From Designing Quantum Processors to Large-Scale Quantum Computing Systems

Carmen G. Almudever [1], Robert Wille [2], Fabio Sebastiano [3,4], Nadia Haider [4] and Eduard Alarcon [5]

[1] Computer Engineering Department, Technical University of Valencia, Spain
[2] Chair for Design Automation, Technical University of Munich, Germany
[3] Department of Quantum and Computer Engineering, Delft University of Technology, The Netherlands
[4] QuTech, Delft University of Technology, The Netherlands
[5] Department of Electronics Engineering, UPC BarcelonaTech, Spain

***Abstract*—Design, simulation, analysis and verification methodologies are crucial for developing electronic circuits and systems at large. Whereas long-standing EDA software is used in the semiconductor technology, there is no counterpart for quantum computing systems yet. Although the quantum computing community started utilizing and adapting some of the already existing EDA tools, for instance, to design quantum processors and control electronics for driving the qubits, or even to solve some quantum computing design tasks, they do not fully use the expertise gained over the last decades in the field of design automation. Current intermediate-scale quantum computers have been designed in an 'adhoc' manner with heterogeneous methods and tools. As we are entering the large-scale era, it is timely and key to further adopt EDA methodologies and software for quantum computing. In this paper, we provide an overview on how full-stack quantum computing systems are being implemented nowadays and discuss which the main challenges are for transitioning from this current scenario to a comprehensive framework encompassing full automated system-wide architecting, design, simulation, verification, and test.**

I. Introduction

The expected and unprecedented computational power that quantum computers will bring is closer to become a reality thanks to the several breakthroughs that have been achieved in the field of quantum computing in the past years. Several demonstrations of quantum advantage, a term referring to a quantum computer solving a problem that is unsolvable by any classical computer in a reasonable amount of time, have been performed based on different quantum devices that include superconducting and photonic processors [1]–[3]. Another relevant achievement is the implementation of quantum error correction (QEC) and fault-tolerant (FT) protocols. As quantum hardware is highly error prone due to environmental noise and imperfect control of qubits, the use of QEC codes and FT techniques is essential to protect quantum information and perform reliable and accurate computations. In the last years, many experiments have shown that errors can be suppressed through quantum error correction achieving lower logical error rates [4]–[7]. Recently, the first implementations of fault-tolerant logical quantum gates and even of FT quantum circuits have been realized [8], [9]. In addition, a crucial aspect refers to the scalability of quantum computing systems. In order to perform FT computations and therefore unleash the computational power of quantum computers, processors need to be scaled to thousands or even millions of qubits. The most advanced and promising technology platforms for large-scale quantum computing are currently trapped ions (e.g. Quantinuum System Model H2 with 32 qubits [10]), neutral atoms (e.g. QuEra Aquila processor with 256 qubits [11]) and superconducting circuits (e.g. IBM Condor processor with 1121 qubits [12]), in which the 1000-qubit barrier has been surpassed.

Although quantum computers are not yet at the scale of solving realistic and practical problems, they already integrate all functional elements or layers of a complete system that goes from applications down to quantum devices. Full-stack quantum computing systems consist of a series of software and hardware components that include: i) programming languages for expressing quantum algorithms and compilers to optimise and modify quantum circuits to make them efficiently executable on current resource and noise-constrained processors [13]–[16]; ii) a low-level quantum instruction set that include the gates supported by a given device and related microarchitecture [17]–[22]; and iii) control electronics for operating and controlling the qubits [23]–[33]. In this early stage of these new systems, which are severely limited by the available resources (i.e qubits) and by noise, codesign among adjacent layers as well as vertical cross-layer design is a must and related optimization is required [34]–[36].

Up to now, intermmediate-size quantum computers have been designed in an 'adhoc' manner, with heterogeneous methods and tools that are becoming more and more sophisticated as the complexity of the systems is increasing. However, there is neither a standard and automated design procedure yet, nor comprehensive design flows that fully exploit the benefits of using design automation. In the coming years, having a full-fledged automatic design framework will be indispensable to address the design of large-scale quantum computing systems given their increasing complexity. Note that this higher complexity is not only due to merely higher qubit counts and the introduction of QEC and FT protocols that will require adding extra functionalities and (runtime)

support, but also because of a change in the processors' architecture (i.e. from single-core to multi-core). More precisely, next quantum computer generations are expected to follow a modular approach in which several quantum processors are combined with classical processors and connected via both classical and quantum communication links [37]–[39].

In this context, there is room for further design-oriented optimization, need for multi-domain integrated simulators, and eventually for a comprehensive framework encompassing full automated system-wide architecting, design, verification, and test. In this paper we will review how full-stack quantum computing systems are being designed nowadays emphasizing which simulation, verification and design tools are used and what the challenges are. The paper is organized as follows. In Sections II and III, an overview on the design of quantum computing processors (focusing on superconducting qubits) and the required classical control electronics to perform quantum gates and qubit measurements is provided. Section IV discusses the software needed to support some tasks that are crucial in quantum computing. Section V provides a full-stack perspective of the design of increasing scale quantum computing systems. Finally, some of the open questions and challenges the quantum computing community is facing and for which the EDA community can play a key role are discussed in Section VI. Conclusions are presented in Section VII.

II. Designing Quantum Processors

Electronic design automation (EDA) plays a key role in advancing the field of various qubits by streamlining and optimizing the design process. In the realm of quantum applications, the reliance on microwave signals and systems spans a broad spectrum. The importance of proper electromagnetic analysis and microwave design is evident in the pursuit of groundbreaking advancements in quantum computing, particularly for large-scale, fault-tolerant quantum information processing units. This is impacting quantum processing chip design processes, particularly for superconducting and spin-based systems.

The complex nature of quantum circuits, particularly those based on superconducting technology, demands sophisticated EDA tools to handle complex qubit geometries and intricate quantum effects. Employing advanced simulation tools, engineers and researchers model the complex interactions of electromagnetic fields within the superconducting circuits to predict and understand the qubit's and resonator's behaviour. These simulations take into account various factors, such as the geometry of qubits and Josephson junctions, the impact of fabrication tolerance, and the coupling between qubits and their surrounding environment. The primary challenges in designing a large-scale quantum processor include developing a scalable chip architecture, ensuring proper signal routing, maintaining acceptable levels of cross-talk and unwanted couplings, and suppressing cavity modes, among other considerations.

Superconducting qubits harness the fundamental principles of superconductivity to achieve quantum information processing [40]. Utilizing superconducting materials that exhibit zero electrical resistance, these qubits are implemented as Josephson junctions—tiny devices capable of carrying a supercurrent. The two quantum states, $|0\rangle$ and $|1\rangle$, correspond to the direction of the supercurrent in the Josephson junction. Through controlled manipulation of these quantum states via microwave pulses, superconducting qubits enable the creation of quantum gates, forming the building blocks for quantum computations. The inherent controllability of superconducting qubits make them promising candidates for the realization of practical quantum processors. From the perspective of microwave engineering, the superconducting qubits, such as transmon [41] and fluxonium [42], are systems of coupled nonlinear LC resonators.

