

Quantum architecture

Yipeng Huang

Rutgers University

November 30, 2022

Where we are in the semester

Full stack quantum computer engineering

1. Fundamentals: superposition and entanglement
2. Canonical algorithms: Shor's factoring algorithm
3. NISQ Algorithms: QAOA & VQE
4. Google Cirq, IBM Qiskit
5. Programming languages, representations
6. Extracting success: quantum computer architecture
7. Prototypes: quantum computer microarchitecture

Table of contents

Challenges of quantum computer architecture

Scheduling

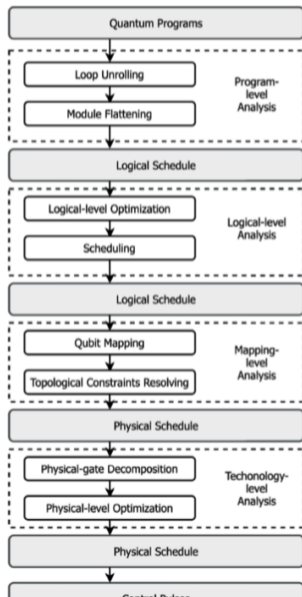
Qubit mapping

Topological constraints resolving

Physical-gate decomposition

Physical-level optimization

From quantum programs to quantum computer operation



Quick summary of the steps so far

- ▶ Loop unrolling
- ▶ Module flattening
- ▶ Logical-level optimization

Table of contents

Challenges of quantum computer architecture

Scheduling

Qubit mapping

Topological constraints resolving

Physical-gate decomposition

Physical-level optimization

Scheduling for maximum parallelism

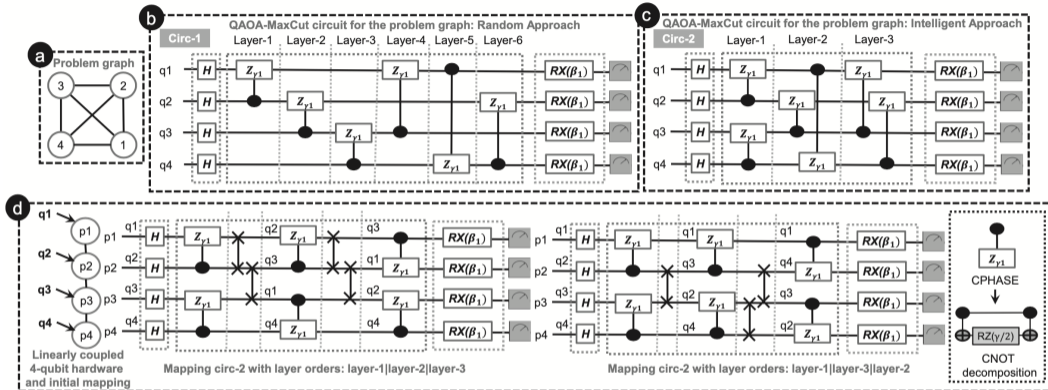


Fig. 1. (a) A 4-node 3-Regular graph, (b) a randomly constructed QAOA-MaxCut instance (circ-1) of the 4-node graph with $p = 1$, (c) an optimized circuit (circ-2) for the problem with reduced number of layers, (d) SWAP addition during circuit compilation for a target hardware with different layer orders.

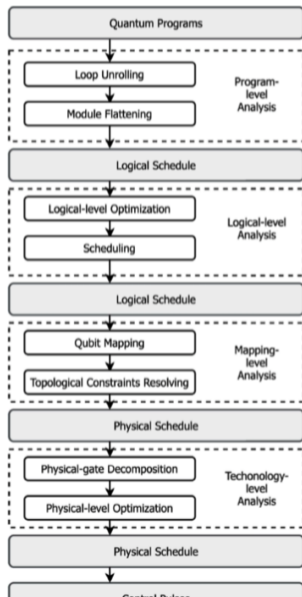
Figure: Credit: [Alam et al., 2020]

Some types of gates commute, so we can move earlier or later.

