Layout Synthesis for Near-Term Quantum Computing

Daniel Bochen Tan
PhD Candidate, CS, UCLA

Outline

- Background: quantum bits and gates, the layout synthesis problem in quantum computing (LSQC)
- QUEKO (Quantum Mapping Examples with Known Optimal) [TC21b]: benchmarks for measuring optimality of LSQC solvers
- OLSQ (Optimal Layout Synthesis for Quantum Computing) [TC20, TC21a]: formulation and implementation for optimal LSQC
- OLSQ-RAA (OLSQ for reconfigurable atom arrays) [TC22]: extending LSQC formulation to a programmable architecture

Background

QUANTUM BITS, GATES, AND LAYOUT SYNTHESIS

Background: Quantum Bits, i.e., Qubits

- A (classical) bit is either 0 or 1.
- A quantum bit, i.e., qubit, is a vector in complex-valued linear space $\mathrm{span}(\{|0\rangle, |1\rangle\})$ where $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$ form a basis.
- A linear space contains also the linear combination of basis states, i.e., $\alpha|0\rangle + \beta|1\rangle = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$

- n bits form a bit-string of length n.
- n-qubit basis states are exactly like the bit-strings: $|00\ldots0\rangle$, …, $|11\ldots1\rangle$.
- n-qubit state is a vector in $\mathrm{span}(\{|00\ldots0\rangle,\ \ldots,\ |11\ldots1\rangle\})$.

- A quantum algorithm produces a state of interest $|\psi\rangle$ from $|00\ldots0\rangle$.
- Measuring $|\psi\rangle$ yields a basis state $|x\rangle$ with probability $|C_x|^2$.

UCLA Samueli School of Engineering

Background: Quantum Gates

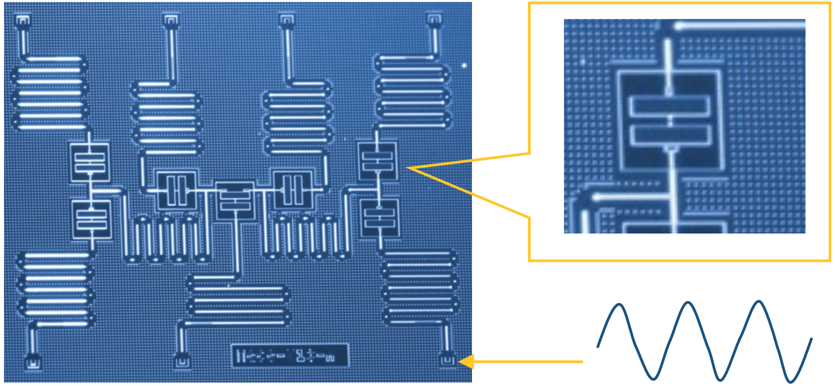
SINGLE-QUBIT GATES

- 'Bit-flip' gate X: $\begin{bmatrix}\alpha\\\beta\end{bmatrix} \mapsto \begin{bmatrix}\beta\\\alpha\end{bmatrix}$, i.e., $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
- 'Phase-flip' gate $Z = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
- Phase-shift gate $R_\phi = \begin{bmatrix}1 & 0\\0 & e^{i\phi}\end{bmatrix}$
  - $Z \equiv R_\pi$, $P \equiv R_{\pi/2}$, $T \equiv R_{\pi/4}$
- Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$

(ENTANGLING) MULTI-QUBIT GATES

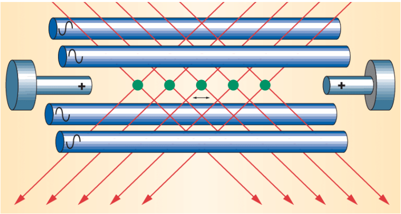
- Controlled-not gate $CNOT$: $\begin{bmatrix}\alpha\\\beta\\\gamma\\\delta\end{bmatrix} \mapsto \begin{bmatrix}\alpha\\\beta\\\delta\\\gamma\end{bmatrix}$
  - $CNOT$ or $CX = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$

Background: Quantum Processers

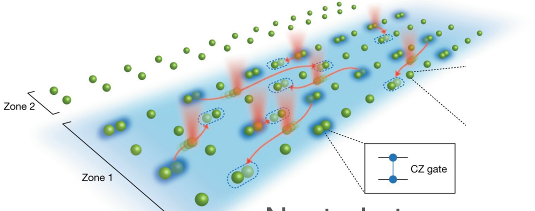


A superconducting QPU by IBM

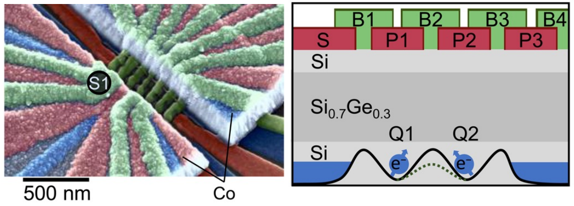
- Quantum registers are physical entities much like classical registers.
- Define two states of a quantum register to be $|0\rangle$ and $|1\rangle$.
- Quantum gates are implemented by signals to the components on QPUs.
- Entangling two-qubit gates requires ‘coupling’ between the two qubits.



Trapped ions



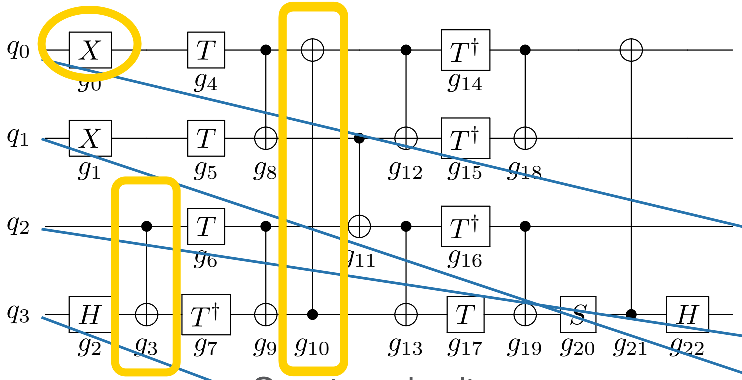
Neutral atom arrays



Silicon spin qubits

Photo credits: Upper left from IBM Newsroom, bottom left from DOI: 10.1109/2.816267, bottom middle from DOI: 10.1038/s41586-022-04592-6, bottom right: DOI: 10.1126/sciadv.abn5130

Background: Layout Synthesis (LSQC)



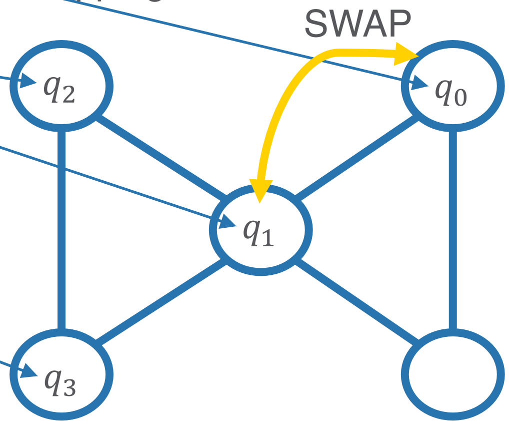
Quantum circuit

```
# Input quantum program
x q[0];
x q[1];
h q[3];
cx q[2], q[3];
t q[0];
…
```

CX on a pair of adjacent qubits, OK.

CX on a pair of non-adjacent qubits!

