Architecture-Based Mapping Algorithm for Multi-Core Quantum Computers

Master Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Pau Escofet i Majoral

In partial fulfilment of the requirements for the
Master's degree in Advanced Telecommunication Technologies

Advisors: Sergi Abadal Cavallé, Carmen García Almudéver
Barcelona, July 2023

Contents

List of Figures 3

List of Tables 4

1 Introduction 6

2 Background 8
2.1 Quantum Computing . . . 8
2.1.1 Quantum Bits . . . 8
2.1.2 Quantum circuits . . . 9
2.2 Quantum Computing Architectures . . . 11
2.2.1 Multi-Core Quantum Computers . . . 12

3 Quantum Circuit Mapping 14
3.1 QMap . . . 14
3.2 FGP-rOEE Algorithm . . . 16
3.2.1 Flaws of FGP-rOEE Algorithm . . . 18
3.3 The need for a complete mapping algorithm . . . 19

4 Methodology 20
4.1 Input . . . 20
4.2 Inter-Core Mapping . . . 20
4.3 Single-Core Circuit Split . . . 20
4.4 Intra-Core Mapping . . . 21
4.5 Evaluation . . . 22

5 Characterization of Non-Local Communications for Quantum Random Circuits 24
5.1 Non-local communications bounds . . . 24
5.2 Analyzing FGP-rOEE performance . . . 27

6 Proposed inter-core mapping algorithm 29

7 Results 34
7.1 Inter-Core Mapping evaluation . . . 34
7.2 Complete Mapping evaluation . . . 35

8 Conclusions and Future Work 38
8.1 Future Work . . . 38

References 40

Appendices 43

List of Figures

1 Bloch sphere . . . 9
2 Quantum circuit example . . . 11
3 Couplimg maps of IBM Boeblingen (left) and IBM Q Tokyo (right) . . . 12
4 Multi-core quantum architecture [1] . . . 12
5 Teleportation protocol . . . 13
6 QMap overview from [2] . . . 15
7 Interaction graphs for the three example timeslices . . . 17
8 2-Partitioning of the interaction graphs . . . 17
9 Communications graph between timeslices 1-2 (left) and 2-3 (right) . . . 18
10 Communications graph between two different timeslices . . . 19
11 Qubit index correction for splitting circuits into core-circuit example . . . 21
12 Quantum core with computation (orange) and communication qubits (blue) 22
13 EPST of a (96, 1000, 0.8)-Random Circuit obtained with the complete mapping pipeline . . . 23
14 Non-local communications for a (96, 5000, 0.8)-Quantum Random Circuit . 27
15 Non-local communications for a (96, 5000, 0.6)-Quantum Random Circuit . 27
16 Non-feasible two-qubit operations . . . 29
17 Unassigned qubits . . . 30
18 One of the unfeasible two-qubit operation assigned to each core . . . 31
19 Valid assignment of qubits into cores . . . 32
20 FGP-rOEE and proposed non-local communications for a (96, 5000, 0.8)-Quantum Random Circuit . . . 34
21 FGP-rOEE and proposed non-local communications for a (96, 5000, 0.6)-Quantum Random Circuit . . . 35
22 Complete mapping using both FGP-rOEE and Our Approach of a (96, 1200, 0.8)-Quantum Random Circuit. Improvement is shown in Table 5 . . . 36
23 Complete mapping using both FGP-rOEE and Our Approach of a (96, 1200, 0.6)-Quantum Random Circuit. Improvement is shown in Table 6 . . . 37
24 Complete mapping using both FGP-rOEE and Our Approach of a (96, 1200, 0.4)-Quantum Random Circuit. Improvement is shown in Table 7 . . . 37
25 Remote quantum operation using classical communications and an entangled quantum state from [3] . . . 39
26 Complete mapping using both FGP-rOEE and Our Approach of a (72, 700, 0.8)-Quantum Random Circuit . . . 43
27 Complete mapping using both FGP-rOEE and Our Approach of a (72, 700, 0.6)-Quantum Random Circuit . . . 44
28 Complete mapping using both FGP-rOEE and Our Approach of a (72, 700, 0.4)-Quantum Random Circuit . . . 44
29 Complete mapping using both FGP-rOEE and Our Approach of a (48, 700, 0.8)-Quantum Random Circuit . . . 45
30 Complete mapping using both FGP-rOEE and Our Approach of a (48, 700, 0.6)-Quantum Random Circuit . . . 45
31 Complete mapping using both FGP-rOEE and Our Approach of a (48, 700, 0.4)-Quantum Random Circuit . . . 46

List of Tables

1 Quantum Gates 10
2 Average fidelities of quantum operations 23
3 Example cost matrix with five operations 30
4 Example cost matrix with one operation 31
5 Improvement (%) of non-local communications and EPST when using the proposed approach for a (96, 1200, 0.8)-Quantum Random Circuit 35
6 Improvement (%) of non-local communications and EPST when using the proposed approach for a (96, 1200, 0.6)-Quantum Random Circuit 35
7 Improvement (%) of non-local communications and EPST when using the proposed approach for a (96, 1200, 0.4)-Quantum Random Circuit 36

Abstract

Current quantum processors must be scaled to match the computational power needed for quantum algorithms that present quantum advantage. It is not feasible to keep integrating more qubits on a single core because qubit crosstalk and decoherence make the computations too noisy. Multi-core approaches have been proposed as a feasible, scalable alternative. Optimizing the mapping of quantum algorithms onto multi-core quantum computers is crucial for improving performance and scalability in quantum computing systems. This thesis presents a pipeline for computing the complete mapping of quantum algorithms into multi-core quantum computers, considering both intra-core and inter-core constraints. The existing state-of-the-art inter-core mapping algorithms are evaluated and found to be suboptimal for achieving near-optimal inter-core communications. An alternative inter-core mapping algorithm is proposed to address this limitation, demonstrating improved effectiveness in allocating quantum operations across multiple cores. A thorough comparison of the proposed algorithm and existing approaches is conducted using the complete mapping pipeline. The findings emphasize the importance of developing efficient inter-core mapping algorithms. The research outcomes contribute to advancing the field of quantum computing and provide valuable insights for optimizing performance and scalability in multi-core quantum computers.

1 Introduction

Quantum computing has emerged as a groundbreaking paradigm with the potential to revolutionize various fields, ranging from cryptography to optimization problems. One key aspect of unlocking the full potential of quantum computing is the scalability of quantum processors [4]. A feasible approach for scaling quantum computers is to go from a single-core quantum computer to a multi-core approach [5], [6], [7]. This comes with new challenges, such as the problem of mapping quantum algorithms onto physical quantum devices, a compilation process that adapts a quantum algorithm to be executed in the desired quantum architecture, which is at the core of this cutting-edge technology. Many mapping techniques have been developed for single-core quantum computers. However, only a few mapping strategies focus on multi-core architectures.

This thesis addresses the critical task of mapping quantum algorithms into multi-core quantum computers. The goal is to propose a comprehensive pipeline that computes the complete mapping of quantum algorithms, allowing for the efficient allocation of quantum circuits across multiple processing cores, considering intra-core and inter-core constraints.

Few inter-core mapping algorithms [8] have been developed to address these challenges, with strong assumptions on the cores' constraints. Their optimality in mapping quantum algorithms across multiple cores has not been rigorously examined. Therefore, a vital objective of this thesis is to assess the effectiveness of current inter-core mappers and identify areas for improvement.

We propose an alternative inter-core mapping algorithm based on the observed limitations of current mapping techniques for multi-core architectures. It is designed to achieve fewer inter-core communications, facilitating improved scalability and performance. The performance of the proposed inter-core mapping algorithm is rigorously compared to existing state-of-the-art approaches using the complete mapping pipeline to obtain a reliable and robust metric.

The goals of this work can be summarized as follows:

- Development of the first complete quantum circuit mapping technique for modular architectures that considers both inter-core and intra-core connectivity constraints.
- Characterization of the non-local communications needed to map quantum random algorithms into a modular architecture, aiming to asses current state-of-the-art inter-core mappers.
- Development of a novel inter-core mapping algorithm to reduce the number of inter-core communications needed when mapping a quantum circuit into a modular architecture.

Regarding the structure of the thesis, Section 2 introduces the basic concepts of quantum computing and provides an overview of both single-core and multi-core quantum computing architectures. Section 3 further discusses the quantum circuit mapping problem and presents two state-of-the-art mapping algorithms, one regarding the distribution of qubits into a multi-core architecture and the other designed for single-core quantum computers. Section 4 proposes a pipeline to compute the complete mapping of a quantum

circuit for a parametrizable modular architecture. Section 5 derives a characterization of the expected number of inter-core communications for random quantum circuits and assesses the state-of-the-art inter-core mapping algorithm discussed in Section 3. Section 6 presents an alternative approach for performing inter-core mapping, which is compared with the state-of-the-art inter-core mapping algorithm (discussed in Section 3) in Section 7. Lastly, Section 8 exposes the conclusions of this thesis and presents future lines of work.

2 Background

This section gives background information on this thesis' various subject areas. We outline the fundamentals of quantum computing, including what quantum bits are, how we can use them for computation, and the foundations upon which quantum algorithms are built. We also review the fundamentals of quantum architectures, emphasizing on the connectivity constraints, a key aspect of this thesis.

2.1 Quantum Computing

Quantum computing is a new paradigm of computation that uses principles of quantum physics, such as superposition and entanglement, to solve computational problems. By leveraging these properties, quantum computers can solve complex problems that are intractable by classical computers. Some fields of application for quantum computers would be cryptography, mathematical optimization, machine learning, and simulation of quantum systems, among others [9], [10], [11], [12].

2.1.1 Quantum Bits

Quantum bits (or qubits) are the fundamental unit of information in quantum computing, analogous to the bit in classical computing. The main difference between bits and qubits is that, while bits can only have two constant values, qubits can be in both states simultaneously thanks to a quantum mechanics property called superposition. A quantum state $|\psi\rangle$ is represented as a linear combination of the basis states $|0\rangle$ and $|1\rangle$ in the form:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad \alpha, \beta \in \mathbb{C} \tag{1}$$

Where the coefficients α and β are the so-called probability amplitudes. Quantum mechanics does not allow us to measure superposition states directly, and when reading a quantum state, this will collapse into either the state $|0\rangle$ or the state $|1\rangle$, with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively. Since the sum of probabilities of a system must be equal to one, it is required that $|\alpha|^2 + |\beta|^2 = 1$.

For a more visual and intuitive understanding of the state of a qubit, a quantum state can also be represented as a vector $|\psi\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$, and by defining the amplitudes α and β in the form of $\alpha = \cos\left(\frac{\theta}{2}\right)$ and $\beta = e^{i\phi}\sin\left(\frac{\theta}{2}\right)$, the quantum state can be displayed on the Bloch sphere (see Figure 1). As the vector $|\psi\rangle$ is 2-dimensional, complex and unitary, the values of α and β are limited, so the quantum state always lies on the surface of the Bloch sphere.

Several qubits can be combined, and the resulting quantum state composed of N qubits has 2^N different amplitudes and can be represented as:

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_3\rangle \otimes ... \otimes |\psi_n\rangle = |\psi_1\psi_2\psi_3...\psi_n\rangle \tag{2}$$

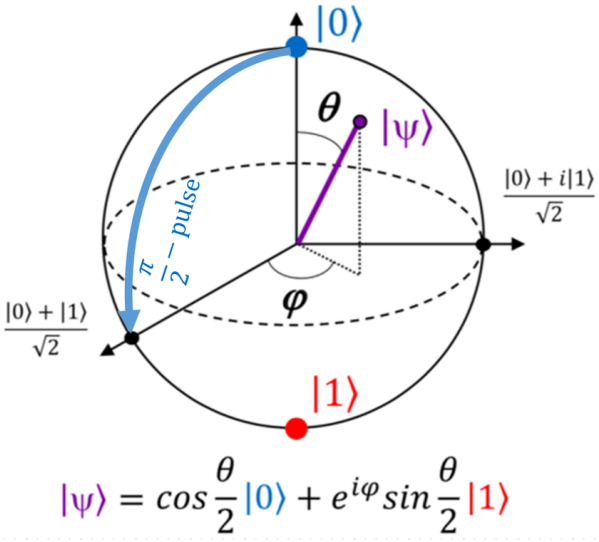


Figure 1: Bloch sphere

Which is equivalent to:

$$|\Psi\rangle = \alpha_1 |0...00\rangle + \alpha_2 |0...01\rangle + \alpha_3 |0...10\rangle + ... + \alpha_{2^N} |1...11\rangle \tag{3}$$

Recall that the sum of the squared amplitudes must be 1, no matter the number of qubits the quantum system has.

$$\sum_{i=1}^{N} |\alpha_i|^2 = |\alpha_1|^2 + |\alpha_2|^2 + ... + |\alpha_N|^2 = 1 \tag{4}$$

Another important quantum property we must review is entanglement. It refers to a relationship between two or more quantum states, where their properties become correlated so they are no longer independent entities. When two quantum particles are entangled, their state cannot be expressed as a tensor product of independent states, as in Equation (2) and must be expressed jointly. Equation (5) shows an example of an entangled state of two qubits. More precisely, it shows the maximally entangled state.

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{5}$$

2.1.2 Quantum circuits

Classical bits are processed by gates such as AND, OR, NOT, XOR and so on. Analogously, quantum states can be modified or changed by applying quantum gates. The resulting quantum state after applying a quantum gate must also be valid. Therefore, quantum gates are unitary operators described as unitary matrices relative to some basis. A matrix U is unitary if it satisfies the following:

$$U^{\dagger}U = UU^{\dagger} = I \tag{6}$$

Quantum gates can be performed to a single qubit or multiple qubits. The size of the unitary matrix for the gate relies on the number of qubits it operates on; it is a square matrix with dimensions of $2^n \times 2^n$, where n represents the number of qubits involved.

