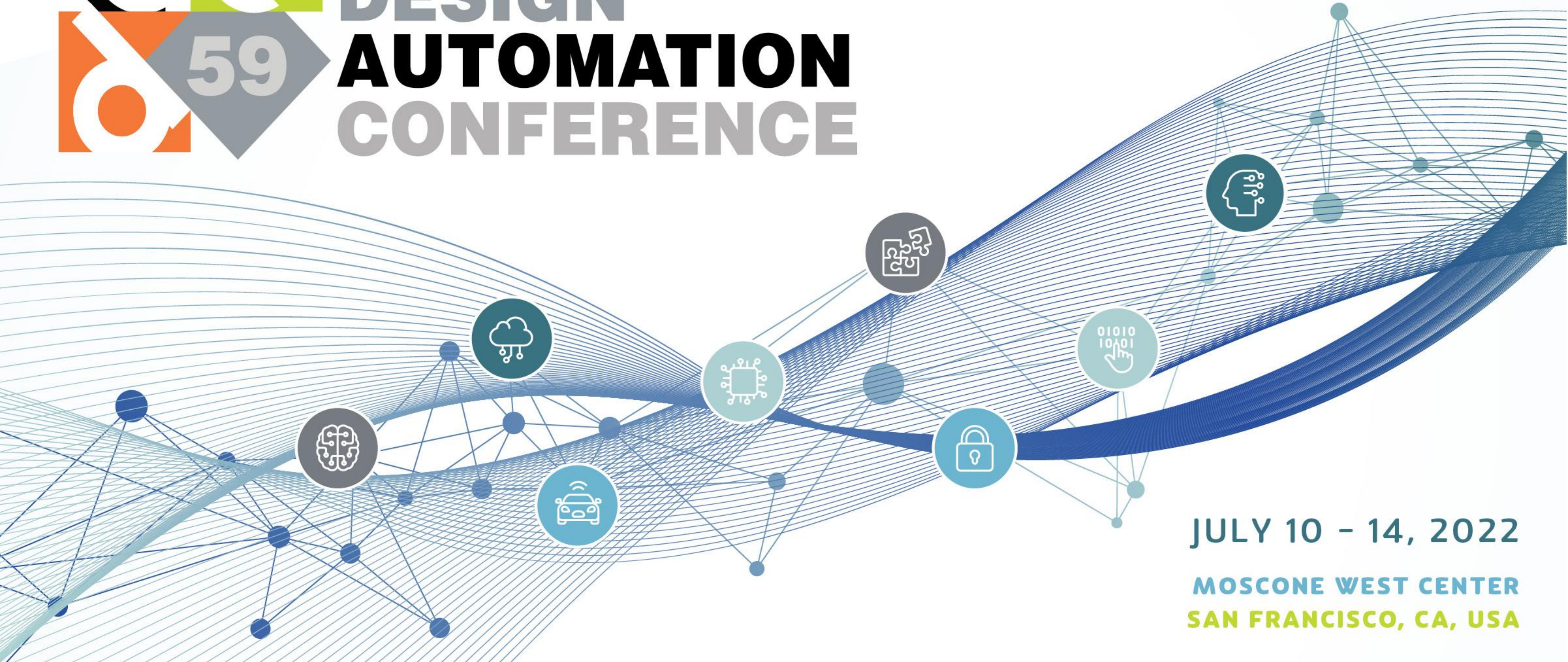
# DESIGN AUTOMATION CONFERENCE

59



JULY 10 - 14, 2022

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

# Qubit Mapping and Scheduling:

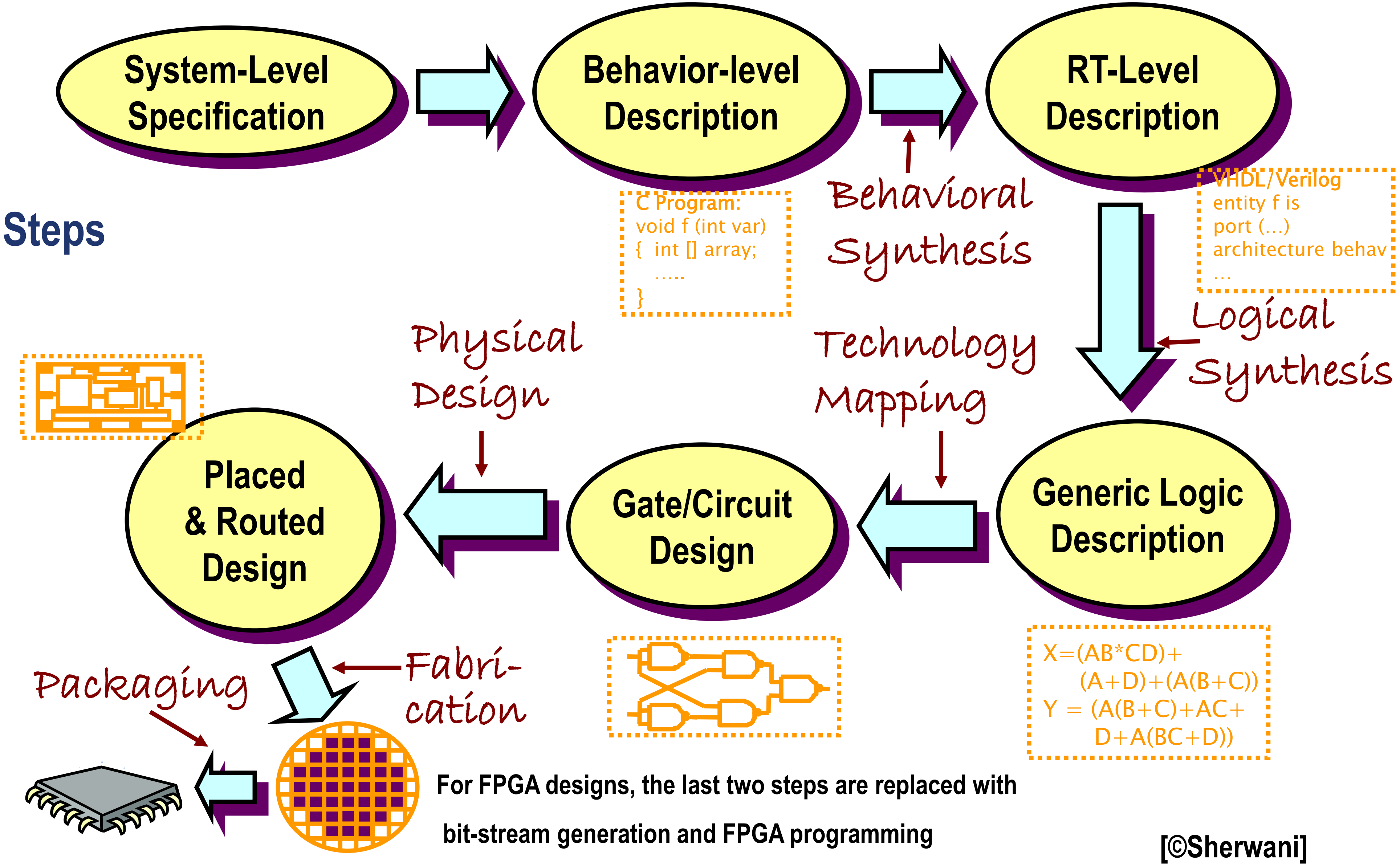
## Gap Analysis and Optimal Solutions

**Jason Cong**
**UCLA Computer Science**
**PhD Student: Daniel Bochen Tan**

**3/9/22**

# VLSI Design Automation Flow

## IC Design Steps



**System-Level Specification**

**Behavior-level Description**

C Program:
void f (int var)
{ int [] array;
…..
}

Behavioral Synthesis

**RT-Level Description**

VHDL/Verilog
entity f is
port (…)
architecture behav
…

Logical Synthesis

**Generic Logic Description**

X=(AB*CD)+
(A+D)+(A(B+C))
Y = (A(B+C)+AC+
D+A(BC+D))

