v. 2, n. 1, 2025

International Journal of
Exact
Sciences

ARCHITECTURE FOR A MULTI-QPU DISTRIBUTED QUANTUM COMPUTING SYSTEM WITH CIRCUIT PARTITIONING

Acceptance date: 12/03/2025

Waldemir Cambiucci
Department of Computer Engineering and Digital Systems University of São Paulo (USP) - São Paulo - SP - Brazil

Regina Melo Silveira
Department of Computer Engineering and Digital Systems University of São Paulo (USP) - São Paulo - SP - Brazil

Wilson Vicente Ruggiero
Department of Computer Engineering and Digital Systems University of São Paulo (USP) - São Paulo - SP - Brazil

Abstract: There is a consensus that the distribution of quantum circuits among processing agents is a viable approach to achieve greater scalability with current hardware technologies of noisy intermediate-scale devices. Thus, new quantum computer architectures with multiple processing units should consider additional circuit partitioning steps, with the generation of subcircuits with lower communication costs between partitions. This article presents a modular multi-QPU quantum computer architecture, as well as results with hypergraph partitioning of circuits, as a permanent layer in future distributed quantum system architectures.

INTRODUCTION

The field of quantum computing promises unprecedented benefits in speeding up solutions to complex problems by using the principles of quantum mechanics. However, many challenges prevent these advantages from being applied in all scenarios. These challenges are present throughout the quantum machine development stack [BANDIC 2022], from the physical implementation of qubits to control hardware [RIESEBOS 2022], from control software to applications in real scenarios. This portrait defines the limitation of the computational power of current machines, *noisy intermediate* scale devices, or *NISQ - Noise Intermediate Scale Quantum* [PRESKILL 2018], with few qubits, high error rates, latency of operations, low port fidelity and short state duration, which prevents the execution of large algorithms. This reality is likely to persist until error-tolerant quantum computers become a reality.

The distribution of circuits between processing agents is a viable approach to achieving greater scalability with current hardware technologies, at the cost of greater control over communication between the agents involved [DIADAMO 2021]. NISQ quantum computer architectures with multiple processing units should be considered with additional quantum circuit partitioning steps, making it possible to generate subcircuits with lower communication costs between partitions. In the literature, we find works that explore different approaches to distributed systems such as [MONROE 2014], [LOKE 2022], as well as discussions on circuit distribution [YIMSIRIWATTANA 2004] and circuit partitioning, through the generation of smaller segments for distribution [DAVARZANI 2020], [DAEI 2020]. These works have paved the way in the area of circuit slicing, outlining different approaches to the challenge of segmenting quantum circuits. However, there is still a need to consolidate these techniques in the context of a modular quantum computer architecture, taking into account the operating costs of communication, latency, circuit coupling and the dedication of qubits for communication between partitions. These are real challenges present in NISQ machines of different technologies for implementing actual physical qubits.

This article completes the experiments on hypergraph partitioning discussed in [CAMBIUCCI 2023], presenting a modular architecture for NISQ quantum systems with multiple processing units, with steps for slicing quantum circuits in distributed systems. A hypergraph approach to the partitioning process was adopted, with the aim of achieving a more efficient distribution, reducing communication costs between partitions and allowing a higher level of circuit abstraction, when compared with approaches from previous work. The article is organized as follows: in section 2 we will see a brief description of the layers of a monolithic architecture; section 3 presents the proposal for a multi-QPU quantum computer architecture, with the partitioning of circuits using hypergraphic heuristics; section 4 presents aspects of hypergraphic partitioning and the results of experiments with benchmark circuits; the article ends with conclusions and references.

LAYERS OF A MONOLITHIC QUANTUM ARCHITECTURE

The literature contains different layouts for the components of a modular quantum computer architecture. Some works such as [JONES 2012], [BERTELS 2020] and [BHARTI 2021] explore important steps for handling and processing quantum algorithms. In [BANDIC 2022] we have a comprehensive description of the layers involved in the processing of NISQ quantum computers and a segmentation between compilation and the *instruction* set of the quantum machine, or *ISA - Instruction Set Architecture*. [JONES 2012] highlights the quantum error correction layer, which is crucial for error-tolerant quantum computing. Currently, this quantum error correction layer is migrating to classical control hardware in some architectures and platforms [RIESEBOS 2022]. In [FU 2017] we find relevant aspects for the construction of a quantum computer microarchitecture, with emphasis on the conversion between the high-level representation and the microinstruction lines, which are responsible for the transformations on qubits in the physical hardware layer.