Table 1 shows the most relevant single and two-qubit gates and their symbol and corresponding matrix.

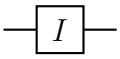
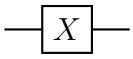
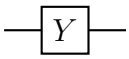
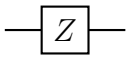
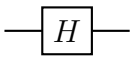
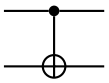
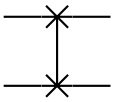
Operator	Gate	Matrix
Identity-I		$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Pauli-X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y		$\begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}$
Pauli-Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Haddamard		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
CNOT		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 1: Quantum Gates

A subset of the two-qubit gates called controlled gates operates only in one qubit, depending on the state of the other. For example, the Controlled-NOT or CNOT applies an X gate to the target qubit ($\bigoplus$) if the control qubit is in state $|1\rangle$. Another example of a controlled gate is the Controlled-Phase (or Controlled-Z).

Another two-qubit gate worth mentioning is the SWAP gate, as it will be highly used in future sections of this thesis. The SWAP gate involves two qubits, swapping each qubit's

quantum state.

Quantum algorithms are often represented as quantum circuits. A quantum circuit is a collection of actions carried out simultaneously and coherently on several qubits. These operations can be initialization, measurements of qubits, or universal quantum gates. Quantum circuits can represent any quantum program or quantum algorithm. Figure 2 illustrates a quantum circuit with four qubits, where several quantum gates are applied to the qubits and all qubits are measured at the end of the circuit.

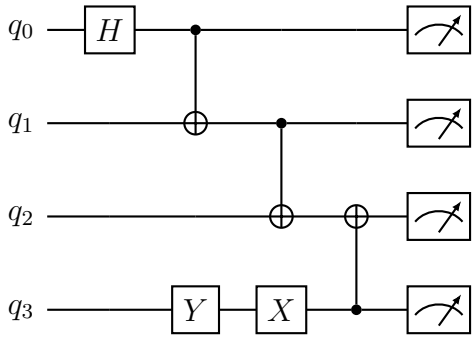


Figure 2: Quantum circuit example

In order to execute a quantum circuit, a quantum architecture with a number of qubits equal to or higher than those used in the circuit is required. We usually refer to the qubits from quantum computers as physical qubits and the qubits from circuits as virtual qubits.

2.2 Quantum Computing Architectures

The development of quantum computers over the past few years has been remarkable, going from theory to reality and from a few qubits to hundreds of qubits [13]. Since quantum computers are still in their infancy, there is still a long way to go. There are several quantum technologies regarding qubit implementation, and there is still uncertainty on whether one will prevail. Although the most advanced qubit technologies in terms of the number of qubits are superconducting [14] and trapped ions [15], there are other technologies that are being used for quantum hardware implementation, such as diamond vacancies [16], quantum dots [17] and neutral atoms [18].

Quantum computing is currently in the so-called noisy intermediate-scale quantum (NISQ) era, characterized by quantum processors with hundreds of qubits. These computers cannot yet do continuous quantum error correction due to the number of qubits they contain.

A NISQ quantum computer can be characterized by the number of physical qubits, the connectivity of those qubits, the decoherence time of the qubits, the gate operation time and the fidelity of each quantum gate. The connectivity of the qubits of the quantum processor (or coupling map) is a graph with the qubits as nodes and an edge between two qubits if the processor supports a two-qubit gate involving that pair of qubits. An example of two coupling maps of IBM's quantum processors are depicted in Figure 3.

The computational power of quantum computers increases as so it does the number of qubits. However, physical constraints do not allow the addition of qubits without limit into a single processor, as crosstalk between qubits increases, making the processor too noisy to be reliable. Various research initiatives are pursuing the objective of developing a quantum processor with significantly greater capacity than current NISQ devices. Among these research initiatives, one particularly promising approach involves combining multiple quantum processors to create a larger and more powerful processor, thus entering into multi-core architectures.

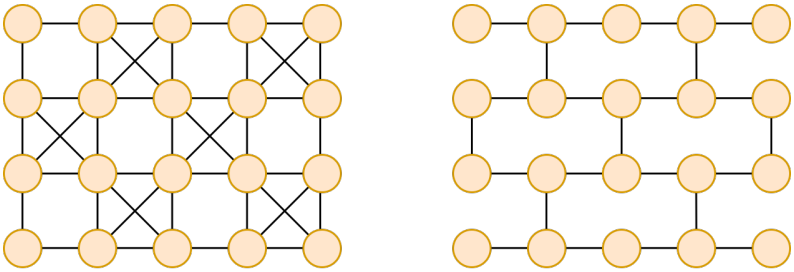


Figure 3: Couplimg maps of IBM Boeblingen (left) and IBM Q Tokyo (right)

2.2.1 Multi-Core Quantum Computers

To solve the scalability problem, several multi-core approaches have been proposed [19, 6, 20, 1, 21, 22]. The main goal is to increase the computational power of the quantum processor without increasing the crosstalk between qubits by having different quantum processors (or cores) connected via classical links (early stages) or by coherent quantum links (later stages). An example of a multi-core architecture proposed in [1, 21] is shown in Figure 4.

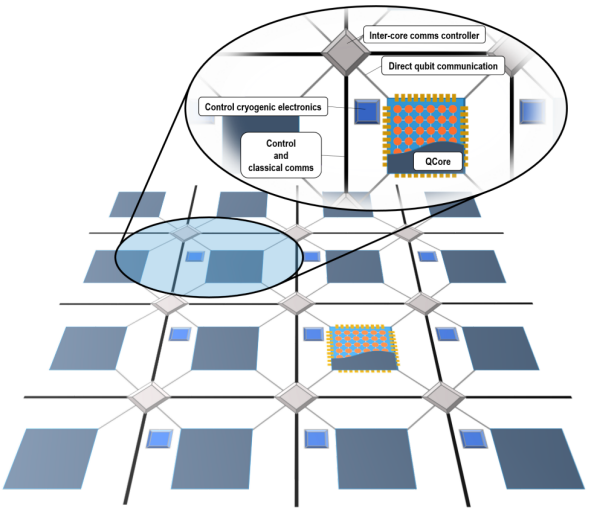


Figure 4: Multi-core quantum architecture [1]

With multi-core architectures, several new challenges arise that include the execution of

two-qubit gates between qubits that are physically located on different quantum processors. As two-qubit gates cannot be performed when qubits are in different cores, we need to find a way for the core to exchange quantum states. The *no-cloning theorem* is a fundamental principle in quantum mechanics that states that it is impossible to create an identical copy of an arbitrary unknown quantum state. Therefore, quantum states will not be copied from one core to another but moved. In the multi-core architecture model from [1], it is proposed to use a maximally entangled state (or EPR pair), as seen in Equation (5), as a communications resource to transmit a quantum state from one core to another. This communication protocol is called quantum teleportation and allows us to move one quantum state between cores by only using classical communications. In the architecture proposed in [1], a quantum link is used to distribute entangled states in different cores.

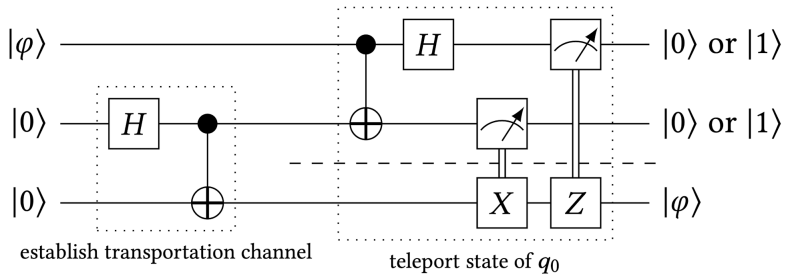


Figure 5: Teleportation protocol

These non-local communications are very costly, and it is a primordial task to decrease the number of inter-core communications needed. We could adapt other approaches to interconnect the cores and communicate among them, like shuttling. However, in the thesis, we will focus on an architectural model with several quantum processors connected among them via classical links and with the ability to distribute entangled pairs among qubits utilizing an EPR-pair generator, as proposed in [1].

3 Quantum Circuit Mapping

Quantum mapping is part of the compilation process that must be applied to a quantum circuit before executing it in the desired architecture. The goal of the mapping is that every time two virtual qubits must interact with each other, their quantum state is stored in neighbouring physical qubits, and therefore, they share an edge from the coupling map, and the two-qubit gate can be executed.

The mapping is composed of two parts, the layout (or initial mapping) and the routing. The initial mapping creates an association of virtual qubits (from the quantum circuit) into physical qubits (from the quantum architecture). However, due to coupling constraints, not all two-qubit gates will be feasible for an initial layout. Therefore we must move the qubits into positions that allow a two-qubit operation among them. That is where the routing comes in. The routing stage adds, over the whole circuit, swap gates to move the quantum states into neighbouring physical qubits so that two-qubit gates can be executed.

An optimal or near-optimal quantum mapping is crucial. Routing will add swap gates, thus increasing the depth of the circuit (number of timeslices of the circuit, where a timeslice is the set of operations that can be executed in parallel), and the fidelity of the whole circuit decreases.

For single-core quantum processors, the circuit to be executed and the coupling constraints of the processor are considered to perform the quantum mapping. There are many different algorithms for qubit mapping, such as [23], [24], [25], [26], and [2].

This problem is much more difficult for multi-core quantum processors, where we have as many coupling maps as cores, and for every two-qubit gate, the involved qubits must be in the same core. Moreover, this is an unexplored topic, and few mapping algorithms focus on multi-core quantum computers.

Intra-core mapping refers to a circuit's initial placement and routing into a single core, adding swap gates to satisfy coupling constraints. On the other hand, inter-core mapping refers to the assignment of qubits to cores, not considering each core's coupling map. The following subsections explain a single-core mapping algorithm and a multi-core mapping algorithm.

3.1 QMap

QMap [2] is a single-core mapper embedded in OpenQL [27]. It follows a step-by-step process, starting with an initial placement of qubits and determining the necessary movement operations to position non-adjacent qubits next to each other for interaction (i.e. two-qubit gates). This ensures a valid sequence of instructions with scheduled operations that respect both the data dependencies and coupling constraints. The objective is to minimize the gate and latency overhead, especially when implementing the algorithm on NISQ processors.

Figure 6 provides an overview of the QMap mapping process. Quantum algorithms and a configuration file containing information about the primitive gates supported by the processor, their execution time, the processor topology, and classical control constraints

are inputted into the mapper. These inputs then undergo a mapping process consisting of three main steps:

1. Initial Placement
2. Routing
3. RC-scheduler

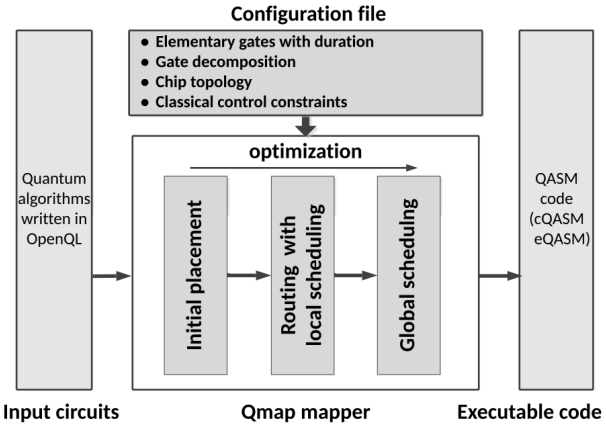


Figure 6: QMap overview from [2]

The output file generated is an executable QASM (Quantum Assembly) code that can be executed on the target quantum processor.

The initial placement is a crucial step in the QMap process, where virtual qubits from the circuit are mapped to the physical qubits of the quantum processor. The goal is to optimize the placement by positioning highly interacting qubits adjacent to each other, minimizing the need for movement operations during two-qubit gates. QMap employs an Integer Linear Programming (ILP) [28] approach to determine the best placement solution.

Routing is the process of determining a sequence of movement operations that allow the execution of two-qubit gates on a quantum processor, particularly when the coupling map of the processor does not connect the involved qubits by identifying a path that requires the fewest number of movements. QMap has different routing strategies. Some routing algorithms focus on minimizing the gates overhead, aiming to add as few gates as possible during the moving operations. Conversely, others prioritize minimizing the latency overhead, selecting a path that optimally interleaves the added movement operations with other operations to reduce execution time.

The next and last step is the RC-scheduler, which reschedules the routed circuit to minimize circuit latency and maximize instruction-level parallelism. It follows an as-late-as-possible (ALAP) strategy, reducing the required lifetime of each qubit and minimizing the decoherence error.

Though QMap is a very complex and parametrizable mapper, only a brief description is

given in this section, as the thesis's primary focus is on inter-core mapping, and QMap serves as the intra-core mapping tool for us.

3.2 FGP-rOEE Algorithm

This section reviews the inter-core mapping algorithm proposed in [8], called Fine Grained Partitioning (FGP-rOEE). This algorithm takes as an input a quantum circuit with q qubits, an architecture with N cores, and assumes each core contains $\frac{q}{N}$ qubits. It also assumes that the coupling map of each core is all-to-all (i.e. all qubits are directly connected to all other qubits in the core), and therefore, no intra-core mapping is needed, no initial placement nor routing.

The algorithm's goal is to obtain a sequence of assignments of qubits to cores, one assignment per timeslice. The circuit is separated into timeslices (the set of quantum gates that can be executed in parallel), and for each of them, a valid assignment of qubits to cores is found. A valid assignment has every pair of qubits interacting (i.e. there is a two-qubit gate involving these qubits) in the timeslice assigned to the same core.

