters' parameters with applied voltage. When initialization phase is finished, buttons are enabled.

In the form, it is possible to save and load files with a specific configuration of the gates using provided SAVE and LOAD buttons. It is also possible to select from a combo box in order to compute theoretical expected output. Finally, by clicking on RUN button, execution phase starts and the commands are actually sent to the circuit.

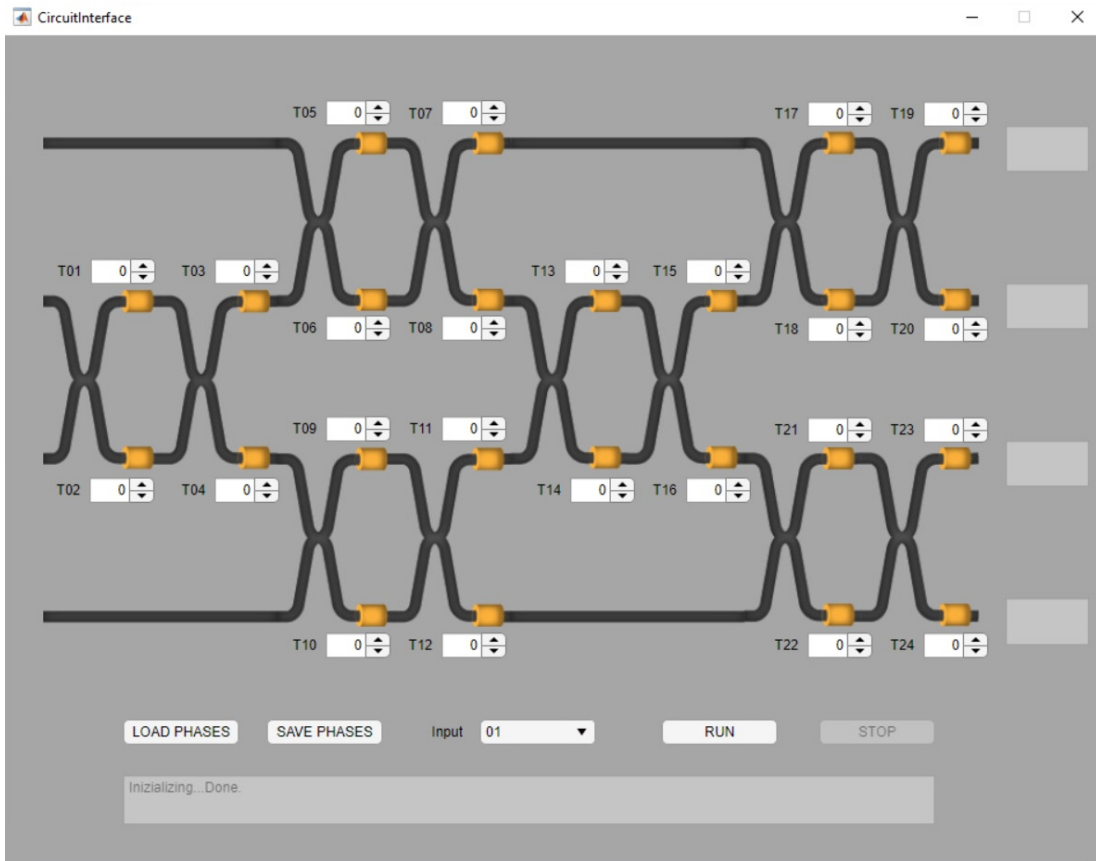


Figure 4.4: User interface at startup

4.3 Interface during execution

During execution phase, all the controls are disabled, apart from STOP button, and the expected module and phase values for the four output modes are shown in the text box on the right on the basis of the theoretical model of the circuit and a specific input selected from the combo box. These values can be compared with the readings on physical detectors (after performing an appropriate normalization of the latter).