By leveraging EM simulation techniques, researchers can predict in advance various performance metrics of such superconducting qubits under different conditions, identify potential sources of decoherence, and optimize the design parameters for enhanced quantum performance. This iterative simulation process is instrumental in fine-tuning the geometrical and material aspects of the qubit, aiding in the development of more robust and efficient quantum processors. In addition to designing and optimizing superconducting chips, electrostatic simulation tools are often used to predict the presence of any unwanted cavity modes that can affect the overall performance of the system.

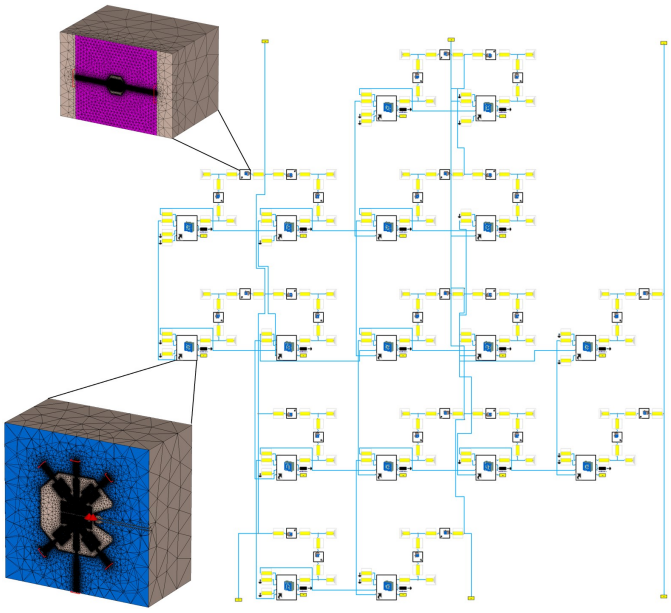


Figure 1: Visual representation of a simulation method used at QuTech [43] for analysing a complete superconducting qubit chip where finite element electromagnetic simulations of the "Starmon" qubits are combined with numerical circuit simulations for accurate and fast computation. Here, CST has been employed as the simulation tool.

Researchers often adapt existing EDA tools, such as HFSS [44], CST [45] and COMSOL [46], or use a combination of tools to meet the specific challenges posed by superconducting qubit designs. In addition, specific tools have been developed by quantum industries and research institutions dedicated to quantum circuit simulation, including Qiskit Metal [47]. In the design of superconducting quantum processors, a common practice involves employing a hybrid simulation approach that combines finite element and circuit simulations. This approach has proven to be a valuable tool for investigating the next generation of chips with numerous qubits. In these simulations, researchers utilize a complex and detailed finite element model for the qubit and a simplified circuit model for the transmission lines as depicted in Fig.1. This strategy is particularly beneficial when analysing large circuits, as the resource-intensive finite element EM simulation of the qubits can be computed in advance. Subsequently, by seamlessly integrating these simulation results into a circuit simulator, much like assembling Lego blocks, researchers can effectively analyse the overall chip performance [48], [49].

For quantum devices based on spin technologies, it has also become essential to have a proper understanding of the microwave performance as the number of qubits in these devices is also rapidly increasing. It is now becoming critical to address the sources of microwave losses, DC and microwave cross-talks. Various research groups within academia and industries are exploring where the simulation works will make a difference by analysing the microwave and electromagnetic performance of current spin-based quantum chips, understand the sources of cross-talk and identify critical components to improve chip performance. Currently, there are design tools like QTCAD [50] that also facilitate electrostatic analysis. Looking ahead, we anticipate the emergence of an expanded array of tools for diverse technologies employed in Quantum Processing Units (QPUs).

III. Electrical Interfaces for Quantum Processors

Quantum processors typically require a classical, i.e., non-quantum, electrical interface to provide the electrical signals driving operations on the qubits, such as single and two-qubit gates, and to perform qubit measurements by reading out the electrical signals from the qubits [52]. Early quantum-computing prototypes primarily relied on commercial bench-top equipment to maximize the flexibility of the experimental setup and to ensure the best possible electronic behaviour so as not to limit the whole computer's performance. While such an approach is feasible for small quantum computers with tens of qubits or less, it quickly becomes unpractical when scaling up the number of qubits, due to the sheer cost of the equipment and its size. To circumvent those limitations, tailor-made electronic solutions have been developed by trading off flexibility for compactness and cost [25], [26], enabling, for instance, the full control of more than 1000 superconducting qubits with a single equipment rack [26]. Although the flexibility and the design cycle of such electronics are not heavily constrained thanks to the extensive use of FPGAs and discrete components, minimizing the target cost function, e.g., cost and size, requires a clear understanding of the impact of the electronic specifications on the qubit performance. The specifications for the required (RF/analog/mixed-signal/digital) electronics could be derived analytically starting from a simplified expression of the quantum Hamiltonian describing the qubits, e.g., as done in [53] for semiconductor spin qubits. This is necessary to optimally budget all the error sources in the electronics but it may be insufficient if the approximations required for an analytical treatment lack accuracy.

As a complementary approach, a co-simulation platform is needed, in which both classical electronics and quantum devices can be simulated to effectively verify the design's functionality and performance and identify any bottlenecks at an early stage (Fig. 2). Adding the simulations of quantum devices as an additional feature in standard EDA tools for electronic circuits should be preferred to exploit their maturity level fully. For instance, a standard Cadence environment for microelectronic design has been enhanced with the simulations of semiconductor spin qubits and superconducting qubits using Verilog-A [51] or equivalent-circuit models [54], respectively. While those approaches can be very useful to co-simulate the electronic interface for a few qubits, they will become rapidly inefficient when scaling up the number of qubits. Simulating the full quantum evolution of a few hundred of qubits will be relevant already in the medium term, since sharing the same electronic controller and read-out over a large number of qubits, e.g., via time-division or frequency-division multiplexing, is a commonly adopted technique to reduce the size of the electronics [24], [31], thus asking for the simultaneous simulation of all the target qubits. Additionally, verifying an even larger system would be required to identify the impact of electronics non-idealities at the quantum-algorithm level, necessarily requiring an

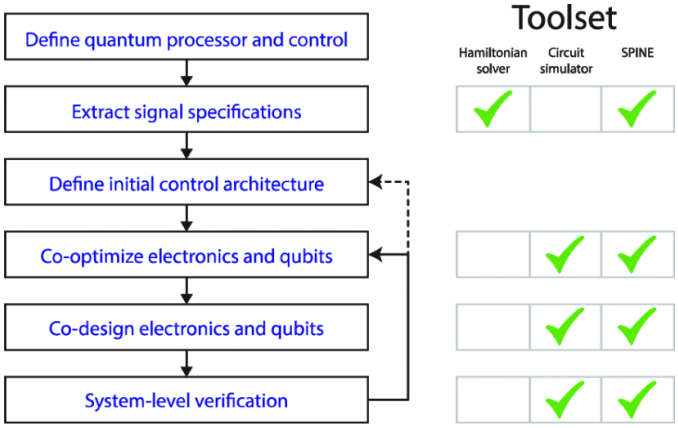


Figure 2: Typical design flow for the electronic interface for a quantum processor, highlighting the need for quantum/classical co-simulations. The specific example shown here for semiconductor spin qubits adopts SPINE (SPIN Emulator) as co-simulation platform. Reproduced from [51].