Parallelism constraints

1. Amount of parallelism available in the instruction stream
2. Safe parallelism despite crosstalk due to spatiotemporal and spectral overlap

From quantum programs to quantum computer operation



Next steps

- ▶ Qubit mapping
- ▶ Topological constraints resolving

Table of contents

Challenges of quantum computer architecture

Scheduling

Qubit mapping

Topological constraints resolving

Physical-gate decomposition

Physical-level optimization

Qubit mapping

- ▶ Ion trap qubits: fully connected topology
- ▶ Superconducting qubits: arbitrary qubits cannot directly interact; needs chain of swap gates

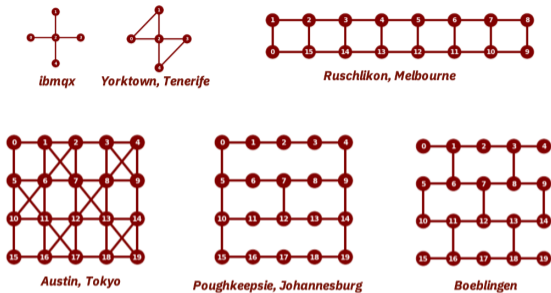


Fig. 1. Examples of several IBM cloud accessible devices. The top left 5-qubit device was the first one made available via the IBM Quantum Experience [40]. The one to the right of it was made available after including additional entangling gates between two pairs of qubits. A 16-qubit device was made available approximately a year after the first device. The devices in the bottom row show three variations of 20-qubit devices available to members of the IBM Q Network [41].

Table of contents

Challenges of quantum computer architecture

Scheduling

Qubit mapping

Topological constraints resolving

Physical-gate decomposition

Physical-level optimization

Topological constraints resolving

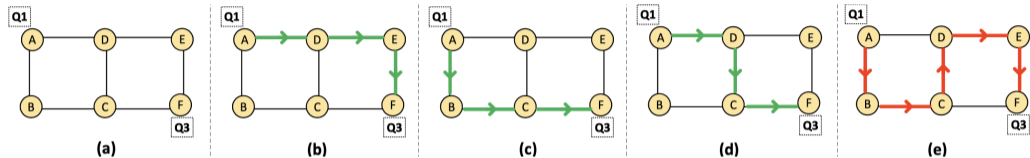


Figure 3. (a) Layout of a 6-qubit quantum computer, (b)-(e) are possible routes from A to F. Note that options (b)(c)(d) have an identical number of swaps and (e) incur higher swaps. An intelligent policy would choose one from (b)(c)(d).

Figure: Credit: [Tannu and Qureshi, 2019]

Superconducting qubits: arbitrary qubits cannot directly interact; needs chain of swap gates

Topological constraints resolving

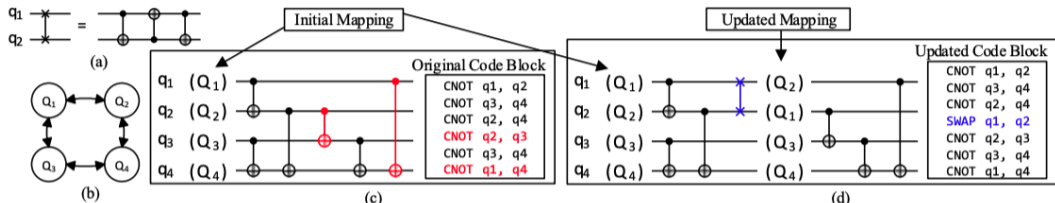


Figure 3. (a) SWAP Gate Decomposition, (b) Physical Qubit Coupling Graph Example, (c) Original Quantum Circuit, (d) Updated Hardware-Compliant Quantum Circuit

Figure: Credit: [Li et al., 2019]

Superconducting qubits: arbitrary qubits cannot directly interact; needs chain of swap gates

