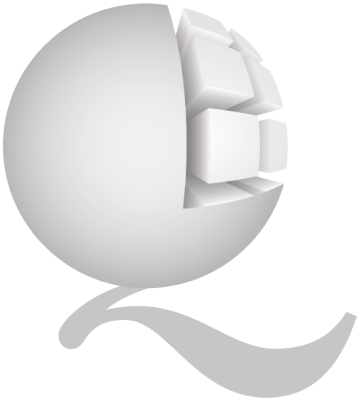
Horizon Europe Programme (2021-2027)
EIC Pathfinder Open
HORIZON-EIC-2022-PATHFINDEROPEN-01



QUADRATURE

Scalable multi-chip architectures enabled by cryogenic wireless/quantum-coherent network-in-package [†]

D5.1: Quantum System Specifications

Contractual Date of Delivery	31/05/2024
Actual Date of Delivery	31/05/2024
Deliverable Dissemination Level	Public
Editor	Artur Garcia (BSC CNS)
Contributors	BSC CNS (leader), UPC, UPV
Quality Assurance	Carmen G. Almudever (UPV), Sergi Abadal (UPC)

[†]This project is supported by the European Commission under the Horizon Europe Framework Programme with Grant agreement no: 101099697.

Document Revisions & Quality Assurance

Deliverable Number	D5.1
Deliverable Responsible	BSC CNS
Work Package	WP5
Main Editor	Artur Garcia (BSC CNS)
Contributors	Eduard Alarcon (UPC), Pau Escofet (UPC), Carmen G. Amudever (UPV)

Internal Reviewers

1. Carmen G. Almudever (UPV)
2. Sergi Abadal (UPC)

Revisions

Version	Date	By	Overview
1.3.0	31/05/2024	*Artur Garcia (BSC CNS)*	Final release.
1.2.0	30/05/2024	*Artur Garcia (BSC CNS)*	Integration of improvements suggested by the quality reviewers (UPC, UPV) and internal revision.
1.1.0	27/05/2024	*Artur Garcia (BSC CNS)*	First draft.

Legal Disclaimer

Executive Summary

The vision of QUADRATURE project aims to explore and validate a new generation of scalable quantum computing architectures featuring distributed quantum cores interconnected via quantum-coherent qubit state transfer links, and orchestrated via an integrated wireless interconnect. This approach thereby supports architectural reconfigurability to serve massive flows of heterogeneous quantum algorithmic demands. On the higher end of the quantum full-stack, the project targets (a) to develop appropriate scalable architectural methods such as mapping, scheduling, and coordination approaches across multiple Qcores, and (b) to demonstrate the scalability of the approach via multi-scale design space optimization for a set of quantum algorithm benchmarks.

This report overviews an overarching double full-stack system architecture encompassing both QC and the enabling communications. The report analyzes the simulation techniques used to study and optimize this architecture. Among these, Design Space Exploration (DSE) and Tensor Networks (TNs) are introduced as the intended methods to study the multi-core structure of the project.

In this context, this report also presents the full quantum system specifications that will provide a top-down driving architectural perspective to the rest of the project. This being exploratory, the report proposes, instead of fixed parameters, parametric ranges to further explore feasibility, performance and resources in a future architectural design space to assess scalable and optimized quantum systems. The report defines and categorizes a set of design variables and technological parameters at different levels of the quantum system, and culminates with a proposed range of system specifications that will serve the basis for future model-based and simulation-based design space explorations.

Abbreviations and Acronyms

CLOPs Circuit Layer Operations Per Second

DMRG Density Matrix Renormalization Group

DSE Design Space Exploration

EPLG Error Per Layered Gate

EPR Einstein Podolsky Rosen state

FoM Figure of Merit

GPU Graphical Processing Unit

HPC High Performance Computing

MAC Media Access Control

MPS Matrix Product State

NISQ Noisy Intermediate-Scale Quantum

NoC Network-on-Chip

QC Quantum Computing

QCore Quantum Core

QEC Quantum Error Correction

QPU Quantum Processing Unit

QV Quantum Volume

RX Receiver

TN Tensor Network

TX Transmitter

The QUADRATURE consortium is composed by

UPV	Coordinator	Spain
UPC	Beneficiary	Spain
TU Delft	Beneficiary	Netherlands
UoS	Beneficiary	Germany
UNICT	Beneficiary	Italy
EQUAL1	Beneficiary	Ireland
BSC CNS	Beneficiary	Spain
NUID UCD	Beneficiary	Ireland
EPFL	Associated partner	Switzerland

Contents

List of Figures

List of Tables

1. Introduction

Current quantum computing (QC) devices are based on different qubit implementation technologies [12, 31, 48]. However, a common approach to all of them is the use of monolithic architectures; that is, a single processor holds all the quantum components in a structure packaging all the quantum features. While this design is appropriate for developing intermediate-size quantum devices, it poses a limitation on their scalability, which is a crucial consideration for the future of quantum technologies. Note that large scale and fault-tolerant QC systems are required to unlock the computational power of this new paradigm and tackle real-world problems.

The multi-core architectural approach proposed in the QUADRATURE project extends this monolithic view, and pushes quantum computers along a road already explored by conventional supercomputers. The result of this exploration has lead to current exascale systems, the most powerful computing devices ever built. Multi-core architectures require a balance among two major processes: computational processing, carried inside each computational cores, and the communication established among these different cores. Communication resolves data dependencies between the processes carried on different cores. In an extreme scenario where communication is not required, each process can progress independently of the work of others. This is a massive parallel situation, where one can obtain massive speedups from supercomputers. In the other extreme case, each process has to wait due to a large amount of dependencies among them, and parallelization brings irrelevant gains.

The vision of the QUADRATURE project is to enable the scalability of quantum computers by adopting a multi-Qcore (Quantum core) architecture, in which multiple quantum processors are combined together to yield a larger computing power [2]. Realizing this vision requires the implementation of a number of different components, as shown in Fig. 1.1, and the development of several innovative technologies. More precisely it implies the exploration of (i) appropriate architectural means to seamlessly manage multiple quantum cores, (ii) an interconnect that supports the quantum and digital communication needs of such architecture, and (iii) a compact integrated quantum-coherent shared medium to realize the quantum side of the interconnect. This architectural shift also calls for redesigning the full-stack quantum system, including both communication and computation, as well as for the use of structured methodologies to perform a cross-layer co-design and analyze the feasibility of the multi-core approach.

This deliverable presents the double full-stack system architecture encompassing both the QC and communications parts and the simulation tools envisaged in the QUADRATURE project required to study the computational capabilities of multi-core quantum computers and its optimization. Furthermore, the full quantum system specifications are introduced. With the aim of exploring the feasibility, performance and

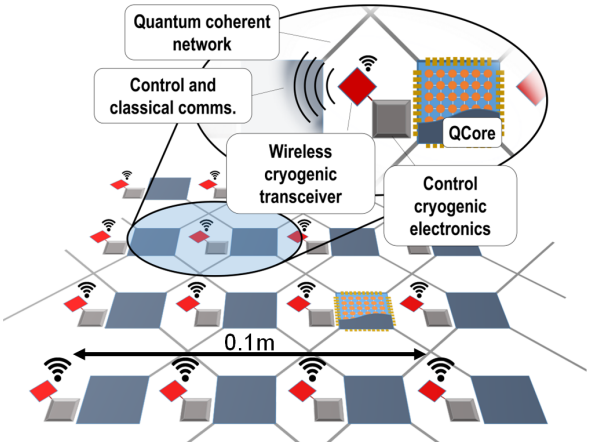


Figure 1.1: The vision of QUADRATURE: several quantum cores are connected via both classical wireless channels and quantum entangling channels to enable the scalability of a quantum computer.

resources of the multi-Qcore architecture, different ranges are proposed for the defined parameters instead of single fixed values.