Technology Mapping

**Gate/Circuit Design**

Physical Design

**Placed & Routed Design**

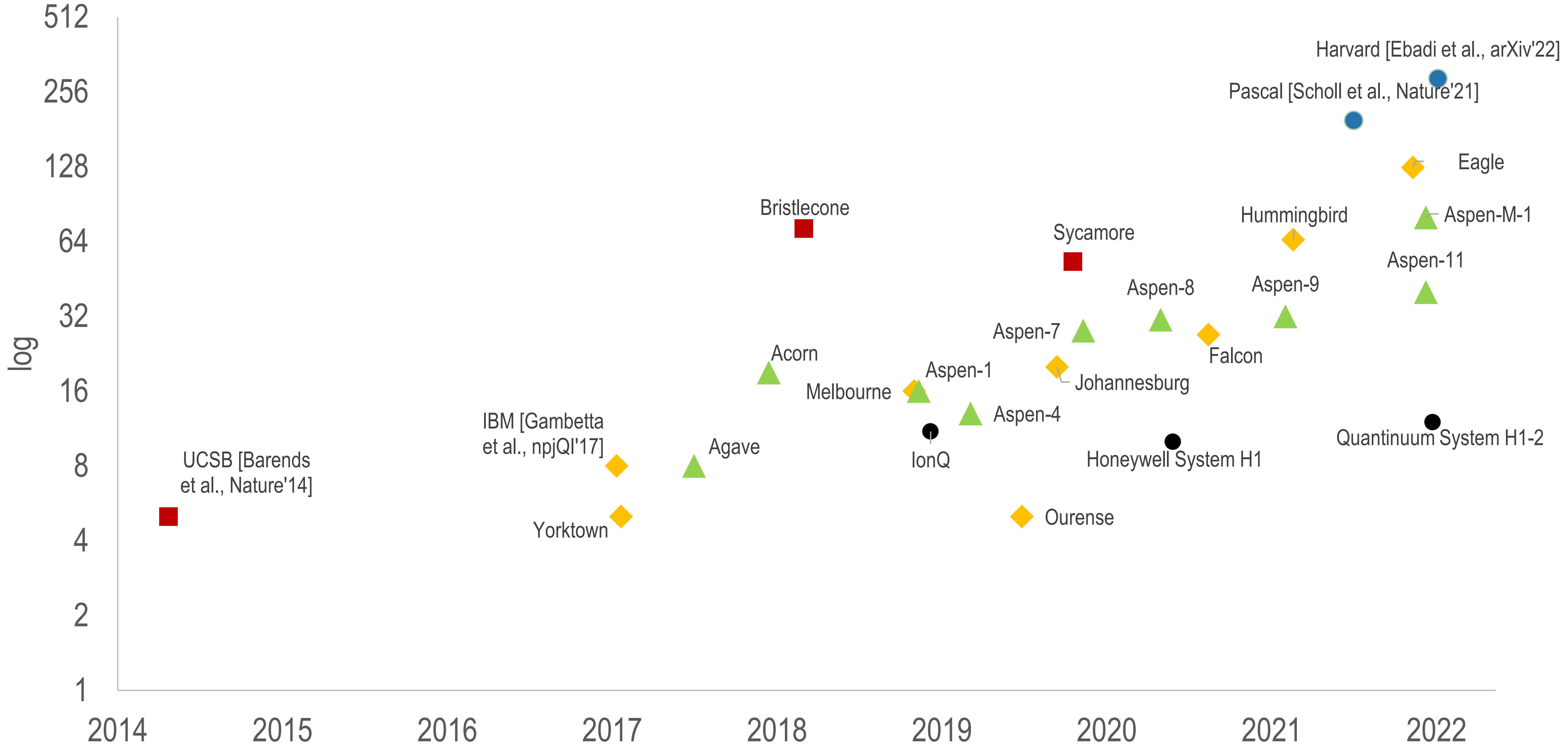
Fabri-cation

Packaging

**For FPGA designs, the last two steps are replaced with bit-stream generation and FPGA programming**

**[©Sherwani]**

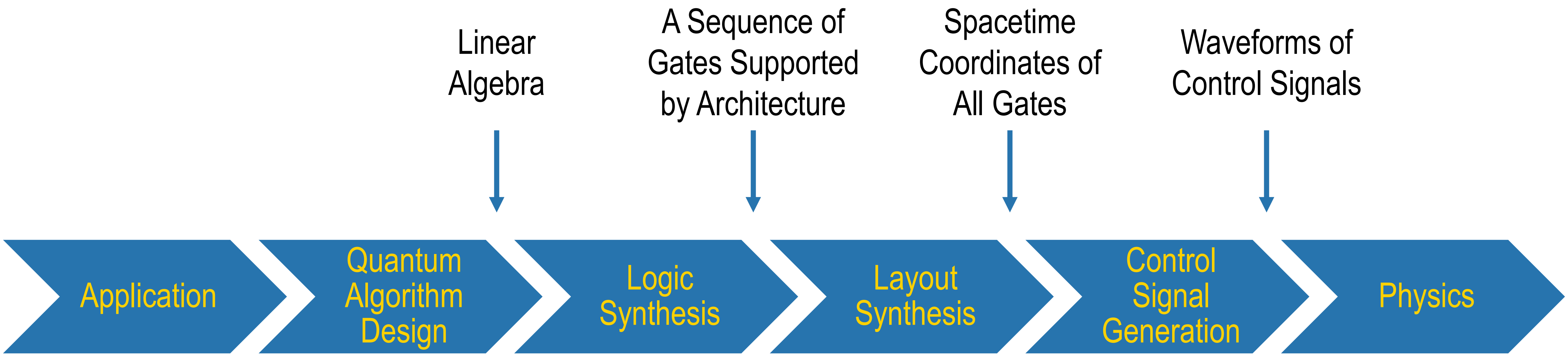
# Advances of Quantum Computing



Physical Qubits

3/9/22

# Flow of Quantum Computing



Linear Algebra

A Sequence of Gates Supported by Architecture

Spacetime Coordinates of All Gates

Waveforms of Control Signals

Application → Quantum Algorithm Design → Logic Synthesis → Layout Synthesis → Control Signal Generation → Physics

- Chemistry
- Optimization
- Machine Learning
- Cryptography

## Our research

**Architecture and Design Automation for Quantum Computing**

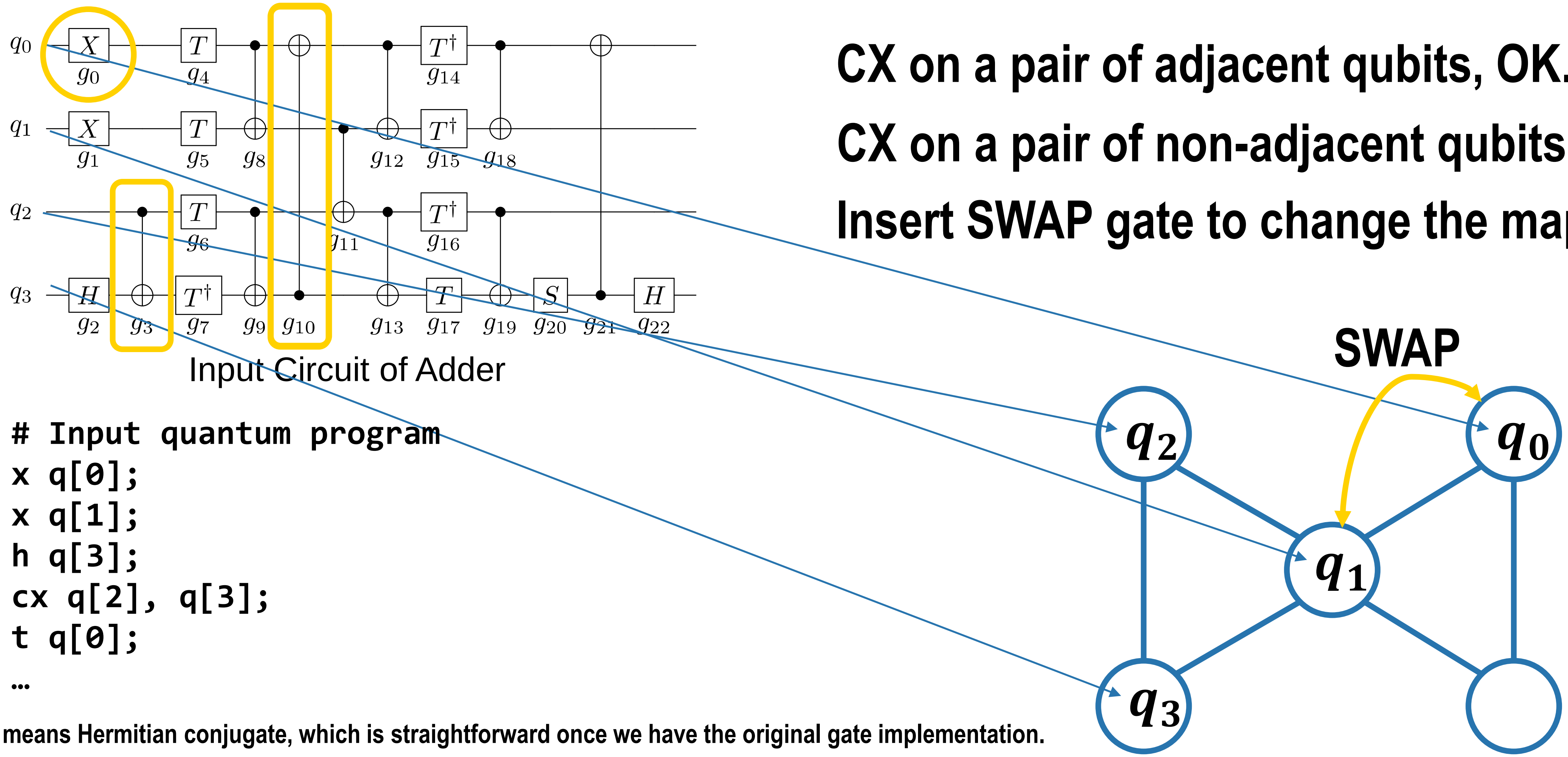
https://vast.cs.ucla.edu/projects/architecture-and-design-automation-quantum-computing

- Superconductor
- Trapped-Ion
- Silicon
- Optics

# Gate Model of Quantum Computation: Quantum Gates

- **Single-qubit gate**
  - “Bitflip” gate $X$: $\begin{pmatrix}\alpha\\\beta\end{pmatrix} \mapsto \begin{pmatrix}\beta\\\alpha\end{pmatrix}$ i.e., $\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$
  - Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$
  - Phase shift gate $R_\phi = \begin{pmatrix}1 & 0\\0 & e^{i\phi}\end{pmatrix}$
    - $P \equiv R_{\frac{\pi}{2}},\ T \equiv R_{\frac{\pi}{4}}$
  - Qiskit $U_3$ gate $U3(\theta,\phi,\lambda) = \begin{pmatrix}\cos(\theta/2) & -e^{i\lambda}\sin(\theta/2)\\e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2)\end{pmatrix}$
- **Two-qubit gate**
  - Controlled-not gate $CX$: $\begin{pmatrix}\alpha\\\beta\\\gamma\\\delta\end{pmatrix} \mapsto \begin{pmatrix}\alpha\\\beta\\\delta\\\gamma\end{pmatrix}$
- All gates are unitary: $AA^{+} = I$
- **The Solovay-Kitaev Theorem:** the gate set $\{H, P, T, CX\}$ is universal for quantum computing! [Nielsen&Chuang, QCQI]

