Structured Optimized Architecting of Full-Stack Quantum Systems in the NISQ era

Carmen G. Almudever* and Eduard Alarcon†

* QuTech, Delft University of Technology
† Technical University of Catalonia, UPC BarcelonaTech

***Abstract*—In the midst of the NISQ era of quantum computers, the challenges are gravitating to encompass both architecting and full-stack engineering aspects, which are inherently algorithm-driven, so that there starts to be a convergence of bottom-up and top down design approaches, what we coin as the Quantum Architecting (QuArch) era. In face of many-fold diverse design proposals, in this paper it is postulated and proposed to apply the so-called Design Space Exploration (DSE) to the full vertical stack of quantum systems as an instrumental methodology to address such design diversity challenge. This structured design means, based upon composing a multidimensional input design space together with compressing the set of output performance metrics into an optimization-oriented overall figure of merit, provides a framework and method for optimization, for performance comparison. It yields as well a way to discriminate among alternative techniques at all layers and across layers, eventually as a structured and comprehensive design-oriented formal framework to address the quantum system design and evaluation complexity. The paper concludes by illustrating instances of this methodology in optimizing and comparing mapping techniques to address the resource-constrained current NISQ quantum chips, and to carry out a quantitative gap analysis of scalability trends aiming many-core distributed quantum architectures.**

I. Introduction

Quantum computing is both an interdisciplinary and multi-disciplinary field of research. It has become an extremely relevant topic, attracting a lot of attention from both academia and industry as it will have a remarkable impact on science and society [1], [2]. Since the 80's when the idea of building a quantum computer and the first physical realizations were proposed, and the subsequent development of the first quantum algorithms in the 90's which promised up to exponential speed-ups compared to their classical counterparts, the field of quantum computing has experienced a very notable progress.

Quantum computing is now a tangible reality. The so-called noisy intermediate-scale quantum (NISQ) processors already exist consisting of up to a few tens of qubits. Some of them are available through the cloud such as the IMB and QuTech quantum chips, in which users can run small quantum algorithms [3], [4]. Furthermore, quantum supremacy has recently been claimed to be demonstrated by Google Research [5]. It is in the recent years that the challenges in this field started gravitating from the more physics-oriented approaches and solutions to the more engineering ones. Realizing a quantum computer demands that quantum technology and device efforts are accompanied by developments in quantum software and architecture and all other research domains that are part of the "full-stack" quantum computing system [1], [6], [7].

Building a quantum computer strongly depends upon but is not only about having quantum devices or processors, analogously to the case of classical computers which are not only based on transistors or integrated circuits. With the advent of functional NISQ devices and the availability of near-term applications such as hybrid optimization quantum algorithms [2], it is timely and critical to address these extra higher-level layers. This research stage is reminiscent of what occurred in conventional computers 40-50 years ago but with an intrinsically different and more sophisticated underlying quantum physical layer. In essence, the era of *quantum architecting* **(QuArch)** is coming into play not only to add extra functional layers, that have been there already for a while, but instead i) to culminate all the previous research and complete the system, ii) to leverage these extra high layers to improve the impairments of the quantum layers underneath in a top-down cross-layer manner, and iii) to allow the exploration of quantum applications and related application-specific architectures.

Due to the still early stage of quantum computing, multiple and different approaches and solutions are being explored and proposed at the different levels of the quantum computing full-stack, encompassing both bottom-up technology and top-down software. The most clear example of such diversity is at the device level. A unique difference compared to classical computing is that, currently, several quantum bit (qubit) implementations are being explored and developed,

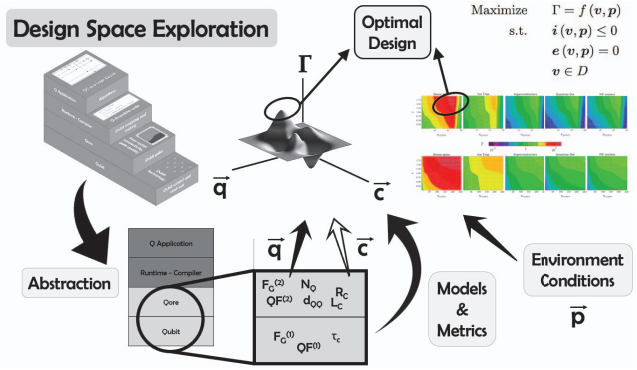

Figure 1: Illustration of the application of DSE to the quantum computing full-stack structured optimized architecting.