1.1 WP5 approach to design, architecting and simulation

Within the QUADRATURE project, WP5 integrates all components and related models of the multi-Qcore architecture developed in other WPs by providing a model-based cross-layer system architecture simulation framework. This requires to first specify a set of parameters and specs for the communication-computation full-stack, and the development of advanced structured methodologies and simulation techniques allowing to explore and simulate complex multi-core quantum architectures executing large quantum algorithms.

This process of exploration, in which different design parameters and variables are swept, feeds back information about the requirements of the system and allows to derive optimal configurations of the double full-stack, dimension guidelines and scalability trends. It addition, it also helps to assess and stress inter-Qcore communications as well as to perform an application-oriented co-design of the computational and communication stacks.

To support the exploration, the novelty of our proposal is the large simulation of the computational processes in HPC architectures. This novel approach has to be performed numerically using state-of-the-art techniques. As an output, the result we obtain is an approximation to the real outcome of the quantum device running the same quantum algorithm. The numerical techniques allow more accurate approximations by increasing the resources. We expect that even in approximate conditions, the insight provided by our approach will allow the further optimization provided by the initial exploration.

The study of the optimal architecture parameters starts in this report, but will continue along the duration of the project. While some aspects of the architecture are determined by the design and fabrication of experimental devices, subject to technol-

ogy limitations, other aspect of the multi-core architecture are open to study. Thus, our research proposes a range of interest of the most relevant parameters of the multi-core QC system. With this approach we aim to:

- Adjust our study to the real conditions on the experimental implementation
- Expand the capabilities of multi-core systems in extreme conditions
- Understand the limitations of single and multi-core architectures

1.2 Organization of the report

This report is organized to present the exploratory ranges after introducing the tools and methodologies used in the project. This Chapter 1 serves as a general introduction to the problem of design in multi-core architectures. Chapter 2 introduces in detail the double full-stack, the central abstraction defining the elements of a multi-core architecture that will be used in this study and on the work of WP5. Chapter 3 presents in a detailed form the methods proposed to materialize the study introduced in this report, namely Design Space Exploration (DSE) and Tensor Network (TN) methods. Chapter 4 exposes the ranges of exploration for the identified relevant parameters. Finally, Chapter 5 summarizes the report and raises some conclusions of the project at this stage.

2. The Double Full-Stack Quantum Computing Architecture

Nowadays, Noisy Intermediate-Scale Quantum (NISQ) computers are implemented as single-chip processors, also referred as single-core quantum processors, in which all qubits are integrated within a single chip. This monolithic architecture is hardly scalable due to challenges in the control electronics and wiring [47], an increase of undesired interactions between qubits (i.e. crosstalk) [19] and a decrease of the device uniformity and yield. To overcome these challenges and solve the scaling problem, modular QC architectures have been already proposed for different qubit implementation technologies [5, 26, 32, 37, 49]. The main idea is to combine multiple quantum processors and connect them via single control systems, classical communication links and ultimately quantum communication technologies [16, 24, 25]. We refer to the latter, in which both classical and quantum communication channels are incorporated as multi-core QC architectures. They will allow performing distributed multi-core QC in which a large algorithm consisting of more qubits than there are in a single processor, is partitioned into smaller instances and executed on several quantum chips.

With this novel architectural approach, new challenges emerge as pointed out in [44] that include: i) the implementation of input/output communication ports for each core (processor) as well as the definition of the ratio of qubits devoted to computation and communication; ii) the development of the technology required for communicating quantum information between chips and corresponding communication protocols; and iii) compilation techniques, including placement and routing of qubits and scheduling of quantum operations, that allow for an efficient distributed multi-core quantum computation. More importantly, it also requires redesigning and extending the so-called full-stack (i.e. different functional layers that connect quantum applications with quantum devices) to incorporate the communication support; that is, a stack in which quantum computation and communication layers are intertwined.

In [44], Rodrigo et al. introduce a general-purpose (i.e. no specific qubit or interconnect technology is assumed) layered stack specific to multi-core QC. We call it a *double full-stack* as it merges the traditional computing stack (application, runtime/compiler, micro-architecture, hardware) with the communication stack (routing qubits among cores, qubit reservation and swapping, etc.).

The full-stack layered architecture vision for multi-core quantum computers that we assume is presented in Fig. 2.1. The different abstractions of the quantum computer at each of the layers are included in the *stairway*: the step treads correspond to elements that configure that specific layer and the step risers its key functions. The whole network layer and the elements included in the red “wedge” correspond to the multi-chip implementation-specific kernel of the stack. Quantum data transfers in multi-core quantum computers affect all the way from the high-level code to the physical opera-

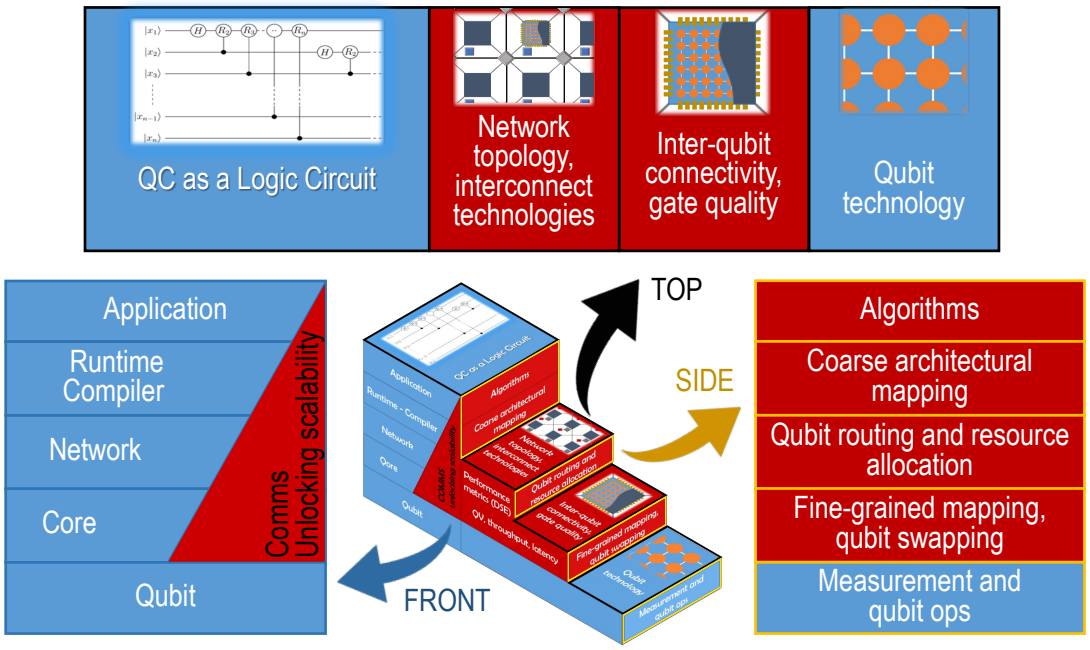


Figure 2.1: A double full-stack multi-core quantum computer vision.

tions performed for measuring a qubit. In the next sections, we do a quick overview of the different layers of the double full-stack architecture.

2.1 Qubit layer

The foundational layer of a quantum computer comprises individual qubits that can be independently controlled and read. This layer can be split into logical and physical sub-layers when using quantum error correction (QEC) protocols [22], where multiple physical qubits function as a single logical qubit.

Although no quantum communication occurs at this level, its performance is crucial as it influences latency and qubit rates in higher communication layers. Quantum communication is akin to "transporting physical qubits", so any factor affecting a qubit impacts overall performance.

Key factors here include decoherence processes, measurement, and gate performance, which depend on the qubit technology and its development stage [3,35]. These factors indirectly but significantly affect upper-layer communications, with coherence time setting a limit on operation duration before quantum information degrades. Long gate latencies and low qubit gate fidelities also impact accurate quantum information transmission.