# Qubit Mapping and Scheduling (Layout Synthesis) for QC (LSQC)



Input Circuit of Adder

**CX on a pair of adjacent qubits, OK.**

**CX on a pair of non-adjacent qubits!**

**Insert SWAP gate to change the mapping**

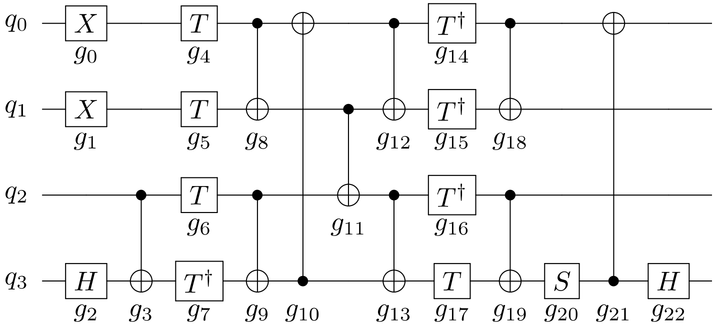
```
# Input quantum program
x q[0];
x q[1];
h q[3];
cx q[2], q[3];
t q[0];
…
```

Coupling Graph of IBM QX 2

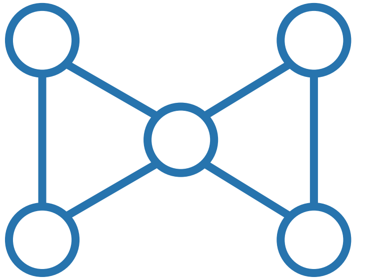
† means Hermitian conjugate, which is straightforward once we have the original gate implementation.

# Layout Synthesis for Quantum Computing (LSQC)

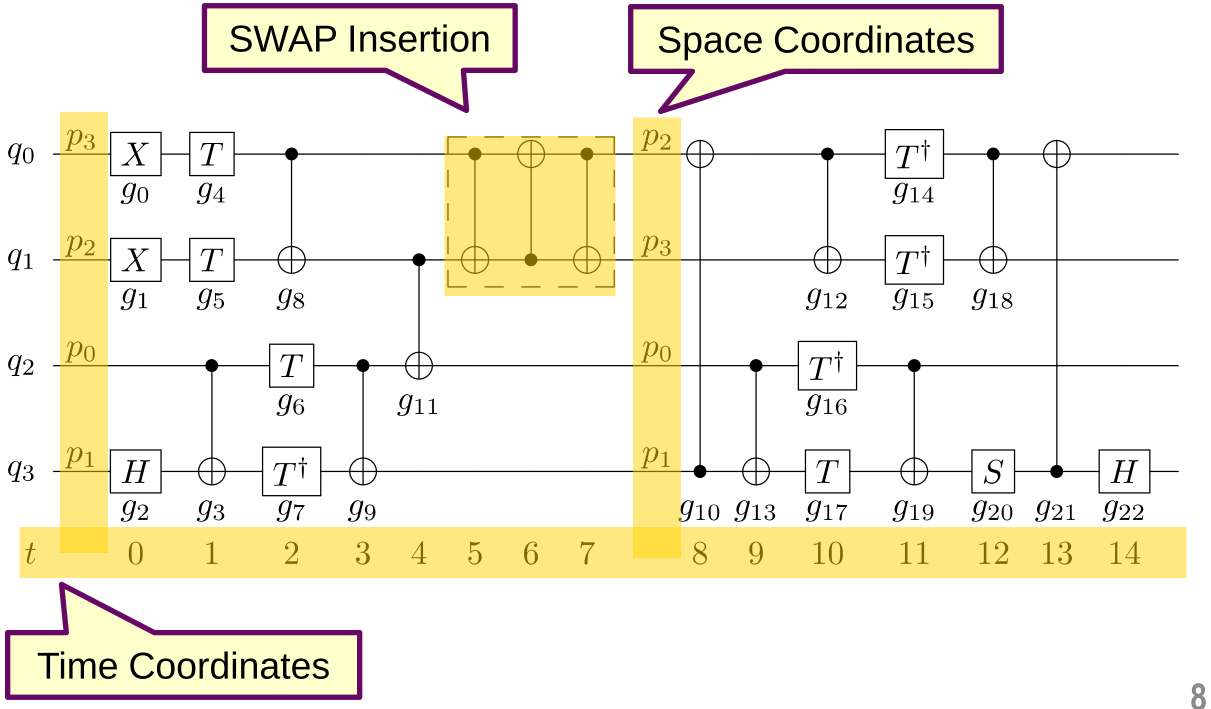
- **Input: quantum circuit/program, coupling graph**
- **Output: spacetime coordinates of all gates, including inserted SWAPs**
- **Objectives: depth, additional SWAP count, fidelity, …**
- **Constraints:**
  - Execute all gates
  - Respect dependencies



Input Circuit of Adder



Coupling Graph of IBM QX 2



SWAP Insertion

Space Coordinates

Time Coordinates

# Outline

- Introduction
- <u>Gap analysis for quantum compilation</u>
- Optimal layout synthesis for quantum computing (OLSQ)
- OLSQ with Gate Absorption

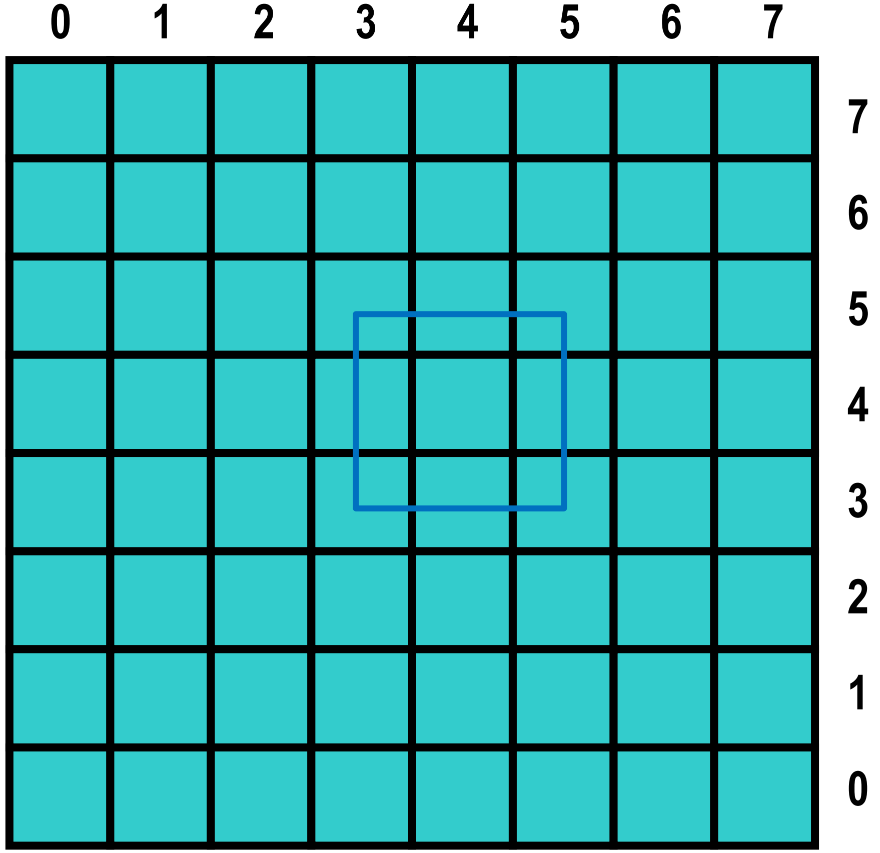
# Previous Works on Layout Synthesis for Quantum Computing

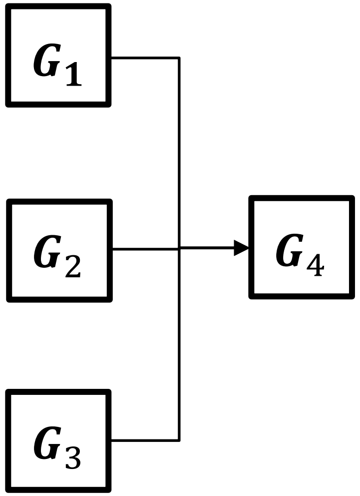
- **Layer-by-layer:**
  - [Maslov et al., TCAD’08], [Zulehner et al., DATE’18]: lookahead search guided by heuristic cost function
  - [Shafaei et al., ASPDAC’14]: optimize the ‘total distance’
- **Gate-by-gate:**
  - [Siraichi et al., CGO’18]: heuristic search for min #SWAPs
  - [Wille et al., DAC’19]: optimize #SWAPs
- **Use dependency:**
  - [Murali et al., ISCA’19]: optimize fidelity upper bound
  - [Li et al., ASPLOS’19]: bi-directional search with cost function concerning both #SWAPs and depth
- **Industry tools: Quilc, Qiskit, t|ket>, Cirq, …**

**Are they good enough?**

# Similar Questions Have Been Asked Before for VLSI Designs

E.g. Circuit placement (an NP-hard problem)





- **Bounding box wire-length**
- $\max_G x_G + \max_G y_G - \min_G x_G - \min_G y_G$
- **5+5-3-3 = 4**

# Construction of Placement Examples with Known Optimal (PEKO) Wirelength [Chang et al., TCAD'04]

- **Up to 2 million placeable objects**
  - *Initial WL gap: 1.6x – 2.5x (2003)*
- **Multiple EE Times articles coverage, e.g.**
  - Placement tools criticized for hampering IC designs [Feb'03]
- **Many downloads from our website**
  - Cadence, IBM, Intel, Magma, Mentor Graphics, Synopsys, …
  - CMU, MIT, SUNY, UCB, UCSB, UCSD, UIC, UMichgan, UWaterloo, …
- **Optimality gap on PEKO was narrowed down to ~20% as of 2007 (from 60% - 150%)**
- **Improvement on real circuits as well**
  - 30+% improvement by mPL placer 2003-06

CMP United Business Media

EETIMES

THE INDUSTRY SOURCE FOR ENGINEERS & TECHNICAL MANAGERS WORLDWIDE

Visit TheWorkCircuit.com today! click here

search go

DEPARTMENTS
- SEMICONDUCTORS
- SYSTEMS & SOFTWARE
- EE DESIGN
- ADVANCED TECHNOLOGY
- THE WORK CIRCUIT
- COMMSDESIGN
- PLANET ANALOG
- EMBEDDED.COM
- iAPPLIANCEWEB

HOT LINKS
- Books

TOP STORY

Updated Wed, 05 Feb 2003 11:26:16 EST

Placement tools criticized for hampering IC designs

Current IC placement algorithms leave so much wire unused that chip designs are essentially several technology generations behind where they could be, according to Jason Cong (left) of the VLSI CAD lab at UCLA. Placement tool vendors disagree with his findings.