978-3-9819263-5-4/DATE21/©2021 EDAA

each of them having specific characteristics (e.g. coherence time, gate fidelity, connectivity, amongst others) and fors and againsts in terms of scalability, footprint, homogeneity, etc. Although some of them seem to be more promising candidates [2], there is no clear case at the outperforming forefront yet. Similarly, at the software level, different ad-hoc solutions are being explored. This diversity at SW level is due to the following concurrent reasons: i) as in classical computing, the higher in the stack, the more choices there are, for which alternative programming languages and compilers can coexist; ii) the field of quantum programming frameworks is an emerging field and therefore it is still organically growing with diverse alternatives; iii) although they all follow a similar flow (a high-level quantum program is compiled into low-level assembly code and ultimately to the corresponding signals to operate and control the qubits) the fact that a real and complete abstraction from the underlying quantum technology is not still possible, makes that different optimization, decomposition, scheduling and mapping strategies are being concurrently explored in the form of physical-aware software.

In this early-stage quantum computing era with multiple and heterogeneous alternative proposals, there is a three-fold need for optimization:

- Firstly, in order to come up with a fair comparison among the proposed approaches, optimization is a requirement.
- Secondly, NISQ devices have quite limited resources (number of qubits) and are highly affected by noise (since qubits and quantum operations are error-prone). Note that quantum error correction and fault-tolerant computation are still not feasible in these devices as they require a large amount of qubits. Therefore, to extract the potential of current and near-term devices given their limited performance and impairments, optimization is a must, not a refinement. More precisely, it is not enough to layer up the different layers, but a cross-layer design and related optimization are required. This particularly calls for a tight co-design among adjacent layers as well as vertical cross-layer design.
- Thirdly, current quantum chips are not at scale and adequate enough for building universal and fault-tolerant quantum computers. In this era, application-specific quantum co-processors will be developed, which also calls for optimization to suit the quantum processor to a given application in the form of its related algorithm.

A compelling means to pursue optimality is the use of structured Design Space Exploration (DSE) methodologies [8]–[10]. Such a methodology is based on the definition of input design variables, performance metrics that are a function of those variables, and an aggregated metric, so-called Figure of Merit (FoM) to be optimised, together with an structured sweep of the multidimensional input design space. The application of this methodology to the quantum computing stack will not only allow optimization but also in turn to: i) observe performance trends in the metric space and hence derive design guidelines, in particular, scalability trends and dimensioning guidelines; ii) a performance comparison across the full input space to identify areas of outperformance or pay-off of a given choice; iii) to partition and categorize the input design space into quantum computer families, potentially per application domains.

DSE exploration methodologies have been extensively applied in different areas such as the design of network-on-chips (NoCs), hardware architectures and circuits, communication full-stacks, and spacecrafts, among others [11]–[13]. However, opposite to other systems where this technique has been used, the purpose of a quantum computer or any kind of computing system, is to efficiently run a given algorithm, and thus comparison stands for benchmark comparison. In other words, in this case there is a extra input to the design problem that is the algorithm itself.

In this paper, we formally and explicitly propose to apply structured DSE methodologies for optimally architecting full-stack quantum systems in the NISQ era (see Figure 1). To this purpose, we will present the current state of quantum computers and revisit DSE methodologies and their outcomes so as to asses their potential use in the quantum computing field. Hence, a discussion on the application of DSE for designing, optimizing and comparing parts of the full-stack will follow with emphasis in the need of defining architectural performance metrics, with two illustrative instances for mapping of quantum circuits and scalability of multi-core quantum architectures. The paper concludes with an extension of DSE to the full-stack and how to deal with complexity and dimensionality.

II. Dealing with diversity on NISQ computers

Quantum computers are now in the NISQ era [14], which refers to error-prone quantum processors consisting of tens to a few hundred of qubits. Although these devices can be used for achieving remarkable milestones such as quantum supremacy [5], fault-tolerance demonstrations [15] and solve some hard problems using hybrid classical-quantum algorithms [2], they are not large in scale and performing enough to accomplish the expected exceptional potential of quantum computing.