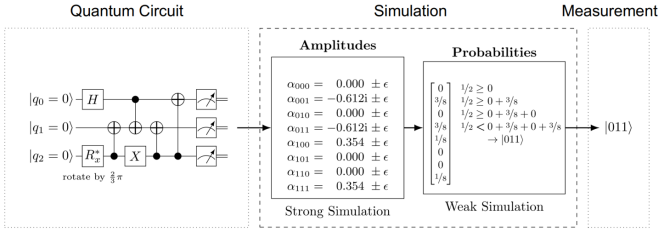
abstraction of both the electronic and qubit layers to make any simulation feasible. With the advent of processors with 1000+ qubits [12], *the EDA community will soon be asked for tools for the electronic/quantum co-simulation and co-design with varying degrees of accuracy depending on the computer scale.*

Computers based on cryogenic qubit platforms will encounter an additional impediment to their scaling: connecting a larger and larger number of cryogenic qubits to a room-temperature electrical interface will require an increasing number of cables, thus incurring into an *interconnect bottleneck* due to the sheer size of the wires, their costs, and their reliability limits. As an alternative, cryogenic electronic controllers have been proposed [23], which is now a widely accepted option both in academia [24], [27]–[30] and industry [31]–[33]. In order to exploit the maturity of the technologies and the EDA tools developed for the semiconductor industry, commercial semiconductor processes are typically adopted, with a widespread preference for commercial CMOS operating at cryogenic temperature (cryo-CMOS) thanks to its VLSI, high performance, and functionality at sub-K temperatures [23]. However, standard device models do not capture several physical effects dominating the device behavior at cryogenic temperatures. Although there is not yet a standard model for cryo-CMOS devices, the research in this field is quite active [55]–[59], even if some aspects of the device physics, such as the subthreshold behavior, the noise, and the mismatch, are not fully understood. Enabling the reliable design of cryogenic circuits for large-scale quantum computers thus demands *the development of robust device models for classical semiconductors devices* and *the adaptation of existing semiconductor EDA tools and flows to take into account the specific features of cryogenic technologies* that will directly affect design styles and methodologies, such as the heavily reduced leakage [60] and cryogenic-aware forward body biasing (FBB) [61].

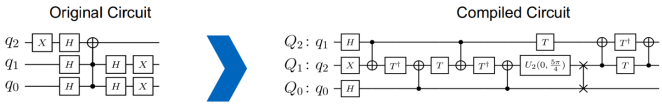
IV. Software for Quantum Computing

In the classical computing realm, software is omnipresent and not only used to realize applications but is essential in the design of electronic circuits and systems themselves. In fact, the ever-increasing complexity of designing those systems put constant pressure on engineers to come up with more efficient and more scalable design software. Over the past decades, this led to sophisticated methods as well as an integration of them into comprehensive design flows. The resulting solutions can handle complexity of impressive scales and are a main reason for the utilization and penetration of (classical) electronic devices into almost all parts of our daily life.

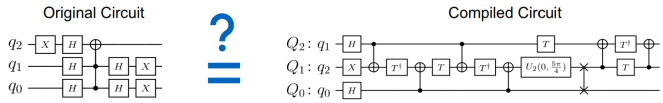
For the quantum realm, similar developments are currently emerging. In fact, a large yet scattered amount of software tools have been developed in the recent years. Prominent examples include IBM's Qiskit [62], Google's Circ [63], Microsoft's Azure Quantum [64], Quantinuum's TKET [65], or Xanadu's Pennylane [66]. They already provide quite



(a) Simulation of quantum circuits.



(b) Compilation of quantum circuits.



(c) Verification of quantum circuits.

Figure 3: Illustration of selected design steps covered by software.

comprehensive software solutions for certain use cases (e.g., to realize specific quantum computing applications).

At the same time, these tools do not (yet) fully utilize experiences gained in the field of design automation over the last decades. This leaves huge potential for further improvement untapped. In fact, many of the design problems considered for quantum computing are of combinatorial and exponential nature (some have even been proven to be NP-complete [67], coNP-hard [68], or QMA-complete [69]). Those are complexities design automation experts are quite familiar with and have developed very efficient solutions for.

In the following, a brief selection of design tasks are listed, for which software and particularly design automation background can be utilized. The list is motivated by developments provided through the *Munich Quantum Toolkit* (MQT) which offers corresponding solutions as open-source implementations (available at https://www.cda.cit.tum.de/research/quantum/mqt/). By this, the list provides an overview of the full spectrum of tasks where dedicated software can be useful and, in fact, is required. Selected references for a more detailed reading is provided for each task. Additionally, some steps are briefly sketched in Fig. 3.

- *Supporting End-Users in Realizing Applications:* In order to realize a quantum computing application, various steps have to be conducted. End-users (usually domain experts in their fields) expect to use the respective platforms without having to understand the specific quantum computing underpinnings. Software can help here in shielding corresponding quantum computing aspects when designing applications (as envisioned, e.g., in [70]) or provide guidance through compilation/design flows (as provided, e.g., in [71]). Besides that, *Quantum Resource Estimation* can substantially help to explore

whether a planned application actually is suitable for quantum computing and/or what costs/requirements would be needed for it [72].