From this literature, we can describe a monolithic architecture through three main layers, as follows:

Programming layer: where we find the coding resources, libraries and frameworks available on the platform for developing quantum programs. Currently, each platform supplier on the market, such as IBM, MICROSOFT, GOOGLE or RIGETTI, offers a wide range of resources for the developer, such as programming tools, simulation, resource estimation and visualization of results.

Compilation layer: from the description of the circuit in a high-level programming language, in this layer the circuit undergoes transformations that make it compatible with the target processor for execution. We can list different related stages such as decomposition, optimization, scheduling, mapping and synthesis. These activities manipulate the gates and quantum operations of the circuit, with various objectives, such as: - preparation of quantum states and efficient initialization of qubits; - physical mapping of qubits according to connectivity in the hardware; - grouping of operations by equivalence and removal of redundancies; - translation of operations into the basic ports supported by the target processor; - decomposition and synthesis of quantum circuits; - and optimization of circuits according to target hardware.

Execution layer: from the synthesis of the optimized quantum circuit, we carry out the translation of the instructions into commands for the electronics involved in controlling the quantum hardware and physical qubits. Currently, many preparation, control and measurement tasks are moving to classical control hardware as well as new definitions for the quantum microarchitecture in use [FU 2017].

MULTIPLE QPU QUANTUM COMPUTER ARCHITECTURE

Considering the needs for circuit partitioning and efficient distribution of processing loads, we propose the reference architecture for quantum computers with multiple processing units in figure 1, with preparation and execution steps for circuit partitioning.

In the proposed design, it is possible to identify the programming, compilation and execution layers, as well as the additional steps for preparing and cutting circuits. As a challenge in the circuit distribution process, we need to address the objective function related to the cost of communication between partitions. The aim is to reduce the number of ebits or qubits dedicated to communication between partitions throughout the execution of the circuit, as well as to optimize the use of links between subpartitions throughout the execution [MARTINEZ 2019] and [DAVARZANI 2020].

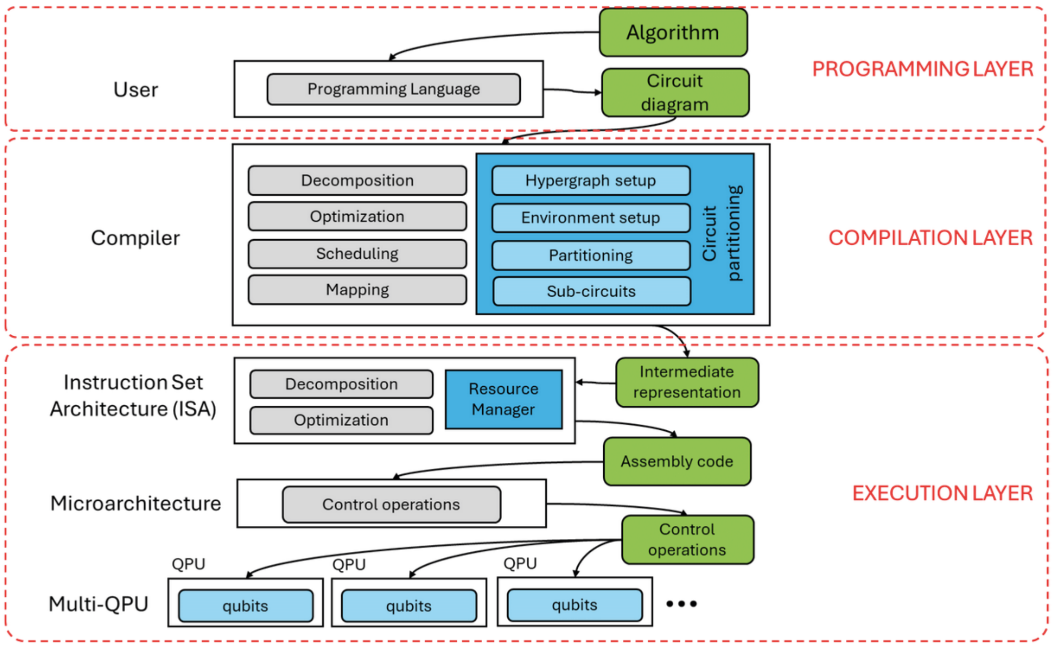

Figure 1. Reference architecture for a quantum computer with multiple processing units and hypergraph partitioning steps for efficient distribution of subcircuits. Source: prepared by the author [2024].