FULL STORY ...

LATEST NEWS
- Broadband modem market to decline through 2008, says report.
- Intel poised to roll cellular baseband with flash

http://cadlab.cs.ucla.edu/~pubbench

# Quantum Mapping Examples with Known Optimal (QUEKO)

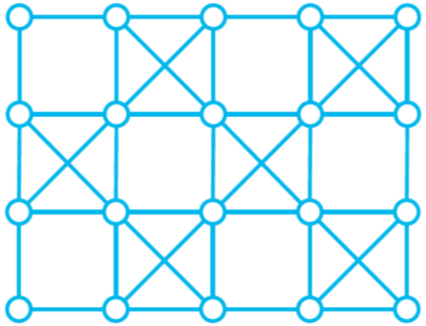
## QUEKO: depth and gate count optimal benchmarks tailored to arbitrary devices for LSQC

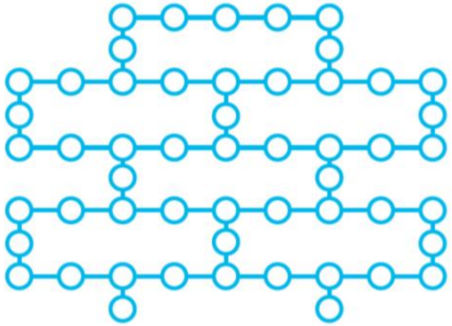
- Input: device graph, target depth, gate density
- Backbone construction: grow a dependency chain
- Sprinkling: match the gate density profile
- Scrambling: challenge the LSQC tools
- Output: OpenQASM file

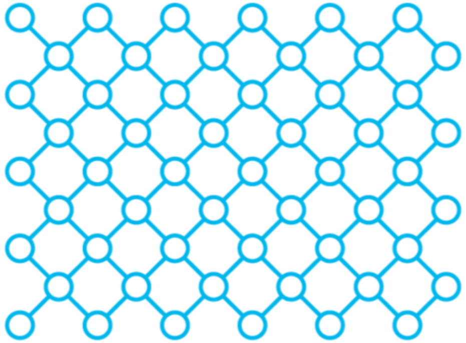
# Evaluating Existing LSQC Tools with QUEKO

- **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 [Arute et al., Nature'19]
- **Tools:**
  - **Cirq (Google)**
  - **Qiskit (IBM)**
  - **tket (Cambridge QC, now Quantinuum)**
  - **[Zulehner et al., DATE'18]**

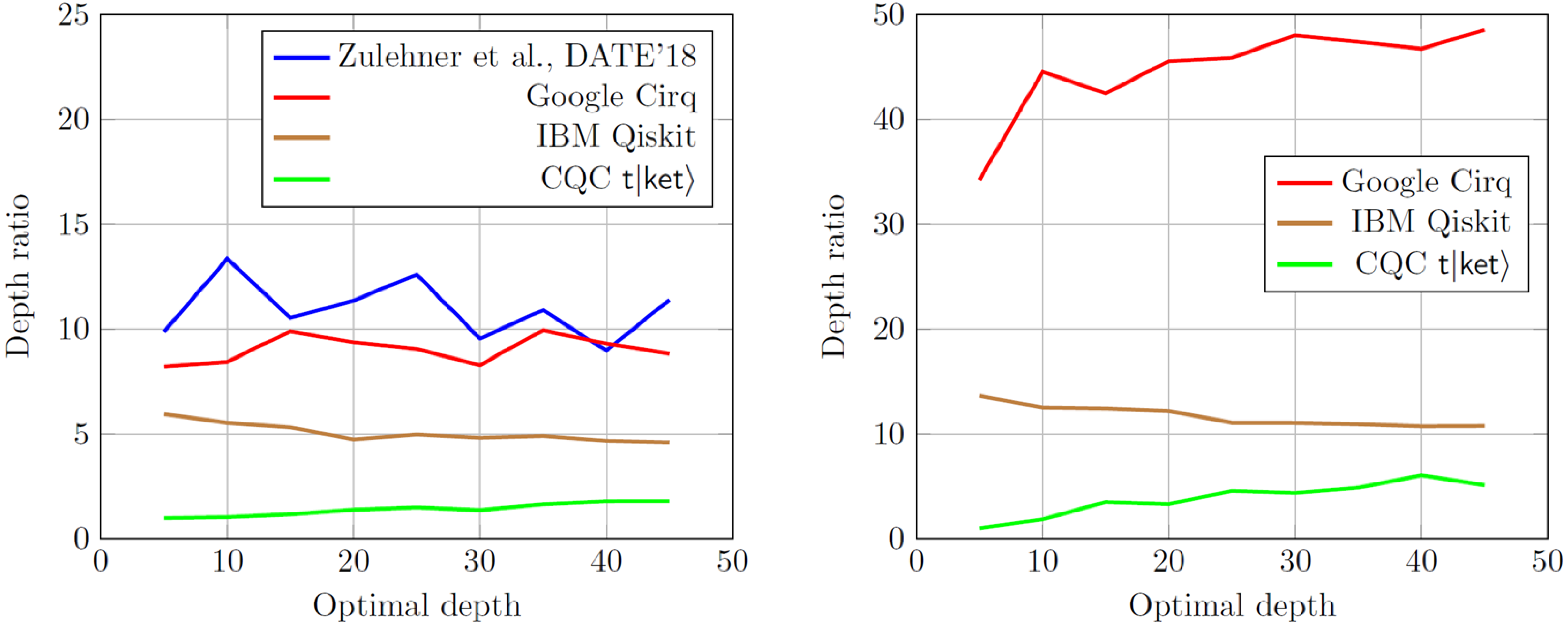

(b) Rigetti's Aspen-4 device graph


(c) IBM's Tokyo device graph


(d) IBM's Rochester device graph


(e) Google's Sycamore device graph

# QUEKO Results: Near-Term Feasible



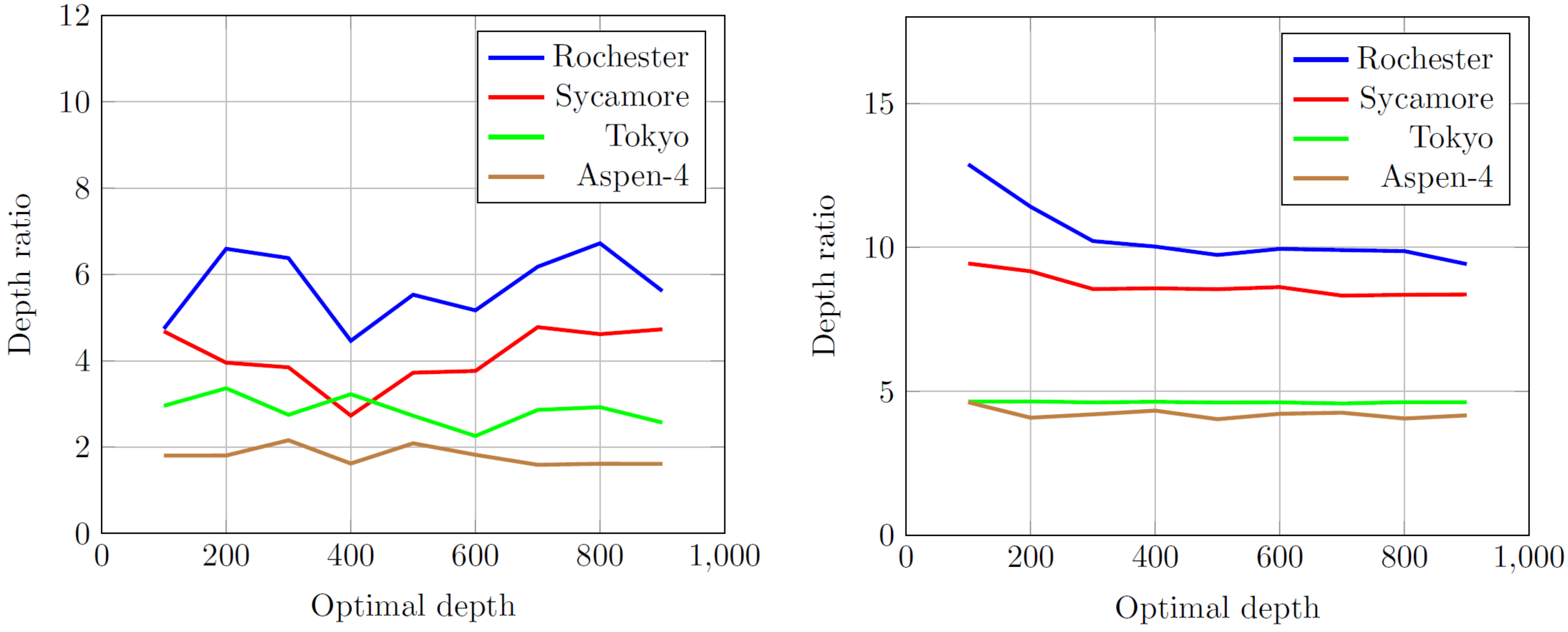
Optimality Gaps of Several Layout Synthesis Tools Revealed by $\mathrm{B_{NTF}}$ QUEKO Benchmarks