- *Quantum Circuit Simulation:* Simulating quantum applications/circuits remains an essential step in every design flow. Originally, this was used to test an idea when corresponding quantum computers are not available. But even with corresponding machines, only simulation will provide the full details such as all amplitudes of a output state of a quantum computing application, whereas the actual machine only provides probabilistic measurement. At the same time, performing quantum circuit simulations requires the multiplication of matrices and vectors of the size $2^n \times 2^n$ and 2^n (n being the number of qubit), respectively, i.e., remains an exponential problem. Various complementary software solutions exist which aim to tackle this complexity [73]–[76]. If additionally noise effects are considered, this complexity even increases further [77].
- *Compilation of Quantum Algorithms:* Quantum computers, as well as their classical counterparts, usually only support a limited set of elementary operations. In addition, they are bounded by further constraints such as limited qubit connectivity which dictates what qubits are allowed to interact with each other, or require a dedicated scheduling/transportation of qubits. All these restrictions makes that high-level descriptions of quantum algorithms have to be compiled through different layers of abstractions before being executable on an actual machine. This includes platform-agnostic steps such as the realization of an oracle, which can be done e.g., by reversible circuit synthesis methods as proposed in [78] as well as dedicated mapping methods (as proposed in [79], [80] for superconducting technologies), shuttling methods (as proposed in [81] for ion-trap technologies), combinations of both (e.g., for neutral atom technologies [82]), and more.
- *Verification of Quantum Circuits:* Throughout all levels of abstraction during a compilation flow, a quantum algorithm/circuit is subject to substantial changes. During this process, it is of utmost importance that the originally intended functionality is preserved. Accordingly, verification methods such as equivalence checkers gain momentum [83], [84]. They offer solutions that automatically check whether a compiled quantum circuit still realizes the same functionality as the originally given application.
- *Quantum Error Correction:* Quantum operations and qubits are fragile and prone to errors. In order to physically implement quantum computers that are capable of conducting reasonable computation, quantum error-correction and fault-tolerance are needed. The principle of QEC is to encode quantum information in a much larger state space (i.e. encoding logical qubits into several physical qubits) and detect and correct errors by using indirect parity check measurements (i.e. error syndromes). While there already is a substantial amount of corresponding codes and theoretical considerations behind it, in which there is an entire community working on quantum error correction [85]–[87], the development of corresponding software just started [88], [89].

V. Towards Designing large-scale full-stack quantum computing systems

Previous sections have discussed current practices and trends for designing quantum devices and circuits, and the role of software both as a design tool and its usage within the higher-end layers of the full-stack. It is of utmost importance to reivindicate a full-stack perspective, with an emphasis in the value of co-design techniques, structured design methods and upscaling to full system design.

Abstractions are commonplace in conventional computing stacks in the form of distinct, self-contained layers with well-defined functions that hold particular information shared exclusively with neighboring layers. A desired crucial feature of a computer, namely the independence of software from the underlying hardware is still valid, despite the purely layered approach has been reexamined with the recent introduction of low-power and AI co-processors with resource scarcity yet stringent performance requirements. Quantum computing systems are not yet ready for such comprehensive abstractions, since higher levels in the stack must be exposed to the low-level physical features of current processors in order to maximize their potential, as these processors are very error-prone and have severely limited resources. Aiming to efficiently execute a quantum algorithm maximizing its success rate leads, for example, to compilation strategies that take into account the quantum chip limited connectivity, operation error rates, error variability across the chip, and crosstalk, amongst other factors. This results in a bottom-up flow of information through the stack that contains pertinent hardware parameters. Furthermore, the compiler can further optimize the quantum circuit by utilizing its application-specific information. It is thus more important than ever to exercise a tight co-design between neighboring layers as well as vertical cross-layer design and related full-stack optimization during this early stage of quantum computing [34], [35], [90]. This co-design needs to happen upfront, at the system conception stage. This approach would have two benefits: first, it would maximize the processing capacity of the restricted quantum system; second, it would establish the foundation for future front-ends that will lead to more layer-oriented abstraction and encapsulation. In this co-design scenario, the community for quantum computing is unclear on which metrics and benchmarks to employ to evaluate and compare the performance of quantum computing systems. In an initial effort, IBM suggested measuring three important aspects of quantum computing performance, namely quality (through Quantum Volume), speed (Circuit Layer Operations Per Second, CLOPS), and scale (qubit count) [91]. Additionally, many quantum benchmark sets have been established

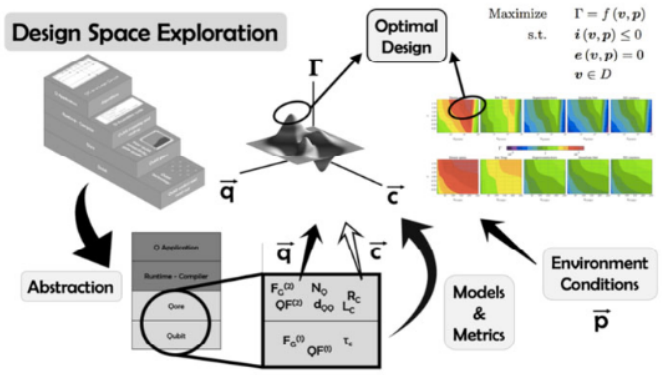


Figure 4: DSE exploration framework for double full-stack modular quantum computing architectures.

[92], [93] for evaluating their overall performance. Furthermore, the absence of design principles from the highest architectural layers down to the physical ones, or even cross-layer co-design, has long been recognized as a requirement by the quantum community. This approach of performing a cross-layer co-design of the full-stack quantum system is particularly key in the architectural bet aiming scalable systems based upon modular architectures, as indicated in the introduction, in which (quantum) communication networks are embedded and enable a multiplicity of quantum computing cores to interact and run larger scale algorithms. Modelling, simulating and optimally architecting and designing such modular systems would benefit from a double-stack formalism, as illustrated in Fig. 4.

Finally, when providing a system's standpoint for designing quantum full-stacks, we do observe the need for a system-wide perspective when addressing larger scale quantum computing systems. When doing so, two main aspects are faced. On the one had, the system becomes more complex, in the sense that not only there will be more subsystems, but the designer will need to address a tighter interplay among and across them. Additionally, such diverse subsystems will be heterogeneous, both in the technologies and hence design techniques incurred, but also the different models and languages to describe them. When aiming system robust scalability, the ultimate aim in design, we do postulate the value of applying structured architecting and design techniques, in the form of formal optimization-centric Design Space Exploration (DSE) methods, as illustrated in Fig. 4. Such system-wide design-oriented optimization framework might be based upon models and simulation tools, or combinations of the former, in the form of multi-domain simulation. The outcome of such DSE framework would be to (a) optimize the different layers of the quantum computing system as well as their vertical integration and optimization, (b) to partition the input design space in different regions of optimality while dimensioning design parameters in each region, as well as (c) to derive scalability trends.

VI. Open Questions

As described in previous sections, having formal design, simulation and verification tools and methodologies is key for developing the next generation of quantum computers, in which the qubit counts will substantially increase and accordingly the complexity of the system. We will therefore summarize some of the open questions and challenges that the quantum computing community is currently facing and for which the expertise of the EDA community will be crucial.

