# CutQC: Using Small Quantum Computers for Large Quantum Circuit Evaluations Extended Abstract

Wei Tang[1], Teague Tomesh[1], Jeffrey Larson[2], Martin Suchara[2], and Margaret Martonosi[1]

[1]Department of Computer Science, Princeton University
[2]Argonne National Laboratory

## 1. Motivation

Quantum computing (QC) has emerged as a promising approach that offers the potential for reduced computational time in several areas [1], including machine learning [2, 3] and computational chemistry [4, 5]. However, these proposed use cases of QC assume the existence of large-scale, fault-tolerant, universal quantum computers.

Instead, today's quantum computers are noisy intermediate-scale quantum (NISQ) devices [6], and major challenges limit their effectiveness. First, because of noise from multiple sources [7, 8, 9, 10], the difficulty of building reliable quantum devices increases dramatically with increasing number of qubits. In fact, larger devices realize significantly worse fidelity than do smaller ones. Second, on a more fundamental level, quantum computers can only execute circuits smaller than the device size. For example, two of the largest current quantum devices have 53 qubits [11, 12] and can execute circuits of that size with only limited fidelity. As a result, both the noise and the intermediate-scale characteristics of NISQ devices present significant obstacles to their practical use.

On the other hand, the only currently viable alternative for QC evaluation—classical simulations of quantum circuits—produces noiseless output but is not tractable. For example, state-of-the-art full-state classical simulations of quantum circuits of merely 45 qubits require tens of hours on thousands of high-performance compute nodes and hundreds of terabytes of memory [13].

To realize quantum advantage in the near future, we need to better utilize near-term NISQ quantum computers. This work develops and demonstrates a comprehensive methodology for cutting large quantum circuits. The resulting subcircuits are mapped onto small quantum computers to expand the reach of small quantum computers with postprocessing techniques that augment small QC platforms with classical computing resources.

## 2. Limitations of the State of the Art

Several promising qubit technologies exist. For example, the largest superconducting quantum computers by IBM [12] and Google [11] have 53 qubits. However, experiments show that these devices can reliably execute circuits with only a few qubits [14, 15]. Other technologies currently have much smaller sizes [16, 17].

Many quantum compilation techniques for NISQ devices have been developed. The recently developed ones include use of real-time device calibration data to improve circuit fidelity by optimally mapping logical qubits to physical ones [18, 19], efficiently scheduling operations to reduce quantum gate counts [20], and repeating circuit executions to mitigate error [21, 22, 23]. These techniques focus on improving a purely quantum computing approach, however, and are intrinsically limited by the size and reliability of NISQ devices. Specifically, these techniques do not allow executions of circuits requiring more qubits than are on the device, and their reliability improvement is limited.

Classical quantum circuit simulations beyond approximately 30 qubits typically use supercomputers, often requiring hundreds to thousands of compute nodes [24, 25, 13], millions of core-hours [26], and a prohibitive amount of memory [27]. In addition, many simulate only a small subset of output states with low fidelity for large quantum circuits, a process called partial state simulation [28, 25, 29]. In general, most of these approaches do not scale. Specifically, even partial state classical simulation beyond 65 qubits is currently difficult [28].

Work done in theoretical physics has considered trading classical and quantum computational resources. These approaches use simple partitioning of qubits [30] or involve exponential postprocessing [31]. Several works manually separate small toy circuits with convenient structures as proof-of-concept numerical demonstrations [32, 33]. However, these theoretical propositions are inflexible, suffer from exponentially high postprocessing costs, and target only a narrow set of quantum circuits. In addition, these works merely prove the mathematical validity but otherwise lack the necessary components for practical implementations. Our work addresses all of these shortcomings, applying efficient partitioning and highly parallelizable postprocessing techniques to realize a practical circuit cutting implementation. This allows for useful trade-offs between classical and quantum compute resources.

## 3. Key Insights

The core insight from this work is that **mixed-integer programming (MIP) can efficiently find cut locations to partition general quantum circuits and distribute the workload between quantum and classical platforms**. The cut locations have a direct and significant impact on the amount of postprocessing required. Therefore, efficient methods for au-

tomatically cutting and distributing the quantum circuit execution workload in a manner that minimizes such postprocessing costs is crucial to its practical application.

The second core insight from this work is that the introduction of the **Dynamic Definition (DD) algorithm enables efficient location of *solution* quantum states and reconstruction of the probability output landscape for large circuits**.

We use both insights to leverage quantum and classical computing resources in a hybrid manner. Leveraging the two platforms together allows us to execute circuits that are much larger than the individual quantum and classical limits. These circuits are also executed much faster than current simulation alternatives and with more reliable outputs than provided by NISQ devices. Our approach, called CutQC, effectively uses small quantum computers as coprocessors in quantum circuit evaluation. CutQC allows sharing the quantum circuit execution workload between the quantum and classical computing platforms more flexibly, with exponentially lower overhead, and can target general quantum circuits.