Toffoli gate density
Rigetti Aspen-4 Device

Quantum supremacy experiment gate density
Google Sycamore device

# QUEKO Results: Scalability Study

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



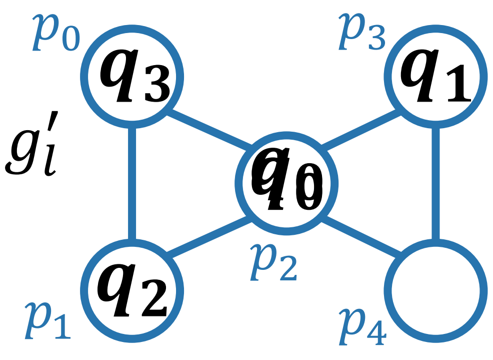
CQC t|ket⟩ Performance

IBM Qiskit Performance

# Outline

- Introduction
- Gap analysis for quantum compilation
- <u>Optimal layout synthesis for quantum computing (OLSQ)</u>
- OLSQ with Gate Absorption

# Large Solution Space of LSQC



- [Wille et al., DAC'19]: $L_2$ two-qubit gates, $M$ logical qubits, $N$ physical qubits
  - Mapping $x_{pq}^{l} = 1$ iff. logical qubit $q$ is mapped to physical qubit $p$ before gate $g_l'$
  - $2^{L_2 MN}$ search space for mapping
  - Permutation variables $y_{\Pi}^{l} = 1$ iff. before gate $g_l'$, qubits have a permutation $\Pi$
  - $L_2 N!$ permutation variables. Needs to pre-compute min cost of each $\Pi$.
  - The example of adder below: >1,500 variables

The SWAP insertion $y_{\Pi}^{4}$ and $y_{\Pi}^{6} = 1$ where $\Pi: 0 \mapsto 0, 1 \mapsto 1, 2 \mapsto 3, 3 \mapsto 2$

For all other $\Pi$'s and $l$'s, $y_{\Pi}^{l} = 0$

Mapping of logical qubit $q_0$

$x_{2,0}^{1} = 1, \qquad x_{i,0}^{1} = 0 \;\; i \neq 2,$

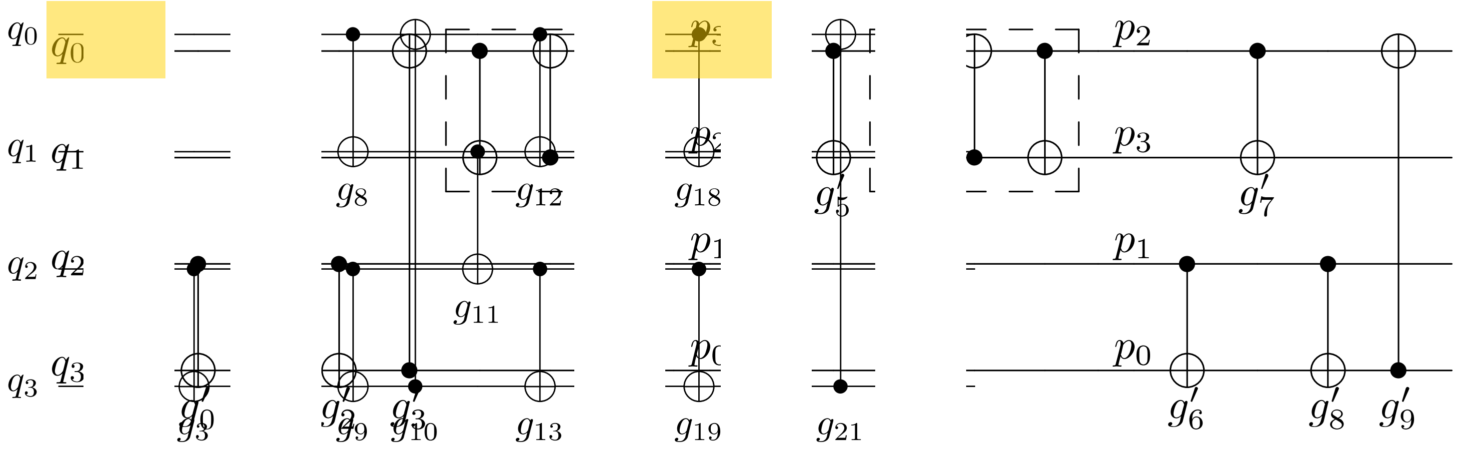
...

$x_{3,0}^{4} = 1, \qquad x_{i,0}^{4} = 0 \;\; i \neq 3,$

$x_{3,0}^{5} = 1, \qquad x_{i,0}^{5} = 0 \;\; i \neq 3,$

$x_{2,0}^{6} = 1, \qquad x_{i,0}^{6} = 0 \;\; i \neq 2,$

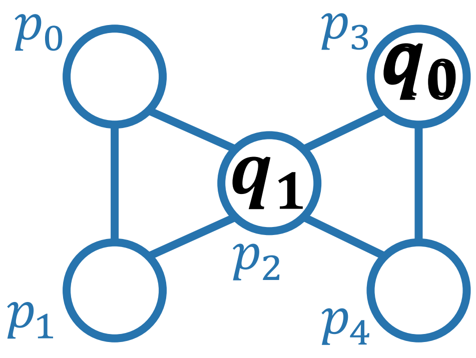
...



Input Circuit of Adder

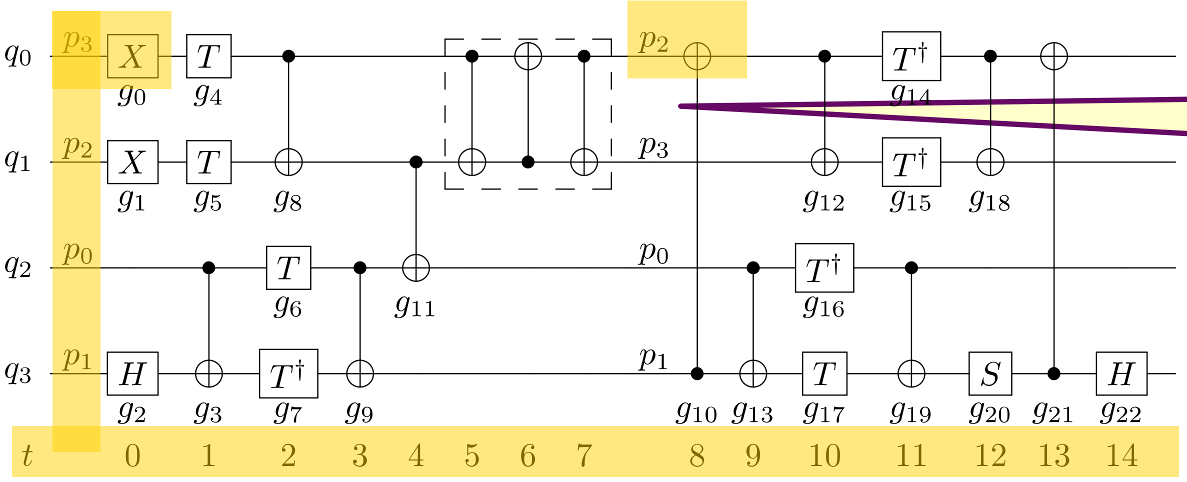
Result of [Wille et al., DAC'19]

# Our Approach: OLSQ (Optimal Layout Synthesis for Quantum Computing)



- **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 $\pi_q^t$: at time $t$, logical qubit $q$ is mapped to the physical qubit $\pi_q^t$
  - Use of SWAP $\sigma_e^t$: $\sigma_e^t = 1$ iff. there is a SWAP on edge $e$ and its last time step is $t$
  - More efficient encoding of search space*: $N^{MT}$

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

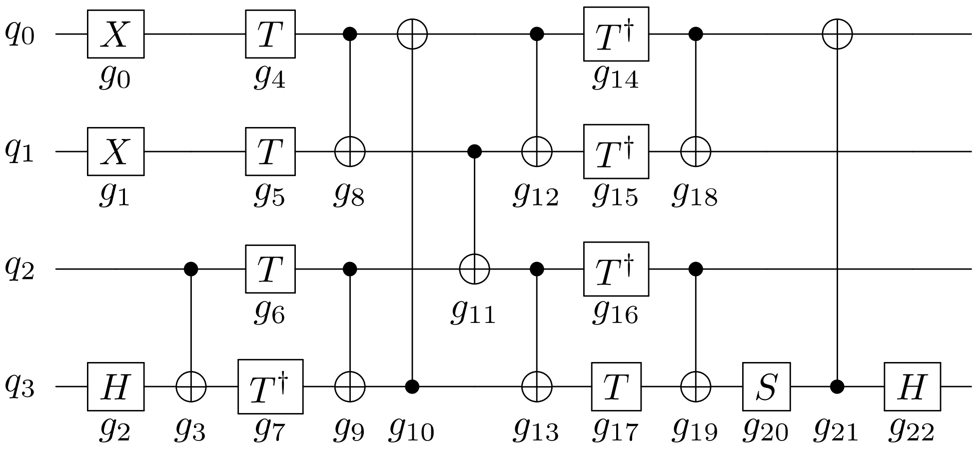
For all other $e$'s and $t$'s: $\sigma_e^t = 0$