Insert additional SWAP gate to change the mapping

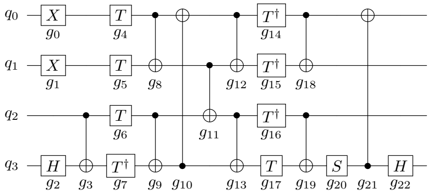


Coupling Graph: vertices are quantum registers, edges means connection for entangling two-qubit gates like CX

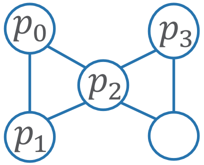
UCLA Samueli School of Engineering

Background: Definition of LSQC

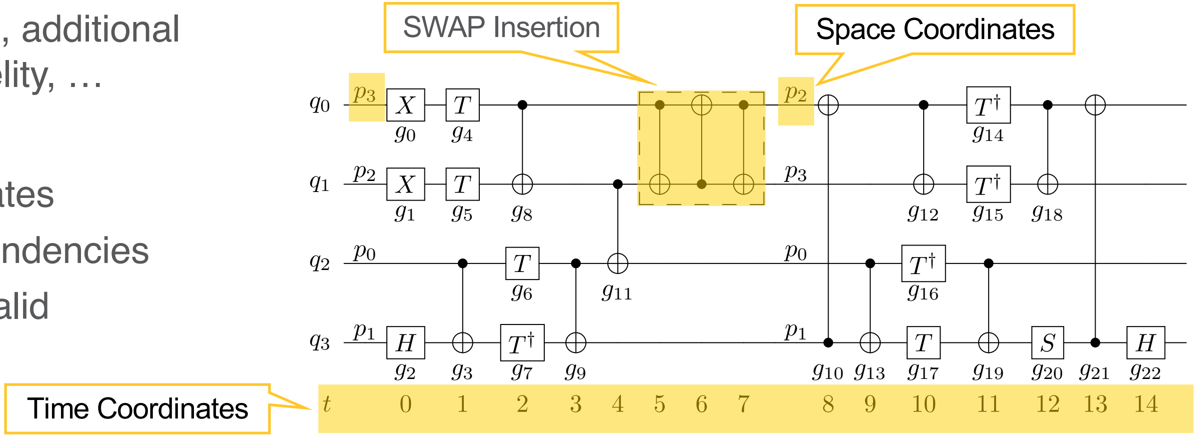
- Input: quantum circuit/program, coupling graph
- Output: spacetime coordinates of all gates, including inserted SWAPs
- Objectives: depth, additional SWAP count, fidelity, …
- Constraints:
  - Execute all gates
  - Respect dependencies
  - SWAPs are valid



Quantum Circuit



Coupling Graph



QUEKO

BENCHMARKS FOR MEASURING LSQC OPTIMALITY

QUEKO: Previous Works on LSQC

- Layer-by-layer:
  - [MFM08], [ZPW18]: lookahead search guided by heuristic cost function
  - [SSP14]: optimize the ‘total distance’
- Gate-by-gate:
  - [SSC18]: heuristic search for min #SWAPs
  - [WBZ19]: optimize #SWAPs
- Use dependency:
  - [MLM19]: optimize fidelity upper bound
  - [LDX19]: bi-directional search with cost function concerning both #SWAPs and depth
- Industry tools: Quilc, Qiskit, tlket> [CDD19], Cirq, …

Are they good enough?

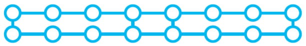
QUEKO: Construction

QUEKO: depth and gate count optimal benchmarks tailored to arbitrary devices for LSQC

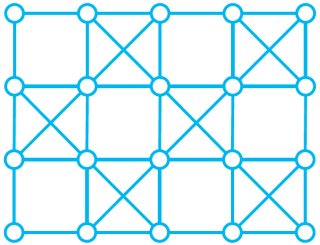
- Input: coupling graph, target depth gate density
- Backbone construction: grow a dependency chain
- Sprinkling: match the gate density profile
- Scrambling: challenge the LSQC tools
- Output: OpenQASM file

Evaluating Existing LSQC Tools with QUEKO

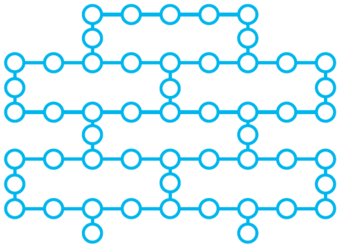
- Devices: Google Sycamore, Rigetti Aspen-4, IBM Q Tokyo, and IBM Q Rochester
- Circuits: QUEKO benchmarks
  - Depth:
    - 5-45 as near-term feasible,
    - 100-900 as scalability study
  - Gate density: profile of Toffoli gate and quantum supremacy experiment [AAB19]
- Tools:
  - Cirq (Google)
  - Qiskit (IBM)
  - tlket> (Quantinuum)
  - [ZPW18]



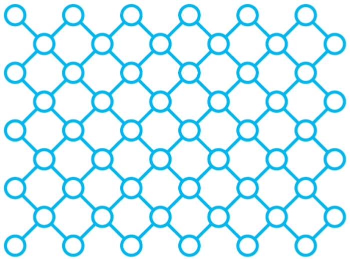
(b) Rigetti's Aspen-4 device graph



(c) IBM's Tokyo device graph



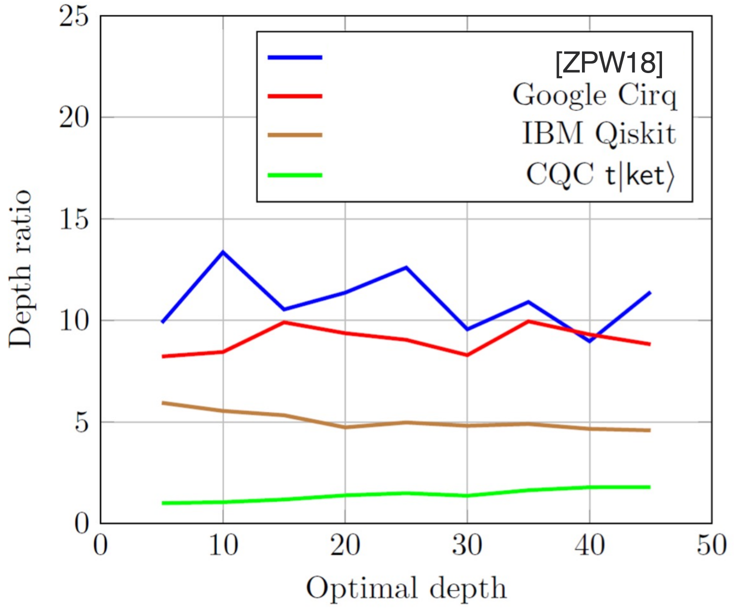
(d) IBM's Rochester device graph



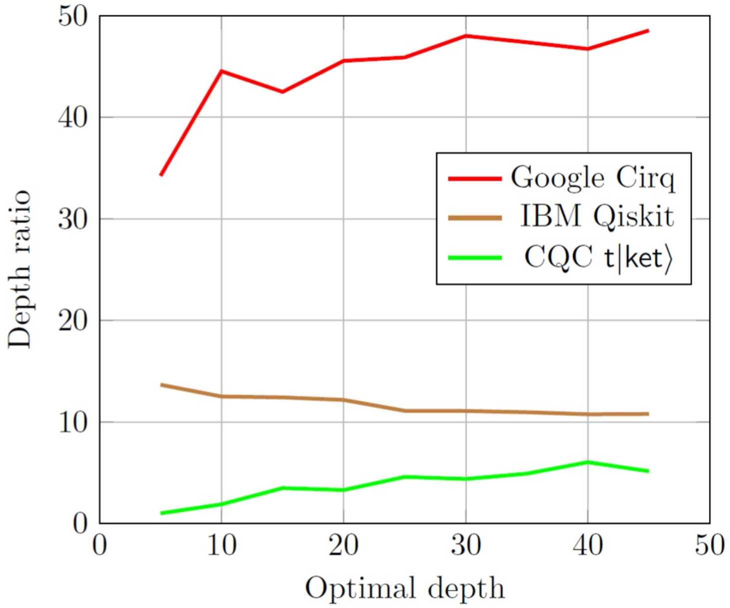
(e) Google's Sycamore device graph

QUEKO: Near-Term Feasible Cases

Optimality Gaps of Several Layout Synthesis Tools Revealed by B_{NTF} QUEKO Benchmarks