For a given timeslice t, FGP-rOEE computes the interaction graph of that timeslice, a graph with the qubits as nodes and an edge between two nodes if the qubits interact with each other in future timeslices. The edges are weighted, representing the immediacy of the interaction. These weights are called look-ahead weights and are computed using Equation (7), where $I(m, q_i, q_j) = 1$ if qubits q_i and q_j interact at timeslice m, and $D(x)$ is a monotonically decreasing, non-negative function that mimics the distance between timeslices, so nearby timeslice has more impact on the look-ahead weights than latter ones. After some experimentation, in [8], it is proposed to use the exponential decay function (Equation (8)) with $\sigma = 1$.

$$w_t(q_i, q_j) = \sum_{t<m\leq T} I(m, q_i, q_j) \cdot D(m - t) \tag{7}$$

$$D(x) = 2^{\frac{-x}{\sigma}} \tag{8}$$

For qubits that interact exactly at timeslice t, an infinity weight is set to the edges. A graph k-partitioning algorithm will be applied to the interaction graph to obtain the assignment of qubits into cores. Qubits that interact at timeslice t must be in the same partition (i.e. core). That is the reason for having infinity weights between qubits.

A k-partitioning algorithm is then applied to the interaction graph. A k-partitioning algorithm partitions the graph into k different disjoint subsets of nodes. For our case, k equals the number of cores N, and it is a requirement that all partitions have the exact same size, as cores have a fixed size, and we can only assign $\frac{q}{N}$ qubits into each core. Because of these requirements, the number of available k-partitioning algorithms is scarce. It is proposed in [8] to use the Overall Extreme Exchange (OEE) algorithm, introduced in [29].

The Overall Extreme Exchange (OEE) algorithm builds upon the KL algorithm [30] and expands its capabilities. In OEE, the process begins with an initial configuration of qubits, and a pair of qubits is selected for sequential exchange. This selection is based on optimizing the minimized exchange cost for the current exchange. If the exchange cost remains positive, further exchanges are performed in an attempt to improve the configuration. The algorithm terminates if the exchange cost becomes zero or negative, indicating no further improvement can be achieved. In [8], a variation of the OEE algorithm is proposed, the rOEE or relaxed Overall Extreme Exchange. rOEE also starts with an assignment and performs exchanges of nodes until a valid partition is reached (i.e. all interacting qubits are in the same core).

This procedure is repeated for every pair of timeslices, using the previous assignment of qubits to cores as input for the rOEE algorithm. With this, a path of valid assignments is found over the whole circuit.

Let us see a simple example for clarification. We start with a circuit with four qubits and three timeslices. The interaction graphs for each timeslice are computed using Equation (7). Figure 7 depicts the circuit (left) and interaction graphs for each one of the timeslices (right).

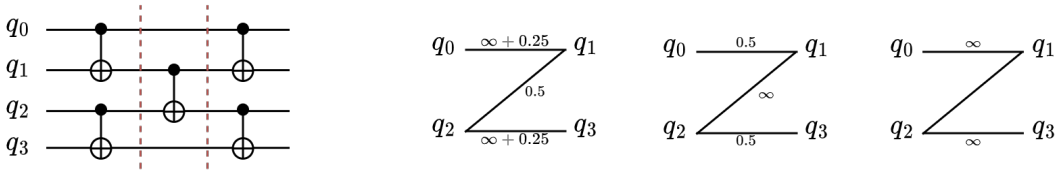


Figure 7: Interaction graphs for the three example timeslices

Then, rOEE is applied sequentially using the previous assignment of qubits to cores as input partition. For the first timeslice, a random initial partition is assumed. Figure 8 depicts the assignments of qubits to cores (blue or green), where each core has a capacity for two qubits.

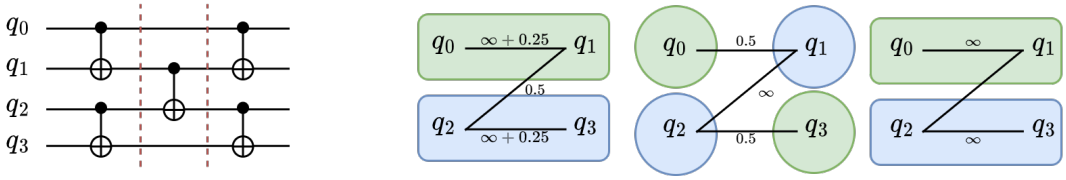


Figure 8: 2-Partitioning of the interaction graphs

Lastly, the inter-core communications graph can be computed as we know how qubits move during the circuit. Figure 9 depicts the communications graph from timeslice 1 to timeslice 2 (left) and from timeslice 2 to timeslice 3 (right).

We can evaluate how good the inter-core mapping is using these communications graphs. In [8], the metric from Equation (9) is used to compute the communications overhead between two timeslices.

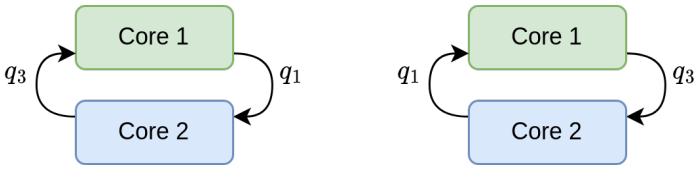


Figure 9: Communications graph between timeslices 1-2 (left) and 2-3 (right)

$$C = r + \sum_{k=2}^{n} (k-1) \cdot c_k \tag{9}$$

Where C is the total communications overhead, c_k is the number of cycles in the communication graph of length k, and r is the number of qubits not involved in a cycle. The sum of the communications overhead over the whole set of timeslices is the total cost of the inter-core mapping.

3.2.1 Flaws of FGP-rOEE Algorithm

In this section, we highlight the flaws or loose assumptions found in the design of the FGP-rOEE algorithm, building the motivation for a correction on how to distribute qubits over cores, and how to evaluate inter-core mapping algorithms.

The qubits to cores assignments (by rOEE) are obtained by the least minimum number of swap operations (among two or more qubits) so that the partition becomes valid for the desired timeslice. The cost of a partition is the sum of the weights of the edges that connect nodes from different partitions, and this cost is minimized when applying rOEE (or any k-partition algorithm). However, the cost of the mapping (from Equation (9)) does not directly relate to the number of edges in the communication graph (the number of qubits that need to be moved between timeslices so both timeslices can be executed). This is counter-intuitive and unnecessary for most multi-core quantum architectures, including [1], where what is essential is the number of non-local communications, not the number of swaps needed.

An example of why the proposed metric does not minimize the number of inter-core communications is illustrated in Figure 10. In this example, we see two interaction graphs for a multi-core architecture of four cores, and though one has more edges (right one) and, therefore, more qubit movements between cores, the cost is lower than the example in the left.

If we want to minimize the number of inter-core communications, we also need to minimize the number of edges in the communications graphs. The metric proposed in [8] is only valid when having only one communication qubit per core. When this happens, a core can not simultaneously send and receive a qubit; therefore, all inter-core communications must involve swaps. One way of performing inter-core swaps is proposed in [3]. However, when having multiple quantum communication qubits (which is feasible as communications play an important role in multi-core quantum architectures), the metric should be the number

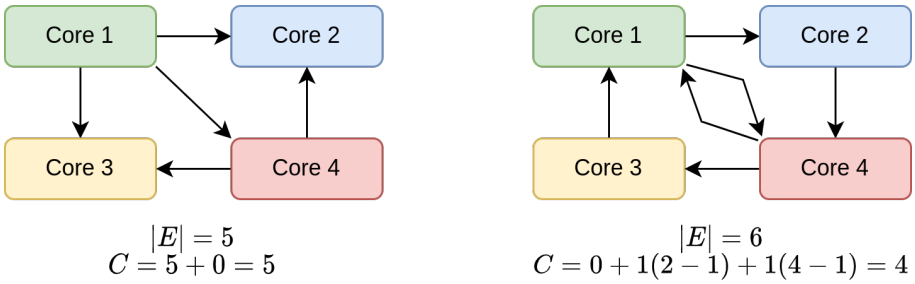


Figure 10: Communications graph between two different timeslices

of edges in the communication graph, as it is the number of inter-core communications we must perform.

A questionable assumption on [8] is to ignore intra-core mapping (i.e. qubits involved in a two-qubit gate must satisfy coupling constraints of the core). Inter-core communications are indeed likely to be much more costly than intra-core communications (i.e. routing). However, assigning qubits to cores is not all we care about when computing the mapping. The following questions arise:

- Can inter-core mapping influence intra-core mapping?
- How does both mapping relate to each other?
- Can we obtain a global metric that evaluates a complete mapping, considering inter-core and intra-core communications?

3.3 The need for a complete mapping algorithm

As discussed before, there are not many inter-core mapping algorithms, yet inter-core mapping only represents half of the problem, as intra-core mapping should not be forgotten. The key to answering the questions posed at the end of the last section begins with the development of a tool that, for a given quantum circuit and quantum architecture with multiple cores, performs a complete mapping of the algorithm, assigning virtual qubits to physical qubits within a core, and routing them (both intra and inter-core) to satisfy all coupling constraints.

By obtaining this complete mapping algorithm, we will be able to compare two inter-core algorithms fairly. We will be able to compare the number of inter-core communications needed using each inter-core mapping algorithm and how the movement of qubits among cores impacts the intra-core mapping of each core. Moreover, it is worth mentioning that this pipeline is the first algorithm that obtains the complete mapping of a quantum circuit into a multi-core architecture, obtaining all the operations needed for the quantum circuit to be executed into a multi-core quantum architecture.

4 Methodology

This section proposes a pipeline to compute a given quantum circuit's complete mapping (assignment of virtual qubits into physical qubits plus intra-core and inter-core routing). The proposed pipeline is composed of the following stages:

1. Input
2. Inter-Core mapping
3. Single-Core circuit splitting
4. Intra-Core mapping
5. Evaluation

4.1 Input

As an input for the proposed pipeline, we will have a quantum circuit using q qubits, the number of cores of the architecture N, the number of qubits per core, and a coupling map for each core (graphs representing qubit topology). Note that the total number of qubits of the architecture should be equal to or higher than the number of qubits in the quantum circuit.

Though our experiments will be performed using a homogeneous set of cores (i.e. the same number of qubits and the same coupling constraints for each core), the pipeline admits having a different number of qubits and different coupling maps for each core.

4.2 Inter-Core Mapping

As inter-core communications are expected to be much more expensive than local communications, the pipeline prioritizes inter-core mapping rather than intra-core mapping. To this end, the first stage of the complete mapping algorithm assigns a qubit to a core for each timeslice, ensuring each two-qubit gate can be executed, as each pair of interacting qubits are allocated in the same core.

For this, we will first partition the circuit into timeslices and then apply the desired inter-core mapping algorithm. Since multi-core quantum computers are still a new concept, the only inter-core mapping algorithm we know of is FGP-rOEE, highly reviewed in Section 3.2. Therefore, the first proposed pipeline uses FGP-rOEE as the inter-core mapping algorithm.

However, any inter-core mapping algorithm could be used in this stage. In future sections, an alternative inter-core mapper is proposed and evaluated by using it in this stage of the mapping pipeline.

4.3 Single-Core Circuit Split

At this point, we have each qubit distributed to a core for each timeslice, ensuring all two-qubit gates are feasible. The next step of the pipeline is to adapt the current quantum

circuit into a set of single-core quantum circuits so that current single-core mapping algorithms can be used.

To this end, and because inter-core mapping has given us how qubits move throughout the circuit execution, we can distribute quantum gates in different sets, depending on which core they are executed. We will then create as many sets of quantum gates as cores our architecture has. With this, we can construct one new quantum circuit for each core, containing the quantum gates of each set, maintaining the order of the gates in the original circuit.

Because of how quantum circuits are currently programmed, qubit index correction is necessary to adapt the qubit index to a suitable index inside the core. For this, we must construct a swap chain of qubits that will be exchanged between timeslices. We will use the swaps to assign indexes inside the core range to each qubit by using the index the previous qubit used. With this, we ensure that the output of the quantum algorithm will be the same as if we used a single-core architecture. Figure 11 depicts an example of qubit index correction between two cores. In this simple example, q_{50} and q_{34} are swapped, and once they arrive at the destination core, we assign to q_{50} the index that q_{34} was using and vice-versa, corresponding to core indexes q_{15} and q_0 respectively. This happens because now we have multiple levels of virtual qubits, the virtual qubits from the original quantum circuit and the virtual qubits of each core circuit.

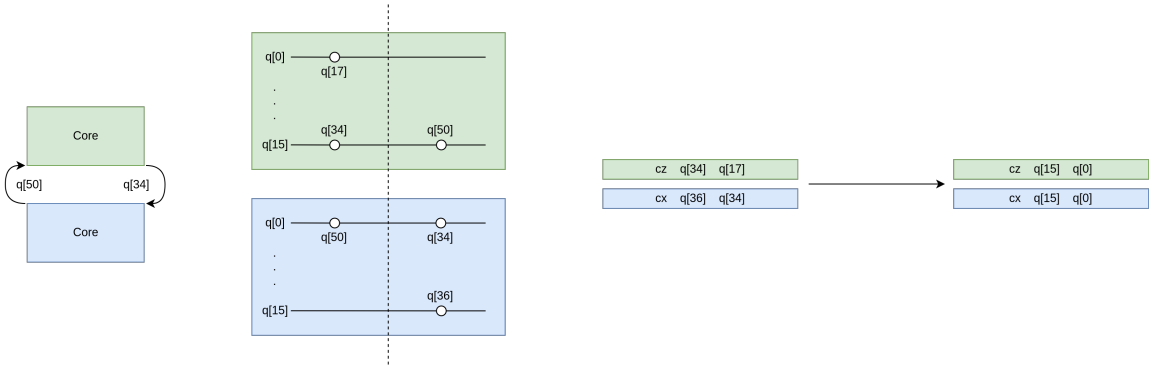


Figure 11: Qubit index correction for splitting circuits into core-circuit example

Because of how FGP-rOEE works, this chain of swaps is constructed by the rOEE partitioning algorithm. However, if other inter-core mappers were to be considered, the chain of swaps must be constructed afterwards the assignment of qubits into cores.