Topological constraints resolving

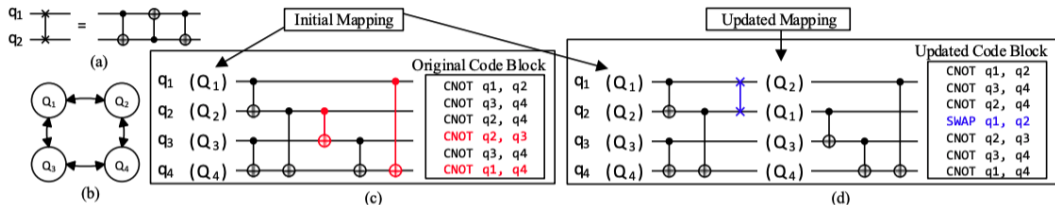
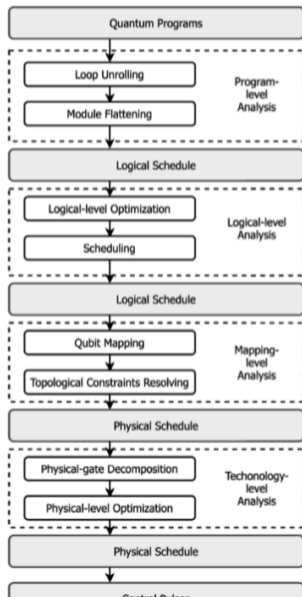


Figure 3. (a) SWAP Gate Decomposition, (b) Physical Qubit Coupling Graph Example, (c) Original Quantum Circuit, (d) Updated Hardware-Compliant Quantum Circuit

Figure: Credit: [Li et al., 2019]

Superconducting qubits: arbitrary qubits cannot directly interact; needs chain of swap gates

From quantum programs to quantum computer operation



Next steps

- ▶ Physical-gate decomposition
- ▶ Physical-level optimization

Table of contents

Challenges of quantum computer architecture

Scheduling

Qubit mapping

Topological constraints resolving

Physical-gate decomposition

Physical-level optimization

Physical-gate decomposition

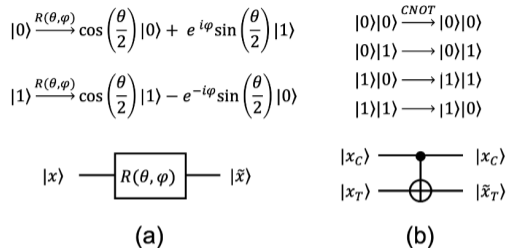


FIG. 1. The rotation and controlled-NOT (CNOT) gates are an example of a universal quantum gate family when available on all qubits, with explicit evolution (above) and quantum circuit block schematics (below). (a) The single-qubit rotation gate $R(\theta, \phi)$, with two continuous parameters θ and ϕ , evolves input qubit state $|x\rangle$ to output state $|\tilde{x}\rangle$. (b) The CNOT (or reversible XOR) gate on two qubits evolves two (control and target) input qubit states $|x_C\rangle$ and $|x_T\rangle$ to output states $|\tilde{x}_C = x_C\rangle$ and $|\tilde{x}_T = x_C \oplus x_T\rangle$, where \oplus is addition modulo 2, or equivalently the XOR operation.

Figure: Credit: [Alexeev et al., 2020]

- ▶ Clifford + T ISA is sensible for an error-corrected machine
- ▶ But for NISQ machine, best two-qubit gate is dependent on native gate set

Physical-gate decomposition

BOX 2. CHOOSING A CNOT GATE DECOMPOSITION

A logical controlled-NOT, or CNOT, gate is an entangling operation that flips the target qubit between 1 and 0 if the control qubit is in the 1 state. It must be decomposed into a sequence of native

quantum gates for the qubit technology to perform the gate operation on the specific qubit system. Two possible decompositions are shown, where RX, RY, and RZ denote rotations around the x-, y- and z-axes, respectively. A system per-

formance simulation could provide metrics to help choose which CNOT to incorporate into the specific design.

Depending on the fidelity of the single- and two-qubit gates in the circuits and on their speed, researchers may want

to choose only one to implement the logical CNOT in the system. Depending on the performance of the qubits available at a particular point in the execution of the algorithm, it is also possible to choose a different logical CNOT sequence.