*$N$ physical qubits,
$M$ logical qubits,
$T$ time steps

# OLSQ: Constraints

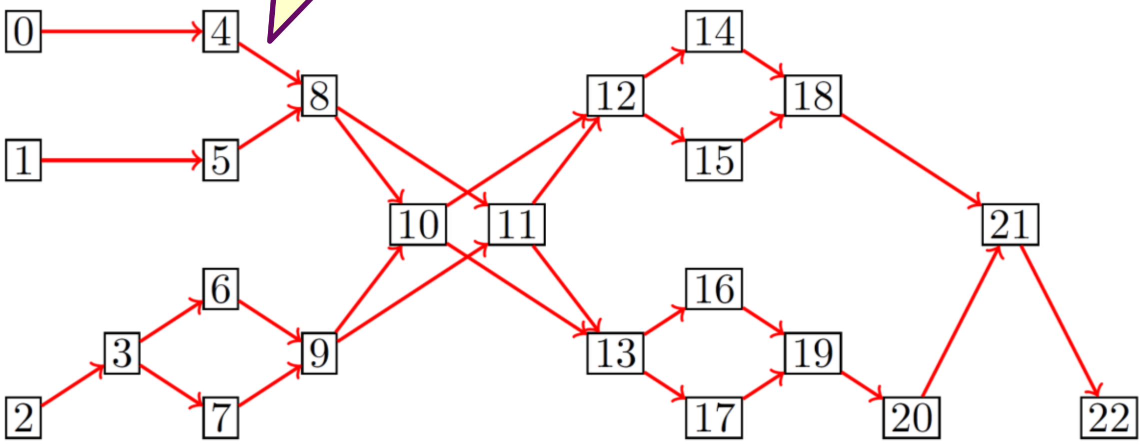
- **Validity**
  - Valid mapping targets: $\forall t, q \quad \pi_q^t \in P$ (all nodes in the coupling graph G)
  - Valid time coordinates: $\forall l \quad 0 \leq t_l < T$ (increase $T$ if no solution)
  - Valid space coordinates: if $g_l$ is a single-qubit gate, $x_l \in P$ ; if a two-qubit gate, $x_l \in E$ (all edges in G)
- **Dependencies:** if $g_l$ depends on $g_{l'}$, then $t_l > t_{l'}$
- **Injective mapping:** $\forall t, q, q' \quad q' \neq q \Rightarrow \pi_q^t \neq \pi_{q'}^t$

- $O(NT)$
  - $O(NT)$
  - $O(L)$
  - $O(L)$
- $O(NT)$
- $O(N^2T)$



Input Circuit of Adder

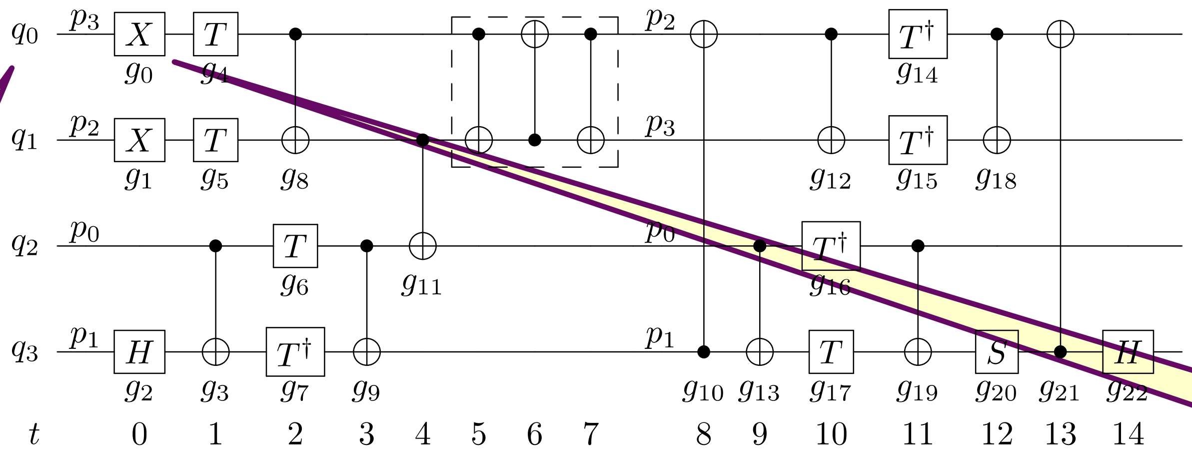
$g_8$ depends on $g_4 \ \Rightarrow t_8 > t_4$



# OLSQ: Constraints

- **Mapping implies spacetime coordinates**
  - $(t_0 = 0 \wedge x_0 = p_3) \Rightarrow (\pi_0^0 = p_3)$

- $O(NTL)$



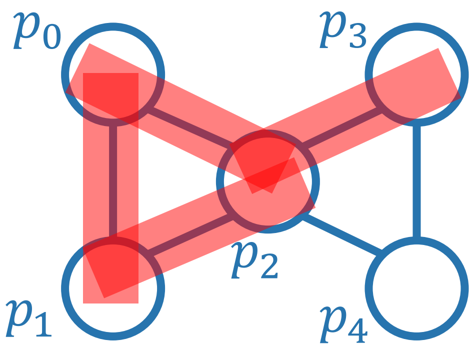
Mapping of logical qubit $q_0$
$\pi_0^0 = 3$

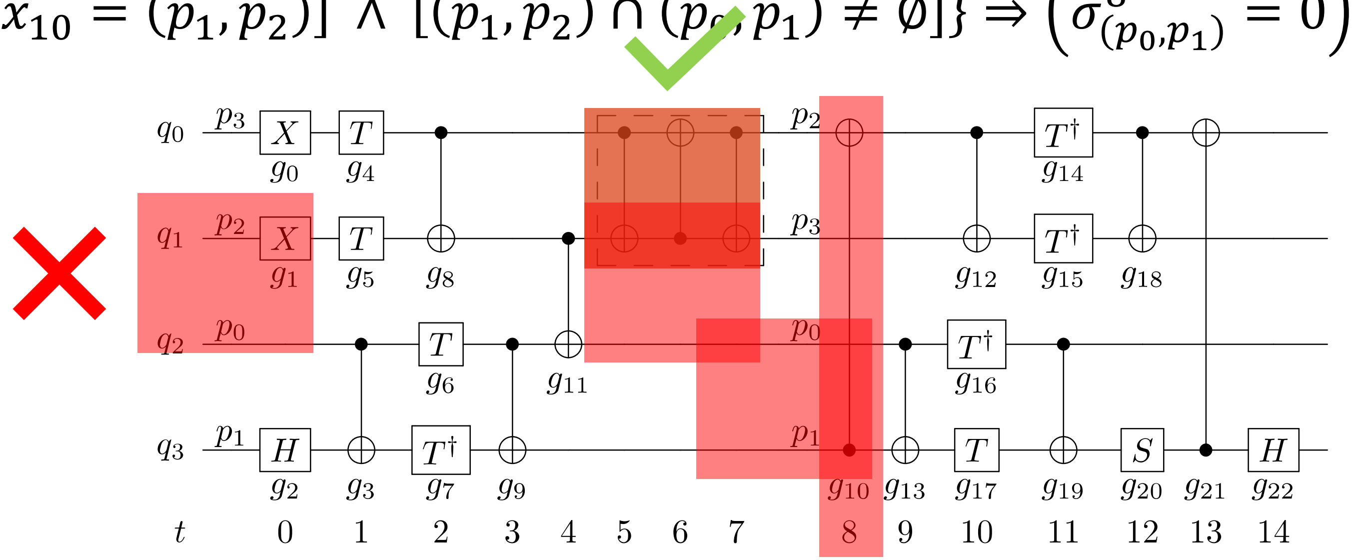
$g_0$ acts on $q_0$, and $q_0$ is mapped to $p_3$ at time 0

Spacetime coordinates for $g_0$ is $t_0 = 0$ and $x_0 = p_3$

# OLSQ: Constraints

- **Legal SWAPs:**
  - Initial SWAP conditions
    - ❖$0 \leq t < 3, \forall e \quad \sigma_e^t = 0$
  - No overlaps between SWAPs
    - ❖$\left(\sigma_{(p_2,p_3)}^7 = 1\right) \Rightarrow \left(\sigma_{(p_2,p_0)}^7 = 0\right)$
  - No overlaps between SWAPs and original gates
    - ❖$\{(t_{10} = 8) \wedge [x_{10} = (p_1, p_2)] \wedge [(p_1, p_2) \cap (p_0, p_1) \neq \emptyset]\} \Rightarrow \left(\sigma_{(p_0,p_1)}^8 = 0\right)$