NISQ devices are being implemented with a variety of quantum technologies that include trapped-ions, superconductors, quantum dots, solid-state spins and majoranas [1], [2], [7]. A common property in any qubit implementation is its fragility, meaning that qubits easily loose their information because of the interaction with the environment. This time-dependent fragility, measured as a qubit lifetime (coherence time) varies from one technology to another and it currently ranges from tens of μs to tens of seconds. Other important and distinct characteristics of quantum technologies are gate fidelity (accuracy of the operations), operation time (time it takes to perform an operation) and qubits connectivity (how qubits are connected and therefore what interactions are feasible). Currently, superconducting and trapped-ion qubits seem to be the most promising technologies due to their qubit counts (qubits per processor) and higher gate fidelities, but it is still quite unclear what the underlying quantum technology will eventually be.

The maturity of some quantum systems in terms of qubit count and controlability has allowed the development of the so-called *full-stacks* that bridge quantum applications

with quantum processors [1], [16]–[18]. However, at the same time, these quantum devices are not mature enough and too constrained for making a complete abstraction of the higher layers from the underlying quantum hardware [16], [19]. In other words, the software layers have to deal with, for instance, the error rates, limited qubit connectivity, and control constraints of current quantum processor for successfully executing a quantum algorithm [20], [21]. The quantum technology diversity together with this propagation of quantum hardware characteristics from the physical layer to the upper ones, has lead to the development of different compilation and optimization approaches at software level, and even to different quantum computer organizations in which to place each of these higher-level processes, so that how much hardware information to expose to them is not completely clear [7], [19].

This disparity and heterogeneity of alternatives at multiple levels calls for a structured comparison thus requiring prior optimization, for which we propose the formalism of Design Space Exploration methodology as an architecting framework. In turn, this will open the door to explore the potential prospective of each solution, not only considering the current performance but also extrapolating it with the so-called forecast gap analysis. This is particular crucial in the NISQ scenario where not only application-specific is relevant but application-optimality and related decision making of alternatives, architecting and dimensioning, are, thereby calling for vertical cross-layer optimization and tight co-design.

III. Revisiting DSE methodology

Design, the core engineering skill, has a direct and crucial relationship to the performance of a given system. The very design process, understood at two levels, namely (a) the educated selection of structural and functional alternatives at all system levels –from technology, to circuit, to architecture to algorithm,- together with (b) the selection of parameters and variables, is the touchstone of a given system to obtain the highest value of its performance. It is particularly challenging the more stringent the specifications are, and the more complex the system is. The latter, higher complexity, formally understood as a higher structural and functional interplay between parts, subsystems and layers, and higher dimensionality of parameters, yields a more subtle, sensitive and intricate design. Considering a system more of a white canvas than a black-box, design decisions encompass the proposal, selection and dimensioning of every single aspect of the system.

Human-centered design exercises insights, and it is instrumental in the conception phase. However, once a given architecture, circuit or device is considered, when facing the proper dimensioning of variables and parameters, humans tend to trade-off open variables versus performance in one dimension at a time, with the risk of toggling performance or reaching functional yet suboptimal designs. In contrast, structured design techniques exist, namely the *Design Space Exploration methodology*, whereby multidimensional tradeoffs are concurrently considered for a wide parametric range, and in pursuit of optimal performance.

The Design Space Exploration methodology itself is a system-wide methodology based upon composing a multidimensional design space collecting a selection of open design variables and parameters, thereby exploring by sweeping such Design Space while obtaining the output system performance in the form of a collection of quantitative performance metrics as a figure of merit or cost function, hence structuring and formalizing the design funnel and allowing in turn optimality [8]–[10], [12]. DSE is systematically applied based upon the following structured steps, namely:

1) Selecting and stating the Design Space by declaring the essential input variables and parameters that compose the multidimensional input space for which combinations result in system points in the design space. DSE requires thus a wide-range comprehensive and structured sweep of variables and parameters combinations so that related system designs are complete and representative of the design possibilities. This wide range exploration, inherent to DSE, and which usually sweeps the input variables across orders of magnitude, differs from a mere sensitivity analysis as in statistical design, which considers small-range variability around the vicinity of a nominal design point. The nature of the design space can be that of a continuous sweep, or cardinals for discrete variables (mainly scalability of structural units) or qualitative attributes that can be applied or not as a knob. If there are mainly attribute knobs, the Design Space is a multidimensional hypercube.

2) Formalize the output system performance in the form of a set of quantitative metrics that indicate and correlate with output performance, the so-called performance metrics.