## 4. Main Artifact

Our main artifact is the first end-to-end hybrid approach that (i) automatically locates efficient positions to cut a large quantum circuit into smaller subcircuits that are (ii) each independently executed using quantum devices with fewer qubits. Via scalable postprocessing techniques, the output of the original circuit can then be reconstructed or sampled efficiently from the subcircuit outputs.

Our QC backend is built on top of IBM's Qiskit [34] package and uses IBM's quantum computers, but we emphasize that this hybrid approach works with any gate-based quantum computing platform. The backend for the automatic cut searcher is implemented in the Gurobi solver [35]. The postprocessing techniques are built as a parallel *C* implementation that utilizes the kernel functions in the Basic Linear Algebra Subprograms package in the Intel Math Kernel Library [36] to optimize the performance on CPUs.

We evaluated our artifact by experimental studies of its runtime and using real-system runs on IBM's quantum devices to demonstrate its fidelity advantage.

## 5. Key Results and Contributions

To evaluate the performance of CutQC, we benchmarked six different quantum circuits that represent a general set of circuits for gate-based QC platforms and promising near-term applications. Figure 6 in the paper shows that CutQC offers an average of 60X to 8600X runtime speedup over classical simulation alternatives for different benchmarks. In addition, Figure 9 demonstrates **executing quantum circuits of up to 100 qubits** on existing NISQ devices with our approach. This is significantly beyond the current reach of either quantum or classical methods alone. Moreover, Figure 10 shows that IBM's quantum computers using CutQC achieve significant $\chi^2$ reduction over state-of-the-art large NISQ devices for various benchmarks.

Specifically, our contributions are as follows:

1. **Expanding the size** of quantum circuits that can be run on NISQ devices and classical simulation by combining the two. Our method allows executions of quantum circuits more than twice the size of the available quantum device backend and significantly beyond the classical simulation limit.
2. **Improving the fidelity** of quantum circuit executions on NISQ devices. We show an average of 21% to 47% improvement to $\chi^2$ loss for different benchmarks by using CutQC with small quantum devices over direct executions on large quantum devices.
3. **Achieving significant speedup** of overall quantum circuit execution over purely classical simulations by orders of magnitude. We use quantum devices as coprocessors to achieve an average of 60X to 8600X runtime speedup over classical simulations for different benchmarks.

## 6. Why ASPLOS

As an emerging computational domain, QC requires hardware-software codesign in order to move toward practical and scalable approaches. This paper identifies shortcomings in NISQ hardware—namely, its constrained and error-prone resources—and develops hardware-software solutions for moving past those shortcomings.

By optimally cutting large quantum circuits, our toolchain identifies opportunities for combining quantum and classical approaches. CutQC executes subcircuits on quantum computing platforms with postprocessing on classical computing platforms. The combination of quantum and classical strategies along with hardware and software strategies are within the scope of the ASPLOS audience.

Likewise, our use of existing compiler approaches such as those in Qiskit along with MIP optimization techniques is within the scope of ASPLOS.

By publishing in ASPLOS, this work will invite future work from the ASPLOS research community that harnesses broad programming languages and architecture expertise to advance this important research domain toward practical and scalable solutions as QC matures.

## 7. Citation for Most Influential Paper Award

By demonstrating how to leverage both quantum and classical computing platforms together to execute quantum algorithms beyond the reach of either one alone, this work pioneered pathways for scalable quantum computing. Even as NISQ machines scaled to larger sizes and as fault-tolerant QC emerged, CutQC's techniques for automatically cutting and efficiently reconstructing quantum circuit executions—and subsequent works building on it—offered the overall, practical strategy for hybrid quantum/classical advantage in QC applications.

# References


[1] Ashley Montanaro. Quantum algorithms: An overview. *npj Quantum Information*, 2:15023, 2016.

[2] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[3] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv:1307.0411*, 2013.

[4] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White. Towards quantum chemistry on a quantum computer. *Nature Chemistry*, 2(2):106, 2010.

[5] Daniel S Abrams and Seth Lloyd. Simulation of many-body Fermi systems on a universal quantum computer. *Physical Review Letters*, 79(13):2586, 1997.

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

[7] PV Klimov, Julian Kelly, Z Chen, Matthew Neeley, Anthony Megrant, Brian Burkett, Rami Barends, Kunal Arya, Ben Chiaro, Yu Chen, et al. Fluctuations of energy-relaxation times in superconducting qubits. *Physical Review Letters*, 121(9):090502, 2018.

[8] 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.

[9] Prakash Murali, David C McKay, Margaret Martonosi, and Ali Javadi-Abhari. Software mitigation of crosstalk on noisy intermediate-scale quantum computers. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1001–1016, 2020.