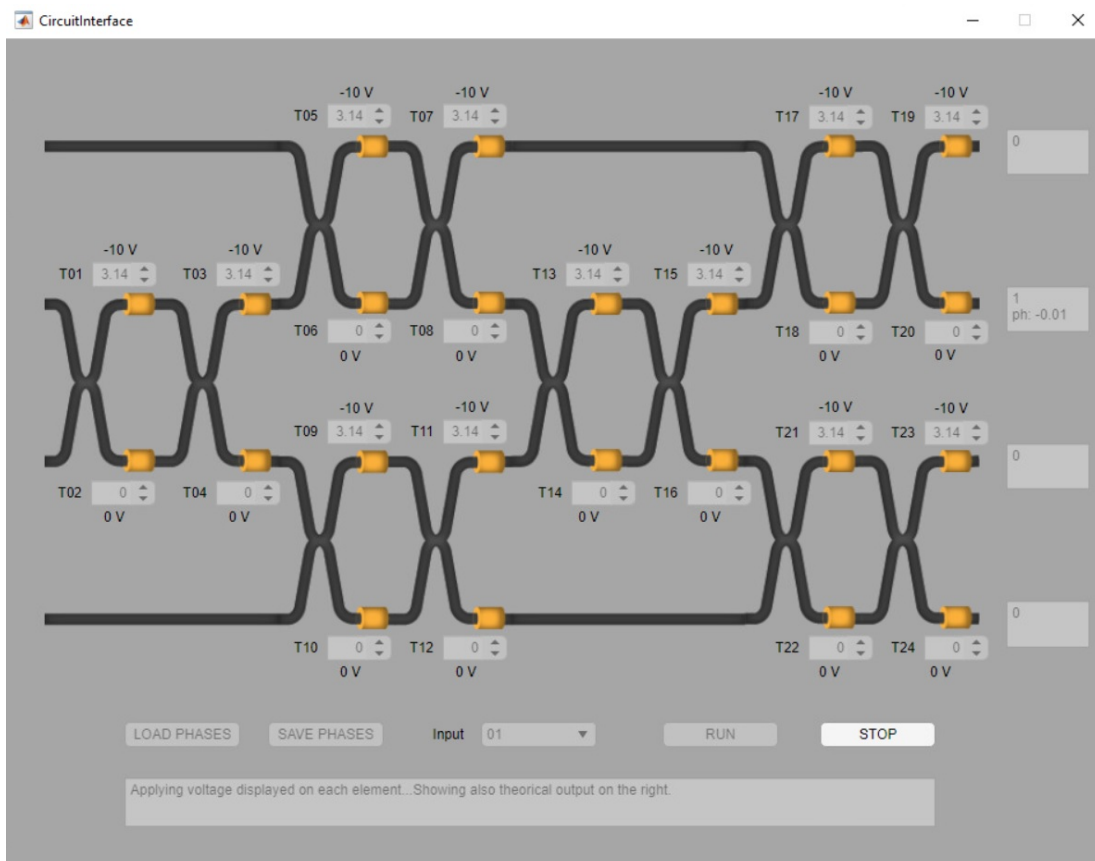


Figure 4.5: User interface while running

4.4 Comparison of results

Diagram in figure 4.6 represents the comparison between calculated output according to the theoretical model described, in red, and values obtained experimentally by varying the control voltage on the top left phase shifter of a single Mach-Zehnder interferometer. In this case, the plots are quite close to each other, but the advantage of the presented software is that it supports the possibility of setting custom voltage-shifting curves for each of the phase shifters to be controlled. The benefit of this approach is linked to the fact that components, although theoretically all having the same response, may have differences due to fabrication process. Therefore, it is possible to calibrate the program for its usage with a specific physical implementation of the circuit.