3) Quantitatively relate the performance metrics to the input design variables and parameters, as a collection of multidimensional to unidimensional functions, which can come either from analytical or behavioral models, simulation-based characterization or experimental characterization. Existing frameworks for modelling and simulation can be leveraged, but instead of characterization, the approach is flipped to a complete design-oriented analysis, design-oriented modelling and design-oriented simulation.

4) Define a single figure of merit or cost function which compresses the output metric space to a single variable, rendering the design problem into a systematic optimization problem.

Complex system designs, those with harsh specifications, or systems dealing with diverse alternatives that need to be compared, renders optimization a must and not a refinement. Indeed, posing the system design problem following the DSE methodology allows for a design framework in which, beyond strict optimality, design trends, a thorough and fair comparison of alternatives, and even partioning the input space into areas for which different design choices outperform diverse candidates can be derived. DSE addresses the intricate problem of design of a complex system with structure, elegance, and compelling evidence of performance in design outcomes. Despite such outcomes of DSE are so valuable as to encompass optimization, design trends, input design space partitioning and scalability and gap analysis, it could be argued that this formalization of the design problem for a complex system makes the approach aseptic and not

intelligible, losing design insights. Indeed the design problem is posed as a multidimensional to one dimensional mathematical optimization, thereby shifting design into modelling. This design-oriented modelling and functional characterization is intrinsic to DSE. This notwithstanding, it does allow a structured and comprehensive answer to the original design problem, while design from the expert human insight still goes into (a) downselection of the most relevant and essential input design variables, and (b) identification and proposal of performance and resource metrics, and the way to aggregate them in a single cost function.

IV. DSE for design and optimization of full-stack quantum systems

In this quantum architecting (QuArch) stage, automatic design toolflows and more structured design procedures are starting to be used to evaluate different design choices at specific layers of the full-stack [22]–[29].

At quantum device level: In [22], the first attempt to apply systematic structured design based on behavioral simulation for designing and evaluating fault-tolerant ion-trap quantum microarchitectures was proposed. To this purpose, their design space included input variables not only from the quantum hardware (e.g. layout and sizing) but also from the microcode that makes use of the underlying quantum device and abstracts the fault-tolerance property to the program. By using this evaluation methodology, they derived several lessons regarding software error correction on ion-trap architectures that are useful for technology developers, quantum theoreticians working on quantum error correction and quantum architects. More recently in [29], a design toolflow to perform an application-driven architectural study on ion-trap NISQ computers was presented. In this work, different ion-trap designs were explored and evaluated by computing application-level metrics such as execution time, algorithm reliability and operation counts as well as device level metrics such as trap heating rates, when running six real quantum applications. By performing this analysis, concrete design guidelines and insights to the best microarchitectural choices were provided. Whereas the previously mentioned works focus on ion-trap technology and different design points are chosen and evaluated under different applications, [26] takes a slightly different approach and proposes to derive quantum superconducting processors designs based on application characteristics. More precisely, they presented an application-driven design flow for generating efficient superconducting quantum processor architectures. In this case, the design of the quantum chip is guided by the application. Note that they introduce the concept of quantum program (algorithm) profiling in which they extract relevant information affecting the quantum chip design that includes the following design variables: topology, bus selection and frequency allocation. In addition, algorithm performance (application metric) and yield rate (device metric) are used as performance metrics.

At control electronics and compilation level: A methodology to assess the efficacy of a quantum instruction set architecture (ISA) to encode quantum algorithms and explore its design space was introduced in [25]. They considered four different ISA designs that were examined under synthetic as well as real quantum circuits and evaluated based on their encoding efficiency and execution time. It is important to mention that the authors proposed a characterization of the quantum circuits to extract relevant structural parameters that impact the ISA encoding efficiency such as gate density, gate diversity and distribution balance. In [24] an architectural simulation framework called SANQ based on behavioral models to explore, design and optimize the classical control system was presented. This work mostly focused on the exploration of the classical control electronics (interface between quantum instructions and the quantum chip) which was evaluated in terms of execution time for twelve different benchmarks. Additionally, the use of two different compiler optimizations on the qubit mapping problem is also analysed by computing the execution fidelity and time. As we will show later the mapping problem can substantially benefit from the application of structured Design Space Exploration techniques. Finally, in [28], the impact of using different quantum-level decompositions on the mapping overhead of reversible circuits was evaluated. They presented an approach to perform a design exploration to obtain quantum circuits with reduced overhead by exploiting information about the targeted quantum hardware (topological constraints) as well as the reversible circuit (MCT decomposition-aware hardware).