2.2 Core layer

The core layer of a quantum computer consists of qubits on a single chip that can perform one- and two-qubit gates. In such setups, some qubits act as transducers or communication ports to connect different cores [11]. The qubit interconnection graph, which can follow various topologies (e.g., all-to-all, ring, or 2D lattice), along with intra-core communication technology, defines intra-core connectivity, communication laten-

cies, and qubit transportation capacities. Control wiring and qubit technology influence the minimum qubit-to-qubit distance, impacting the core's area and communication latencies.

Two-qubit operations are typically constrained to adjacent locations, necessitating frequent qubit movement or swapping, making two-qubit gate quality metrics critical for performance. These metrics include two-qubit gate latency (operation time) and two-qubit gate fidelity (operation accuracy), with long latencies and low fidelities degrading quantum information over multiple transfers.

The performance of communications is also influenced by the interconnection topology, the number of qubits per core, and inter-qubit spacing, which affects travel distance and duration.

2.3 Network layer

This layer manages inter-core connections in a quantum computer, integrating both classical and quantum networks for control messages and quantum state transfers (e.g., qubit shuttling, quantum teleportation) [28, 43]. It optimizes inter-core communication through medium access protocols, entanglement distribution, resource reservation, and network scheduling, all of which depend on the technology used.

Key factors include inter-core topologies and interconnect technologies [21], which define core-to-core distances, communication latencies, and qubit transfer rates. Parameters like the number and fidelity of EPR generators for teleportation are crucial.

Inter-core quantum communications involve qubit transportation for local two-qubit gates (qubit routing) and potentially remote gates, necessitating efficient network policies and protocols to meet the stringent computational requirements of lower layers.

2.4 Runtime/Compiler layer

This layer is the first logical layer, abstracting the physical elements from their logical representation in a multi-core quantum computer. It compiles code to quantum assembly, coordinates instruction execution, and performs coarse architectural mapping, similar to algorithm partitioning in classical many-core architectures, aiming for optimized processing.

It applies offline optimizations considering the architecture's limited resources and specific characteristics, such as inter-core network capacity, topology, core features, and qubit technology. At this level, the quantum computer is viewed as a set of connected quantum cores or quantum processing units (QPUs).

Inter-core communications and details about the multi-core platform's capabilities and topology are integral to the compilation and mapping process. Qubit traffic, inherently deterministic due to its direct computation relationship, is loosely scheduled at this layer but ultimately controlled by the network layer.

2.5 Application layer

The uppermost layer corresponds to the code description of the quantum algorithm to be run on the quantum computer. This layer is agnostic to hardware, meaning that

low-level architectural details or constraints are not considered. There is no reference to any explicit communication operation unless it is performed within a large-scale quantum network such as the Quantum Internet.

In any case, the code might include some compiler directives enabling optimized qubit distribution and instructions execution, as it is already done in multi-core classical computing.

In order to evaluate and optimise this double full-stack architecture, quantum system specifications that include parameters and variables from both computational and communication stacks need to be defined. Furthermore, simulation techniques and structured methodologies that consider such architectural parameters are required as it will be discussed in the next sections.

3. Methods for Architecture Exploration and Simulation

As previously mentioned, multi-core modular architectures are a promising approach for scaling quantum computers. In this project, a full-fledged multi-core QC system will not be implemented, some of the key components will be experimentally validated. Still, we have the opportunity to evaluate, through simulation, the potential of multi-core QC potential to unlock the scalability issue of monolithic designs.

As depicted in Section 3.1, by using structured DSE, we will be able to explore the entire design space without being limited by the "intuition" and designer's previous experience that might hinder the way to the optimal (but maybe not intuitive) solution. In addition, we will be able to identify design trends and guidelines. Applying DSE requires either underlying models or quantum system simulators over which the design space can be explored. However, deriving models or simulation is itself a challenge.

As described further in Section 3.2, simulating quantum systems to obtain the expected output of quantum algorithms running in such systems is a computationally hard task due to the exponentially increasing size of the vector space that allocates the representation of quantum states. Due to this exponential cost, direct simulation techniques require the full memory space of a large supercomputer to represent moderately small systems of a number of qubits around $N \approx 50$. A number of different techniques aim to reduce this computational cost by performing an approximate simulation, representing only the relevant degrees of freedom of the system.

3.1 Structured design space exploration methodology

DSE is a structured design methodology that allows optimizing a system by finding extreme points of a given cost function or Figure of Merit (FoM) based on some parameters of interest that describe the performance, quality, or overall cost of the solution [23, 27]. This optimization relies on modeling the interdependencies among the different performance metrics and the variables describing the system. This modeling process might include analytic/theoretical expressions, behavioral models, computer-based simulations, or their zone-wise combinations. However, it is important to note that DSE is used to design, not just to optimize. Performance metrics optimization is in fact just one of the DSE use cases, as it is also useful for rapid prototyping or system integration with no need for analytical metrics [23]. Indeed, DSE uses the optimization framework to control the design process by looking for trends and guidelines in the system performance when varying the available parameters. Whatever the design problem is, if the analysis is correctly prepared, DSE will not blindly look for "the extreme-case highest-performing scenario", which could be unpractical or ignore sub-

optimal options that may suffice for the actual context or the resources constraints. Rather, the main virtue of DSE is to be able to consider system-wide trade-offs and different metrics that may also affect the design optimality.

For example, a DSE analysis of a network deployment will not optimize the average throughput of the entire network but will take into account deployment costs and qualitative characteristics such as network reliability or flexibility. DSE achieves this by letting the designer concurrently sweep all the open variables in the design space, instead of "manually" tweaking them in a one-by-one approach and consolidating several performance/cost metrics into a single FoM, which is then optimized.

DSE optimization can be generically formulated as

$$max_v \quad \Gamma = f(J(v,p)) \tag{3.1}$$
$$s.t. \quad i(v,p) \leq 0 \tag{3.2}$$
$$e(v,p) = 0 \tag{3.3}$$
$$v \in D \tag{3.4}$$

where Γ is an objective function to optimize, and $J(v,p)$ is the vector containing the different optimization metrics considered. Each point in the solution space is represented by the pair (v,p), where v is the vector containing the decision variables that determine the system design, and p the vector of fixed parameters that specify the environment/scenario the system is placed on. A system of equalities in vector e and inequalities in vector i constrain the problem optimization. The solution v belongs to the feasible domain D.

When facing a design problem, we are not going to evaluate each solution in the explored design space by means of an experimental prototype and measurement. To enable such an extensive exploration, metrics are extracted for each solution through previously obtained models. They might come from analytical models (either theoretical or behavioral), simulations, or experimental data from a subset of the solution space.

The flexibility of this framework is what makes DSE a powerful approach for complex design optimization problems. In summary, this methodology facilitates the task by:

- Exploring the entire design space given a predefined set of parameters and variables.
- Providing design trends and guidelines.
- Being valid also for early design decisions, when there are no experimental data sets, computer simulations, or even analytical models for the performance metrics of the system.

3.1.1 Applying DSE to multi-core QC systems

DSE provides a resilient design flow that fits the characteristics of any design problem within QC: the largely unexplored design space, with still many design decisions left open, the high cost and current unfeasibility of experimenting with physical implementations, and the novelty of the quantum realm (the quantum weirdness), requiring multi-disciplinary collaboration [15]. For that reason, DSE has been already used

for different problems in the field of QC: optimal quantum arithmetic reversible circuit synthesis [50]; optimizing the parameters of the Quantum Approximate Optimization Algorithm (QAOA), a quantum-classical hybrid technique to solve NP-hard problems in NISQ quantum computers [1]; and mapping, either by reducing circuit overhead for specific target hardware [40], or by benchmarking existing mapping approaches and deriving optimal strategies for specific quantum algorithms and quantum processors [8, 56]. Most interestingly, DSE has already been applied also for architecting QC systems. Focusing on NISQ ion trap QCCD architectures, in [38], Murali et al. gave out some optimal parameters, operation implementation of gates and communications, as well as topology choices for improving scalability in the near term. This would be a qubit-technology-specific case study covering a very small space in our overarching approach on scalability and communications analysis for going beyond NISQ through modular architectures.