4.4 Intra-Core Mapping

As we have as many single-core circuits as cores in the multi-core architecture, we can compile them using mapping algorithms for single-core quantum computers, such as QMap, explained in Section 3.1.

By using the core's coupling map, we will obtain a modified circuit with swap gates inserted, so each two-qubit gate can be executed and does not violate any coupling constraint. Because of the limitations of current single-core mappers, we assumed the com-

munication qubits are connected to all computation qubits of the core. Therefore, the teleportation protocol can be executed no matter which physical qubit the quantum state is. Figure 12 shows the coupling map of the assumed architecture.

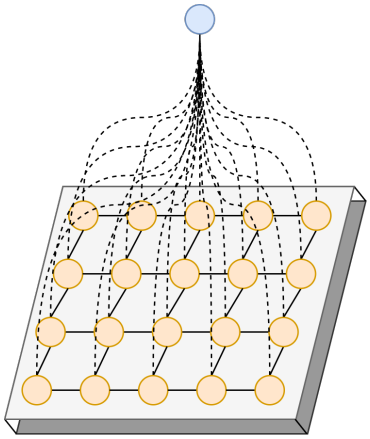


Figure 12: Quantum core with computation (orange) and communication qubits (blue)

This assumption can be dropped, and communication qubits can only connect to some computation qubits (instead of all of them). However, the intra-core mapping algorithm (QMap in our case) must be modified so the initial placement of qubits and the routing guarantees that communication qubits are always found in the desired physical qubits, which is out of this project's scope.

Other intra-core mappers, such as [23], [24], [25], [26], and [2] can be used.

4.5 Evaluation

Current single-core mapping techniques use the number of swap gates added by the router as the evaluation metric for the mapping algorithm. This makes sense as the more swap gates are added, the more time it takes to run the quantum algorithm and the more the system is prone to errors. However, we have two types of movements for multi-core mapping: intra-core (swap gates) and inter-core (quantum teleportation). Because of this, we must use a metric that combines both kinds of movements as well as the resources used (entanglement generation in the teleportation case).

We propose to use the Estimated Probability of a Successful Trial (EPST), shown in Equation (10).

$$EPST = {f_{2q}}^{|2q|} \cdot {f_{1q}}^{|1q|} \cdot {f_{meas}}^{|meas|} \cdot {f_{ent}}^{|ent|} \tag{10}$$

Where f_{2q} is the average fidelity of two-qubit gates, and $|2q|$ is the number of two-qubit gates of the mapped circuit (quantum algorithm, inter-core mapping, and intra-core mapping). The same goes for one-qubit gates ($1q$), measurements ($meas$) and entanglements (ent) needed for the complete execution in a multi-core quantum computer. With this, we consider the fidelity of each type of movement as the product of each gate fidelity.

In the context of quantum gates, fidelity is used to evaluate the accuracy and effectiveness of the gate's operation in transforming an input quantum state to the desired output state. A high-fidelity gate indicates that its operation closely approximates the ideal transformation, meaning it retains the coherence and integrity of the quantum information encoded in the input state.

Figure 13 shows a complete compilation of a Quantum Random Circuit (with 96 qubits, 1000 gates, and a 0.8 two-qubit gate frequency) for a different number of cores (from one to forty-eight cores), values in the x-axis. The top plot shows the non-local and local communications needed during the complete mapping pipeline. The bottom plot shows the Estimated Probability of a Successful Trial of the mapped circuit.

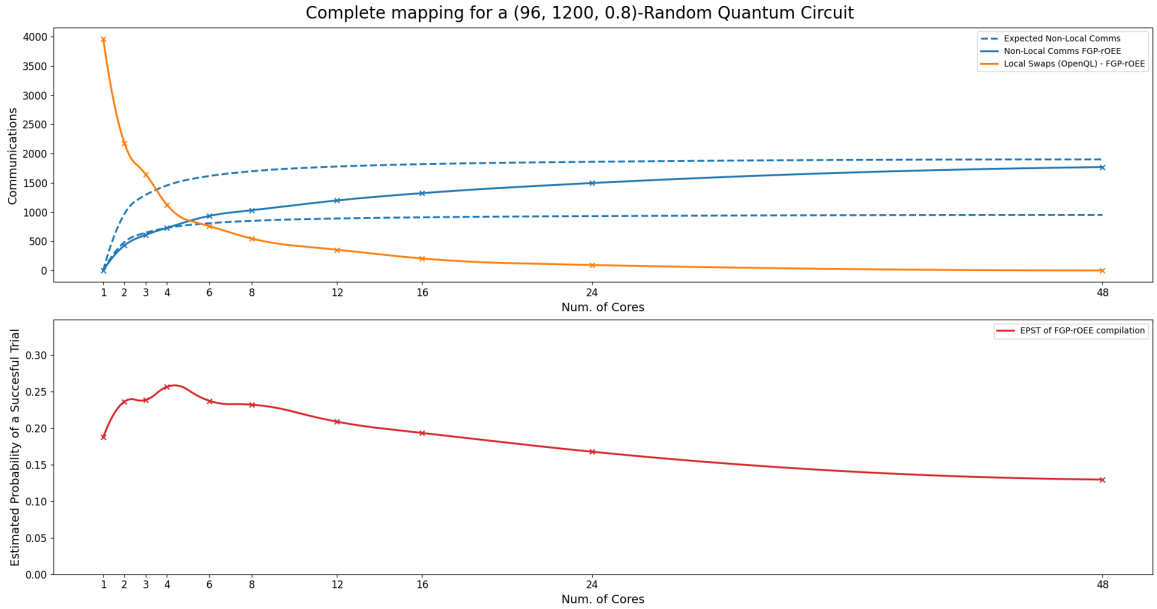


Figure 13: EPST of a (96, 1000, 0.8)-Random Circuit obtained with the complete mapping pipeline

The average error rates for each type of operation have been extracted from IBM [31]. Each gate's error probability has been decreased by a factor of ten to match the expected improvement over time of gate fidelity. Table 2 shows the values used to compute the EPST in Figure 13.

	Gate Fidelity
Two-Qubit Gate	0.9998711
Single-Qubit Gate	0.999991456
Measurement	0.99983
Entanglement	0.99966

Table 2: Average fidelities of quantum operations

5 Characterization of Non-Local Communications for Quantum Random Circuits

In this section, we characterize the non-local communications needed for the correct execution of a Random Quantum Circuit in a given architecture.

A Quantum Architecture is characterized by q, the number of qubits in the architecture, and N, the number of partitions (cores) of the quantum architecture. For this characterization, we assume cores are connected all-to-all (i.e. a core can transmit a qubit to any other core), and there is a uniform distribution of qubits per core, setting the number of qubits per core to $\frac{q}{N}$.

A Random Quantum Circuit is characterized by q, the number of qubits of the circuit, g, the number of gates of the circuit (also known as the size of the circuit), and f, the fraction of two-qubit gates of the quantum circuit.

5.1 Non-local communications bounds

Theorem 1. *For a given (q, N)-Quantum Architecture, and a (q, g, f)-Quantum Random Circuit, the expected number of non-local communications without using future qubit interactions is:*

$$\frac{(N-1)gfq}{N(q-1)} \leq \texttt{non-local comms} \leq \frac{2(N-1)gfq}{N(q-1)} \tag{11}$$

Proof. We first characterize the probability of, for a given two-qubit gate $cx(q_i, q_j)$, the probability of q_i and q_j of being in the same core.

Let $\mathcal{Q}$ be the set of qubits in the whole architecture and $\mathcal{C}(q_i)$ be the set of qubits in the same core as q_i. We define the probability of q_i and q_j of being in the same core as the number of qubits different than q_i, that are in the same core as q_i, over the total number of qubits different than q_i:

$$\frac{|\{q_k \in \mathcal{C}(q_i) : q_k \neq q_i\}|}{|\{q_k \in \mathcal{Q} : q_k \neq q_i\}|} = \frac{\frac{q}{N} - 1}{q - 1} \tag{12}$$

The expected number of two-qubit gates is given by $g \cdot f$. Let t be the number of timeslices the circuit has. Note that t is a new random variable to which we do not have access, and therefore it must disappear by the end of the proof. The expected number of two-qubit gates in each timeslice is:

$$\frac{g \cdot f}{t} \tag{13}$$

And therefore, as each qubit interacts at most one time per timeslice, the number of qubits involved in a two-qubit operation per timeslice is:

$$2\frac{g \cdot f}{t} \tag{14}$$

For a given timeslice t_i, the probability of q_1 being involved in a two-qubit operation is the number of interacting qubits in the timeslice over the total number of qubits:

$$\frac{2\frac{g \cdot f}{t}}{q} \tag{15}$$

Without loss of generality, let's see the possible scenarios of q_1 when going from timeslice t_{i-1} to timeslice t_i:

- q_1 interacts in t_i with a qubit q_k in the same core it currently is. No non-local communication is needed.

$$\mathcal{C}_{t_{i-1}}(q_1) = \mathcal{C}_{t_{i-1}}(q_k) = \mathcal{C}_{t_i}(q_1) = \mathcal{C}_{t_i}(q_k) \tag{16}$$

  The probability for the first scenario is given by the probability of q_1 interacting in timeslice t_i times the number of qubits in $\mathcal{C}_{t_{i-1}}(q_1)$ different than q_1, over the number of all qubits different than q_1:

$$\frac{2\frac{g \cdot f}{t}}{q} \cdot \frac{\frac{q}{N} - 1}{q - 1} \tag{17}$$

- q_1 interacts in t_i with a qubit q_k in a different core than it currently is. Two non-local communications are needed, one to send q_1, and the other to make space for the arriving qubit.

$$\mathcal{C}_{t_{i-1}}(q_1) \neq \mathcal{C}_{t_{i-1}}(q_k) \qquad \mathcal{C}_{t_i}(q_1) = \mathcal{C}_{t_i}(q_k) \tag{18}$$

  The probability for this second scenario is given by the probability of q_1 interacting in timeslice t_i times the number of qubits not in $\mathcal{C}_{t_{i-1}}(q_1)$ different than q_1, over the number of all qubits different than q_1:

$$\frac{2\frac{g \cdot f}{t}}{q} \cdot \frac{(N-1)\frac{q}{N}}{q - 1} \tag{19}$$

- q_1 does not interact in t_i. No non-local communications are needed. It may happen that q_1 is moved to another core to make room for an arriving qubit that needs to interact in the core, but this non-local communication has already been considered for the interacting qubit's second case.

Therefore, at timeslice t_i, the expected number of non-local communications caused by q_1 interacting is defined as:

$$0 \cdot \frac{2\frac{g \cdot f}{t}}{q} \cdot \frac{\frac{q}{N} - 1}{q - 1} + 2 \cdot \frac{2\frac{g \cdot f}{t}}{q} \cdot \frac{(N-1)\frac{q}{N}}{q - 1} + 0 \cdot (1 - \frac{2\frac{g \cdot f}{t}}{q}) = 2 \cdot \frac{2\frac{g \cdot f}{t}}{q} \cdot \frac{(N-1)\frac{q}{N}}{q - 1} \tag{20}$$

Generalizing for all qubits and timeslices, the expected number of non-local communications is defined as:

$$t \cdot q \cdot 2 \cdot \frac{2\frac{g \cdot f}{t}}{q} \cdot \frac{(N-1)\frac{q}{N}}{q - 1} = \frac{4(N-1)gfq}{N(q-1)} \tag{21}$$

In the above scenarios, we consider all cases where q_1 is involved. When we multiply the obtained expression by the total number of qubits, we are considering each two-qubit gate twice (i.e. when we were considering q_2 scenarios, we have taken into account the probability f q_1 interacting with, for example, q_2, but, by multiplying the obtained equation by the number of qubits q, we have also taken into account the probability of q_2 interacting with q_1). However, if two qubits need to interact and are in different cores, we only need to move one. Therefore, the obtained equation needs to be divided by two.

$$\texttt{Non-local comms upper bound} = \frac{2(N-1)gfq}{N(q-1)} \tag{22}$$

This gives us the upper bound, where a non-interacting qubit must also be moved to make space for each arriving interacting qubit. Probabilistically, we will have cycles among partitions (i.e. core two must send a qubit to core three, and core three must send a qubit to core two). If that is the case, there is no need for the movement of a non-interacting qubit. With this, we obtain the lower bound on the number of non-local communications needed by assuming that no non-interacting qubits will be moved and that we will always have 2-cycles for each non-local communication. Obtaining thus the lower bound:

$$\texttt{Non-local comms lower bound} = \frac{\frac{2(N-1)gfq}{N(q-1)}}{2} = \frac{(N-1)gfq}{N(q-1)} \tag{23}$$

□

A good inter-core mapping algorithm stays very close to the lower bound. Because we are only managing two-qubit gates, if, for a given timeslice, we have q_{int} qubits that interact with a qubit located in another core, the assignment can be solved using $\frac{q_{int}}{2}$ non-local communications if and only if each core has an even number of interacting qubits. Otherwise, when having an odd number of interacting qubits in a core, a non-interacting qubit will be moved out of the core to make space for an arriving interacting qubit, thus increasing the number of non-local communications. In an ideal scenario, each core will contain an even number of interacting qubits, and the lower bound (Equation (23)) will be matched. On the other hand, in the worst-case scenario, each core contains exactly one interacting qubit at each timeslice, forcing us to move one non-interacting qubit per two-qubit gate, matching the upper bound (Equation (22)).