Toffoli gate density
Rigetti Aspen-4 Device



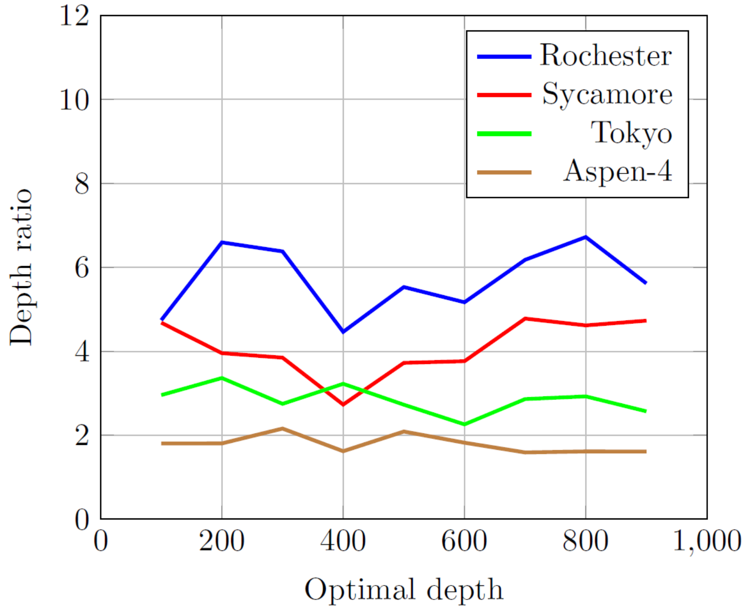
Quantum supremacy experiment gate density
Google Sycamore device

Depth ratio: depth achieved / optimal depth, consistently >1.5x

UCLA Samueli School of Engineering

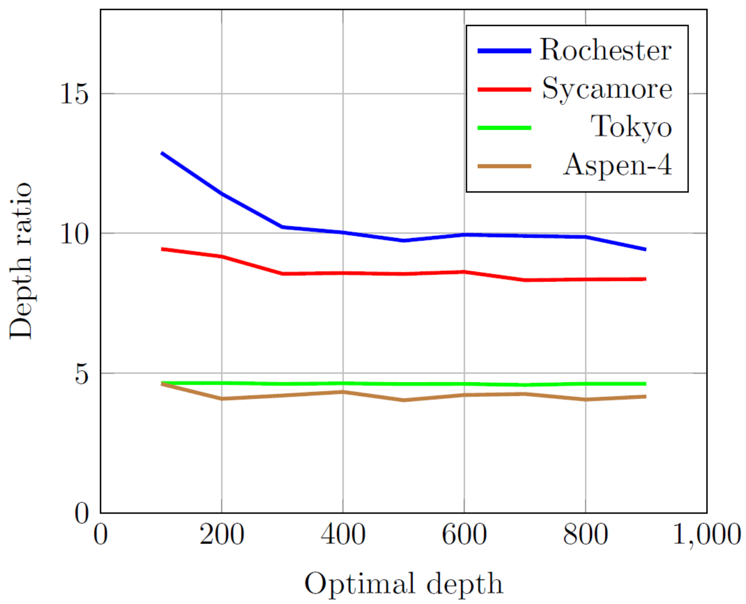
QUEKO: Results in Scalability Study

Optimality Gaps of Two Layout Synthesis Tools Revealed by B_{SS} QUEKO Benchmarks



>1.8x

CQC t|ket⟩ Performance



~4.8x

IBM Qiskit Performance

OLSQ

FORMULATION AND IMPLEMENTATION FOR OPTIMAL LSQC

OLSQ: Variables

Variables in OLSQ

- Spacetime Coordinates (x_l, t_l) for every gate g_l
  - If g_l is a single-qubit gate, x_l is a physical qubit; if g_l is a two-qubit gate, x_l is an edge
- Mapping π_q^t: at time t, logical qubit q is mapped to the quantum register π_q^t
- Use of SWAP σ_e^t: $\sigma_e^t = 1$ iff. there is a SWAP on edge e and its last time step is t

- More efficient encoding than [WBZ19]: N^{MT} (N quantum registers, M qubits, T time steps)

Mapping of logical qubit q_0

$\pi_0^0 = 3,$

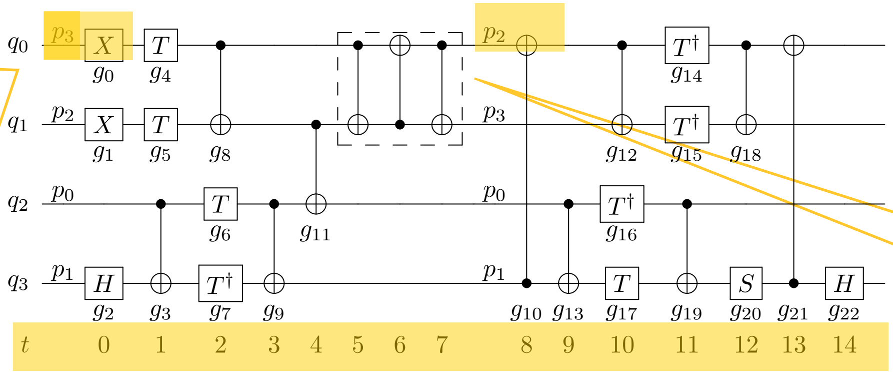
$\pi_0^1 = 3,$

…

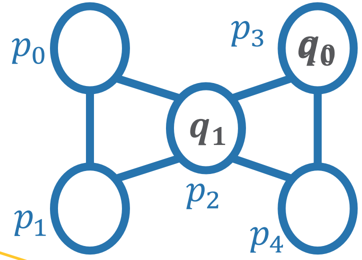
$\pi_0^8 = 2,$

…

$\pi_0^{14} = 2$



Result of OLSQ



The SWAP insertion $\sigma^7_{(p_2,p_3)} = 1$

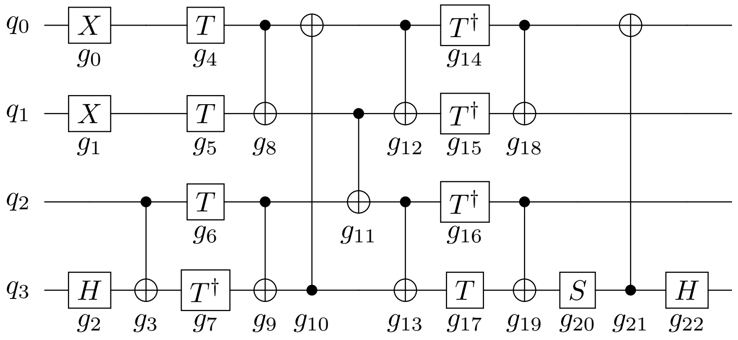
For all other e's and t's: $\sigma_e^t = 0$