- *The quest from current co-design techniques onto future encapsulation of layers.* Recently, the use of co-design techniques is becoming more prominent so as to extract optimal performance vs resource trade-offs per layer while doing so for tighter vertical integration. Intriguingly, the NISQ era will experiment an impasse, we observe, in migrating from co-design techniques to more pervasive abstractions and functional encapsulations until a full-fledged abstracted full-stack is reached. The current need to exchange design variables and performance metrics vertically across the stack will serve as predecessors of such layer front-ends that will precede full abstractions, thereby facilitating application of more automated design methods. DSE frameworks can also be progressed to ease incorporation of more automation within design end-to-end processes.
- *Driving the development of future large-scale and FT quantum computers.* Although the current developments are mainly steered by the quantum processors as the most precious resource (i.e. following a bottom-up approach), the situation may drastically change in the future when a more holistic optimization may be required or other higher layers in the quantum stack, e.g., the interface electronics or the microarchitecture, may impose more stringent requirements driving the computer architecture and design. Furthermore, the introduction of quantum error correction and fault-tolerant mechanisms will add an extra layer of complexity to the full system. It will have a substantial impact on the computer's organization as new functionalities and support will need to be integrated in the stack.
- *Machine-learning techniques for optimized synthesis and design.* AI-based heuristics are currently intensely being explored in various areas of silicon system-on-chip design [94], [95], from logic design, logic synthesis, physical design down to fabrication. Examples encompass automated synthesis of large digital architectures, automated floorplanning, energy-aware networks-on-chip, improved placement and routing, and subsystem optimization, by virtue of their enhanced performance in optimization tasks handled hitherto through heuristics. We do foresee a preponderance of analogous ML-centric methods for quantum system automated design that constitute hard problems, including optimized synthesis of the connectivity topologies in quantum chips,

floorplanning of large-scale quantum systems and ML-learning enabled mapping techniques for algorithm-architecture co-design.

- *HPC design-oriented simulation and quantum approaches for increasing-scale quantum computing systems.* Within the current dispute between the quantum computer community and the HPC community in the quest to show quantum advantage, we do identify as a challenge and opportunity to leverage HPC computing platform as a design-oriented large-scale simulation framework for optimal design of increasing scale quantum computing systems. An example of such approach would be the use of tensor network formalisms running in a super-computer [96], that can provide lossy approximations of overall fidelity of a given quantum system, to explore scalability trends. Analogously, an opportunity arises in applying quantum algorithms, for instance quadratic unconstrained binary optimization, that can naturally be mapped in quantum annealers to optimize certain aspects of the quantum system, both at the off-line design and architecting phase, or during operation, such as applying QUBO optimization for enhanced algorithm-architecture mapping [97]
- *Quantum computing needs to get more accessible for their end-users.* As covered above, the end-users of quantum computers will be domain experts from various applications areas; with most of them having no background in quantum physics or quantum computing. Considering that we already have a workforce problem in the community, it is hence essential to provide solutions and interfaces that shield the end-users as much as possible from the intricacies of quantum computing. A clear example that introducing such interfaces is possible can be found in classical computing which literally can be used by anyone, even though most of them are no electrical or computer engineers. Some ideas how this can be accomplished are discussed, e.g., in [70]. The challenge remains to realize and establish such solutions.
- *A closer interaction between the design automation and the quantum computing community is needed.* As reviewed above, several design problems in quantum computing have a similar nature and complexity as their counterparts in classical computing. However, because of the interdisciplinary of the topic, both communities did not yet consolidated on proper abstractions, constraints, objective functions, etc. that are precise enough to properly reflect the reality but abstract enough so that they can be processed by design automation methods. For example, mapping a quantum algorithm to an actual constrained platform has huge similarities with classical scheduling, mapping, and routing. In addition, once a simplified qubit model is available, analog/RF circuit designers could develop very helpful qubit electrical interfaces and, by this, enable the verification for full quantum systems. All that, however, only works if both communities work together and develop a proper and common language/terminology that is understood and can be used by both "sides".
- *Education of the next generation of quantum engineers.* As in other emerging technological fields, it is necessary to train a new generation of quantum engineers and scientists to keep the steady and remarkable progress that quantum computing has made in the last years. The main challenge here is to have a critical mass of experts that are able not only to train and educate these new generations but also to articulate comprehensive and multidisciplinary higher education programs. In that educational context, we bet on exposing quantum technologies students to EDA tools, as is currently commonplace in electronics and computer engineering graduate programs.

VII. Conclusions

The multidisciplinary field of quantum computing is steadily progressing through advancements in the different disciplines that required to build up a full-stack system at scale. Although current quantum computers are still limited by the amount of resources and noise, they already integrate all functional elements to efficiently execute quantum algorithms. Such computers are designed up to now using diverse methods and tools that have been specifically developed by quantum industries and research institutions or/and adapted from existing EDA tools used in the semiconductor technology. In this scenario, the paper has discussed current practices and trends for designing quantum devices and control circuits, the role of software both as a design tool and its usage within the higher-end layers of the full-stack and system-level co-design techniques and structured design. The paper culminates with a proposal of several challenges, encompassing: the introduction of more abstractions and functional encapsulation, migrating towards a top-down system-wide full-stack perspective, exploring the use of machine learning, HPC and quantum approaches for more automated and optimized design, increase end-user accessibility, and fostering interaction between the EDA and quantum computing communities, at both research and education levels.

Acknowledgment

N.H. expresses gratitude for the support received from Leo Di Carlo and his group, as well as Intel Corporation and IARPA (U.S. Army Research Office Grant No. W911NF-16-1-0071) for their contributions to the research funds. C.G.A., F.S. and E.A. acknowledge support from EU, grant HORIZON-EIC-2022-PATHFINDEROPEN-01-101099697. E.A acknowledges support from the QCOMM-CAT-Planes Complementarios: Comunicacion Cuántica and Generalitat de Catalunya. C.G.A also acknowledges support from MICIIN and European ERDF under grant PID2021-123627OB-C51. R.W.'s work received funding from the European Research Council (ERC) under the European Union's Horizon 2020

research and innovation program (grant agreement No. 101001318).

References

[1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[2] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, et al. Strong quantum computational advantage using a superconducting quantum processor. *Physical review letters*, 127(18):180501, 2021.

[3] Lars S Madsen, Fabian Laudenbach, Mohsen Falamarzi Askarani, Fabien Rortais, Trevor Vincent, Jacob FF Bulmer, Filippo M Miatto, Leonhard Neuhaus, Lukas G Helt, Matthew J Collins, et al. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, 2022.

[4] Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614(7949):676–681, 2023.

[5] C Ryan-Anderson, NC Brown, MS Allman, B Arkin, G Asa-Attuah, C Baldwin, J Berg, JG Bohnet, S Braxton, N Burdick, et al. Implementing fault-tolerant entangling gates on the five-qubit code and the color code. *arXiv preprint arXiv:2208.01863*, 2022.

[6] Youwei Zhao, Yangsen Ye, He-Liang Huang, Yiming Zhang, Dachao Wu, Huijie Guan, Qingling Zhu, Zuolin Wei, Tan He, Sirui Cao, et al. Realization of an error-correcting surface code with superconducting qubits. *Physical Review Letters*, 129(3):030501, 2022.

[7] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, et al. Realizing repeated quantum error correction in a distance-three surface code. *Nature*, 605(7911):669–674, 2022.