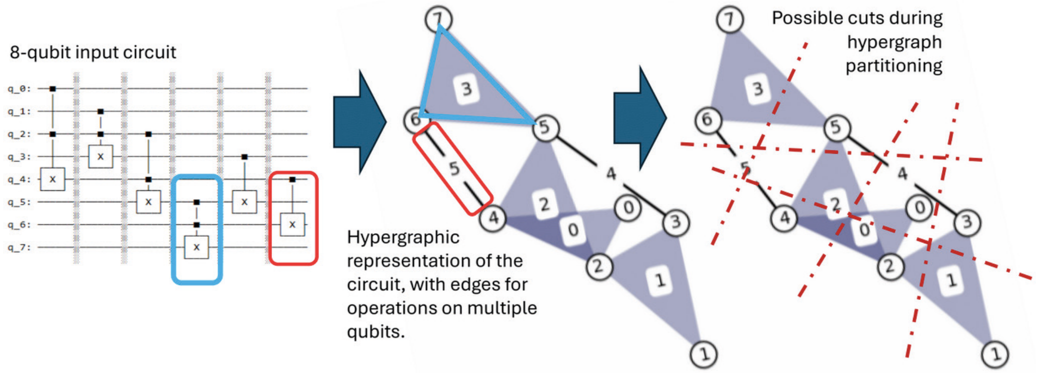
8-qubit input circuit
q_0:
q_1:
q_2:
q_3:
q_4:
q_5:
q_6:
q_7:
Hypergraphic representation of the circuit, with edges for operations on multiple qubits.
Possible cuts during hypergraph partitioning

Figure 2 - Hypergraph representation of quantum circuits and possible cuts during the partitioning process. Source: prepared by the author [2024].

Circuit	n	d	cnots	bipart	fm	red.%	Circuit	n	d	cnots	bipart	fm	red.%
QFT	4	31	18	5	4	20%	QPE	4	28	11	4	2	50%
QFT	6	47	39	12	9	25%	QPE	6	54	34	12	8	33%
QFT	8	63	68	22	16	27%	QPE	8	84	65	22	16	27%
QFT	10	79	105	35	25	29%	QPE	10	94	98	35	24	31%
QFT		119	231	77	56		QPE	15	168	231	77	56	
QFT	30	239	915	330	225	32%	QPE	30	342	910	330	225	32%
QFT	50	399	2525	925	625	32%	QPE	50	588	2522	925	625	32%
QFT	100	799	10050	3725	2500	33%	QPE	100	1188	10047	3725	2500	33%
QFT			14460	5370	3600		QPE			14457	5370	3600	

Table 1. Bipartition for circuits with n qubits and depth d, varying the number of cnots; bipart is the total number of ebits between partitions; fm is the total number of ebits obtained with Fiduccia-Mattheyses; red.% is the communication reduction obtained with hypergraphic heuristics.

HYPERGRAPHIC PARTITIONING OF CIRCUITS

As a strategy for partitioning circuits, we explored the hypergraph representation of the input circuit, for subsequent use of partitioning heuristics in two configurations: the first is based on **weighted hypergraph partitioning heuristics**, using the circuit's coupling ratio as a weight in hyperedges; the second configuration considers **the connectivity restrictions between qubits** present in the target machine topologies, generating weights in hyperedges of the hypergraph. As a circuit impact metric, the coupling ratio can be calculated from the number of quantum gates between multiple qubits, over the total number of qubits present in the circuit under consideration. This measure gives an indication of cost of partitioning the circuit: the higher the coupling ratio, the higher the potential cost generated by cutting the circuit, which requires optimized heuristics for more efficient partitioning.

Numerous partitioning heuristics have been explored in the literature, such as KL [KERNIGHAN-LIN 1970] and SW [STOER-WAGNER 1997]. We focused on the FM heuristic [FIDUCCIA-MATTHEYSES 1982] as the basis for the hypergraph partitioning process, due to its simple implementation and reduced execution time, with a Big-O (n) function, i.e. linear time. Its choice was motivated by the results discussed in the work of [CAMBIUCCI 2023], based on a comparison of different graph and hypergraph partitioning heuristics in the context of quantum circuits.