At application level: In [23], DSE and design automation were used for generating different versions of quantum arithmetic circuits. They presented a design flow capable of translating irreversible classical circuits into reversible algorithms and ultimately to alternative quantum circuits optimised based on different metrics (e.g. number of qubits, numer of operations or runtime). This methodology allows to accommodate the cost of arithmetic circuits to quantum device architectures.

Recently, a more formal description and application of structured DSE methodologies to other quantum related challenges such as the mapping of quantum algorithms and the scalability of quantum processors has been presented by the authors in [30] and [31], respectively.

A. Two illustrative instances of DSE for quantum systems

As previously mentioned, given the current state of quantum devices (resource-constrained and error-prone) it is not possible to make a complete abstraction of the higher layers

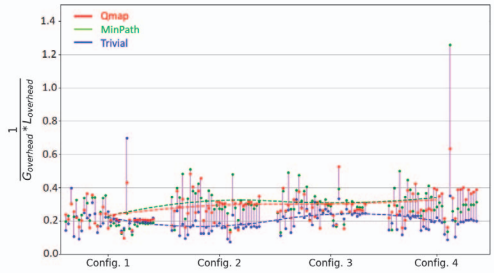

Figure 2: Applying DSE to the Qmap mapper. Each set of three connected vertical points represents a benchmark. Config. 1-4 are the different configurations of the mapper ordered by increasing complexity.

from the underlying quantum device and different hardware-aware compilation solutions have been developed. One specific example is the mapping process, where the quantum circuit representing the quantum algorithm is modified to respect the quantum processor constraints such as qubit connectivity, primitive gate set and shared control electronics [20]. In [30], the formal application of structured DES methodologies to this mapping problem, in which multiple and diverse alternatives are being explored, is proposed. As stated in the paper, this will allow not only to have a more in depth and structured analysis of their performance but also to identify what features are key and worth to implement. More precisely, by using DSE techniques we will be able to: i) determine in which regimes some mapping solutions outperform others; ii) derive optimal mapping strategies for specific quantum algorithms and quantum processors; and iii) perform an scalability analysis. As a first attempt to articulate some structure in the analysis and assessment of a quantum circuit mapper (the Qmap [21]) using DSE, a very simple FoM was defined (see Y axis of Figure 2) and computed for different benchmarks. As shown in Figure 2, the FoM was obtained for three three mapping strategies (design points) that were also swept in complexity. By complexity we mean adding more features to the mapping procedure (e.g. looking ahead, insertion of MOVES and SWAPs) by playing with some internal knobs. As it can be observed, adding more features not always leads to better performance, in this case, higher FoM values.

Another explicit example of using DSE for quantum systems is presented in [31]. It is used as a methodology to address the scalability challenge in quantum processors and explore whether multi-core quantum architectures (vs. current monolithic architectures) may effectively unleash the full quantum computer potential as shown in 3. This work shows how DSE can serve as a means to explore quantum architectures scalability while comparing qubit technologies. However, by exploring the design space in other ways, DSE will allow to determine which is the best performing multi-core design in terms of interconnects technology or intra- and inter-core topology, or even qubit technology. With the aim of determining the "decision threshold" between monolithic

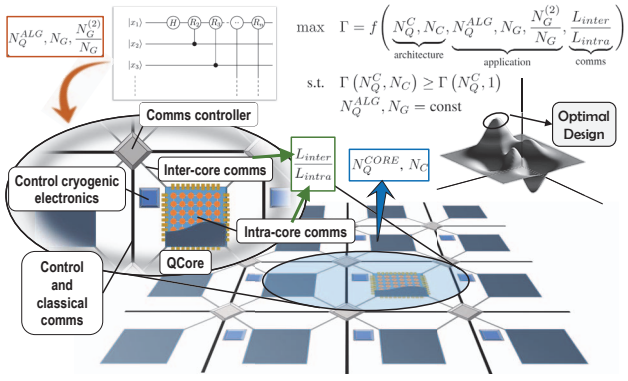

Figure 3: Design Space Exploration for multi-core quantum computing architectures.