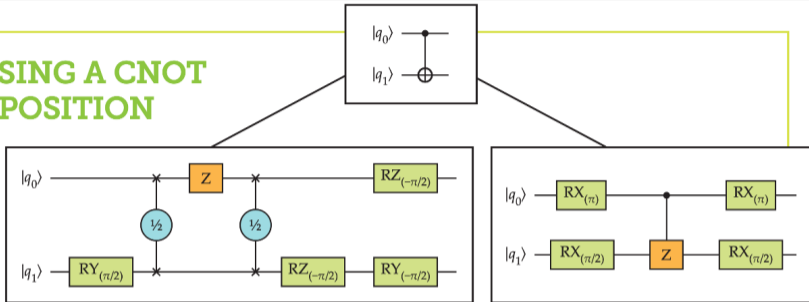


Figure: Credit: [Matsuura et al., 2019]

Physical-gate decomposition

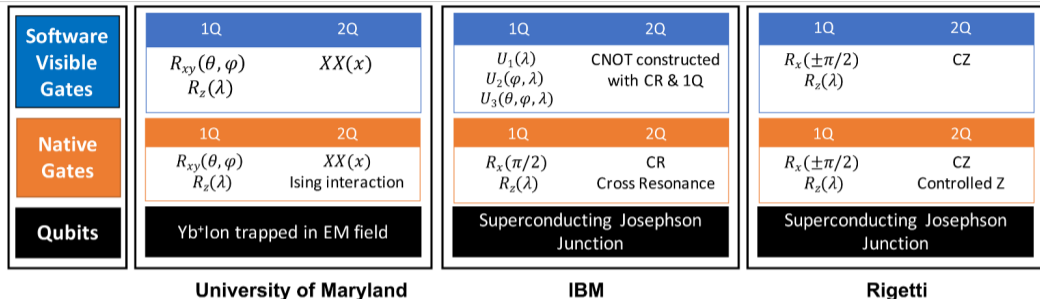


Figure 1. Hardware qubit technology, native gate set, and software-visible gate set in the systems used in our study. Each qubit technology lends itself to a set of native gates. For programming, vendors expose these gates in a software-visible interface or construct composite gates with multiple native gates.

Figure: Credit: [Murali et al., 2019]

Two qubit gates remain dominant sources of errors.

Table of contents

Challenges of quantum computer architecture

Scheduling

Qubit mapping








Topological constraints resolving

Physical-gate decomposition

Physical-level optimization

Primary sources

- ▶ [Ding and Chong, 2020, Chapters 4,6,7]
- ▶ [Córcoles et al., 2020, Section III.B]
- ▶ [National Academies of Sciences, Engineering, and Medicine, 2019, Chapter 6.5]
- ▶ [Martonosi and Roetteler, 2019, Chapter 6]

-  Alam, M., Ash-Saki, A., and Ghosh, S. (2020).
Circuit compilation methodologies for quantum approximate optimization algorithm.
2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO).
-  Alexeev, Y., Bacon, D., Brown, K. R., Calderbank, R., Carr, L. D., Chong, F. T., DeMarco, B., Englund, D., Farhi, E., Fefferman, B., Gorshkov, A. V., Houck, A., Kim, J., Kimmel, S., Lange, M., Lloyd, S., Lukin, M. D., Maslov, D., Maunz, P., Monroe, C., Preskill, J., Roetteler, M., Savage, M., and Thompson, J. (2020).
Quantum computer systems for scientific discovery.
-  Córcoles, A. D., Kandala, A., Javadi-Abhari, A., McClure, D. T., Cross, A. W., Temme, K., Nation, P. D., Steffen, M., and Gambetta, J. M. (2020).
Challenges and opportunities of near-term quantum computing systems.
Proceedings of the IEEE, 108(8):1338–1352.
-  Ding, Y. and Chong, F. T. (2020).
Quantum computer systems: Research for noisy intermediate-scale quantum computers.
Synthesis Lectures on Computer Architecture, 15(2):1–227.
-  Li, G., Ding, Y., and Xie, Y. (2019).
Tackling the qubit mapping problem for nisq-era quantum devices.
In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 1001–1014.
-  Martonosi, M. and Roetteler, M. (2019).
Next steps in quantum computing: Computer science’s role.
arXiv preprint arXiv:1903.10541.
-  Matsuura, A., Johri, S., and Hogaboam, J. (2019).