The FM heuristic is an effective method for dividing a set of vertices in a hypergraph into two groups or partitions, minimizing the relationships between them. The algorithm works in iterative steps, where each vertex is moved from one partition to another if this reduces the number of edges between the subpartitions. We have created a variation of this algorithm, where a weighted telemetry is applied to guide the vertex movement process. This telemetry considers weights on the hyperedges, based on the number of connections between a vertex and a partition. The weights are dynamically updated when over iterations, allowing the algorithm to prioritize moves that result in significant reduction in communication between partitions. By considering not only the structure of the circuit, but also the weights in the hyperedges, the algorithm is able to achieve a more efficient partition, reducing the communication costs between partitions for distributed systems. Centralized bipartite partitioning was addressed because it is widely used as a calibration test, and is carried out by centrally cutting the width of the input circuit, i.e. each of the two partitions has the same number of qubits. Table 1 illustrates the results of this experiment:

With the hypergraph approach used, the reduction in cuts in binary gates was around 35% compared to random central partitioning, when we simply cut the circuit to its width without reordering qubits. Quantum Fourier transform (QFT) and Quantum Phase Estimation (QPE) benchmarks were used, generated using the MQTBench tool [QUETSCHLICH 2023]. The machine used was a 16GB RAM 2.20GHz notebook with 12 cores and 16 processors.

CONCLUSIONS

This paper proposes a modular multi-QPU quantum computer architecture design, with a hypergraph approach based on FIDUCCIA-MATTHEYSES heuristics to optimize circuit partitioning, reducing communication costs between partitions, as previously discussed in [CAMBIUCCI 2023].

REFERENCES

BANDIC, Medina; FELD, Sebastian; ALMUDEVER, Carmen G. Full-stack quantum computing systems in the NISQ era: algorithm-driven and hardware-aware compilation techniques. In: 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2022. p. 1-6.

BERTELS, K. O. E. N. et al. Quantum computer architecture: Towards full-stack quantum accelerators. In: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2020. p. 1-6.

BHARTI, Kishor et al. Noisy intermediate-scale quantum (NISQ) algorithms. arXiv preprint arXiv:2101.08448, 2021.

Brian W KERNIGHAN and Shen LIN. An efficient heuristic procedure for partitioning graphs. The Bell system technical journal, 49(2):291–307, 1970.

CAMBIUCCI, W., SILVEIRA, R. M., and RUGGIERO, W. V., "Hypergraphic Partitioning of Quantum Circuits for Distributed Quantum Computing," 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), Bellevue, WA, USA, 2023, pp. 268-269, doi: 10.1109/QCE57702.2023.10237.

DAEI, Omid; NAVI, Keivan; ZOMORODI-MOGHADAM, Mariam. Optimized quantum circuit partitioning. International Journal of Theoretical Physics, v. 59, n. 12, p. 3804-3820, 2020, https://arxiv.org/abs/2005.11614

DAVARZANI, Zohreh et al. A dynamic programming approach for distributing quantum circuits by bipartite graphs. Quantum Information Processing, v. 19, n. 10, p. 1-18, 2020. https://arxiv.org/abs/2005.01052

DIADAMO, Stephen; GHIBAUDI, Marco; CRUISE, James. Distributed quantum computing and network control for accelerated VQE. arXiv preprint arXiv:2101.02504, 2021.

FIDUCCIA,C.M., MATTHEYSES, R. M., "A Linear-Time Heuristic for Improving Network Partitions," 19th Design Automation Conference, 1982, pp. 175-181, doi: 10.1109/DAC.1982.1585498.

FU, Xiang et al. An experimental microarchitecture for a superconducting quantum processor. In: Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture. 2017. p. 813-825.

JONES, N. Cody et al. Layered architecture for quantum computing. Physical Review X, v. 2, n. 3, p. 031007, 2012.

LOKE, Seng W. From Distributed Quantum Computing to Quantum Internet Computing: an Overview. arXiv preprint arXiv:2208.10127, 2022.

MARTINEZ, Pablo; HEUNEN, Chris. Automated distribution of quantum circuits via hypergraph partitioning. Physical Review A, v. 100, n. 3, p. 032308, 2019.

MONROE, C. et al. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. Physical Review A, v. 89, n. 2, 2014.

PRESKILL, John. Quantum computing in the NISQ era and beyond. Quantum, v. 2, p. 79, 2018. https://arxiv.org/abs/1801.00862

QUETSCHLICH, Nils; BURGHOLZER, Lukas; WILLE, Robert. MQT Bench: Benchmarking software and design automation tools for quantum computing. Quantum, v. 7, p. 1062, 2023.

RIESEBOS, Leon et al. Modular software for real-time quantum control systems. In: 2022 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE, 2022. p. 545-555.

STOER, M., WAGNER, F. (1997). "A Simple Min-Cut Algorithm." Journal of the ACM, 44(4), 585-591.

YIMSIRIWATTANA, Anocha; LOMONACO JR, Samuel J. Distributed quantum computing: A distributed Shor algorithm. In: Quantum Information and Computation II. SPIE, 2004. p. 360-372. https://arxiv.org/abs/quant-ph/0403146.