Note that when applying DSE to a given problem, one needs to determine the solution domain (i.e. the variables and parameters we can tweak in order to optimize the design), the performance/quality metrics, and the aggregation function that groups them into the FoM. So far, just a few system-wide performance metrics like Quantum Volume (QV) [18], Circuit Layer Operations Per Second (CLOPS) [54], and Error Per Layered Gate (EPLG) [36] have been proposed [4] that measure the quality and speed of monolithic quantum computers.

3.2 Tensor Networks

Among the techniques used to simulate quantum circuits, computational developments in recent years have established Tensor Networks (TNs) as the standard technique. The main reason is the flexibility of this methodology to approximate at different levels the representation of a quantum computation. In the following sections we outline the methodology of TNs, and its use in this project to approximate quantum circuits.

3.2.1 Introduction to Tensor Networks

TNs are a quantum-inspired novel framework originally developed to study condensed matter systems which provides a general way to systematically perform controlled truncations of high-dimensional linear algebra problems (see [17, 41, 45] for reviews). This represents a formidable tool, which has allowed to study with unprecedented accuracy a wide variety of quantum many-body systems, a task which, if performed with traditional methods, would involve an exponential cost in computational resources as the number of constituents of the system grows.

In particular, the development in 1992 of the Density Matrix Renormalization Group (DMRG) [55] and its subsequent derivation in terms of Matrix Product States (MPS) promoted TNs as the de facto best approach for studying ground-state properties of strongly correlated lattice systems. Later, subsequent works introduced novel algorithms for studying the thermodynamic behavior as well as the dynamics of these systems [42].

The efficient description of quantum many-body systems with local interactions in terms of TNs revolves around the concept of correlations and entanglement: when the components of a system are mostly uncorrelated, it is possible to work with highly com-

pressed representations of the wave function of the system, essentially determined by a parameter, the so-called bond dimension, governing the size of the tensors involved. For a finite bond dimension, this results in a cost which only scales polynomially with the number of constituents. This kind of representation has been rigorously shown to provide an excellent description [52] of ground states of gapped 1D Hamiltonians and thermal states, and over the past few years numerous algorithms and prescriptions for optimizing these states have appeared in the literature [9]. At the same time, the introduction of novel structures such as Project Entangled Pair States has allowed to employ the powerful TN machinery to higher-dimensional systems [9, 41].

In spite of the great success when dealing with equilibrium properties, the description of the dynamics of quantum many-body systems with matrix product states still presents significant challenges: performing a naive time evolution of a many-body wave function quickly runs into the so-called entanglement barrier, corresponding to an exponential increase of the bond dimensions required for an accurate state description [46]. A recent breakthrough in this direction has been the proposal of a novel series of algorithms based on a transverse contraction of the TN corresponding to the time evolution of the system, which -at least in some circumstances- seem to be capable of circumventing this barrier [7, 39]. This has led to the introduction of the ideas of temporal MPS, and the corresponding temporal entanglement [14, 20, 33], a concept which is currently the object of intense investigations both from the numerical and theoretical side. Given the non-hermitian structure of the transfer matrices involved in these calculations, many standard optimization algorithms which are traditionally used in the literature cannot be straightforwardly applied to study this problem, and crude power methods are still widely used for this purpose. A new series of algorithms backed by analytical insights on the properties of these states are thus urgently needed in order to provide a proper understanding of temporal entanglement and shed light on the long-standing puzzle of thermalization of quantum systems. The study of both discrete and continuous time evolution of many-body systems is also one of the battlegrounds where the competition between classical and quantum computers is taking place, as recent claims of quantum advantage in this context [30] have been challenged by TN simulations performed on classical computers [10, 34, 51].

Beyond the traditional condensed matter applications, TNs provide a powerful framework which can be exploited in a much wider context. Indeed, a large number of numerical physics problems are based on expensive linear algebra operations, and the tools developed by the TN community can be applied with profit to those as well. The fundamental strategy here is to express the problems using appropriate degrees of freedom, trying to minimize the amount of correlations between them. It is then possible to resort again to highly compressed representations such as MPS for the relevant equations, which can be solved in an efficient way.

3.2.2 TNs methods

The fundamental element used in the TN representation are Tensors. This family of mathematical objects includes scalars, vectors, matrices, and $N-$dimensional objects in the general case. Tensors are grouped in TNs by establishing a collection of connections among the different tensors. These connections connect different dimensions of a pair of tensors each. The resulting structure can be represented by a graph structure (see Fig. 3.1).

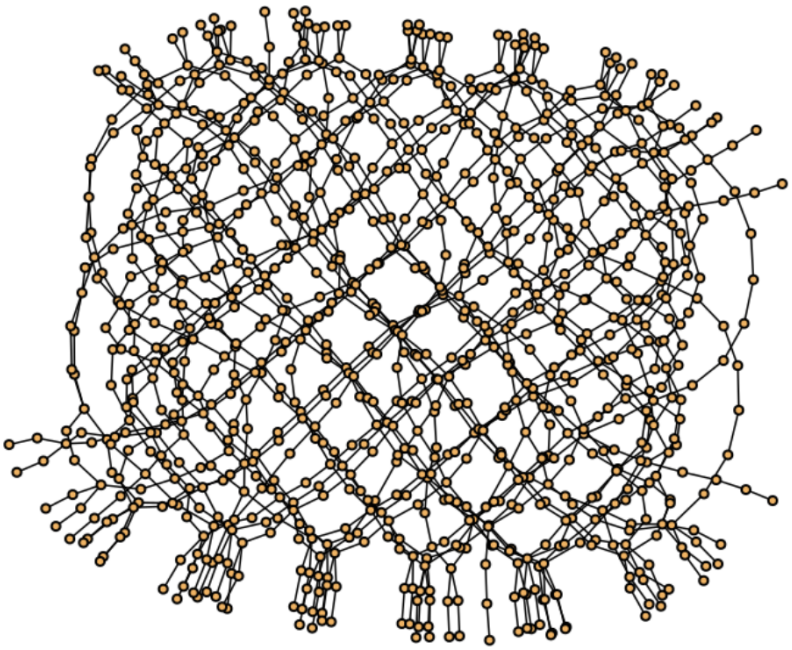


Figure 3.1: A Tensor Network representing a random circuit on the Sycamore architecture with $N = 53$ qubits.

To facilitate the description and manipulation of TNs, a graphical notation has been established. This approach simplifies the representation of complex TNs, and includes the relevant information of each object. To start with this notation language, first we introduce the Tensor Objects, namely vectors ($1-$order tensors), matrices ($2-$order tensors) and $3-$order tensors:

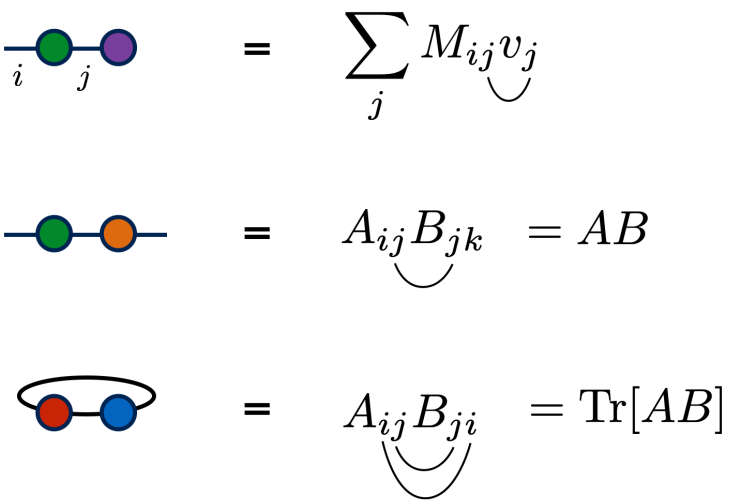
vector	v_j	
matrix	M_{ij}	
3-index tensor	T_{ijk}	