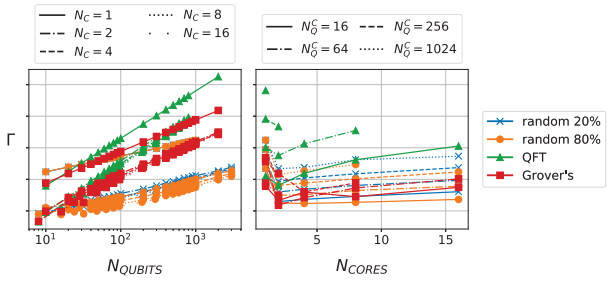

Figure 4: Several benchmarks' scalability on different multi-cores architectures: two different random benchmarks (varying the fraction of two qubit-gates (20%–80%), QFT and Grover's search.

single-core and multi-core quantum architectures, a figure of merit Γ was defined as the quotient of algorithm size over the communications overhead (see Figure 3). Note that it depends on three different aspects: the architecture configuration, the application-specific parameters, and the technology specifications. Figure 4 shows the FoM vs. number of qubits (left) and number or cores (right) for different benchmarks.

B. Extending the DSE to the full-stack: dealing with complexity and dimensionality

The works discussed so far already show the potential of the application of structured DSE methodologies but are focused on a specific layer of the full-stack or challenge. However, more broadly, DSE methods can be extended to all layers of the full-stack. By doing so, optimal full-stack system designs for a specific kind of application, that is, application-driven full-stacks can be derived by performing a DSE-based vertical cross-layer co-design of the full-stack quantum computer, which as previously stated is crucial in the NISQ era. In turn, we will also be able to fairly compare different quantum systems based on the same or a different quantum technology, identify bottlenecks and areas of outperformance or pay-off of a given choice, and provide design guidelines for the implementation of current, near and future quantum computers.

The main challenge here is how to deal with the huge input design space of a multi-layer design in which several variables might be concurrently swept. The curse of dimensionality together with increasing complexity in the form of a network of interplays among system parts would require pruning the design space and coarse-fine multi-step Design Space Exploration efficient techniques [8], [32], [33].

V. Conclusions

This position paper has proposed the use of structured Design Space Exploration methodologies to provide a formal and comprehensive solution to challenges of the QuArch era of NISQ systems, namely the need to cope with diversity at various levels within the system, to compare performance among alternatives, and pursue design optimality, as a driver for next generation architecting and design aspects for application-specific quantum co-processors. We aimed at providing solutions to some of the architecting challenges

identified in the vision report [7]. An annotated survey of state-of-the-art in quantum full-stacks and challenges therein, followed by a review of previous attempts to provide structure to the design of quantum computers, allowed us to postulate and propose DSE as a way to optimize designs, obtain multi-scale design trends in the metric space, to compare alternatives in a large design space, to segment and categorize such design space, to provide parameter dimensioning guidelines, and to assess scalability trends and quantitative forecast gap analysis.

Future research efforts by the community could encompass an in-depth discussion on which architectural performance metrics are more suitable to DSE of quantum systems. The multidisciplinary research community and their expertise in the different layers and subsystems, from device to circuit up to architecture, would also consider DSE-oriented behavioral models of each part of the quantum full-stack. Finally, given the complexity and dimensionality of DSE-oriented models needed in this QuArch era and the lack of all-pervasive models, a fertile future area would be to address machine learning surrogate models to interpolate and extrapolate data from sparse measurements or simulations thus providing design-oriented functions in pursuit of optimality and benchmark comparison to drive the next generation of quantum computing systems.

Acknowledgment

The authors sincerely appreciate scientific discussions with Sergi Abadal (UPC), Hans van Someren (TU Delft), Santiago Rodrigo (UPC) and Medina Bandic (TU Delft).

References

[1] Salonik Resch and Ulya R Karpuzcu. Quantum computing: an overview across the system stack. *arXiv preprint arXiv:1905.07240*, 2019.

[2] Philipp Gerbert and Frank Rueß. The next decade in quantum computing and how to play. *Boston Consulting Group, November*, 2018.

[3] IBM. Quantum experience, 2017.

[4] QuTech. Quantum inspire, 2020.

[5] 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.

[6] Carmen G Almudever, Lingling Lao, Xiang Fu, Nader Khammassi, Imran Ashraf, Dan Iorga, Savvas Varsamopoulos, Christopher Eichler, Andreas Wallraff, Lotte Geck, et al. The engineering challenges in quantum computing. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 836–845. IEEE, 2017.

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

[8] Luigi Nardi, David Koeplinger, and Kunle Olukotun. Practical design space exploration. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 347–358. IEEE, 2019.