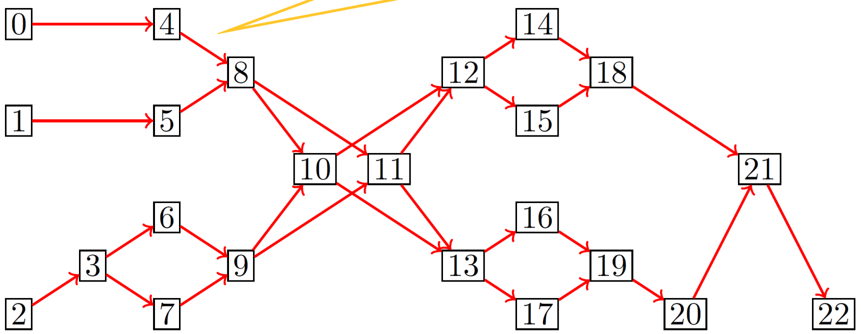
UCLA Samueli School of Engineering

OLSQ: Constraints

- Validity
  - Valid space coordinates: if g_l is a single-qubit gate, $x_l \in P$; if a two-qubit gate, $x_l \in E$ (all edges in G)
  - ...
- Injective mapping: $\forall t, q, q' \quad q' \neq q \Rightarrow \pi_q^t \neq \pi_{q'}^t$
- Dependencies: if g_l depends on $g_{l'}$, then $t_l > t_{l'}$

g_8 depends on $g_4 \ \Rightarrow t_8 > t_4$

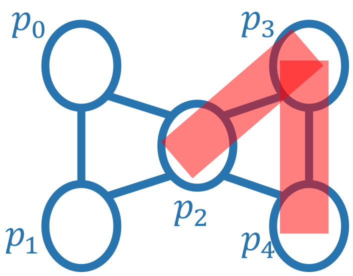




OLSQ: Constraints

Mapping transformed by SWAPs

- $\left[(\pi_0^0 = p_3) \wedge \left(\sum_{e:\, p_3 \in e} \sigma_e^0 = 0\right)\right] \Rightarrow (\pi_0^1 = p_3)$
- $\left[(\pi_0^7 = p_3) \wedge \left(\sigma_{(p_2,p_3)}^7 = 1\right)\right] \Rightarrow (\pi_0^8 = p_2)$

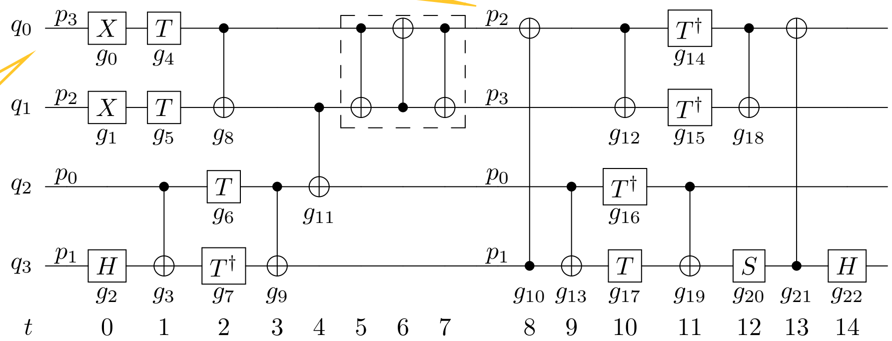


There is a SWAP on (p_2, p_3) finishing at time 7.

Mapping of q_0 changes at time 8.

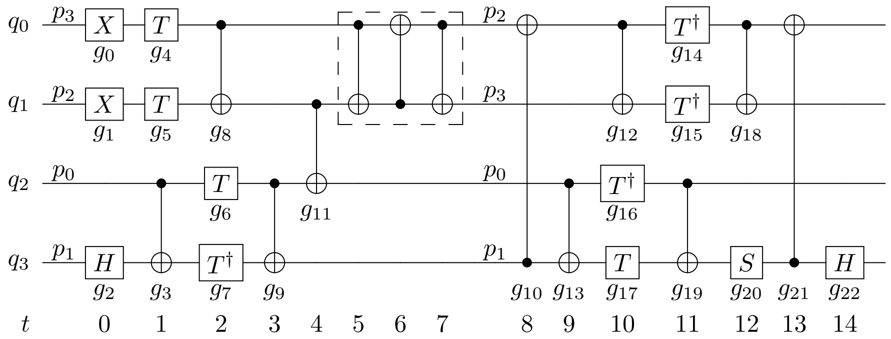
There are no SWAPs ending at time 0 on any edge connecting p_3.

Mapping of q_0 at time 1 is the same with that at time 0.



OLSQ: Objectives

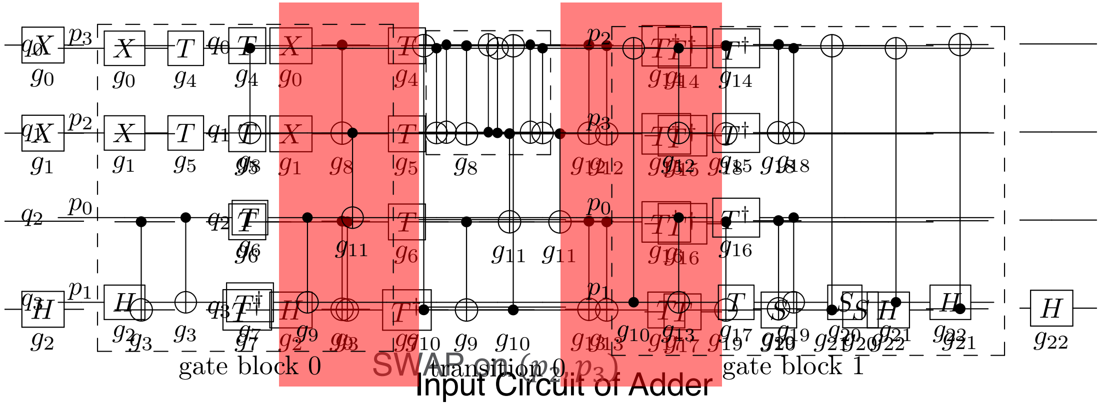
- Depth = $\max t_l$
- #SWAP = $\sum \sigma_e^t$, or/and
- Fidelity = $\prod_q f_{\mathrm{m}}(\pi_q^T) \cdot \prod_{l_1} f_1(x_{l_1}) \cdot \prod_{l_2} f_2(x_{l_2}) \cdot \prod_{e,t} f_S(e)^{\sigma_e^t}$
  - f_{m}, f_1, f_2, and f_S are measurement, single-qubit gate, two-qubit gate, and SWAP fidelity.
  - π_q^T is the final mapping. l_1 goes over all single-qubit gates; l_2 goes over all two-qubit gates.