(3.5)

The TN representation is equivalent to the algebraic operations of vector-matrix and matrix-matrix multiplications. For this, the operation along a particular pair of common indices between two tensors, as in a matrix-vector product

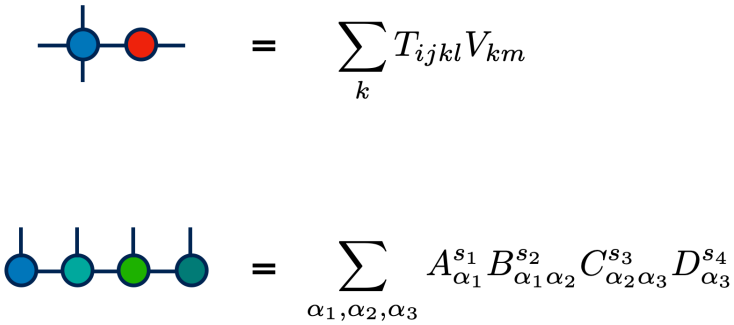
$$A_i = \sum_j M_{i,j} v_j \tag{3.6}$$

is graphically represented as the connection between tensors. Other operations with indices, such as the trace, are also represented with this notation:



(3.7)

The graphical notation allows a simple representation of increasingly complex operations between a large number of tensors. The following diagrams involve a number of tensors with different dimensionalities:



(3.8)

A key observation in order to apply the TN methodology to the study and simulation of quantum circuits is the representation of a gate operation along a quantum computation. We apply a gate G onto an initial state $|\Psi\rangle$. The resulting state $|\Psi'\rangle$ is

$$|\Psi'\rangle = G|\Psi\rangle \tag{3.9}$$

One qubit gates are $2-$order tensors, G^i_j, and 2 qubit gates are $4-$order tensors, $G^{i,j}_{k,l}$, where each index is of dimension $= 2$. If we represent the initial vector as

$$|\Psi\rangle = \sum_{i_1,i_2,\ldots i_N} c_{i_1,i_2,\ldots i_N} |i_1, i_2, \ldots i_N\rangle \tag{3.10}$$

then the operation of applying the gate into the initial state

$$|\Psi'\rangle = G|\Psi\rangle = \sum_{i_1,i_2,\ldots i_N} G^{i_k}_{i'_k} c_{i_1,i_2,\ldots i_N} |i_1, i_2, \ldots i_N\rangle \tag{3.11}$$

is a TN.

The key observation in TN applied to the multi-core problem is the approximation of a quantum state coefficients as, in MPS form:

$$c_{i_1,i_2,\ldots i_N} = \sum_{\alpha_k=1}^{\chi} A^{i_1}_{\alpha_1,\alpha_2} \ldots A^{i_N}_{\alpha_N,\alpha_{N+1}} \tag{3.12}$$

This well known matrix product form can be approximated reducing the range of each α_k to $\chi' < \chi$.

$$c_{i_1,i_2,\ldots i_N} \approx \sum_{\alpha_k=1}^{\chi'} A^{i_1}_{\alpha_1,\alpha_2} \ldots A^{i_N}_{\alpha_N,\alpha_{N+1}} \tag{3.13}$$

However, in the multi-core approach, this approximation is used only in inter-core connections, while the representation inside each core can be exact:

$$c_{i_1,i_2,\ldots i_N} \approx \sum_{intercore}^{\chi'} \sum_{intracore}^{\chi} A^{i_1}_{\alpha_1,\alpha_2} \ldots A^{i_N}_{\alpha_N,\alpha_{N+1}} \tag{3.14}$$

With the combination of exact representation *intra-core* and approximations *inter-core* –where the communication and entangling operations are less strict– we propose TNs as a perfect tool for the simulation of multi-core systems due to the controllable balance of accuracy and performance.

3.2.3 Tensor Network technologies

The TN methods introduced above allow a revision of the simulation of quantum circuits. In this representation, operation of quantum gates is equivalent to algebraic operations among Tensors. The full contraction of the TN is equivalent to calculating the expectation value of some local operators, or the probability amplitude of a certain outcome.

Algebraic operations are easily parallelized using numerical libraries, and accelerated using modern processors and GPUs. However, as stated above, the full representation of a TN is also a large numerical structure requiring the operation of large portions of memory. An important observation allows the approximation of TN by reducing the local dimensions, while preserving the result of a tensor-tensor operation. In this way, the overall size of each tensor can be reduced, the computational cost is also reduced, but the numerical result is potentially unchanged. This approximation will happen to be accurate under some conditions.

In order to simulate the multi-core structures proposed by the QUADRATURE project, we implement the circuits and algorithms using the library TENET[1] under development by the Barcelona Supercomputing Center. This tool allows the simplification, approximation and execution of large circuit simulations. TENET is already implementing simulation functionalities in HPC systems such as the Marenostrum 5 supercomputer (see Fig 3.2).

3.2.4 TN simulation of the double full-stack

The TN methodology is an algebraic representation of the operations occurring in the Hilbert space. It does not represent the underlying technology used for the construction of qubits nor the communication lines. This limitation allows the representation