With this, we can, for the first time, evaluate how an inter-core mapping algorithm performs and decide if it performs well or badly by comparing the number of inter-core communications of the mapper with the bounds obtained.

It is essential to understand that the proposed lower bound in Equation (23) is not a hard bound on the number of non-local communications since, when deriving it, no future interactions of qubits are considered. An approach that uses future interactions of qubits as information could achieve fewer non-local communications than the one proposed, by positioning the qubits to reduce the number of unfeasible two-qubit gates.

5.2 Analyzing FGP-rOEE performance

In this section, we compare the number of non-local communications needed when using the FGP-rOEE algorithm to the derived upper and lower bounds. For this, we will use a random circuit with 96 qubits, 5000 gates, and a two-qubit gate fraction of 0.6 and 0.8. Figures 14 and 15 show the number of non-local communications needed (y-axis) depending on the number of cores of the architecture (x-axis).

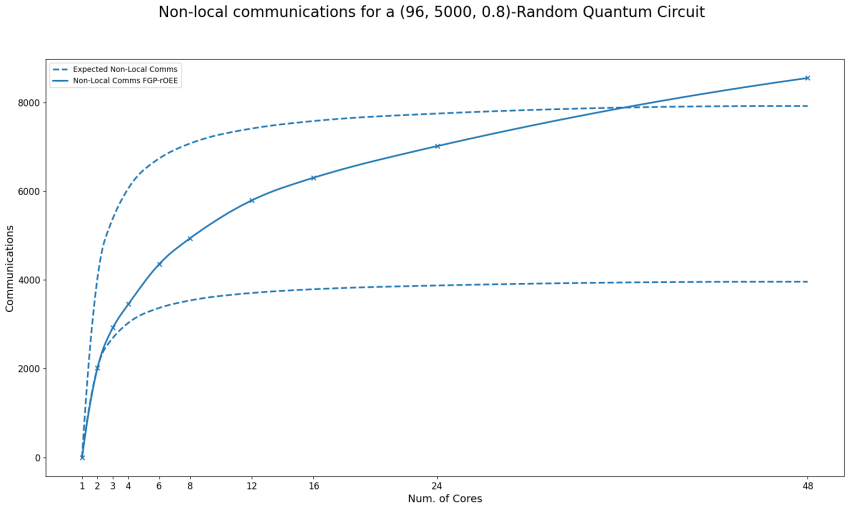


Figure 14: Non-local communications for a (96, 5000, 0.8)-Quantum Random Circuit

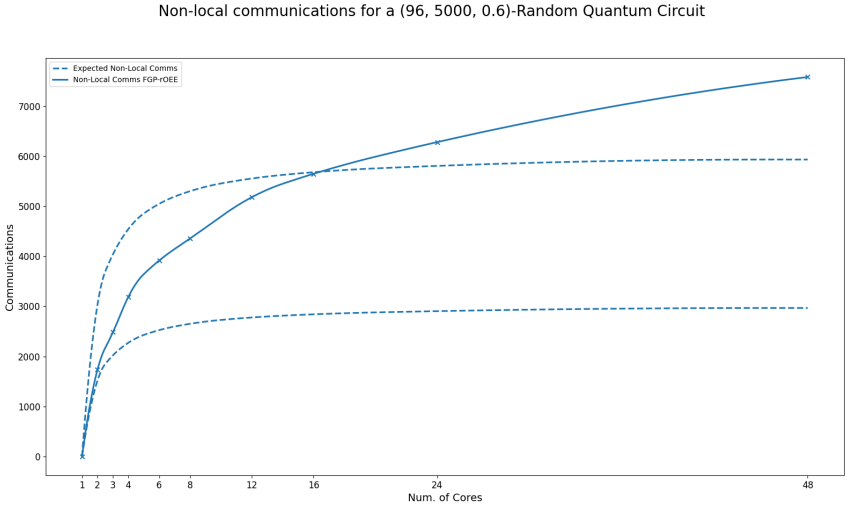


Figure 15: Non-local communications for a (96, 5000, 0.6)-Quantum Random Circuit

It can be seen how, for a low number of cores, FGP-rOEE manages to stay close to the lower bound, but as the number of cores increases, the number of non-local communications gets close to the upper bound, sometimes surpassing it. This proves that FGP-rOEE

does not find an optimal or near-optimal number of non-local communications for the inter-core mapping problem.

As inter-core communications are highly costly, having an inter-core mapper that achieves the lowest possible number of non-local communications is mandatory. This builds up the motivation for designing a new inter-core mapping algorithm to minimize the non-local communications achieved by FGP-rOEE.

6 Proposed inter-core mapping algorithm

In this section, we propose an alternative inter-core mapper to FGP-rOEE. Similar to FGP-rOEE, this algorithm describes how to assign qubits to cores in between timeslices and is generalized to map the whole algorithm. However, instead of distributing qubits into cores, this algorithm distributes two-qubit operations into cores, a much easier task than the former. As each two-qubit operations within a timeslice involve two distinct qubits, the algorithm's output will be a valid assignment for that particular timeslice.

We start with a valid assignment of qubits to cores for timeslice i. The following timeslice $(i+1)$ has a set of unfeasible two-qubit gates involving qubits that are currently located in different cores. This is depicted in Figure 16, where we can see five unfeasible two-qubit operations for an example architecture of four cores.

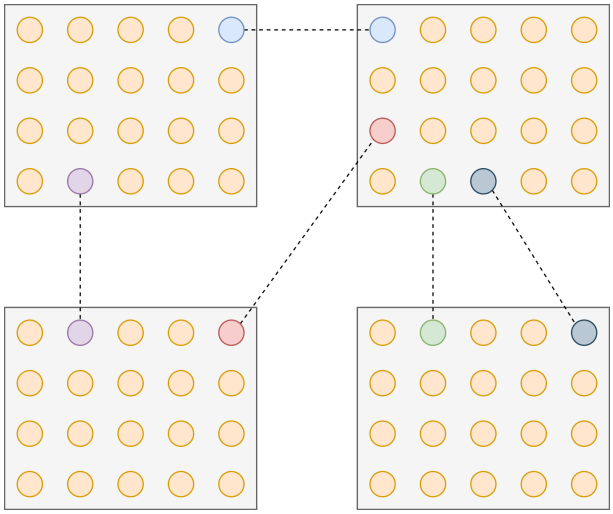


Figure 16: Non-feasible two-qubit operations

The qubits involved in the unfeasible two-qubit operations are then removed from the assignment and placed in an auxiliary vector of unassigned qubits, as depicted in Figure 17. The task now is to assign each operation to a core that has enough space to assign the qubits. For this assignment, we will construct a cost matrix using the cost function in Equation (24), which assigns a cost to each pair of unfeasible two-qubit operation and core based on how many non-local communications are needed to place both qubits involved in the operation into the destination core (i.e. one if a qubit involved in the two-qubit gate is already placed in the core, two otherwise). The cost matrix for the example at hand is shown in Table 3.

$$C_t(op_i, c_j) = \begin{cases} \infty & \text{if } c_j \text{ is full} \\ 1 & \text{if } op_{i_1} \in c_j \text{ or } op_{i_2} \in c_j \\ 2 & \text{otherwise} \end{cases} \tag{24}$$

The cost matrix is used to assign a single operation to each core. To do so, we use the Hungarian algorithm [32], a method used to solve optimization problems, specifically the

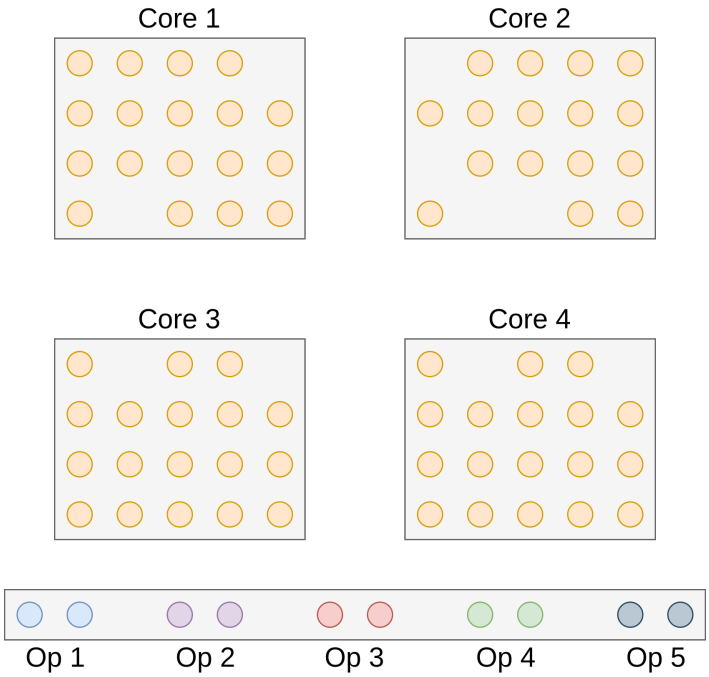


Figure 17: Unassigned qubits

	Core 1	Core 2	Core 3	Core 4
Operation 1	1	1	2	2
Operation 2	1	2	1	2
Operation 3	2	1	1	2
Operation 4	2	1	2	1
Operation 5	2	1	2	1

Table 3: Example cost matrix with five operations

assignment problem. The goal of the assignment problem is to find the best way to assign a set of tasks (operations) to a group of resources (cores) while minimizing the total cost.

When an operation is assigned to a core, both qubits involved are placed in the free spaces of the core. Therefore, the number of free spaces in the core decreases by two every time an operation is assigned to it. By placing both qubits in each operation into the same core, we ensure the two-qubit gate will be feasible in the following timeslice.

When having more operations than cores, the Hungarian algorithm only assigns one operation per core, so some operations will remain unassigned. A new cost matrix will be computed for those unassigned operations, considering the new free spaces of each core, setting a weight of infinity for those cores already complete. By doing this, we ensure that no core exceeds its capacity and that the resulting assignment will be valid.

For the example at hand, the Hungarian algorithm assigns operation one to the first core, operation two to the third core, operation three to the second core, and operation four to

the fourth core, leaving operation five unassigned. This assignment is depicted in Figure 18, and the cost matrix for the unassigned operations is shown in Table 4.

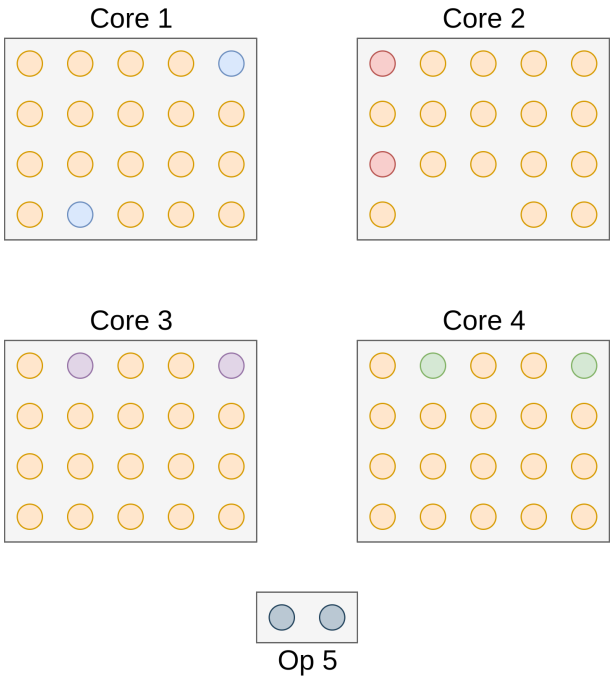


Figure 18: One of the unfeasible two-qubit operation assigned to each core

	Core 1	**Core 2**	**Core 3**	**Core 4**
Operation 5	∞	1	∞	∞

Table 4: Example cost matrix with one operation

Iteratively, the cost matrix is computed, and one two-qubit operation is assigned to each core with free space until all qubits are assigned to a core. Figure 19 depicts the example's final assignment.

There is a problem in the proposed algorithm whenever a core has an odd number of qubits involved in unfeasible operations (i.e. the free space of a core when assigning operations to cores is odd). If a core had an odd number of free spaces, the last unfeasible operation could not be assigned to any core, as it takes two free spaces, but there are two cores with one free space each. An auxiliary two-qubit operation involving one qubit from each core containing an odd number of free spaces is created to solve this problem. The two qubits involved in the auxiliary operation will be selected as the pair that interacts the most in future timeslices.

To quantify how much each pair of qubits interact in future timeslices, an approach similar to in [8] is selected.

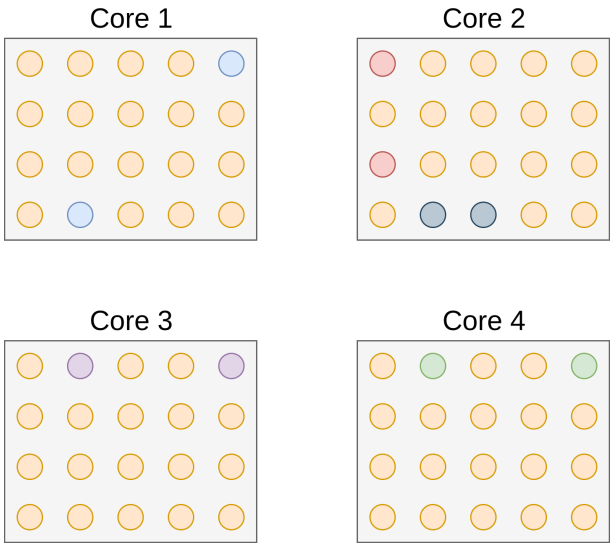


Figure 19: Valid assignment of qubits into cores

$$w_t(q_i, q_j) = \sum_{t<m\leq T} I(m, q_i, q_j) \cdot 2^{-(m-t)} \tag{25}$$

Where lookahead weights are computed for each pair of qubits using Equation (25), where $I(m, q_i, q_j) = 1$ if qubits q_i and q_j interact at timeslice m. With this, an even number of free spaces per core is guaranteed, making the assignment of all two-qubit operations possible.