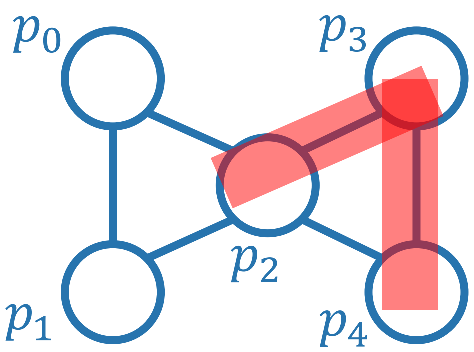


- $O(NTL)$
  - $O(N)$
  - $O(NT)$
  - $O(NTL)$

# OLSQ: Constraints

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

- $O(N^2T)$
  - $O(N^2T)$
  - $O(N^2T)$

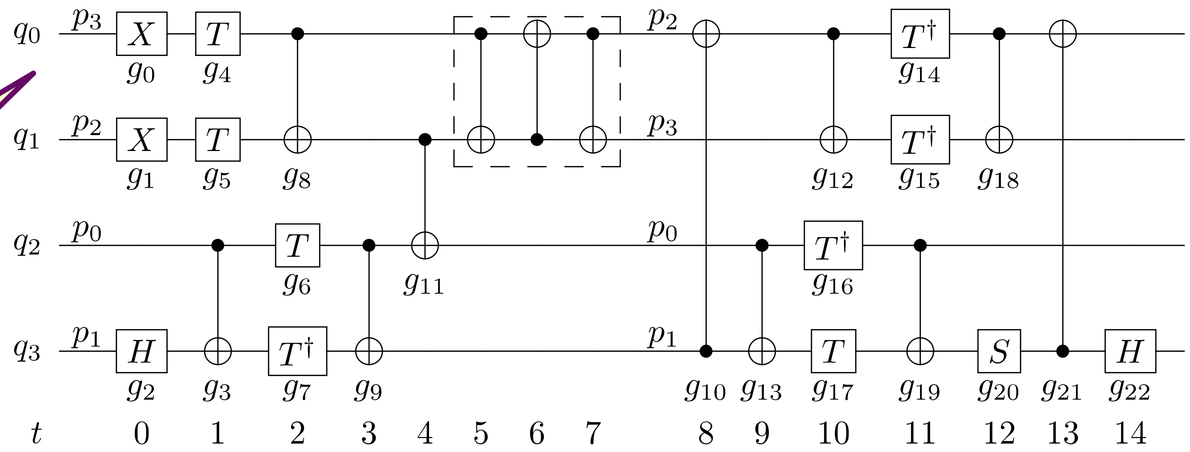


There is a SWAP on $(p_2, p_3)$ finishing at time 7.

Mapping of $q_0$ changes at time 8.

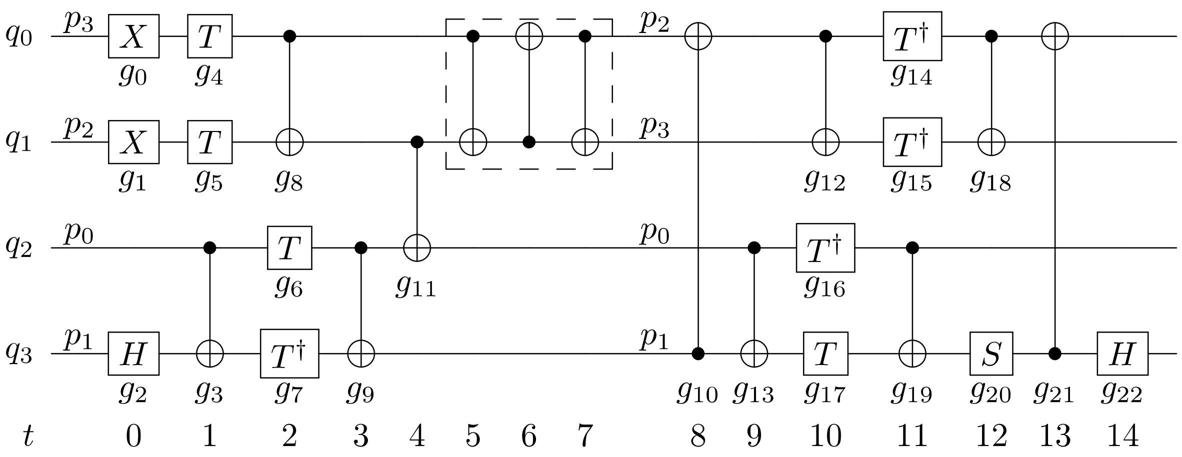
There are no SWAPs ending at time 0 on any edge connecting $p_3$.

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



# OLSQ Optimization Objectives

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



Result of OLSQ

3/9/22

# Solved Using SMT (Satisfiability Modulo Theories)

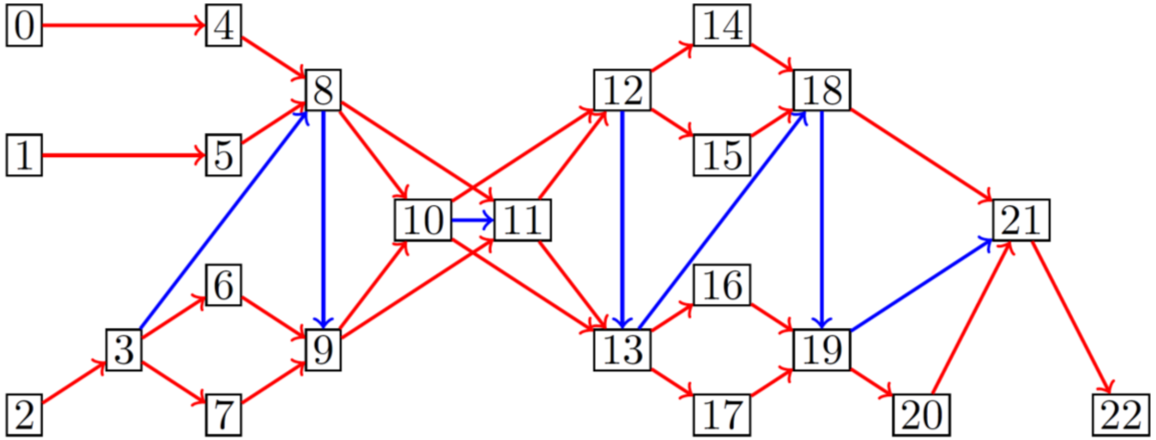
- **SAT (Boolean Satisfiability): given a conjunctive normal form, whether there is an assignment such that it is true. E.g.,**
  - $\boldsymbol{a} \land (\overline{\boldsymbol{a}} \lor \boldsymbol{b}) \land \boldsymbol{c}$
  - $\boldsymbol{Solution}$: $\boldsymbol{a} = \boldsymbol{b} = \boldsymbol{c} = \mathbf{True}$
- **SMT generalizes SAT to more complex formulas involving real numbers, integers, lists, arrays, bit-vectors, etc. E.g.**
  - $\boldsymbol{a} \coloneqq \boldsymbol{x} + \boldsymbol{y} < \mathbf{3},\ \boldsymbol{b} \coloneqq \boldsymbol{x} < \mathbf{4} - \boldsymbol{y},\ \boldsymbol{c} \coloneqq \boldsymbol{x} > \mathbf{0}.$
  - Then, $\boldsymbol{x} = \boldsymbol{y} = \mathbf{1}$ makes the model satisfiable.
- **SMT is very expressive, widely used in compilation, programming language, formal verification, etc.**
- **There are efficient SMT solvers, such as Z3 (and we can further customize for OLSQ)**

# Summary of Constraints for OLSQ

| Constraints | OLSQ |
|---|---|
| Validity | $O(NT)$ |
| Injective Mapping | $O(N^2T)$ |
| Dependency | $O(NT)$ |
| Mapping constrains Spacetime Coordinates | $O(NTL)$ |
| No Overlap with Other SWAPs | $O(NT)$ |
| No Overlap with Original Gates | $O(NTL)$ |
| Mapping transformed by SWAPs | $O(N^2T)$ |
| In total | $O(NTL)$ |

# Key Advantages of OLSQ

- **Efficiency: $O(NT)$ vars & $O(NTL)$ constraints versus $O(L_2N!)$ & $O(L_2MN!)$**
- **Complexity result: a polynomial certificate → quantum LS is in NP**
- **Optimality: independent from input gate order**

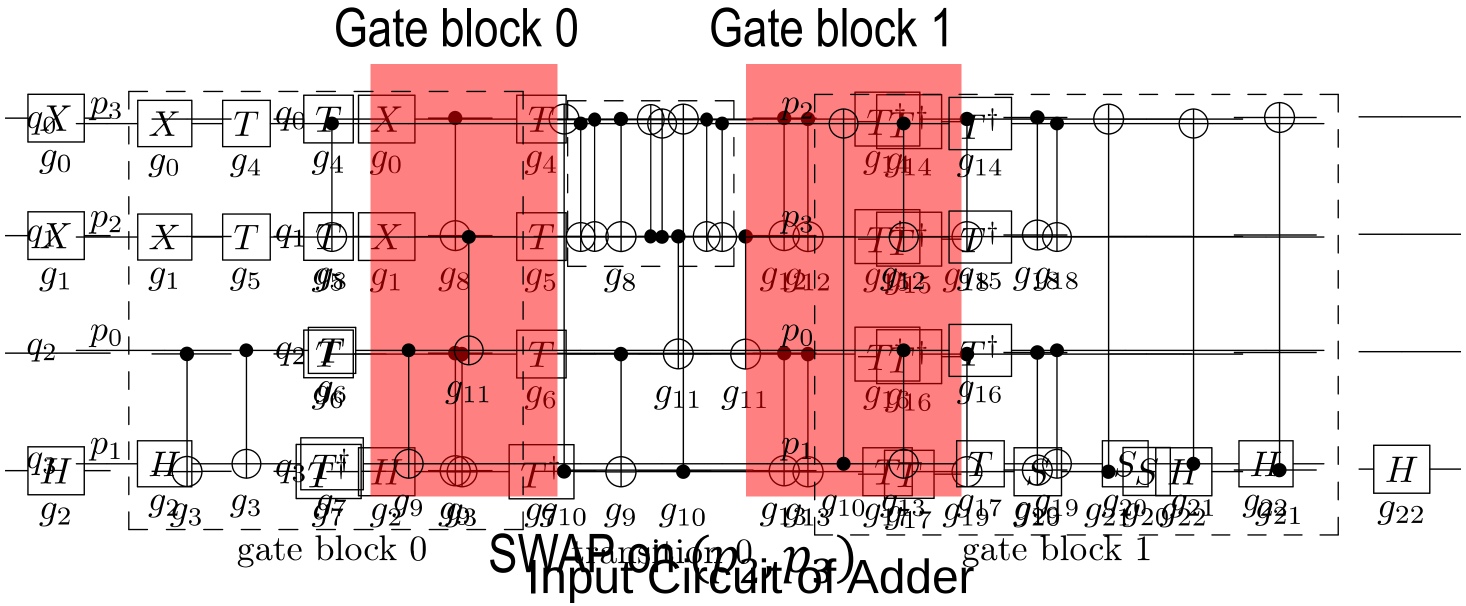


Side-effect of gate-by-gate processing in [Wille et al., DAC'19]