[1] Access via https://github.com/bsc-quantic/Tenet.jl

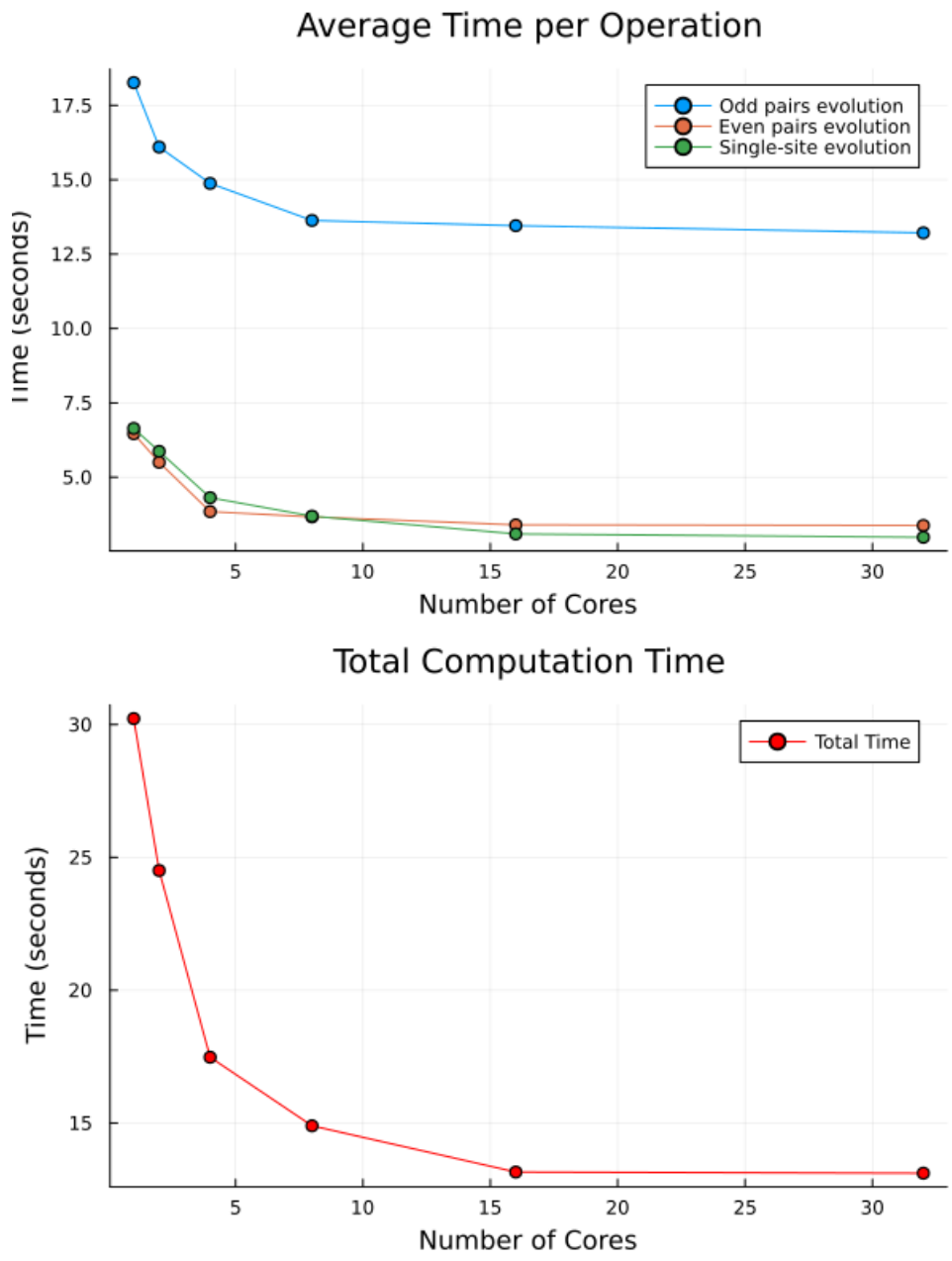


Figure 3.2: Results of the simulation of a Quantum algorithm using an approximate representation in an HPC system (Marenostrum 5 supercomputer) composed of multiple cores executing parallel threads. These results obtained using the library TENET to simulate quantum circuits, obtaining an speedup due to the parallel execution of threads.

of the top layers of the computational stack introduced above, but not the effective Hamiltonian description of the system.

With the TN, we can partition the algorithms following the multi-core architecture. This imposes a hierarchical structure in the TN structure: some tensors represent parts of the system inside a given core, while others are the communication lines among different cores. Classical communication is established with $1-$order connections, while quantum communication requires channels with an order > 1.

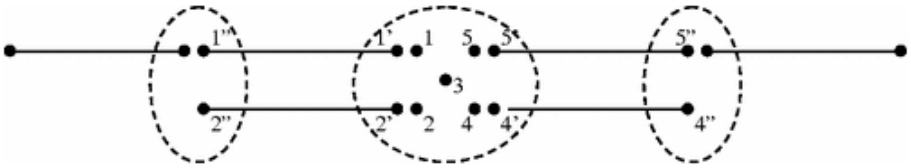


Figure 3.3: (Taken from [53]) Connection between the parameter χ and a collection of teleportation channels established between parts of the system. Each *inter-core* channel can be used to transmit information across cores. This operation can be used in a local state preparation and transmission of the state along the Quantum channels.

There exists a direct connection between the parameters χ used in the representation of the quantum state as a TN (see Fig.3.3) and the communication requirements. The capacity of the connection is equivalent to a number of χ EPR pairs, and one can modify this capacity by adding communication resources. Creating EPR pairs for teleportation between cores is handled in the TN picture by modifying χ.

4. Quantum Computing System Specifications for Exploration

As previously mentioned, a set of technological parameters and design variables at different levels of the quantum system needs to be defined. This section therefore outlines the key system specifications that will be utilized in our simulation/exploration framework and that will provide a top-down driving architectural perspective to the rest of the project. This being exploratory we use, instead of fixed values, parametric ranges that will serve the basis for future model-based and simulation-based design space explorations.

Note that to account for anticipated technological advancements, we employ a gap analysis approach to project future values. Using current error and latency values would not accurately reflect the potential of emerging technologies, so we base our projections on future estimations to better capture expected improvements in QC technologies. Therefore, the values for quantum gate errors, EPR pair generation frequency, and qubit decoherence times have been sourced from the work of Kim et al., as detailed in [29]. In their study, Kim et al. estimate these values for future quantum systems by extrapolating from real results obtained in recent years. This approach provides a forward-looking perspective on the expected performance and capabilities of emerging quantum technologies, allowing us to base our specifications on well-founded predictions.

4.1 Parameters and ranges

We have have selected the following technological and architectural parameters that will be used in our exploration and simulations:

- **Coherence time:** It determines how long a qubit can maintain its coherence (i.e. qubit's lifetime). It can modelled by means of the T_1 and T_2 constants, also called amplitude and phase damping, respectively.
- **Operation fidelity:** How accurate operations such as single-qubit gate, two-qubit gates and measurement can be performed. These are defined from their error probabilities as single-qubit gate error e_1, two-qubit gate error e_2 and measurement error e_r.
- **EPR rate and error probability:** The communication between Qcores will require the generation of Quantum communication channels using entanglement. Key factors are the the entanglement generation error probability e_{EPR} and entanglement generation rate R_{EPR}, that is, how often entanglement succeeds.

- **Number of cores:** Monolithic quantum chips count in a given multi-core architecture, noted as N_{CORES}.
- **Number of qubits per core:** Qubit counts in each single quantum processor, noted as N_Q^{CORE}.
- **Total number of qubits:** Sum of the number of qubits in the quantum computer, N_Q. It is calculated as $N_Q = N_Q^{CORE} * N_{CORES}$.

These parameters are summarized in Table 4.1. They are critical for defining the performance and capabilities of our multi-core QC architecture. These parameters encompass both hardware configurations, such as the number of cores and qubits, as well as operational characteristics, including error probabilities and damping times. By specifying these parameters, we ensure a comprehensive understanding of the system's capabilities and limitations, thereby facilitating accurate modelling and analysis.

Figures 4.1 and 4.2 illustrate the exploratory range of architecture sizes we are considering. These architectures range from 1 to 100 cores, with each core containing between 9 to 100 qubits. Consequently, the total number of qubits spans from 9 to 10,000. This wide range allows us to examine the performance and scalability of various configurations, providing insights into the potential capabilities and limitations of future QC systems.

Table 4.1: System Specifications

Notation	**Meaning**	**Value**	
N_{CORES}	Number of cores/chips	1 - 100	
N_Q^{CORE}	Number of qubits per chip	9 - 100	
N_Q	Total number of qubits in the computer	$N_{CORES} \cdot N_Q^{CORE}$	
		Current	**Projected**
R_{EPR}	EPR pair generation rate	$1 \cdot 10^6$ Hz [5]	$1 \cdot 10^8$ Hz
e_{EPR}	EPR pair generation error probability	0.2 [5]	$2.07 \cdot 10^{-3}$ [29]
e_1	Single-qubit gate error probability	0.015 [6]	$1.4 \cdot 10^{-5}$ [29]
e_2	Two-qubit gate error probability	0.036 [6]	$7.65 \cdot 10^{-5}$ [29]
e_r	Measurement/readout error probability	0.031 [6]	$8.44 \cdot 10^{-4}$ [29]
T_1	Amplitude damping for the memory noise model	$2 \cdot T_2$ [13]	
T_2	Phase damping for the memory noise model	100μs [13]	4.46 ms [29]

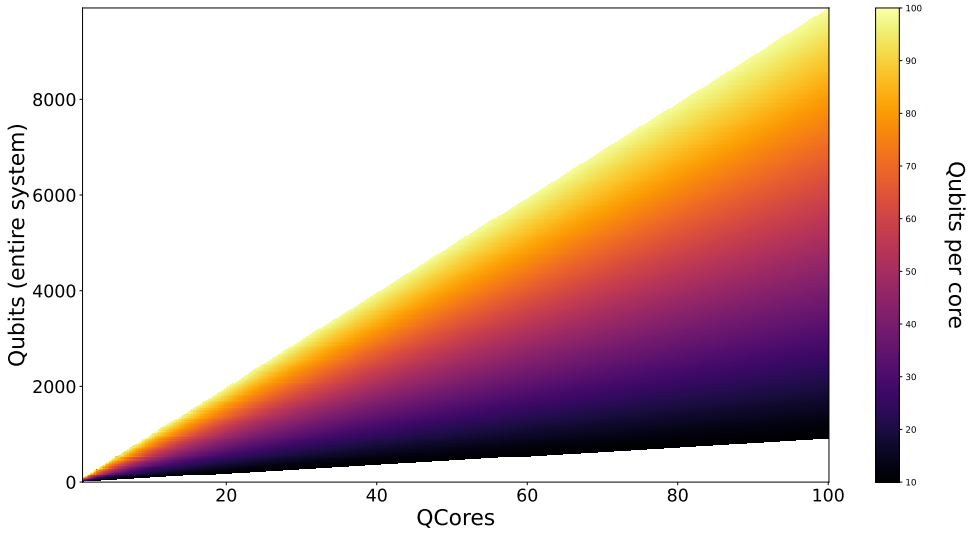


Figure 4.1: Exploration range of architecture sizes.Qubits in each core for a specific number of cores (x-axis) and qubits in the entire system (y-axis).