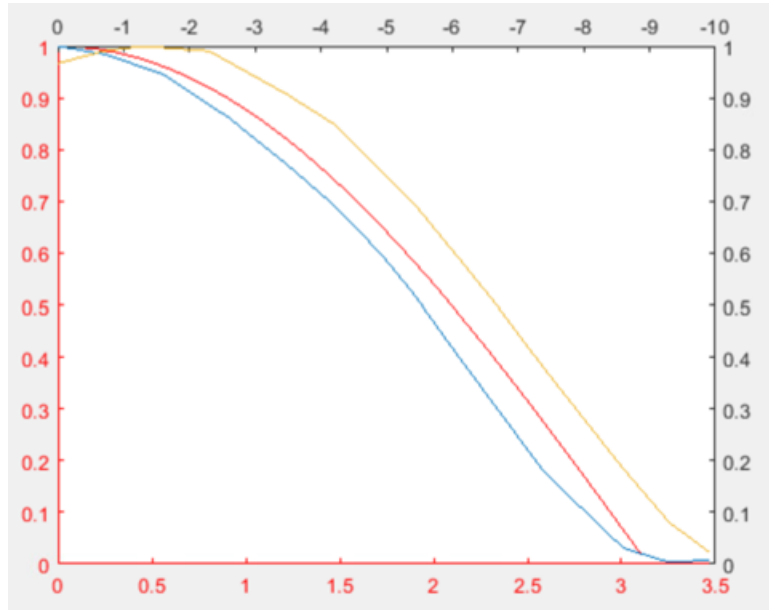


Figure 4.6: Comparison of theoretical and experimental response

Red plot shows theoretical output as a function of the shifter parameter for a single MZI, yellow and blue plots represent experimental result as a function of applied voltage.

Chapter 5

Conclusions

In conclusion, the developed application constitutes a proof-of-concept, showing a potential of quantum circuits based on linear optics, which is the possibility of creating a general purpose system and reconfiguring it to implement the algorithm of interest from time to time.

A practical application that can actually be used requires an overall miniaturization of the system, from the generation of the input to the detection of the output and to be able to work with a greater number of qubits while dealing with the propagation of errors.

Although there are these significant technological challenges, research is making progress on several fronts, mostly because the development of quantum computers would have a major impact in a lot of areas of great relevance, including cryptography, secure communications and optimization problems.

Bibliography

- [1] R. Feynman. Quantum mechanical computers. *Optics News*, 11:11–20, 1985.
- [2] Peter Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings of 35th Annual Symposium on Foundations of Computer Science*, 10 1996.
- [3] Pieter Kok, William Munro, Kae Nemoto, T. Ralph, Jonathan Dowling, and G. Milburn. Review article: Linear optical quantum computing. *Review of Modern Physics*, 79, 01 2006.
- [4] Todd Pittman, Bryan Jacobs, and J. Franson. Quantum computing using linear optics. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 25, 07 2004.
- [5] Timothy P. Spiller, William J. Munro, Sean D. Barrett, and Pieter Kok. An introduction to quantum information processing: applications and realizations. *Contemporary Physics*, 46(6):407–436, 2005.
- [6] Nicholas C. Harris, Jacques Carolan, Darius Bunandar, Mihika Prabhu, Michael Hochberg, Tom Baehr-Jones, Michael L. Fanto, A. Matthew Smith, Christopher C. Tison, Paul M. Alsing, and Dirk Englund. Linear programmable nanophotonic processors. *Optica*, 5(12):1623–1631, Dec 2018.
- [7] Michael Reck, Anton Zeilinger, Herbert Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical review letters*, 73:58–61, 08 1994.

- [8] William Clements, Peter Humphreys, Benjamin Metcalf, W. Kolthammer, and Ian Walmsley. An optimal design for universal multiport interferometers. *Optica*, 3, 03 2016.
- [9] Fabio Antonio Bovino. On chip intrasystem quantum entangled states generator. In *2017 Conference on Lasers and Electro-Optics Europe and European Quantum Electronics Conference (CLEO/Europe-EQEC)*, pages 1–1, 2017.
- [10] Fabio Antonio Bovino. Intrasystem entanglement generator and unambiguos bell states discriminator on chip. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7993–7997, 2019.
- [11] Fabio Antonio Bovino. Practical quantum communication and processing. In *2021 Conference on Lasers and Electro-Optics Europe and European Quantum Electronics Conference (CLEO/Europe-EQEC)*, pages 1–1, 2021.

Ringraziamenti

Ringrazio l'Aeronautica Militare ed AFCEA, per avermi dato la possibilità di frequentare questo Master.

Inoltre, un sentito ringraziamento va ai docenti e a tutta l'organizzazione, per gli insegnamenti e per averci garantito una flessibilità tale da consentirci di seguire le lezioni e portare avanti gli impegni lavorativi.

Infine, ringrazio mio padre Lorenzo, mia madre Maria Teresa e mia sorella Paola, che mi sostengono sempre.

Roma, 26 Luglio 2022

Pietro De Lellis