Up to this point, the proposed mapping algorithm does not consider how qubits interact in future timeslices, obtaining, in the best-case scenario, non-local communications matching the lower bound derived in Theorem 1. However, future interactions of qubits can be added to the cost matrix to perform an assignment that further reduces the number of non-local communications. For this, for each qubit, an attraction force to each core is computed using Equations (25) and (26). Where $I_t(q_{i'}, c_j) = 1$ if qubit $q_{i'}$ is in core c_j at timeslice t.

$$\texttt{attr}_t(q_i, c_j) = \sum_{i'=0}^{q} I_t(q_{i'}, c_j) \cdot w_t(q_i, q_{i'}) \tag{26}$$

The new cost matrix is then computed using the costs given in Equation (27), where the number of non-local communications needed and the attraction forces are combined. Note that, as each operation op_i involves two qubits (op_{i_1} and op_{i_2}), the operation attraction force to a core is the average attraction of each of the involved qubits. With this, we aim to reduce the number of non-local communications needed, with the potential to obtain an even lower number than the one derived with the lower bound in Theorem 1, as now future interactions are considered.

$$C_t(op_i, c_j) = \begin{cases} \infty & \text{if } j \text{ is full} \\ 1 - \frac{\texttt{attr}_t(op_{i_1}, c_j) + \texttt{attr}_t(op_{i_2}, c_j)}{2} & \text{if } op_{i_1} \in j \text{ or } op_{i_2} \in c_j \\ 2 - \frac{\texttt{attr}_t(op_{i_1}, c_j) + \texttt{attr}_t(op_{i_2}, c_j)}{2} & \text{otherwise} \end{cases} \tag{27}$$

Moreover, the proposed approach cost function is completely tunable, allowing for more complex variations of the problem. For example, if not all cores are connected to each other, the number of non-local communications to send a qubit to a core may be more than one, as different hops need to be made. The cost function can be applied to this case, and many others, giving a much more robust and applicable inter-core mapping algorithm than FGP-rOEE.

7 Results

This section compares FGP-rOEE and the proposed approach, using the inter-core communications (Section 7.1), and the complete EPST (Section 7.2). For this, we will use the pipeline proposed in Section 4, adding the operations needed for inter-core and intra-core mapping to the circuit.

This section presents a selection of the most significant and representative experiments. These experiments involve a higher number of qubits to highlight the key outcomes and trends observed throughout the evaluation process. Additionally, to ensure the completeness and transparency of the research, a number of supplementary experiments have been included in the appendix. The appendix contains additional experiments that offer additional evidence, further validating the conclusions drawn in the Results section.

7.1 Inter-Core Mapping evaluation

Figures 20 and 21 compares the performance of FGP-rOEE (solid, blue line), the proposed approach without using future interactions of qubits (dashed green line) and the proposed approach using lookahead weights to characterize the interaction of qubits in future timeslices (solid green line), as well as the upper and lower bound on the non-local communications derived in Theorem 1.

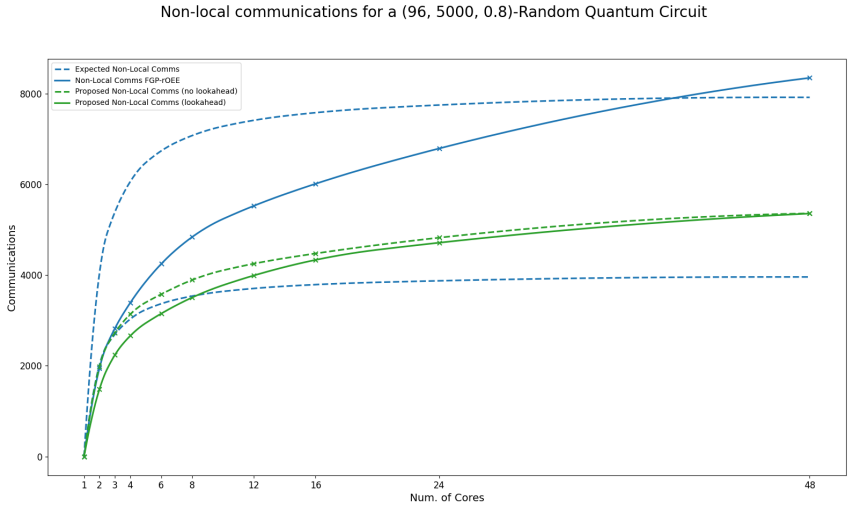


Figure 20: FGP-rOEE and proposed non-local communications for a (96, 5000, 0.8)-Quantum Random Circuit

It can be seen that both proposed approaches perform much better than FGP-rOEE, where the non-local communications needed are much closer to the lower bound. The approach using future interactions of qubits reduces the non-local communications needed even more, obtaining fewer communications than the lower bound dictates. Recall that the lower bound only applies when not using lookahead weights.

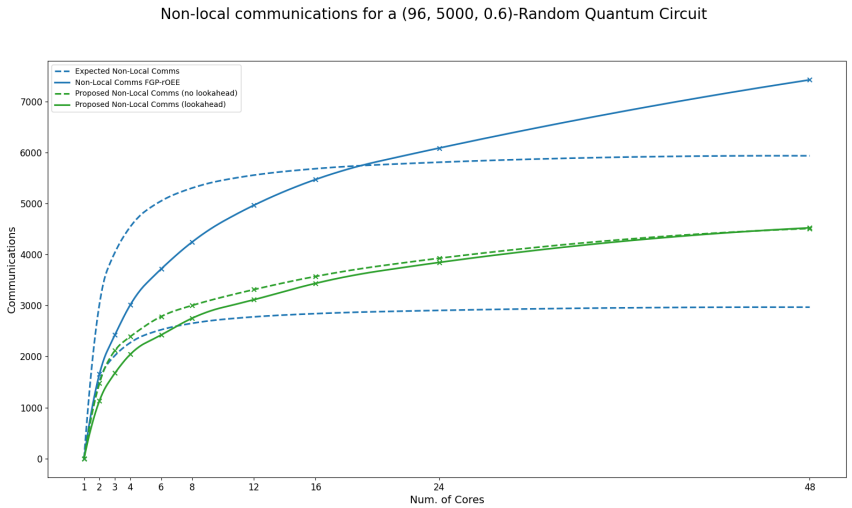


Figure 21: FGP-rOEE and proposed non-local communications for a (96, 5000, 0.6)-Quantum Random Circuit

7.2 Complete Mapping evaluation

Figures 22 to 24 and Tables 5 to 7 show a comparison of the different compiled random circuits. The bottom right plot shows the percentage improvement of our inter-core mapping approach compared to FGP-rOEE, assessing both the number of non-local communications needed and global EPST. We have used a different number of qubits, gates and two-qubit gate frequencies to explore the whole spectrum of random circuits. However, regarding the number of cores, for a circuit with q qubits, we must have a number of cores N such that $\frac{q}{N}$ is even. Otherwise, a two-qubit gate might be impossible to allocate when having an odd number of qubits per core.

Number of Cores	**1**	**2**	**3**	**4**	**6**	**8**	**12**	**16**	**24**	**48**
Non-local communications (%)	0	21	14	16	20	20	22	23	27	30
EPST (%)	0	11	13	13	22	24	34	38	54	76

Table 5: Improvement (%) of non-local communications and EPST when using the proposed approach for a (96, 1200, 0.8)-Quantum Random Circuit

Number of Cores	**1**	**2**	**3**	**4**	**6**	**8**	**12**	**16**	**24**	**48**
Non-local communications (%)	0	34	36	32	32	33	36	36	36	35
EPST (%)	0	18	32	29	33	39	55	62	73	88

Table 6: Improvement (%) of non-local communications and EPST when using the proposed approach for a (96, 1200, 0.6)-Quantum Random Circuit

It can be seen how, for all the experiments performed, the proposed approach leads to a lower number of inter-core communications and a higher Estimated Probability of a

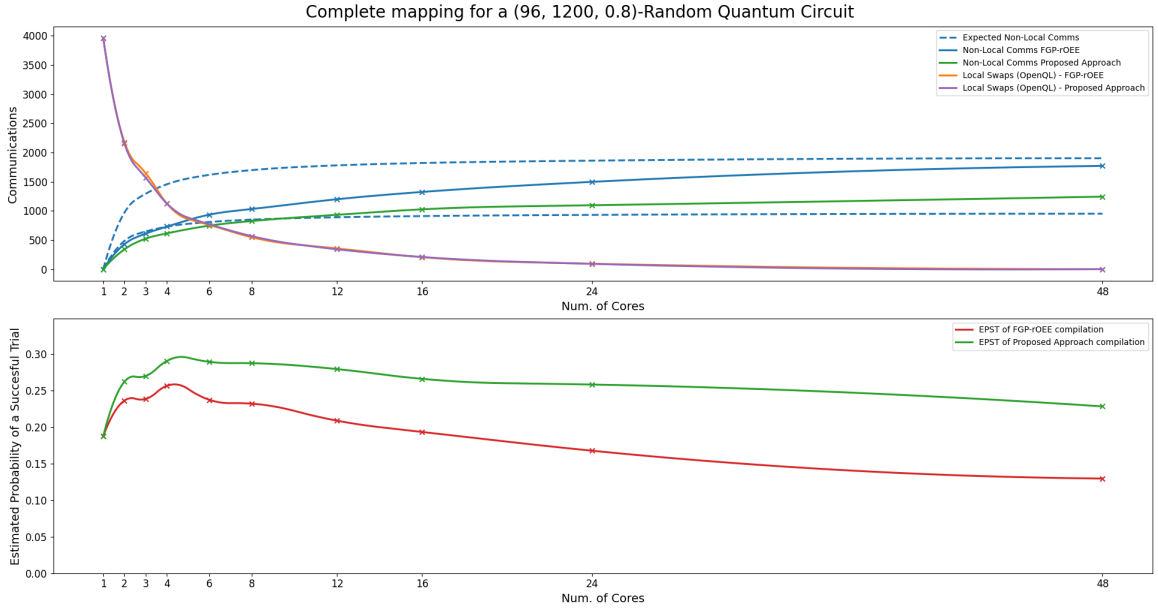


Figure 22: Complete mapping using both FGP-rOEE and Our Approach of a (96, 1200, 0.8)-Quantum Random Circuit. Improvement is shown in Table 5

Number of Cores	**1**	**2**	**3**	**4**	**6**	**8**	**12**	**16**	**24**	**48**
Non-local communications (%)	0	42	29	38	38	38	41	38	40	38
EPST (%)	0	10	17	24	32	38	51	49	58	68

Table 7: Improvement (%) of non-local communications and EPST when using the proposed approach for a (96, 1200, 0.4)-Quantum Random Circuit

Succesful Trial (EPST). Using this last metric, we consider the complete mapping of the quantum algorithm, making it much more reliable than the number of non-local communications. Therefore, if the inter-core mapping had some effect on the intra-core mapping (which does not look like it for random circuits), it would also be considered within the EPST metric.

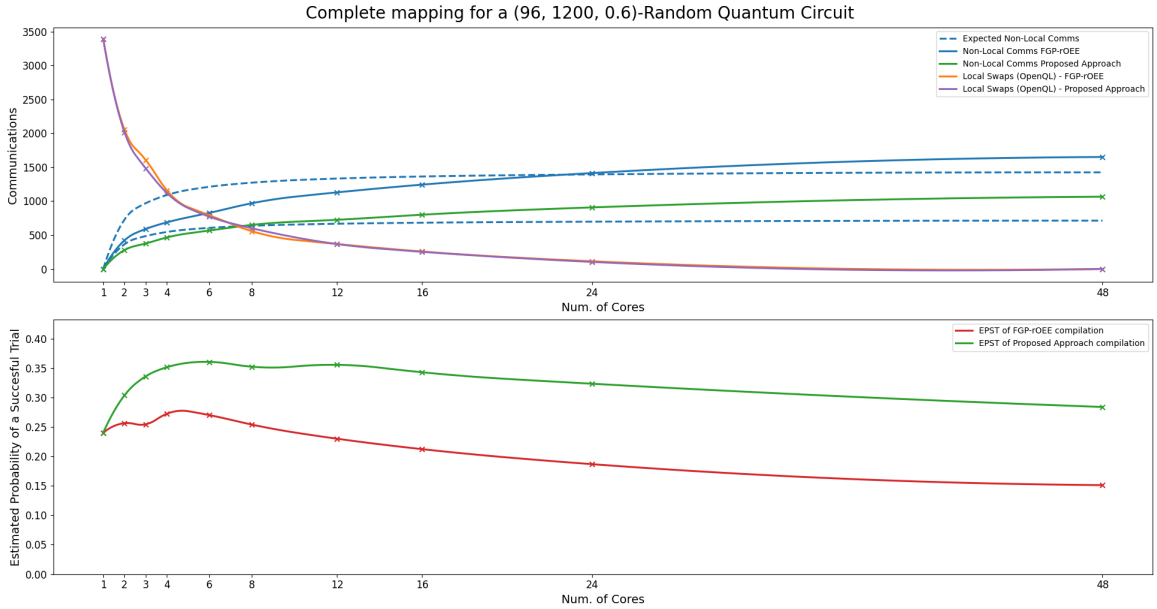


Figure 23: Complete mapping using both FGP-rOEE and Our Approach of a (96, 1200, 0.6)-Quantum Random Circuit. Improvement is shown in Table 6