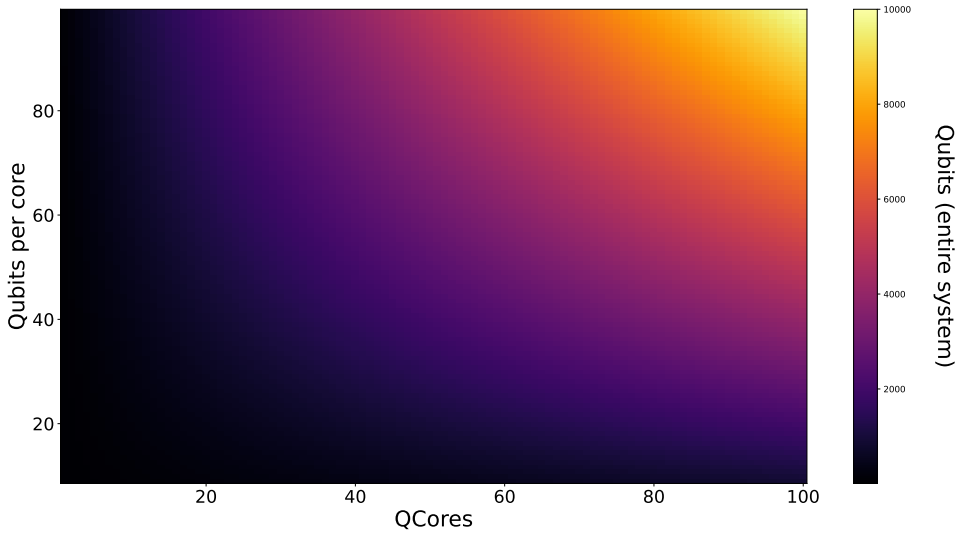


Figure 4.2: Exploration range of architecture sizes. Total number of qubits in the system for a specific number of cores (x-axis) and qubits per core (y-axis).

4.2 System topology

Besides the quantitative specifications, our system also includes important qualitative details. These involve how the cores or chips are connected to each other, known as inter-core connectivity, and how the qubits are connected within each core, referred to as intra-core connectivity. These connections are crucial for efficient communication between cores and for reliable operations within each core, ensuring the system performs well and maintains robust quantum computations.

Figure 4.3 depicts the different inter-core topologies considered in this work. It is important to note that all topologies, except the Grid topology (Line, Ring, Star, and All-to-all), can accommodate any number of cores without requiring structural modifications. In the case of Grid topology, instead, we will only evaluate systems with N^2 QCores in a $N \times N$ grid. Regarding intra-core connectivities, a grid topology (or square mesh) and a heavy hex topology will be assumed.

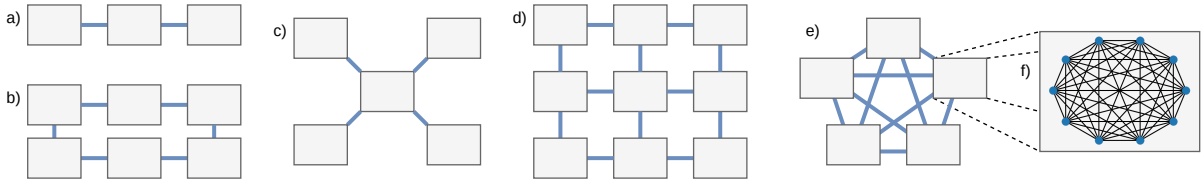


Figure 4.3: Used inter-core topologies. a) Line topology: QCores are connected in a 1D array manner. b) Ring topology: QCores are connected in a circular way. c) Star topology: All QCores are connected only to the central one. d) Grid topology: QCores are connected in a 2D array manner. e)All-to-all topology: All QCores are connected to all other QCores. f) Intra-Core topology. Inside each QCore, all qubits are directly connected to all other qubits in the same QCore.

5. Conclusions

This report has proposed the full quantum system specifications as an overarching top-down driving architectural perspective to QUADRATURE. The report has proposed parametric ranges, in lieu of just metric specifications, to subsequently explore architectural feasibility, performance and resources in a future architectural design space. The final aim is to assess scalability and optimize the design of QUADRATURE quantum systems.

The report has posed and categorized a set of design variables and technological parameters across quantum system level, preceded by a double full-stack system overall system architecture conceptual framework encompassing both QC and the enabling communications, including the qubit layer, core layer, network layer, runtime/compiler layer and application layer. It has been then revisited current methods for architectural simulation, with a particular emphasis on our QUADRATURE bet on tensor network formalism, including a formal self-contained introduction to TN and exploration of their suitability to multi-core distributed communications-enabled large quantum compute architectural double full-stacks. The report finally culminates with a proposed range of system specifications that will serve the basis for future model-based and simulation-based architectural design space explorations.

Bibliography

[1] M. Alam, A. Ash-Saki, and S. Ghosh. Design-space exploration of quantum approximate optimization algorithm under noise. In *2020 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–4. IEEE, 2020.

[2] E. Alarcón, S. Abadal, F. Sebastiano, M. Babaie, E. Charbon, P. H. Bolívar, M. Palesi, E. Blokhina, D. Leipold, B. Staszewski, A. Garcia-Sáez, and C. G. Almudever. Scalable multi-chip quantum architectures enabled by cryogenic hybrid wireless/quantum-coherent network-in-package, 2023.

[3] C. G. Almudever, L. Lao, X. Fu, N. Khammassi, I. Ashraf, D. Iorga, S. Varsamopoulos, C. Eichler, A. Wallraff, L. Geck, et al. The engineering challenges in quantum computing. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 836–845. IEEE, 2017.

[4] M. Amico, H. Zhang, P. Jurcevic, L. S. Bishop, P. Nation, A. Wack, and D. C. McKay. Defining standard strategies for quantum benchmarks. *arXiv preprint arXiv:2303.02108*, 2023.

[5] J. Ang, G. Carini, Y. Chen, I. Chuang, M. A. DeMarco, S. E. Economou, A. Eickbusch, A. Faraon, K.-M. Fu, S. M. Girvin, et al. Architectures for multinode superconducting quantum computers. *arXiv preprint arXiv:2212.06167*, 2022.

[6] F. Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[7] M. C. Bañuls, M. B. Hastings, F. Verstraete, and J. I. Cirac. Matrix product states for dynamical simulation of infinite chains. *Phys. Rev. Lett.*, 102:240603, Jun 2009.

[8] M. Bandic, H. Zarein, E. Alarcon, and C. G. Almudever. On structured design space exploration for mapping of quantum algorithms. In *2020 XXXV conference on design of circuits and integrated systems (DCIS)*, pages 1–6. IEEE, 2020.

[9] M. C. Bañuls. Tensor network algorithms: A route map. *Annual Review of Condensed Matter Physics*, 14(1):173–191, Mar. 2023.