Transition-Based (TB-) OLSQ

- Motivation: many mapping variables are redundant in the lack of SWAPs.
- Solution: gate blocks + transitions.
- Variables: mapping, spacetime, SWAP *for each block* instead for each time step
    - 2 blocks versus 14 time steps
- After SWAP insertion, we can use ASAP (as soon as possible) scheduling

Gate block 0 Gate block 1



TB-OLSQ: Summary of Constraints

Constraints	TB-OLSQ Revision
Validity	Change bounds to #blocks
Injective Mapping	No change
Dependency	Change $>$ to $\geq$
Mapping constrains Spacetime Coordinates	No change
No Overlap with Other SWAPs	No change
No Overlap with Original Gates	Not required anymore
Mapping transformed by SWAPs	No change

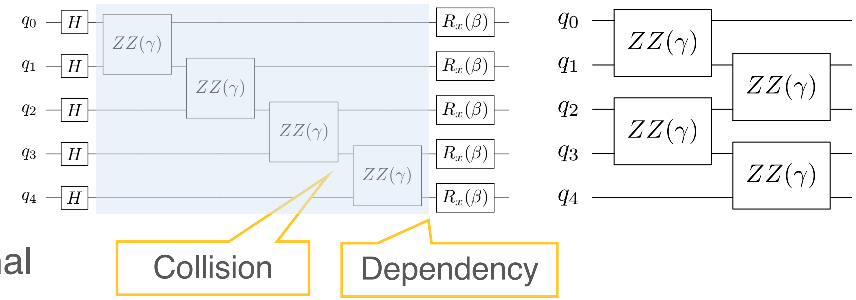
TB-OLSQ: Evaluations

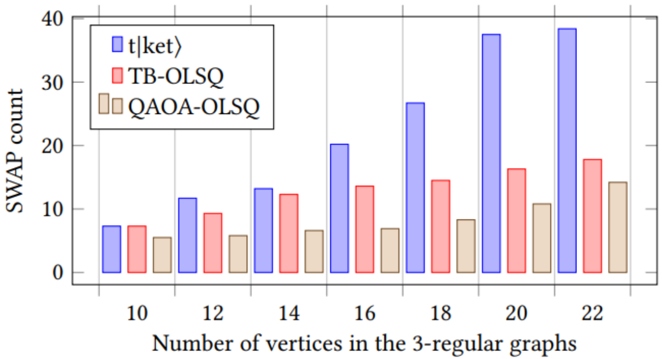
Comparison with OLSQ >400x speedup (geomean)

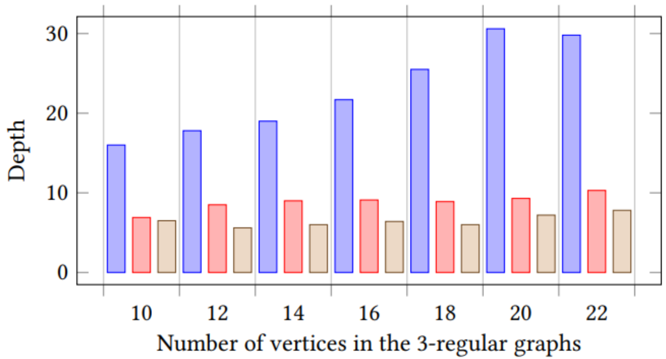
Benchmarks	TB-OLSQ optimizing SWAP vs. tlket> [CDD19]	TB-OLSQ optimizing fidelity vs. TriQ [MLM19]
Small circuits to verify optimality	(reduction geomean) 76%	1.07X
Larger arithmetic circuits	57%	1.02X
QUEKO circuits	100%	2.10X

OLSQ on QAOA: TB + Commutation

- QAOA [FGG14] a promising application
- ZZ-phase = $\begin{bmatrix} e^{-i\gamma} & 0 & 0 & 0 \\ 0 & e^{i\gamma} & 0 & 0 \\ 0 & 0 & e^{i\gamma} & 0 \\ 0 & 0 & 0 & e^{-i\gamma} \end{bmatrix}$
- Commutable, i.e., AB=BA, since diagonal







Result: 70% depth reduction, 54% SWAP reduction compared to tlket>.

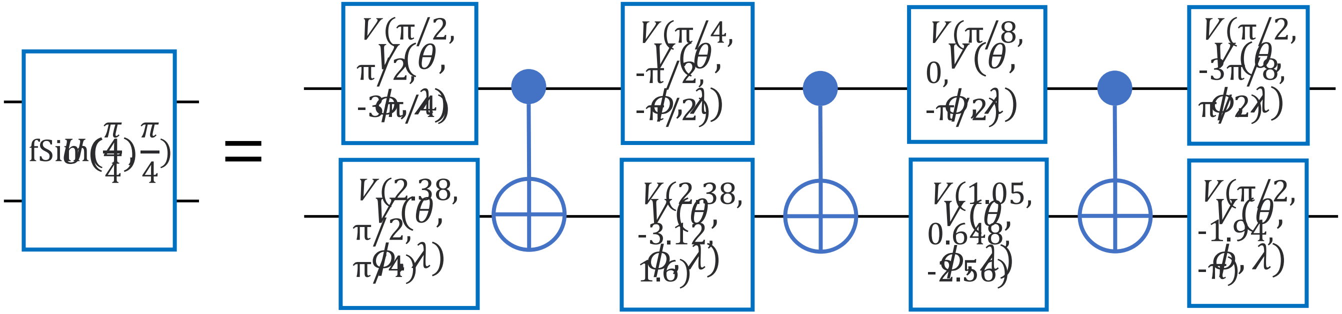
UCLA Samueli School of Engineering

OLSQ-GA: Programmable Two-Qubit Gate

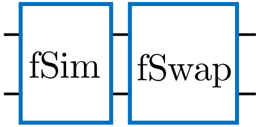
- A programmable single-qubit gate can be configured to be any matrix in $U(2)$

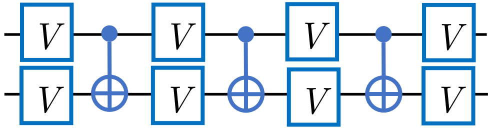
$$V(\theta,\phi,\lambda)=\begin{bmatrix}\cos(\theta/2) & -e^{i\lambda}\sin(\theta/2)\\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2)\end{bmatrix}$$

- A programmable two-qubit gate can be configured to any matrix in $U(4)$
- KAK Decomposition [VW04]: any $U(4)$ to 3 CNOT's and some $U(2)$
- Many quantum programs expressed with $U(4)$ gates: $\mathrm{fSim}(\theta,\phi)$ in chemistry simulation [KMW18], QAOA, and quantum convolutional neural networks [CCL19]



OLSQ-GA: Gate Absorption





$\times 2$

$$\mathrm{fSim}(\theta,\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -i\sin\theta & 0 \\ 0 & -i\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{bmatrix}$$

$$\mathrm{fSwap} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\mathrm{fSwap} \cdot \mathrm{fSim}(\theta,\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -i\sin\theta & \cos\theta & 0 \\ 0 & \cos\theta & -i\sin\theta & 0 \\ 0 & 0 & 0 & -e^{-i\phi} \end{bmatrix}$$

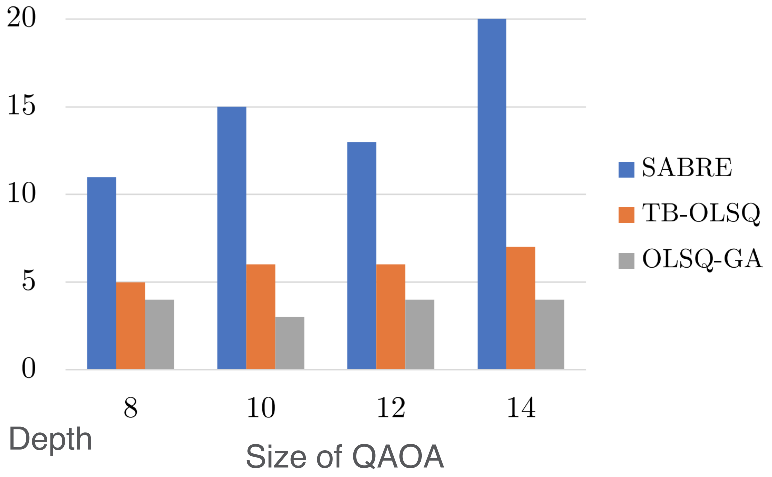
Formulation:

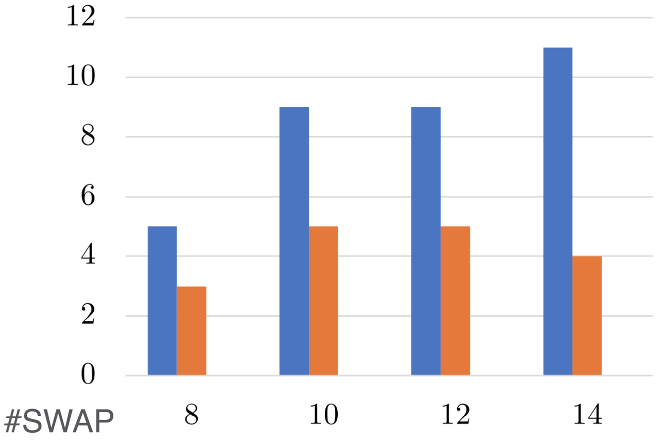
- Use of absorbed SWAP $\alpha_e^t = 1$ iff. there is an absorbed SWAP on edge e at time t
- Mapping transformed by both absorbed and explicit SWAPs α_e^t and σ_e^t

UCLA Samueli School of Engineering

OLSQ-GA: Evaluation on SWAPs and Depth

- Similar QAOA instances of size 8 to 14 like in leading QAOA experiments [HSN21]
- SABRE [LDX19]: leading heuristic mapper, recently adopted in Qiskit
- OLSQ-GA (considers commutation) reduced depth up to 80%, absorbed all the SWAPs

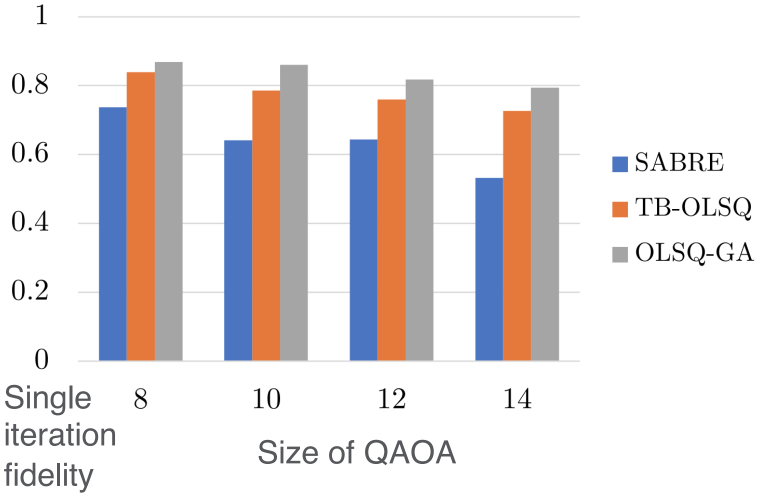


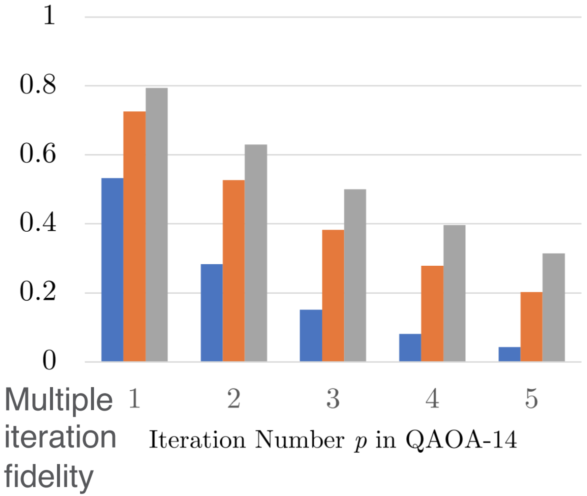


Size of QAOA (There are no gray bars since OLSQ-GA absorbed all the SWAPs)

OLSQ-GA: Evaluation on Fidelity

- OLSQ-GA improves fidelity by up to 49% for 1 iteration, 636% for 5 iterations.



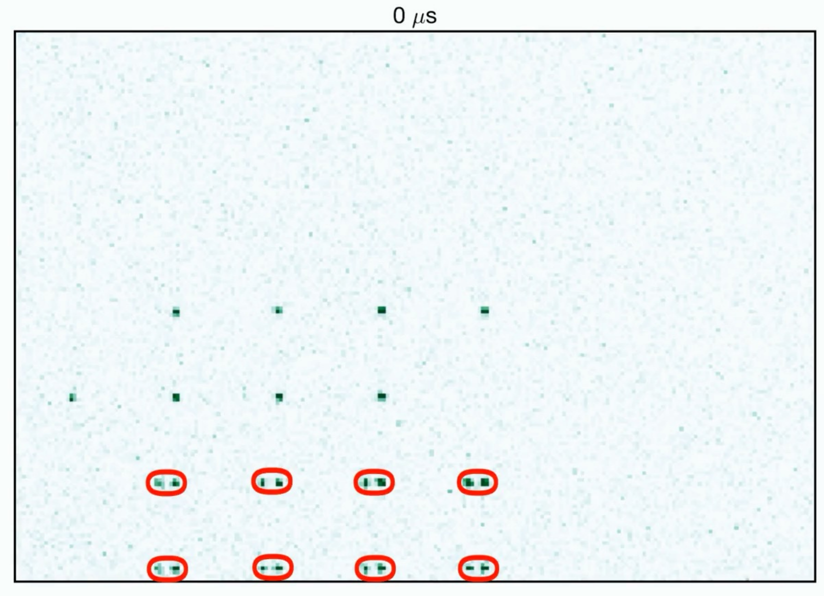


OLSQ-RAA

LSQC FOR RECONFIGURABLE ATOM ARRAYS

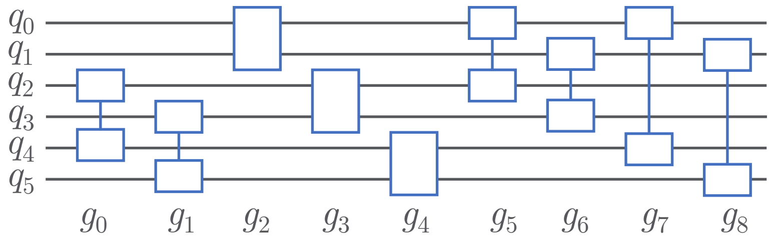
OLSQ-RAA: Reconfigurable Architectures

- Neutral atom arrays [BLS22]: 1) many quantum registers (>200 possible), 2) reconfigurability
- When atoms (qubits) are close, two-qubit gates by illuminating with laser
- AOD traps are mobile
- SLM traps are stationary
- How can we execute any program?