[8] Lukas Postler, Sascha Heuβen, Ivan Pogorelov, Manuel Rispler, Thomas Feldker, Michael Meth, Christian D Marciniak, Roman Stricker, Martin Ringbauer, Rainer Blatt, et al. Demonstration of fault-tolerant universal quantum gate operations. *Nature*, 605(7911):675–680, 2022.

[9] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. Logical quantum processor based on reconfigurable atom arrays. *Nature*, pages 1–3, 2023.

[10] SA Moses, CH Baldwin, MS Allman, R Ancona, L Ascarrunz, C Barnes, J Bartolotta, B Bjork, P Blanchard, M Bohn, et al. A race track trapped-ion quantum processor. *arXiv preprint arXiv:2305.03828*, 2023.

[11] Jonathan Wurtz, Alexei Bylinskii, Boris Braverman, Jesse Amato-Grill, Sergio H Cantu, Florian Huber, Alexander Lukin, Fangli Liu, Phillip Weinberg, John Long, et al. Aquila: Quera's 256-qubit neutral-atom quantum computer. *arXiv preprint arXiv:2306.11727*, 2023.

[12] IBM. The hardware and software for the era of quantum utility is here, 2023.

[13] Andrew Cross. The ibm q experience and qiskit open-source quantum computing software. In *APS March Meeting Abstracts*, volume 2018, pages L58–003, 2018.

[14] Ryan LaRose. Overview and comparison of gate level quantum software platforms. *Quantum*, 3:130, 2019.

[15] Nader Khammassi et al. Openql: A portable quantum programming framework for quantum accelerators. *arXiv preprint arXiv:2005.13283*, 2020.

[16] Seyon Sivarajah et al. t| ket>: A retargetable compiler for nisq devices. *Quantum Science and Technology*, 2020.

[17] Xiang Fu, Leon Riesebos, MA Rol, Jeroen Van Straten, J Van Someren, Nader Khammassi, Imran Ashraf, RFL Vermeulen, V Newsum, KKL Loh, et al. eqasm: An executable quantum instruction set architecture. In *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 224–237. IEEE, 2019.

[18] Lingling Lao, Prakash Murali, Margaret Martonosi, and Dan Browne. Designing calibration and expressivity-efficient instruction sets for quantum computing. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 846–859. IEEE, 2021.

[19] Cupjin Huang, Tenghui Wang, Feng Wu, Dawei Ding, Qi Ye, Linghang Kong, Fang Zhang, Xiaotong Ni, Zhijun Song, Yaoyun Shi, et al. Quantum instruction set design for performance. *Physical Review Letters*, 130(7):070601, 2023.

[20] Xiang Fu, Michiel Adriaan Rol, Cornelis Christiaan Bultink, J Van Someren, Nader Khammassi, Imran Ashraf, RFL Vermeulen, JC De Sterke, WJ Vlothuizen, RN Schouten, et al. An experimental microarchitecture for a superconducting quantum processor. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 813–825, 2017.

[21] Mengyu Zhang, Lei Xie, Zhenxing Zhang, Qiaonian Yu, Guanglei Xi, Hualiang Zhang, Fuming Liu, Yarui Zheng, Yicong Zheng, and Shengyu Zhang. Exploiting different levels of parallelism in the quantum control microarchitecture for superconducting qubits. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 898–911, 2021.

[22] Nader Khammassi, Randy W Morris, Shavindra Premaratne, Florian Luthi, Felix Borjans, Satoshi Suzuki, Robert Flory, Linda Patricia Osuna Ibarra, Lester Lampert, and Anne Y Matsuura. A scalable microarchitecture for efficient instruction-driven signal synthesis and coherent qubit control. *arXiv preprint arXiv:2205.06851*, 2022.

[23] Bishnu Patra, Rosario M. Incandela, Jeroen P. G. van Dijk, Harald A. R. Homulle, Lin Song, Mina Shahmohammadi, Robert Bogdan Staszewski, Andrei Vladimirescu, Masoud Babaie, Fabio Sebastiano, and Edoardo Charbon. Cryo-cmos circuits and systems for quantum computing applications. *IEEE Journal of Solid-State Circuits*, 53(1):309–321, 2018.

[24] Bagas Prabowo, Guoji Zheng, Mohammadreza Mehrpoo, Bishnu Patra, Patrick Harvey-Collard, Jurgen Dijkema, Amir Sammak, Giordano Scappucci, Edoardo Charbon, Fabio Sebastiano, Lieven M. K. Vandersypen, and Masoud Babaie. 13.3 a 6-to-8ghz 0.17mw/qubit cryo-cmos receiver for multiple spin qubit readout in 40nm cmos technology. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 64, pages 212–214, 2021.

[25] Joseph Bardin. Beyond-classical computing using superconducting quantum processors. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 65, pages 422–424, 2022.

[26] George Zettles, Scott Willenborg, Blake R. Johnson, Andrew Wack, and Brian Allison. 26.2 design considerations for superconducting quantum systems. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 65, pages 1–3, 2022.

[27] Kiseo Kang, Donggyu Minn, Seongun Bae, Jaeho Lee, Seokhyeong Kang, Moonjoo Lee, Ho-Jin Song, and Jae-Yoon Sim. A 40-nm cryo-cmos quantum controller ic for superconducting qubit. *IEEE Journal of Solid-State Circuits*, 57(11):3274–3287, 2022.

[28] Yatao Peng, Andrea Ruffino, Jad Benserhir, and Edoardo Charbon. A cryogenic sige bicmos hybrid class b/c mode-switching vco achieving 201dbc/hz figure-of-merit and 4.2ghz frequency tuning range. In *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 65, pages 364–366, 2022.

[29] Gengnanyang Zhang, Haichuan Lin, and Cheng Wang. 34.5 a calibration-free 12.8-16.5ghz cryogenic cmos vco with 202dbc/hz fom for classic-quantum interface. In *2023 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 512–514, 2023.

[30] Yanshu Guo, Qichun Liu, Yaoyu Li, Wenqiang Huang, Tian Tian, Siqi Zhang, Nan Wu, Songyao Tan, Ning Deng, Zhihua Wang, Hanjun Jiang, Tiefu Li, and Yuanjin Zheng. A polar-modulation-based cryogenic transmon qubit state controller in 28 nm bulk cmos for superconducting quantum computing. *IEEE Journal of Solid-State Circuits*, 58(11):3060–3073, 2023.

[31] Jongseok Park et al. A fully integrated cryo-cmos soc for state manipulation, readout, and high-speed gate pulsing of spin qubits. *IEEE Journal of Solid-State Circuits*, 56(11):3289–3306, 2021.

[32] Sudipto Chakraborty et al. A cryo-cmos low-power semi-autonomous transmon qubit state controller in 14-nm finfet technology. *IEEE Journal of Solid-State Circuits*, 57(11):3258–3273, 2022.