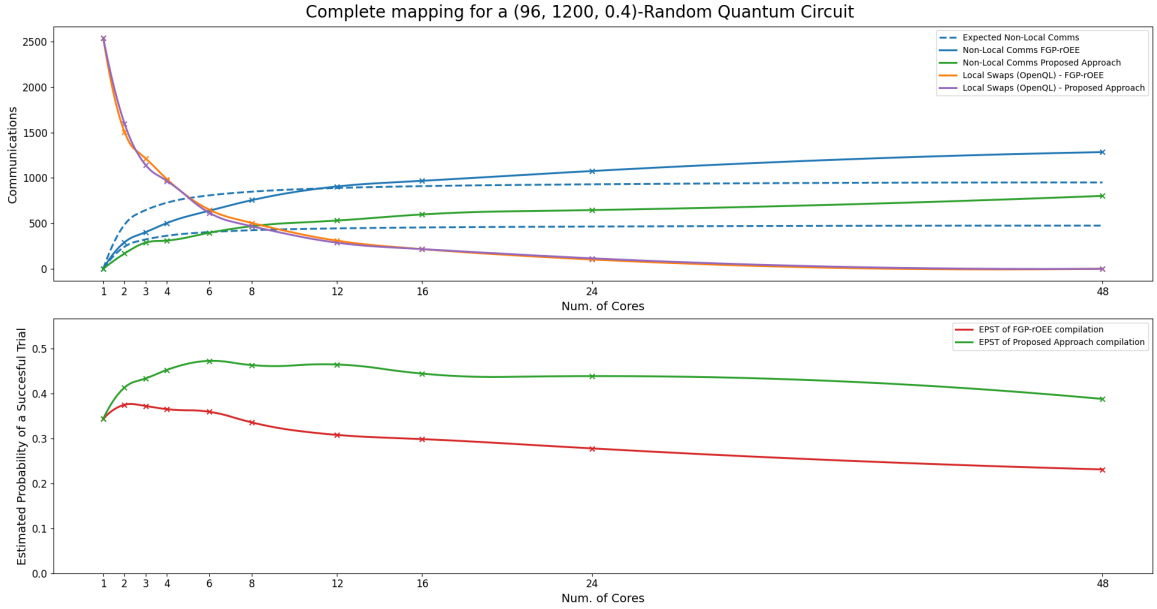


Figure 24: Complete mapping using both FGP-rOEE and Our Approach of a (96, 1200, 0.4)-Quantum Random Circuit. Improvement is shown in Table 7

8 Conclusions and Future Work

In this thesis, a comprehensive pipeline for computing the complete mapping of quantum algorithms has been proposed. The mapping of quantum algorithms into physical quantum devices is a critical step in the execution of quantum computing algorithms.

In this thesis, a groundbreaking mapping algorithm has been introduced, marking its distinction as the first-ever algorithm of its kind. The proposed pipeline enables the specification of inter-core and intra-core constraints, facilitating the mapping of quantum circuits onto multi-core quantum architectures. This achievement is truly remarkable as it expands the existing knowledge in the field of quantum computing.

One of this thesis' primary contributions is identifying and demonstrating the suboptimal performance of the current state-of-the-art inter-core mappers. Through rigorous analysis and evaluation, it has been shown that these existing mappers do not achieve optimality or near-optimality in terms of non-local communications needed. This finding underscores the need for further research and development in this area.

An alternative algorithm has been proposed to overcome the limitations of the current inter-core mapping algorithms. The proposed algorithm offers improved effectiveness in mapping quantum algorithms across multiple cores. Extensive experiments and simulations have been performed to validate the performance of this algorithm, providing evidence of its superiority over existing approaches. The complete mapping pipeline has been employed to evaluate the different inter-core mapping approaches to ensure a fair and reliable comparison. Using the complete mapping pipeline, the evaluation process captures the intricacies and complexities of mapping quantum algorithms, allowing for a comprehensive and meaningful comparison between the proposed algorithm and existing methods.

The findings of this thesis highlight the significance of developing efficient and optimized inter-core mapping algorithms for quantum computing systems. The proposed pipeline and alternative algorithm contribute to the body of knowledge in this field and provide valuable insights for researchers and practitioners working towards improving the performance and scalability of quantum computing systems.

In conclusion, this thesis has successfully proposed a comprehensive pipeline for computing the complete mapping of quantum algorithms, identified the limitations of existing inter-core mappers, presented an alternative algorithm to address these limitations, and conducted a thorough comparison using the complete mapping pipeline. The research outcomes contribute to advancing the field of quantum computing and pave the way for further research and development in inter-core mapping algorithms.

8.1 Future Work

The only quantum communication protocol considered in this compiler design was quantum teleportation (depicted in Figure 5). However, other quantum protocols should also be considered, as they have the potential to obtain a better complete mapping when combined with quantum teleportation and intra-core mapping.

One example of this mentioned quantum protocols is the execution of a two-qubit gate involving two qubits in different cores through classical communication and an entangled quantum system. By doing this, moving the qubit from one core to another is unnecessary, potentially reducing the number of operations needed. Such protocol is depicted in Figure 25, from [3].

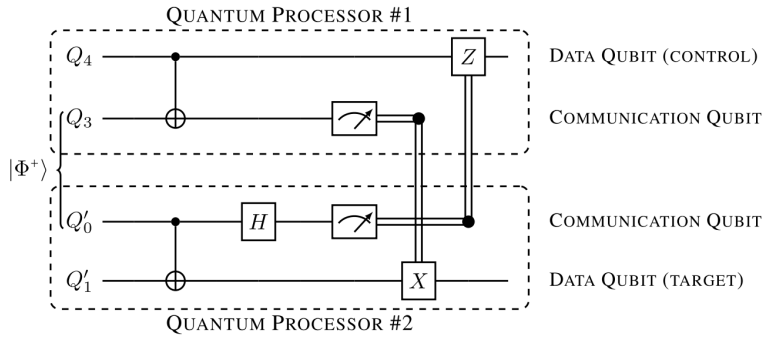


Figure 25: Remote quantum operation using classical communications and an entangled quantum state from [3]

We strongly believe that a complete mapping algorithm that uses remote operations and quantum teleportation will outperform approaches only considering teleportation.

References

[1] Santiago Rodrigo, Sergi Abadal, Eduard Alarcón, and Carmen G Almudever. Exploring a double full-stack communications-enabled architecture for multi-core quantum computers. *arXiv preprint arXiv:2009.08186*, 2020.

[2] Lingling Lao, Hans van Someren, Imran Ashraf, and Carmen G. Almudever. Timing and Resource-Aware Mapping of Quantum Circuits to Superconducting Processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41:359–371, 02 2022.

[3] Davide Ferrari, Angela Sara Cacciapuoti, Michele Amoretti, and Marcello Caleffi. Compiler Design for Distributed Quantum Computing. *IEEE Transactions on Quantum Engineering*, 2:1–20, 2021.

[4] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 06 1982.

[5] Nicholas Laracuente, Kaitlin Smith, Poolad Imany, Kevin Silverman, and Frederic Chong. Modeling Short-Range Microwave Networks to Scale Superconducting Quantum Computation.

[6] Christopher Monroe, Robert Raussendorf, A Ruthven, Kevin K Brown, Peter Maunz, Luming Duan, and Jung-Hyun Kim. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. 89, 02 2014.

[7] Jay Gambetta. Expanding the IBM Quantum roadmap to anticipate the future of quantum-centric supercomputing, 05 2022.

[8] Jonathan M. Baker, Casey Duckering, Alexander Hoover, and Frederic T. Chong. Time-sliced quantum circuit partitioning for modular architectures. *Proceedings of the 17th ACM International Conference on Computing Frontiers*, 05 2020.

[9] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439:553–558, 12 1992.

[10] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26:1484–1509, 10 1997.

[11] Dmitry A Fedorov, Bo Peng, Niranjan Govind, and Yuri Alexeev. VQE Method: A Short Survey and Recent Developments, 2021.

[12] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm, 2014.

[13] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, 08 2018.

[14] Fei Yan, Philip Krantz, Youngkyu Sung, Morten Kjaergaard, Daniel L. Campbell, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. Tunable Coupling

Scheme for Implementing High-Fidelity Two-Qubit Gates. *Physical Review Applied*, 10, 11 2018.

[15] C. Ospelkaus, C. Langer, J. Amini, K. Brown, D. Leibfried, and D. Wineland. Trapped-Ion Quantum Logic Gates Based on Oscillating Magnetic Fields. *Physical Review Letters*, 101, 08 2008.

[16] Maximilian Ruf, Noel H. Wan, Hyeongrak Choi, Dirk Englund, and Ronald Hanson. Quantum networks based on color centers in diamond. *Journal of Applied Physics*, 130:070901, 08 2021.

[17] Tong Wu and Jing Guo. Variability and Fidelity Limits of Silicon Quantum Gates Due to Random Interface Charge Traps. *IEEE Electron Device Letters*, 41.7:1–1, 01 2020.

[18] Dolev Bluvstein, Harry LeVine, Giulia Semeghini, Tout T Wang, Sepehr Ebadi, M Kalinowski, Alexander Keesling, Nishad Maskara, Hannes Pichler, Markus Greiner, Vladan Vuletic, and Mikhail D Lukin. A quantum processor based on coherent transport of entangled atom arrays. *Nature*, 604:451–456, 04 2022.

[19] Nicholas LaRacuente, Kaitlin N Smith, Poolad Imany, Kevin L Silverman, and Frederic T Chong. Modeling Short-Range Microwave Networks to Scale Superconducting Quantum Computation, 2022.

[20] Kaitlin N Smith, Gokul Subramanian Ravi, Jonathan M Baker, and Frederic T Chong. Scaling Superconducting Quantum Computers with Chiplet Architectures. 10 2022.

[21] Santiago Rodrigo, Sergi Abadal, Eduard Alarcón, Medina Bandic, Hans van Someren, and Carmen G. Almudéver. On Double Full-Stack Communication-Enabled Architectures for Multicore Quantum Computers. *IEEE Micro*, 41:48–56, 09 2021.

[22] Santiago Rodrigo, Sergi Abadal, Eduard Alarcón, and Carmen G Almudever. Will quantum computers scale without inter-chip comms? a structured design exploration to the monolithic vs distributed architectures quest. page 1–6, 2020.

[23] Gadi Aleksandrowicz, Thomas G Alexander, Panagiotis Kl. Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M Chow, Antonio D Córcoles-Gonzales, Abigail J Cross, Andrew W Cross, Juan Cruz-Benito, Chris Culver, Salvador De, E de, Delton Ding, Eugene F Dumitrescu, Ivan Duran, Pieter T Eendebak, Mark J Everitt, Ismael Faro Sertage, A Frisch, Andreas Fuhrer, Jay M Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Martín-Fernández, Douglas McClure, David S McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Rodriguez, Giacomo Nannicini, Paul Nation, Pauline J Ollitrault, L ORiordan, Hanhee Paik, Jesús M Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia E Rice, Abdón Rodríguez Davila, Raymond Harry

Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Camilo Silva, Yukio Siraichi, Seyon Sivarajah, John A Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan Wildstrom, Jessica Wilson, Erick Winston, Christopher G Wood, Stephen J Wood, Stefan Wörner, Ismail Akhalwaya, and Christa Zoufal. Qiskit: An Open-source Framework for Quantum Computing. 01 2019.

[24] Matthew Amy and Vlad Gheorghiu. staq—A full-stack quantum processing toolkit. 5:034016–034016, 06 2020.

[25] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. t—ket⟩: a retargetable compiler for NISQ devices. 6:014003–014003, 11 2020.

[26] Mohammad Javad Dousti, Alireza Shafaei, and Massoud Pedram. Squash: a scalable quantum mapper considering ancilla sharing. *Proceedings of the 24th edition of the great lakes symposium on VLSI*, 05 2014.

[27] Nader Khammassi, Imran Ashraf, J. van Someren, Razvan Nane, A M Krol, Michiel Adriaan Rol, Lingling Lao, Koen Bertels, and Carmen G Almudever. OpenQL: A Portable Quantum Programming Framework for Quantum Accelerators. 18:1–24, 12 2021.

[28] L Lao, B van Wee, I Ashraf, J van Someren, N Khammassi, K Bertels, and C G Almudever. Mapping of lattice surgery-based quantum circuits on surface code architectures. *Quantum Science and Technology*, 4:015005, 09 2018.

[29] Taehoon Park and Chae Y. Lee. Algorithms for partitioning a graph. *Computers Industrial Engineering*, 28:899–909, 10 1995.

[30] B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal*, 49:291–307, 02 1970.

[31] Quantum Lab - IBM Quantum, 2023.

[32] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics*, 52:7–21, 02 2005.