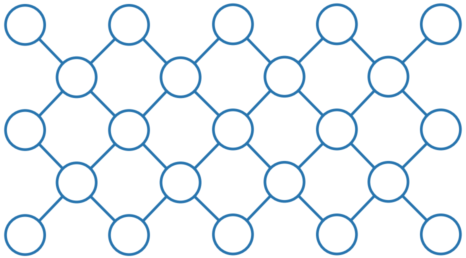


OLSQ-RAA: Example

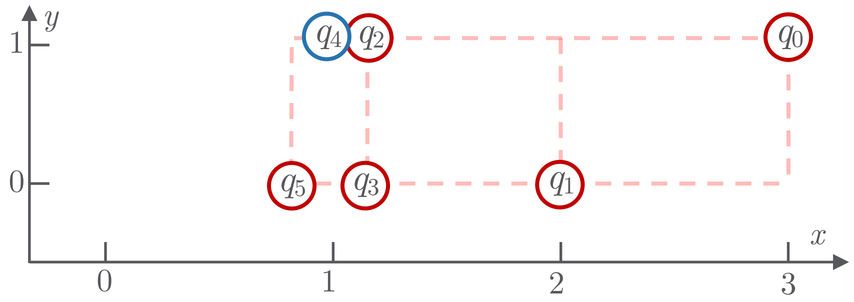
Compiling a QAOA program:



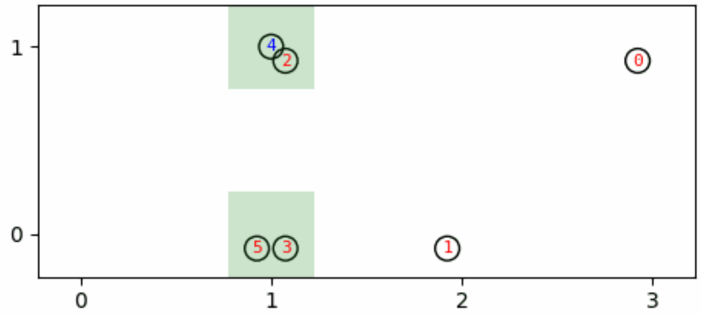
3 SWAPs on a fixed architecture



(Part of) Google Sycamore

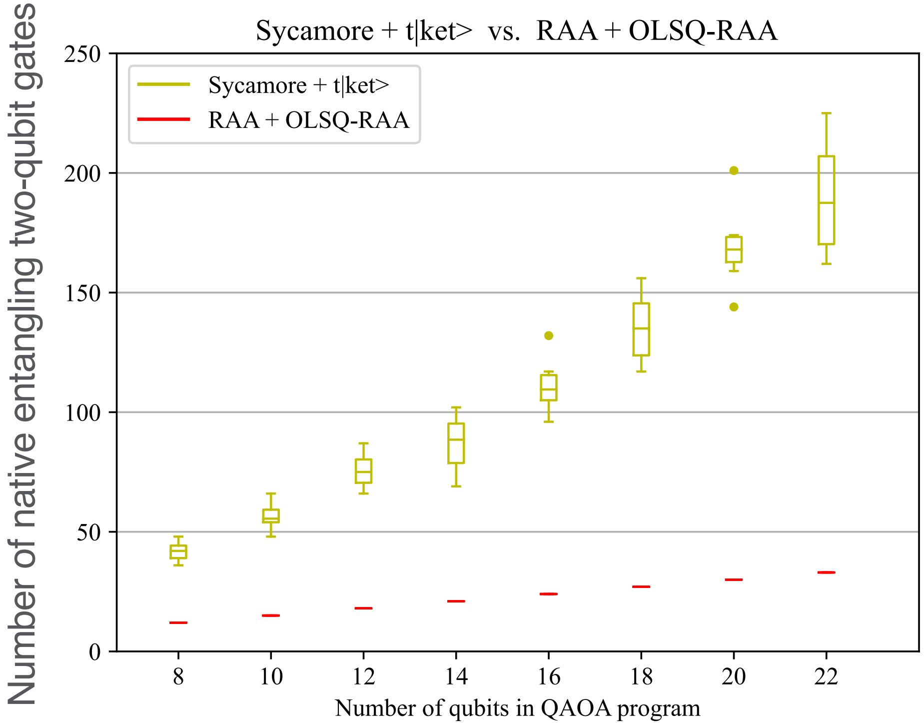


$t = 0$



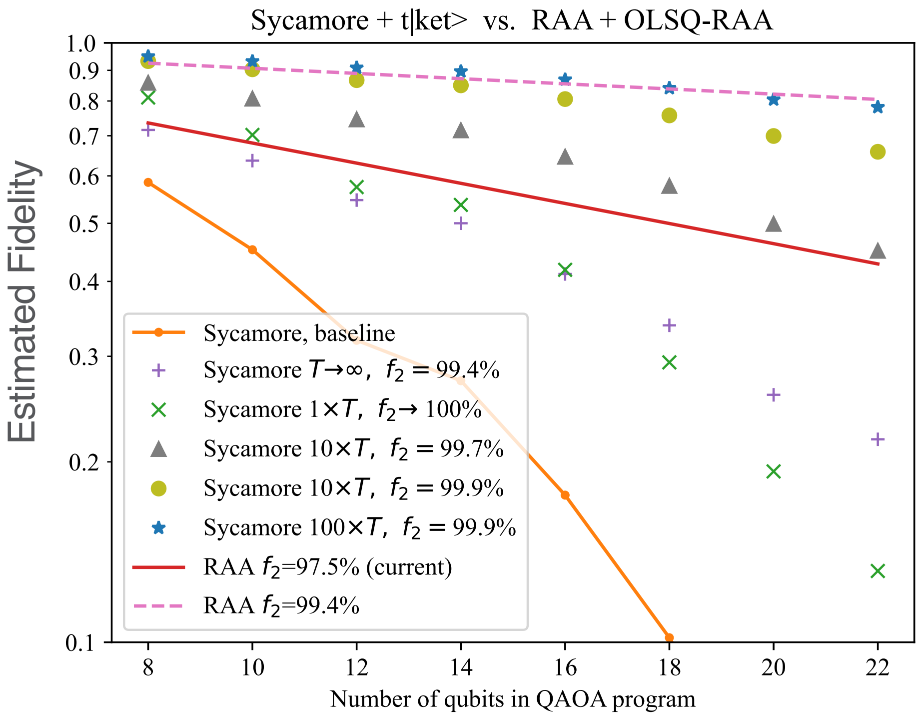
0 SWAP on RAA

OLSQ-RAA: Evaluations on Number of Gates



- Sycamore + tIket> represents the previous leading experiment [HSN21].
- For RAA, all the 'routing' is done by array movements.
- 5.72x less gates for QAOA-22.

OLSQ-RAA: Evaluation on Fidelity



- f_2 two-qubit gate fidelity, T coherence time
- Currently, on QAOA-22, RAA + OLSQ-RAA has 14.4x higher fidelity than Sycamore + tlket>.
- The pure effect of reconfigurability is 1.96x.
- RAA with $f_2 = 99.4\%$ and current T ~= Sycamore + tlket> with $f_2 = 99.9\%$ and 100x current T

UCLA Samueli School of Engineering

Q&A

Publications

[TC20] B. Tan and J. Cong, “Optimal Layout Synthesis for Quantum Computing,” in *2020 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Virtual Event, USA, Nov. 2020, doi: 10.1145/3400302.3415620. arXiv: cs.AR/2007.15671

[TC21a] B. Tan and J. Cong, “Optimal Qubit Mapping with Simultaneous Gate Absorption,” in *2021 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Virtual Event, USA, Nov. 2021, doi: 10.1109/ICCAD51958.2021.9643554 arXiv: 2109.06445

[TC21b] B. Tan and J. Cong, “Optimality Study of Existing Quantum Computing Layout Synthesis Tools,” *IEEE Transactions on Computers*, vol. 70, no. 9, pp.1363–1373 , 2021. doi: 10.1109/TC.2020.3009140. arXiv: quant-ph/2002.09783

[TBL22] B. Tan D. Bluvstein, M. D. Lukin, and J. Cong, “Qubit Mapping for Reconfigurable Atom Arrays,” in *2022 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Virtual Event, USA, Nov. 2022, doi: 10.1145/3508352.3549331

[TC22] B. Tan and J. Cong “Layout Synthesis for Near-Term Quantum Computing: Gap Analysis and Optimal Solution” To appear in Rasit O. Topaloglu, editor, *Design Automation of Quantum Computers*, Springer, 2022.