[33] Juhwan Yoo, Zijun Chen, Frank Arute, Shirin Montazeri, Marco Szalay, Catherine Erickson, Evan Jeffrey, Reza Fatemi, Marissa Giustina, Markus Ansmann, Erik Lucero, Julian Kelly, and Joseph C. Bardin. Design and characterization of a <4-mw/qubit 28-nm cryo-cmos integrated circuit for full control of a superconducting quantum processor unit cell. *IEEE Journal of Solid-State Circuits*, 58(11):3044–3059, 2023.

[34] Teague Tomesh and Margaret Martonosi. Quantum codesign. *IEEE Micro*, 41(5):33–40, 2021.

[35] Carmen G Almudever and Eduard Alarcon. Structured optimized architecting of full-stack quantum systems in the nisq era. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 762–767. IEEE, 2021.

[36] Gushu Li et al. Software-hardware co-optimization for computational chemistry on superconducting quantum processors. *arXiv preprint arXiv:2105.07127*, 2021.

[37] Sergey Bravyi, Oliver Dial, Jay M Gambetta, Darío Gil, and Zaira Nazario. The future of quantum computing with superconducting qubits. *Journal of Applied Physics*, 132(16), 2022.

[38] James Ang, Gabriella Carini, Yanzhu Chen, Isaac Chuang, Michael Austin DeMarco, Sophia E Economou, Alec Eickbusch, Andrei Faraon, Kai-Mei Fu, Steven M Girvin, et al. Architectures for multinode superconducting quantum computers. *arXiv preprint arXiv:2212.06167*, 2022.

[39] Santiago Rodrigo, Sergi Abadal, Eduard Alarcon, Medina Bandic, Hans Van Someren, and Carmen G Almudéver. On double full-stack communication-enabled architectures for multicore quantum computers. *IEEE micro*, 41(5):48–56, 2021.

[40] Jens Koch, M Yu Terri, Jay Gambetta, Andrew A Houck, David I Schuster, Johannes Majer, Alexandre Blais, Michel H Devoret, Steven M Girvin, and Robert J Schoelkopf. Charge-insensitive qubit design derived from the cooper pair box. *Physical Review A*, 76(4):042319, 2007.

[41] AJ Berkley, H Xu, RC Ramos, MA Gubrud, FW Strauch, PR Johnson, JR Anderson, AJ Dragt, CJ Lobb, and FC Wellstood. Entangled macroscopic quantum states in two superconducting qubits. *Science*, 300(5625):1548–1550, 2003.

[42] Marta Pita-Vidal, Arno Bargerbos, Chung-Kai Yang, David J Van Woerkom, Wolfgang Pfaff, Nadia Haider, Peter Krogstrup, Leo P Kouwenhoven, Gijs De Lange, and Angela Kou. Gate-tunable field-compatible fluxonium. *Physical Review Applied*, 14(6):064038, 2020.

[43] QuTech. [online], https://qutech.nl/.

[44] Inc. (2021) ANSYS. Hfss: High-frequency structure simulator. version 2021 r2. [online], https://www.ansys.com/products/electronics/ansys-hfss.

[45] Computer Simulation Technology. (2022). Cst studio suite. version 2022. [online], https://www.cst.com.

[46] COMSOL Inc. (2022). Comsol multiphysics. version 5.7. [online], https://www.comsol.com.

[47] IBM Quantum. (2022). Qiskit metal. version 0.0.6. [online], https://qiskit.org/metal/.

[48] N. Haider, M. Beekman, P. Kaminski, and L. Dicarlo. Design and analysis of superconducting quantum processors for surface code. *APS March Meeting Abstracts*, 2022.

[49] J. F. Marques, B. M. Varbanov, M. S. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider, W. Vlothuizen, A. Bruno, B. M. Terhal, and L. DiCarlo. Dlogical-qubit operations in an error-detecting surface code. *Nat. Phys*, 18:80–86, 2022.

[50] Nanoacademic. (2022). Qtcad [online], https://nanoacademic.com/solutions/qtcad/.

[51] Jeroen van Dijk, Andrei Vladimirescu, Masoud Babaie, Edoardo Charbon, and Fabio Sebastiano. A co-design methodology for scalable quantum processors and their classical electronic interface. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 573–576, 2018.

[52] J.P.G. van Dijk, E. Charbon, and F. Sebastiano. The electronic interface for quantum processors. *Microprocess. Microsyst.*, 66(C):90–101, apr 2019.

[53] J.P.G. van Dijk, E. Kawakami, R.N. Schouten, M. Veldhorst, L.M.K. Vandersypen, M. Babaie, E. Charbon, and F. Sebastiano. Impact of classical control electronics on qubit fidelity. *Phys. Rev. Appl.*, 12:044054, Oct 2019.

[54] Benjamin Gys, Rohith Acharya, Steven Van Winckel, Kristiaan De Greve, Georges Gielen, and Francky Catthoor. A co-simulation methodology for the design of integrated silicon spin qubits with their control/readout cryo-cmos electronics. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 12(3):685–693, 2022.

[55] Rosario M. Incandela, Lin Song, Harald Homulle, Edoardo Charbon, Andrei Vladimirescu, and Fabio Sebastiano. Characterization and compact modeling of nanometer cmos transistors at deep-cryogenic temperatures. *IEEE Journal of the Electron Devices Society*, 6:996–1006, 2018.

[56] Christian Enz, Arnout Beckers, and Farzan Jazaeri. Cryo-cmos compact modeling. In *2020 IEEE International Electron Devices Meeting (IEDM)*, pages 25.3.1–25.3.4, 2020.

[57] Shivendra Singh Parihar, Victor M. van Santen, Simon Thomann, Girish Pahwa, Yogesh Singh Chauhan, and Hussam Amrouch. Cryogenic cmos for quantum processing: 5-nm finfet-based sram arrays at 10 k. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(8):3089–3102, 2023.

[58] Amol D. Gaidhane, Rakshith Saligram, Wriddhi Chakraborty, Suman Datta, Arijit Raychowdhury, and Yu Cao. Design exploration of 14 nm finfet for energy-efficient cryogenic computing. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 9(2):108–115, 2023.

[59] Victor M van Santen, Marcel Walter, Florian Klemme, Shivendra Singh Parihar, Girish Pahwa, Yogesh S Chauhan, Robert Wille, and Hussam Amrouch. Design automation for cryogenic cmos circuits. In *Design Automation Conference*, pages 1–6, 2023.

[60] Rob A. Damsteegt, Ramon W.J. Overwater, Masoud Babaie, and Fabio Sebastiano. A benchmark of cryo-cmos 40-nm embedded sram/drams for quantum computing. In *ESSCIRC 2023- IEEE 49th European Solid State Circuits Conference (ESSCIRC)*, pages 165–168, 2023.