[10] T. Begušić, J. Gray, and G. K.-L. Chan. Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance. *Science Advances*, 10(3), Jan. 2024.

[11] K. R. Brown, J. Kim, and C. Monroe. Co-designing a scalable quantum computer with trapped atomic ions. *npj Quantum Information*, 2(1):1–10, 2016.

[12] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, 05 2019.

[13] J. Bylander, S. Gustavsson, F. Yan, F. Yoshihara, K. Harrabi, G. Fitch, D. G. Cory, Y. Nakamura, J.-S. Tsai, and W. D. Oliver. Dynamical decoupling and noise spectroscopy with a superconducting flux qubit. *arXiv preprint arXiv:1101.4707*, 2011.

[14] S. Carignano, C. R. Marimón, and L. Tagliacozzo. On temporal entropy and the complexity of computing the expectation value of local operators after a quench, 2023.

[15] F. T. Chong. Technical perspective: Applying design-space exploration to quantum architectures. *Communications of the ACM*, 65(3):100–100, 2022.

[16] J. M. Chow. Quantum intranet. *IET Quantum Communication*, 2(1):26–27, 2021.

[17] J. I. Cirac, D. Pérez-García, N. Schuch, and F. Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Reviews of Modern Physics*, 93(4), Dec. 2021.

[18] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3):032328, 2019.

[19] Y. Ding, P. Gokhale, S. F. Lin, R. Rines, T. Propson, and F. T. Chong. Systematic crosstalk mitigation for superconducting qubits via frequency-aware compilation. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, oct 2020.

[20] K. Doi, J. Harper, A. Mollabashi, T. Takayanagi, and Y. Taki. Timelike entanglement entropy. *Journal of High Energy Physics*, 2023(5), May 2023.

[21] P. Escofet, S. B. Rached, S. Rodrigo, C. G. Almudever, E. Alarcón, and S. Abadal. Interconnect fabrics for multi-core quantum processors: A context analysis. In *Proceedings of the 16th International Workshop on Network on Chip Architectures*, NoCArc '23, page 34–39, New York, NY, USA, 2023. Association for Computing Machinery.

[22] D. Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. In *Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics*, volume 68, pages 13–58, 2010.

[23] M. Gries. Methods for evaluating and covering the design space during early design development. *Integration*, 38(2):131–183, 2004.

[24] IBM. Ibm roadmap, 2022. https://www.ibm.com/quantum/roadmap.

[25] IBM. Ibm roadmap 2025, 2022. https://research.ibm.com/blog/ibm-quantum-roadmap-2025.

[26] H. Jnane, B. Undseth, Z. Cai, S. C. Benjamin, and B. Koczor. Multicore quantum computing. *arXiv preprint arXiv:2201.08861*, 2022.

[27] E. Kang, E. Jackson, and W. Schulte. An approach for effective design space exploration. In R. Calinescu and E. Jackson, editors, *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems*, pages 33–54, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[28] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger. Shuttling-based trapped-ion quantum information processing. *AVS Quantum Science*, 2(1):014101, 2020.

[29] J. Kim, D. Min, J. Cho, H. Jeong, I. Byun, J. Choi, J. Hong, and J. Kim. A fault-tolerant million qubit-scale distributed quantum computer. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 1–19, New York, NY, USA, 2024. Association for Computing Machinery.

[30] Y. Kim, A. Eddins, S. Anand, K. Wei, E. Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618:500–505, 06 2023.

[31] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 06 2019.

[32] N. LaRacuente, K. N. Smith, P. Imany, K. L. Silverman, and F. T. Chong. Short-range microwave networks to scale superconducting quantum computation, 2022.

[33] A. Lerose, M. Sonner, and D. A. Abanin. Influence matrix approach to many-body floquet dynamics. *Phys. Rev. X*, 11:021040, May 2021.

[34] H.-J. Liao, K. Wang, Z.-S. Zhou, P. Zhang, and T. Xiang. Simulation of ibm's kicked ising experiment with projected entangled pair operator, 2023.

[35] M. Martonosi and M. Roetteler. Next steps in quantum computing: Computer science's role. *arXiv preprint arXiv:1903.10541*, 2019.

[36] D. C. McKay, I. Hincks, E. J. Pritchett, M. Carroll, L. C. Govia, and S. T. Merkel. Benchmarking quantum processor performance at scale. *arXiv preprint arXiv:2311.05933*, 2023.

[37] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Physical Review A*, 89(2), feb 2014.

[38] P. Murali, N. M. Linke, M. Martonosi, A. J. Abhari, N. H. Nguyen, and C. H. Alderete. Architecting noisy intermediate-scale quantum computers: A real-system study. *IEEE Micro*, 40(3):73–80, 2020.

[39] A. Müller-Hermes, J. Ignacio Cirac, and M. C. Bañuls. Tensor network techniques for the computation of dynamical observables in one-dimensional quantum spin systems. *New Journal of Physics*, 14(7):075003, July 2012.

[40] P. Niemann, A. A. de Almeida, G. Dueck, and R. Drechsler. Design space exploration in the mapping of reversible circuits to ibm quantum computers. In *2020 23rd Euromicro Conference on Digital System Design (DSD)*, pages 401–407. IEEE, 2020.

[41] R. Orús. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, Aug. 2019.

[42] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck, and C. Hubig. Time-evolution methods for matrix-product states. *Annals of Physics*, 411:167998, Dec. 2019.

[43] S. Pirandola, J. Eisert, C. Weedbrook, A. Furusawa, and S. L. Braunstein. Advances in quantum teleportation. *Nature photonics*, 9(10):641–652, 2015.

[44] S. Rodrigo, S. Abadal, E. Alarcón, M. Bandic, H. Van Someren, and C. G. Almudéver. On double full-stack communication-enabled architectures for multicore quantum computers. *IEEE micro*, 41(5):48–56, 2021.

[45] U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, Jan. 2011.

[46] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac. Entropy scaling and simulability by matrix product states. *Phys. Rev. Lett.*, 100:030504, Jan 2008.

[47] F. Sebastiano, J. Van Dijk, B. Patra, J. van Staveren, X. Xue, C. Almudever, G. Scappucci, M. Veldhorst, L. Vandersypen, A. Vladimirescu, et al. Cryo-cmos interfaces for large-scale quantum computers. In *2020 IEEE International Electron Devices Meeting (IEDM)*, pages 25–2. IEEE, 2020.

[48] S. Slussarenko and G. J. Pryde. Photonic quantum information processing: A concise review. *Applied Physics Reviews*, 6(4):041303, 10 2019.

[49] K. N. Smith, G. S. Ravi, J. M. Baker, and F. T. Chong. Scaling superconducting quantum computers with chiplet architectures, 2022.

[50] M. Soeken, M. Roetteler, N. Wiebe, and G. De Micheli. Design automation and design space exploration for quantum computers. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 470–475. Ieee, 2017.

[51] J. Tindall, M. Fishman, E. M. Stoudenmire, and D. Sels. Efficient tensor network simulation of ibm's eagle kicked ising experiment. *PRX Quantum*, 5(1), Jan. 2024.

[52] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Physical Review B*, 73(9), Mar. 2006.

[53] F. Verstraete, D. Porras, and J. I. Cirac. Density matrix renormalization group and periodic boundary conditions: A quantum information perspective. *Phys. Rev. Lett.*, 93:227205, Nov 2004.

[54] A. Wack, H. Paik, A. Javadi-Abhari, P. Jurcevic, I. Faro, J. M. Gambetta, and B. R. Johnson. Scale, quality, and speed: three key attributes to measure the performance of near-term quantum computers. *arXiv preprint arXiv:2110.14108*, 2021.

[55] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992.

[56] H. Zarein. Design space exploration for mapping in quantum computers. Master's thesis, Universitat Politècnica de Catalunya, 2020.