[LTN22] W.-H. Lin, B. Tan, M. Y. Niu, J. Kimko, and J. Cong “Domain-Specific Quantum Architecture Optimization” To appear in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2022. arXiv: cs.AR/2207.14482

References

[NC10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge, United Kingdom: Cambridge University Press, 2010.

[BKM14] R. Barends *et al.*, "Superconducting quantum circuits at the surface code threshold for fault tolerance," *Nature*, vol. 508, no. 7497, pp. 500–503, Apr. 2014, doi: 10.1038/nature13171.

[GCS17] J. M. Gambetta, J. M. Chow, and M. Steffen, "Building logical qubits in a superconducting quantum computing system," *npj Quantum Inf*, vol. 3, no. 1, p. 2, Dec. 2017, doi: 10.1038/s41534-016-0004-0.

[SSW21] P. Scholl *et al.*, "Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms," *Nature*, vol. 595, no. 7866, pp. 233–238, Jul. 2021, doi: 10.1038/s41586-021-03585-1.

[EKC22] S. Ebadi *et al.*, "Quantum optimization of maximum independent set using Rydberg atom arrays," *Science*, vol. 0, no. 0, p. eabo6587, May 2022, doi: 10.1126/science.abo6587.

[MFM08] Maslov, Dmitri, Sean M. Falconer, and Michele Mosca. "Quantum Circuit Placement." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27, no. 4 (April 2008): 752–63. https://doi.org/10.1109/TCAD.2008.917562.

[ZPW18] A. Zulehner, A. Paler, and R. Wille, "Efficient mapping of quantum circuits to the IBM QX architectures," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, Germany, Mar. 2018, pp. 1135–1138, doi: 10.23919/DATE.2018.8342181.

[SSP14] A. Shafaei, M. Saeedi, and M. Pedram. "Qubit Placement to Minimize Communication Overhead in 2D Quantum Architectures." In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 495–500. Singapore: IEEE, 2014. https://doi.org/10.1109/ASPDAC.2014.6742940.

[SSC18] M. Y. Siraichi, V. F. dos Santos, S. Collange, and F. M. Q. Pereira. "Qubit Allocation." In *Proceedings of the 2018 International Symposium on Code Generation and Optimization - CGO 2018*, 113–25. Vienna, Austria: ACM Press, 2018. https://doi.org/10.1145/3168822.

References

[WBZ19] R. Wille, L. Burgholzer, and A. Zulehner, “Mapping Quantum Circuits to IBM QX Architectures Using the Minimal Number of SWAP and H Operations,” in *Proceedings of the 56th Annual Design Automation Conference 2019 on - DAC ’19*, Las Vegas, NV, USA, 2019, pp. 1–6, doi: 10.1145/3316781.3317859.

[MLM19] P. Murali, N. M. Linke, M. Martonosi, A. J. Abhari, N. H. Nguyen, and C. H. Alderete, “Full-stack, real-system quantum computer studies: architectural comparisons and design insights,” in *Proceedings of the 46th International Symposium on Computer Architecture - ISCA ’19*, Phoenix, Arizona, 2019, pp. 527–540, doi: 10.1145/3307650.3322273.

[LDX19] G. Li, Y, Ding, and Y. Xie. “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 - ASPLOS ’19*, 1001–14. Providence, RI, USA: ACM Press, 2019. https://doi.org/10.1145/3297858.3304023.

[CDD19] A. Cowtan, S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, and S. Sivarajah, “On the qubit routing problem,” in *14th conference on the theory of quantum computation, communication and cryptography (TQC 2019)*, Dagstuhl, Germany, 2019, vol. 135, p. 5:1–5:32, doi: 10.4230/LIPIcs.TQC.2019.5.

[CCR04] C.-C. Chang, J. Cong, M. Romesis, and M. Xie, “Optimality and Scalability Study of Existing Placement Algorithms,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 4, pp. 537–549, Apr. 2004, doi: 10.1109/TCAD.2004.825870.

[AAB19] F. Arute *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019, doi: 10.1038/s41586-019-1666-5.

[FGG14] E. Farhi, J. Goldstone, and S. Gutmann, “A Quantum Approximate Optimization Algorithm,” *arXiv:1411.4028 [quant-ph]*, Nov. 2014. Available: http://arxiv.org/abs/1411.4028.

[VW04] F. Vatan and C. Williams, “Optimal quantum circuits for general two-qubit gates,” *Phys. Rev. A*, vol. 69, no. 3, p. 032315, Mar. 2004, doi: 10.1103/PhysRevA.69.032315.

References

[KMW18] I. D. Kivlichan *et al.*, “Quantum Simulation of Electronic Structure with Linear Depth and Connectivity,” *Phys. Rev. Lett.*, vol. 120, no. 11, p. 110501, Mar. 2018, doi: 10.1103/PhysRevLett.120.110501.

[CCL19] I. Cong, S. Choi, and M. D. Lukin, “Quantum convolutional neural networks,” *Nat. Phys.*, vol. 15, no. 12, Art. no. 12, Dec. 2019, doi: 10.1038/s41567-019-0648-8.

[HSN21] M. P. Harrigan *et al.*, “Quantum approximate optimization of non-planar graph problems on a planar superconducting processor,” *Nat. Phys.*, vol. 17, no. 3, pp. 332–336, Mar. 2021, doi: 10.1038/s41567-020-01105-y.

[BLS22] D. Bluvstein *et al.*, “A quantum processor based on coherent transport of entangled atom arrays,” *Nature*, vol. 604, no. 7906, Art. no. 7906, Apr. 2022, doi: 10.1038/s41586-022-04592-6.

[BLL21] N. Bjørner, M. Levatich, N. P. Lopes, A. Rybalchenko, and C. Vuppalapati, “Supercharging Plant Configurations Using Z3,” in *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, vol. 12735, P. J. Stuckey, Ed. Cham: Springer International Publishing, 2021, pp. 1–25. doi: 10.1007/978-3-030-78230-6_1.

[CCS05] T. Chan, J. Congy, and K. Sze, “Multilevel generalized force-directed method for circuit placement,” in *Proceedings of the 2005 international symposium on physical design - ISPD ’05*, San Francisco, California, USA, 2005, p. 185. doi: 10.1145/1055137.1055177.

[FFM12] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Phys. Rev. A*, vol. 86, no. 3, Art. no. 3, Sep. 2012, doi: 10.1103/PhysRevA.86.032324.

[Lit19] D. Litinski, “A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery,” *Quantum*, vol. 3, p. 128, Mar. 2019, doi: 10.22331/q-2019-03-05-128.

[AG04] S. Aaronson and D. Gottesman, “Improved Simulation of Stabilizer Circuits,” *Phys. Rev. A*, vol. 70, no. 5, Art. no. 5, Nov. 2004, doi: 10.1103/PhysRevA.70.052328.