[61] Ramon W.J. Overwater, Masoud Babaie, and Fabio Sebastiano. Cryogenic-aware forward body biasing in bulk cmos. *IEEE Electron Device Letters*, pages 1–1, 2023.

[62] Qiskit contributors. Qiskit: An Open-source Framework for Quantum Computing.

[63] Cirq: a python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits.

[64] Azure Quantum.

[65] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. t|ket⟩: a retargetable compiler for NISQ devices. 6(1):014003.

[66] Ville Bergholm et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations.

[67] Adi Botea, Akihiro Kishimoto, and Radu Marinescu. On the complexity of quantum circuit compilation. In *Symp. on Combinatorial Search*, pages 138–142, 2018.

[68] Mathias Soeken, Robert Wille, Oliver Keszocze, D Michael Miller, and Rolf Drechsler. Embedding of large boolean functions for reversible logic. 12(4):41, 2016.

[69] Dominik Janzing, Pawel Wocjan, and Thomas Beth. "Non-identity check" is QMA-complete. *Int'l Journal of Quantum Information*, 03(03):463–473, 2005.

[70] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. Towards an automated framework for realizing quantum computing solutions. In *International Symposium on Multiple-Valued Logic, pages=134–140, year=2023, organization=IEEE.*

[71] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. MQT predictor: Automatic device selection with device-specific circuit compilation for quantum computing. *arXiv preprint arXiv:2310.06889*, 2023.

[72] Michael E Beverland, Prakash Murali, Matthias Troyer, Krysta M Svore, Torsten Hoefler, Vadym Kliuchnikov, Guang Hao Low, Mathias Soeken, Aarthi Sundaram, and Alexander Vaschillo. Assessing requirements to scale to practical quantum advantage. *arXiv preprint arXiv:2211.07629*, 2022.

[73] Alwin Zulehner and Robert Wille. Advanced simulation of quantum computations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(5):848–859, 2018.

[74] Alwin Zulehner and Robert Wille. Matrix-vector vs. matrix-matrix multiplication: Potential in dd-based simulation of quantum computations. In *Design, Automation & Test in Europe Conference*, pages 90–95. IEEE, 2019.

[75] Stefan Hillmich, Igor L Markov, and Robert Wille. Just like the real thing: Fast weak simulation of quantum computation. In *Design Automation Conference*, pages 1–6, 2020.

[76] Trevor Vincent, Lee J O'Riordan, Mikhail Andrenkov, Jack Brown, Nathan Killoran, Haoyu Qi, and Ish Dhand. Jet: Fast quantum circuit simulations with parallel task-based tensor-network contraction. *Quantum*, 6:709, 2022.

[77] Thomas Grurl, Jürgen Fuß, and Robert Wille. Noise-aware quantum circuit simulation with decision diagrams. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(3):860–873, 2022.

[78] Alwin Zulehner and Robert Wille. One-pass design of reversible circuits: Combining embedding and synthesis for reversible logic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(5):996–1008, 2017.

[79] Alwin Zulehner et al. An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[80] Gushu Li et al. Tackling the qubit mapping problem for NISQ-era quantum devices. In *International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1001–1014, 2019.

[81] Daniel Schoenberger, Stefan Hillmich, Matthias Brandl, and Robert Wille. Using boolean satisfiability for exact shuttling in trapped-ion quantum computers. *arXiv preprint arXiv:2311.03454*, 2023.

[82] Ludwig Schmid, David F Locher, Manuel Rispler, Sebastian Blatt, Johannes Zeiher, Markus Müller, and Robert Wille. Computational capabilities and compiler development for neutral atom quantum processors: Connecting tool developers and hardware experts. *arXiv preprint arXiv:2309.08656*, 2023.

[83] Lukas Burgholzer and Robert Wille. Advanced equivalence checking for quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(9):1810–1824, 2020.

[84] Lukas Burgholzer, Rudy Raymond, and Robert Wille. Verifying results of the ibm qiskit quantum circuit compilation flow. In *IEEE International Conference on Quantum Computing and Engineering*, pages 356–365. IEEE, 2020.

[85] Pavel Panteleev and Gleb Kalachev. Degenerate quantum ldpc codes with good finite length performance. *Quantum*, 5:585, 2021.

[86] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical ldpc codes. In *ACM SIGACT Symposium on Theory of Computing*, pages 375–388, 2022.

[87] Samuel Stein, Sara Sussman, Teague Tomesh, Charles Guinn, Esin Tureci, Sophia Fuhui Lin, Wei Tang, James Ang, Srivatsan Chakram, Ang Li, et al. Hetarch: Heterogeneous microarchitectures for superconducting quantum systems. In *International Symposium on Microarchitecture*, pages 539–554, 2023.

[88] Lucas Berent, Lukas Burgholzer, and Robert Wille. Software tools for decoding quantum low-density parity-check codes. In *Asia and South Pacific Design Automation Conference*, pages 709–714, 2023.

[89] Tyler LeBlond, Ryan S Bennink, Justin G Lietz, and Christopher M Seck. Tiscc: A surface code compiler and resource estimator for trapped-ion processors. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 1426–1435, 2023.

[90] Yunong Shi et al. Resource-efficient quantum computing by breaking abstractions. *Proceedings of the IEEE*, 108(8):1353–1370, 2020.

[91] Andrew Wack, Hanhee Paik, Ali Javadi-Abhari, Petar Jurcevic, Ismael Faro, Jay M Gambetta, and Blake R Johnson. Quality, speed, and scale: three key attributes to measure the performance of near-term quantum computers. *arXiv preprint arXiv:2110.14108*, 2021.

[92] Robin Blume-Kohout and Kevin C Young. A volumetric framework for quantum computer benchmarks. *Quantum*, 4:362, 2020.

[93] Daniel Mills et al. Application-motivated, holistic benchmarking of a full quantum computing stack. *arXiv preprint arXiv:2006.01273*, 2020.

[94] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

[95] Deepthi Amuru, Andleeb Zahra, Harsha V Vudumula, Pavan K Cherupally, Sushanth R Gurram, Amir Ahmad, and Zia Abbas. Ai/ml algorithms and applications in vlsi design and technology. *Integration*, 2023.

[96] Sergio Sanchez-Ramirez, Javier Conejero, Francesc Lordan, Anna Queralt, Toni Cortes, Rosa M Badia, and Artur Garcia-Saez. Rosnet: A block tensor algebra library for out-of-core quantum computing simulation. In *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*, pages 1–8. IEEE, 2021.

[97] Medina Bandic, Luise Prielinger, Jonas Nüßlein, Anabel Ovide, Santiago Rodrigo, Sergi Abadal, Hans van Someren, Gayane Vardoyan, Eduard Alarcon, Carmen G Almudever, et al. Mapping quantum circuits to modular architectures with qubo. *arXiv preprint arXiv:2305.06687*, 2023.