[10] Sarah Sheldon, Easwar Magesan, Jerry M Chow, and Jay M Gambetta. Procedure for systematically tuning up cross-talk in the cross-resonance gate. *Physical Review A*, 93(6):060302, 2016.

[11] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[12] IBM. On "quantum supremacy", 2019. https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/.

[13] Xin-Chuan Wu, Sheng Di, Emma Maitreyee Dasgupta, Franck Cappello, Hal Finkel, Yuri Alexeev, and Frederic T Chong. Full-state quantum circuit simulation by using data compression. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–24, 2019.

[14] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B Buckley, David A Buell, et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *arXiv preprint arXiv:2004.04197*, 2020.

[15] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B Buckley, David A Buell, et al. Hartree-Fock on a superconducting qubit quantum computer. *arXiv preprint arXiv:2004.04174*, 2020.

[16] IonQ. Introducing the world's most powerful quantum computer, 2020. https://ionq.com/posts/october-01-2020-most-powerful-quantum-computer.

[17] Honeywell. Honeywell system model h0, 2020. https://www.honeywell.com/us/en/company/quantum/quantum-computer.

[18] Prakash Murali, Jonathan M Baker, Ali Javadi-Abhari, Frederic T Chong, and Margaret Martonosi. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1015–1029, 2019.

[19] Swamit S Tannu and Moinuddin K Qureshi. Not all qubits are created equal: a case for variability-aware policies for NISQ-era quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 987–999, 2019.

[20] Jeff Heckey, Shruti Patil, Ali JavadiAbhari, Adam Holmes, Daniel Kudrow, Kenneth R Brown, Diana Franklin, Frederic T Chong, and Margaret Martonosi. Compiler management of communication and parallelism for quantum computation. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 445–456, 2015.

[21] Abhinav Kandala, Kristan Temme, Antonio D Córcoles, Antonio Mezzacapo, Jerry M Chow, and Jay M Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491, 2019.

[22] Swamit S Tannu and Moinuddin Qureshi. Ensemble of diverse mappings: Improving reliability of quantum computers by orchestrating dissimilar mistakes. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 253–265, 2019.

[23] Swamit S Tannu and Moinuddin K Qureshi. Mitigating measurement errors in quantum computers by exploiting state-dependent bias. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 279–290, 2019.

[24] Thomas Häner and Damian S Steiger. 5 petabyte simulation of a 45-qubit quantum circuit. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–10, 2017.

[25] Benjamin Villalonga, Dmitry Lyakh, Sergio Boixo, Hartmut Neven, Travis S Humble, Rupak Biswas, Eleanor G Rieffel, Alan Ho, and Salvatore Mandrà. Establishing the quantum supremacy frontier with a 281 pflop/s simulation. *Quantum Science and Technology*, 5(3):034003, 2020.

[26] Benjamin Villalonga, Sergio Boixo, Bron Nelson, Christopher Henze, Eleanor Rieffel, Rupak Biswas, and Salvatore Mandrà. A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. *npj Quantum Information*, 5:1–16, 2019.

[27] Edwin Pednault, John A Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, Erik W Draeger, Eric T Holland, and Robert Wisnieff. Breaking the 49-qubit barrier in the simulation of quantum circuits. *arXiv preprint arXiv:1710.05867*, 2017.

[28] Zhao-Yun Chen, Qi Zhou, Cheng Xue, Xia Yang, Guang-Can Guo, and Guo-Ping Guo. 64-qubit quantum circuit simulation. *Science Bulletin*, 63(15):964–971, 2018.

[29] Igor L Markov, Aneeqa Fatima, Sergei V Isakov, and Sergio Boixo. Quantum supremacy is both closer and farther than it appears. *arXiv preprint arXiv:1807.10749*, 2018.

[30] Sergey Bravyi, Graeme Smith, and John A Smolin. Trading classical and quantum computational resources. *Physical Review X*, 6(2):021043, 2016.

[31] Tianyi Peng, Aram Harrow, Maris Ozols, and Xiaodi Wu. Simulating large quantum circuits on a small quantum computer. *arXiv:1904.00102*, 2019.

[32] Xiao Yuan, Jinzhao Sun, Junyu Liu, Qi Zhao, and You Zhou. Quantum simulation with hybrid tensor networks. *arXiv preprint arXiv:2007.00958*, 2020.

[33] Fergus Barratt, James Dborin, Matthias Bal, Vid Stojevic, Frank Pollmann, and Andrew G Green. Parallel quantum simulation of large systems on small quantum computers. *arXiv preprint arXiv:2003.12087*, 2020.

[34] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago,

Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. Qiskit: An open-source framework for quantum computing, 2019.

[35] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2020.

[36] Endong Wang, Qing Zhang, Bo Shen, Guangyong Zhang, Xiaowei Lu, Qing Wu, and Yajuan Wang. Intel math kernel library. In *High-Performance Computing on the Intel® Xeon Phi™*, pages 167–188. Springer, 2014.