Appendices

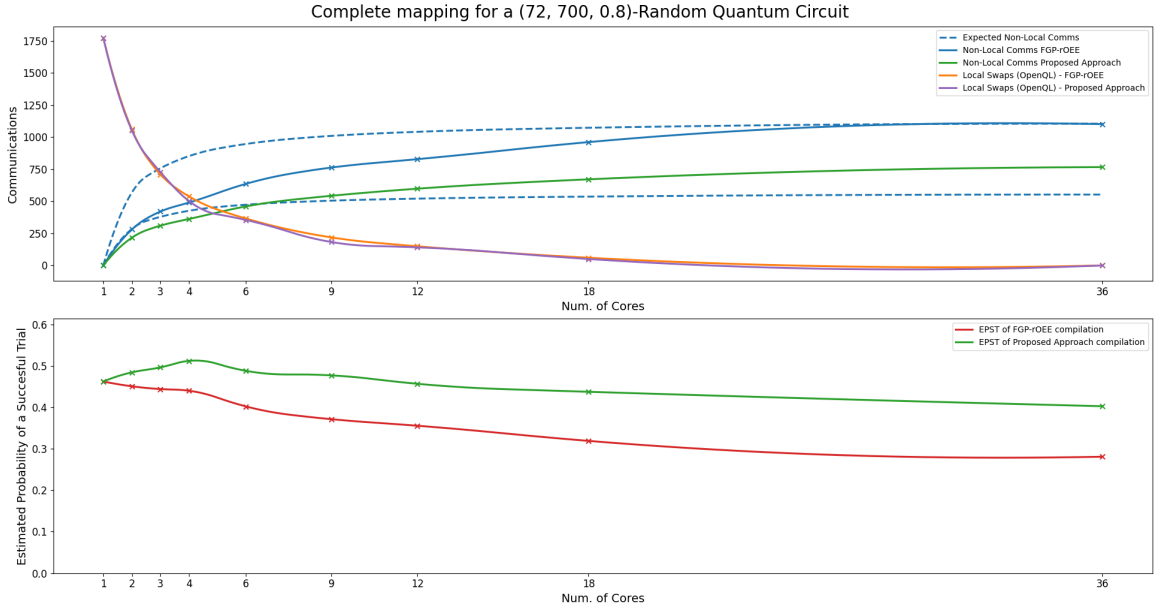


Figure 26: Complete mapping using both FGP-rOEE and Our Approach of a (72, 700, 0.8)-Quantum Random Circuit

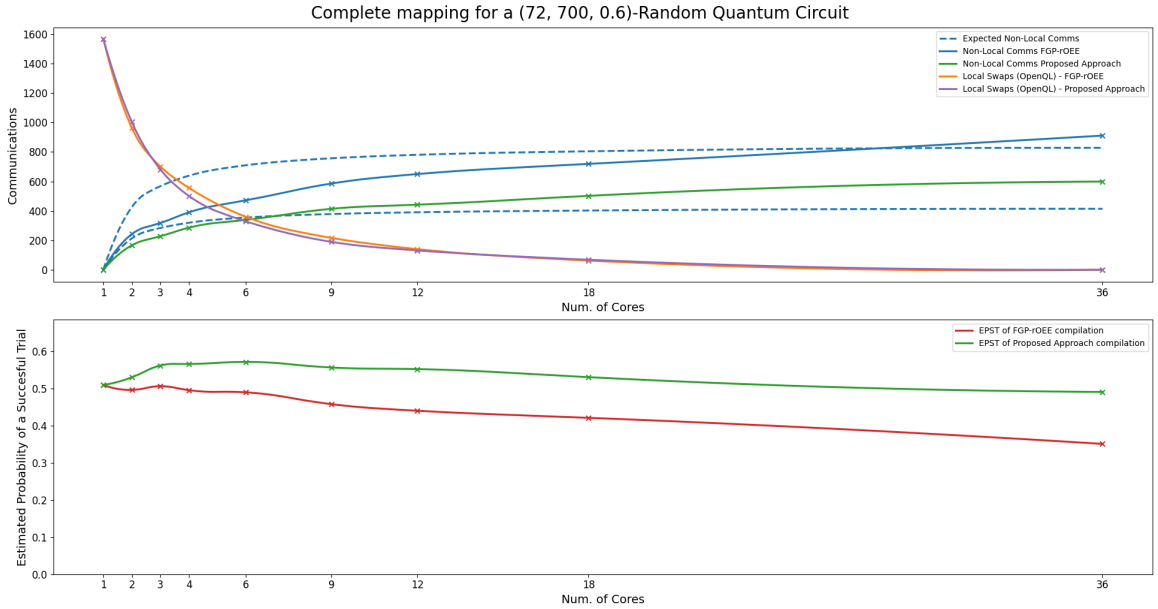


Figure 27: Complete mapping using both FGP-rOEE and Our Approach of a (72, 700, 0.6)-Quantum Random Circuit

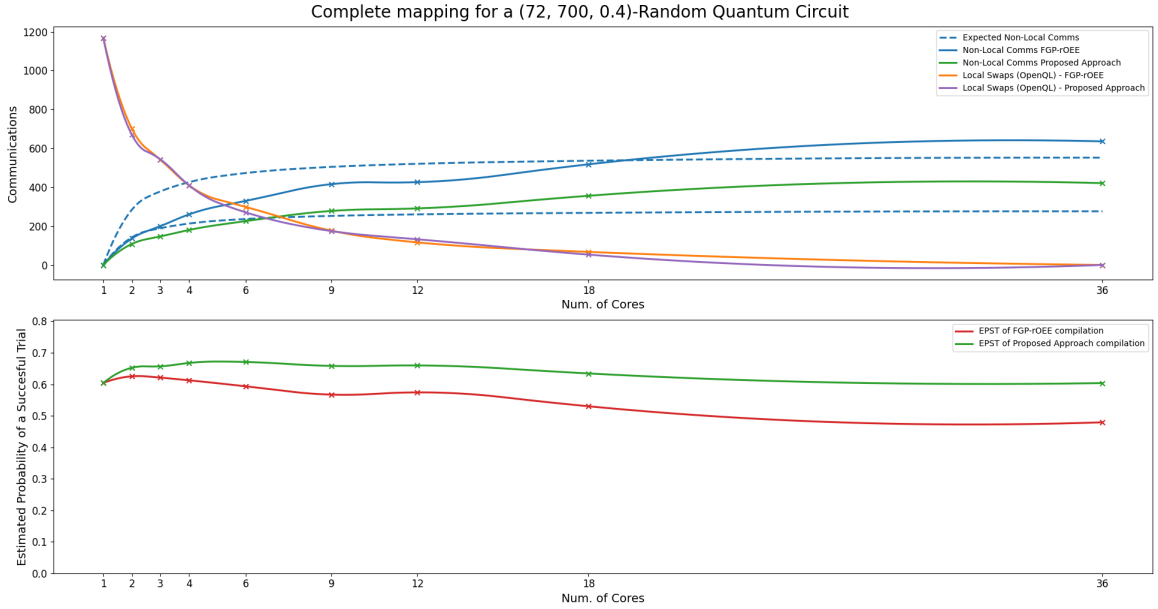


Figure 28: Complete mapping using both FGP-rOEE and Our Approach of a (72, 700, 0.4)-Quantum Random Circuit

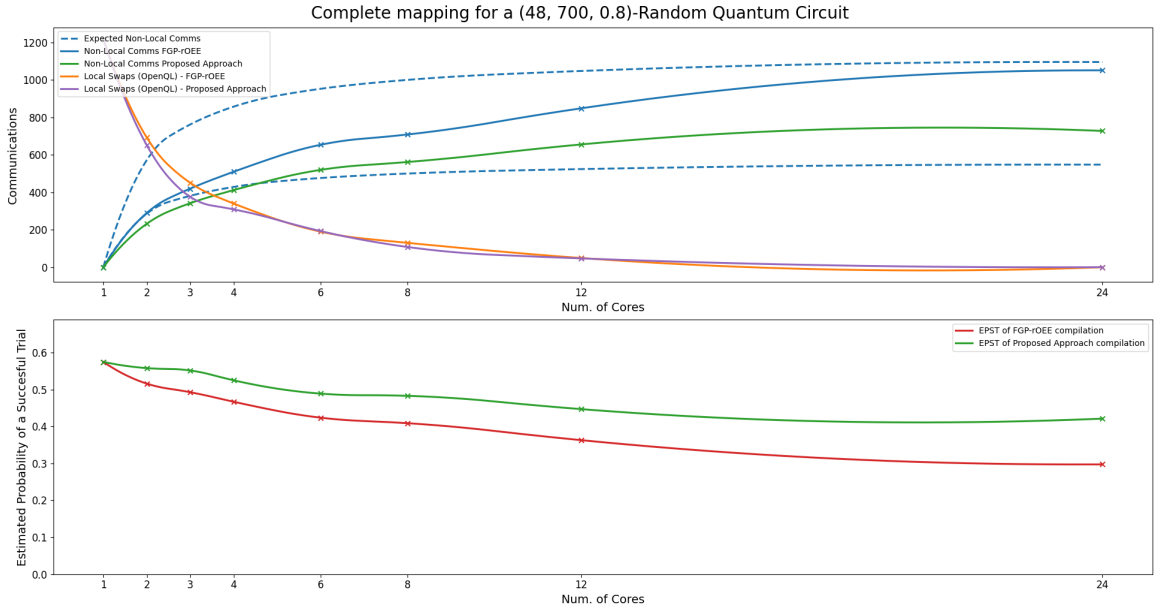


Figure 29: Complete mapping using both FGP-rOEE and Our Approach of a (48, 700, 0.8)-Quantum Random Circuit

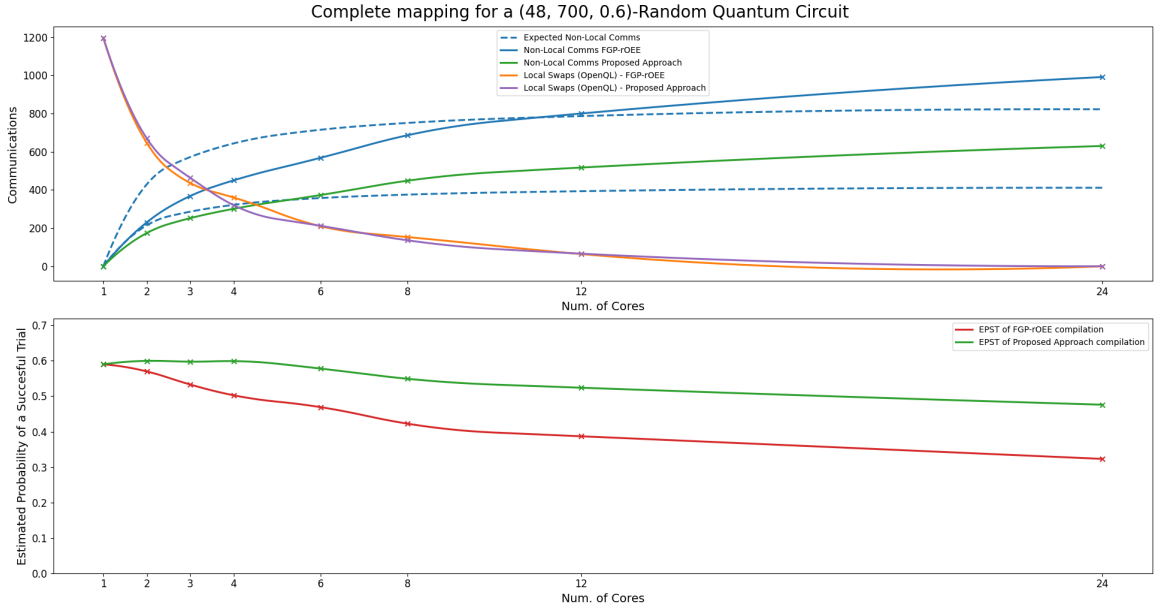


Figure 30: Complete mapping using both FGP-rOEE and Our Approach of a (48, 700, 0.6)-Quantum Random Circuit

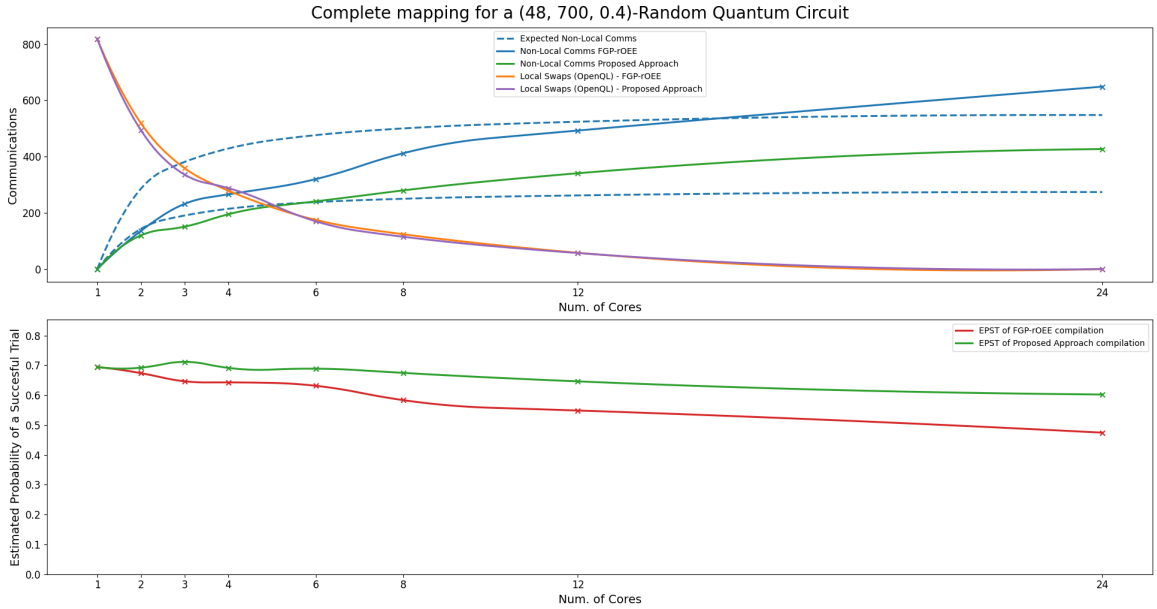


Figure 31: Complete mapping using both FGP-rOEE and Our Approach of a (48, 700, 0.4)-Quantum Random Circuit