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

[10] Alexander Feldman, Johan de Kleer, and Ion Matei. Design space exploration as quantified satisfaction. *arXiv preprint arXiv:1905.02303*, 2019.

[11] Alessandro Bertoni, Sophie I Hallstedt, Siva Krishna Dasari, and Petter Andersson. Integration of value and sustainability assessment in design space exploration by machine learning: an aerospace application. *Design Science*, 6, 2020.

[12] Sergi Abadal, Mario Iannazzo, Mario Nemirovsky, Albert Cabellos-Aparicio, Heekwan Lee, and Eduard Alarcón. On the area and energy scalability of wireless network-on-chip: A model-based benchmarked design space exploration. *IEEE/ACM Transactions on Networking*, 23(5):1501–1513, 2014.

[13] Ye Yu, Yingmin Li, Shuai Che, Niraj K Jha, and Weifeng Zhang. Software-defined design space exploration for an efficient dnn accelerator architecture. *IEEE Transactions on Computers*, 2020.

[14] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[15] Robin Harper and Steven T Flammia. Fault-tolerant logical gates in the ibm quantum experience. *Physical review letters*, 122(8):080504, 2019.

[16] Frederic T Chong, Diana Franklin, and Margaret Martonosi. Programming languages and compiler design for realistic quantum hardware. *Nature*, 549(7671):180–187, 2017.

[17] 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.

[18] Antonio D Córcoles, Abhinav Kandala, Ali Javadi-Abhari, Douglas T McClure, Andrew W Cross, Kristan Temme, Paul D Nation, Matthias Steffen, and Jay M Gambetta. Challenges and opportunities of near-term quantum computing systems. *arXiv preprint arXiv:1910.02894*, 2019.

[19] Yunong Shi, Pranav Gokhale, Prakash Murali, Jonathan M. Baker, Casey Duckering, Yongshan Ding, Natalie C. Brown, Christopher Chamberland, Ali Javadi-Anhari, Andrew W. Cross, David I. Schuster, Kenneth R. Brown, Margaret Martonisi, and Frederic T. Chong. Resource-efficient quantum computing by breaking abstractions. *Proceedings of the IEEE*, 108(8):1353–1370, 2020.

[20] Carmen G Almudever, Lingling Lao, Robert Wille, and Gian G Guerreschi. Realizing quantum algorithms on real quantum computing devices. *arXiv preprint arXiv:2007.01000v1*, 2020.

[21] Lingling Lao, Hans van Someren, Imran Ashraf, and Carmen G Almudever. Timing and resource-aware mapping of quantum circuits to superconducting processors. *arXiv preprint arXiv:1908.04226*, 2019.

[22] Lucas Kreger-Stickles and Mark Oskin. Microcoded architectures for ion-tap quantum computers. *ACM SIGARCH Computer Architecture News*, 36(3):165–176, 2008.

[23] Mathias Soeken, Martin Roetteler, Nathan Wiebe, and Giovanni 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.

[24] Gushu Li, Yufei Ding, and Yuan Xie. Sanq: A simulation framework for architecting noisy intermediate-scale quantum computing system. *arXiv preprint arXiv:1904.11590*, 2019.

[25] Anastasiia Butko, George Michelogiannakis, Samuel Williams, Costin Iancu, David Donofrio, John Shalf, Jonathan Carter, and Irfan Siddiqi. Understanding quantum control processor capabilities and limitations through circuit characterization. *arXiv preprint arXiv:1909.11719*, 2019.

[26] Gushu Li, Yufei Ding, and Yuan Xie. Towards efficient superconducting quantum processor architecture design. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1031–1045, 2020.

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

[28] Philipp Niemann, Alexandre AA de Almeida, Gerhard Dueck, and Rolf 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.

[29] Prakash Murali, Dripto M Debroy, Kenneth R Brown, and Margaret Martonosi. Architecting noisy intermediate-scale trapped ion quantum computers. *arXiv preprint arXiv:2004.04706*, 2020.

[30] Medina Bandic, Hossein Zarein, Eduard Alarcon, and Carmen 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.

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

[32] Eunsuk Kang, Ethan Jackson, and Wolfram Schulte. An approach for effective design space exploration. In *Monterey Workshop*, pages 33–54. Springer, 2010.

[33] Songqing Shan and G Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and multidisciplinary optimization*, 41(2):219–241, 2010.