Dependencies enforced by OLSQ

# Transition-Based 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



# Constraints for TB-OLSQ

| Constraints | TB-OLSQ Revision |
|---|---|
| Validity | |
| Injective Mapping | |
| Dependency | |
| Mapping constrains Spacetime Coordinates | |
| No Overlap with Other SWAPs | |
| No Overlap with Original Gates | |
| Mapping transformed by SWAPs | |

# Summary of Constraints for TB-OLSQ

| Constraints | OLSQ | [Wille et al., DAC'19] |
|---|---|---|
| Validity | $O(NT)$ | |
| Injective Mapping | $O(N^2T)$ | |
| Dependency | $O(NT)$ | |
| Mapping constrains Spacetime Coordinates | $O(NTL)$ | |
| No Overlap with Other SWAPs | $O(NT)$ | |
| No Overlap with Original Gates | $O(NTL)$ | |
| Mapping transformed by SWAPs | $O(N^2T)$ | |
| In total | $O(NTL)$ | $O(L_2MN!)$ |

# TB-OLSQ Evaluation

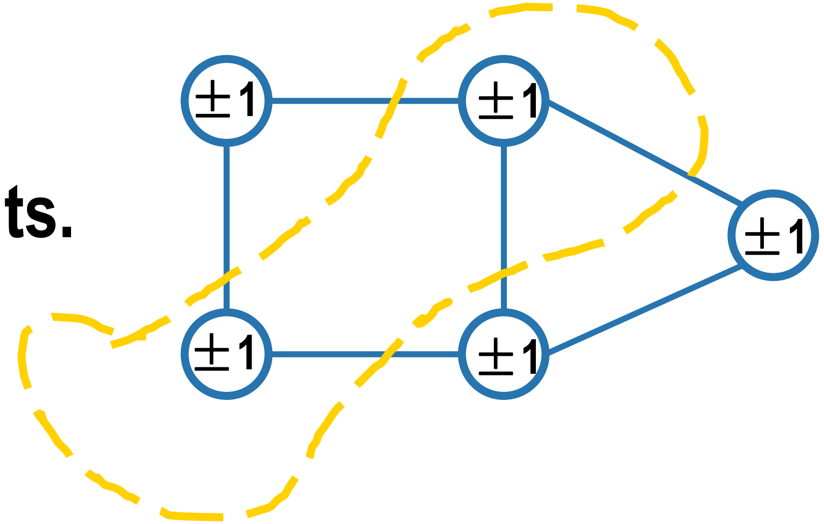
- Comparison with OLSQ >400x speedup (geomean)

| Benchmarks |
|---|
| Small circuits to verify optimality |
| Larger arithmetic circuits |
| QUEKO circuits |

- A more recent work [Zhang et al., ASPLOS'21] uses A* search with an admissible heuristic, which runs faster with depth-optimal solutions (but cannot optimize other objectives, e.g. fidelity).

* TriQ is from [Murali et al., ISCA'19]

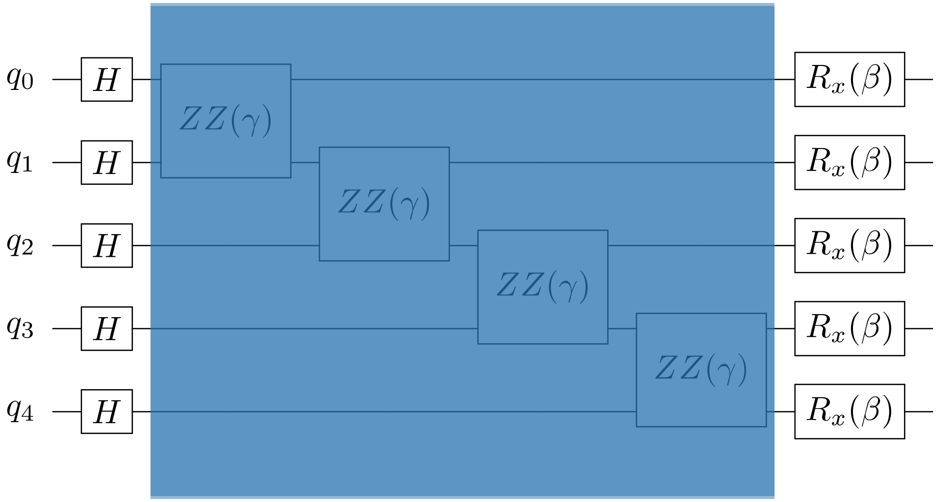
# Quantum Approximate Optimization Algorithm (QAOA)

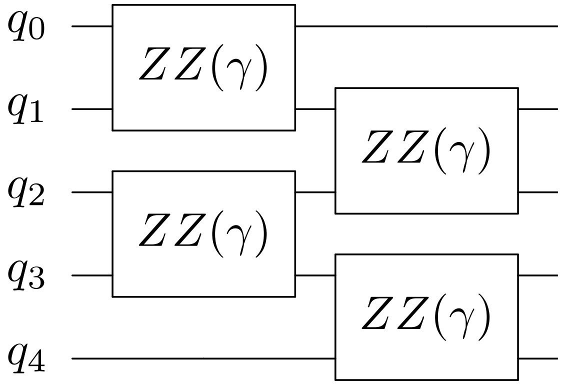


- **Aiming optimization with binary variables**
- **Quantize the problem by changing variables to qubits.**
- **Example: MAX-CUT problem on $G = (V, E)$**
- **Assign $\pm 1$ variables $z_i$ to vertices**
- **MAX-CUT = Maximize $\sum_{(v_j, v_k) \in E} \frac{1 - z_j z_k}{2}$**
- **$z_j z_k$ has a corresponding two-qubit gate, ZZ-Phase.**

# The ZZ-Phase Gate for Each Edge

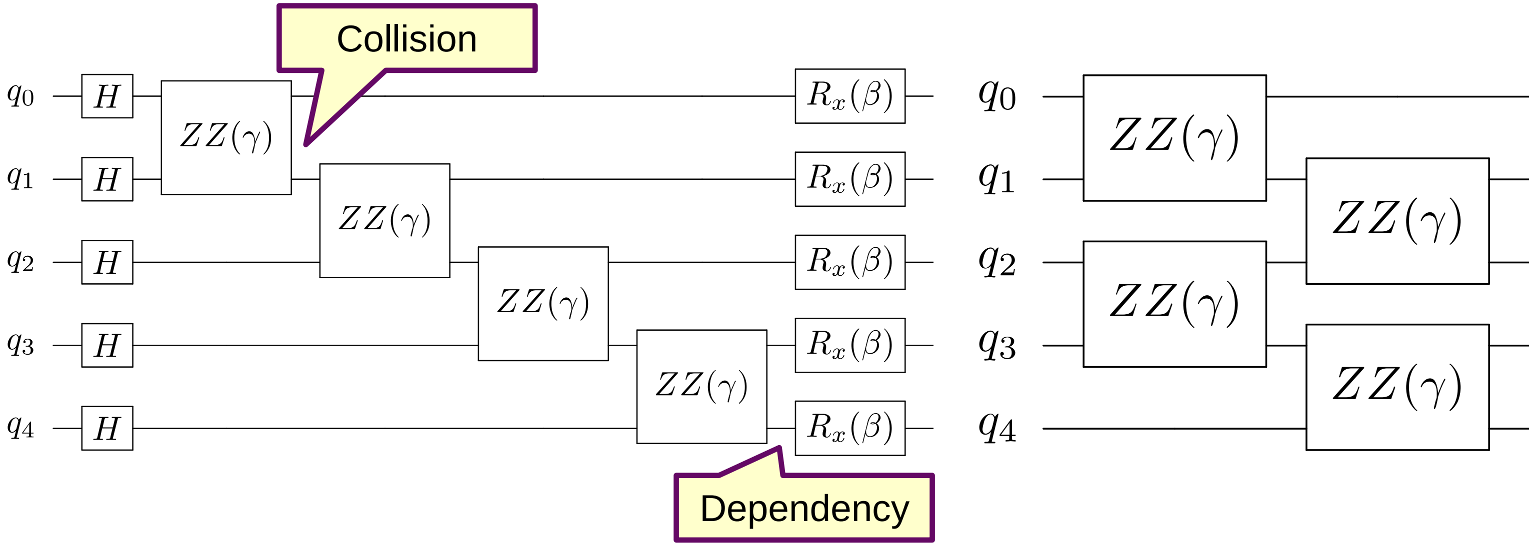
- $\begin{pmatrix} e^{-i\gamma} & 0 & 0 & 0 \\ 0 & e^{i\gamma} & 0 & 0 \\ 0 & 0 & e^{i\gamma} & 0 \\ 0 & 0 & 0 & e^{-i\gamma} \end{pmatrix}$
- **Commutable, i.e., AB=BA, since diagonal**





# QAOA-OLSQ

- **Observation:** some ‘dependencies’ are not real, according to commutation.



- **Solution:** make a distinction between dependency and collision
- **Result:** 70% depth reduction, 54% SWAP reduction compared to tket.

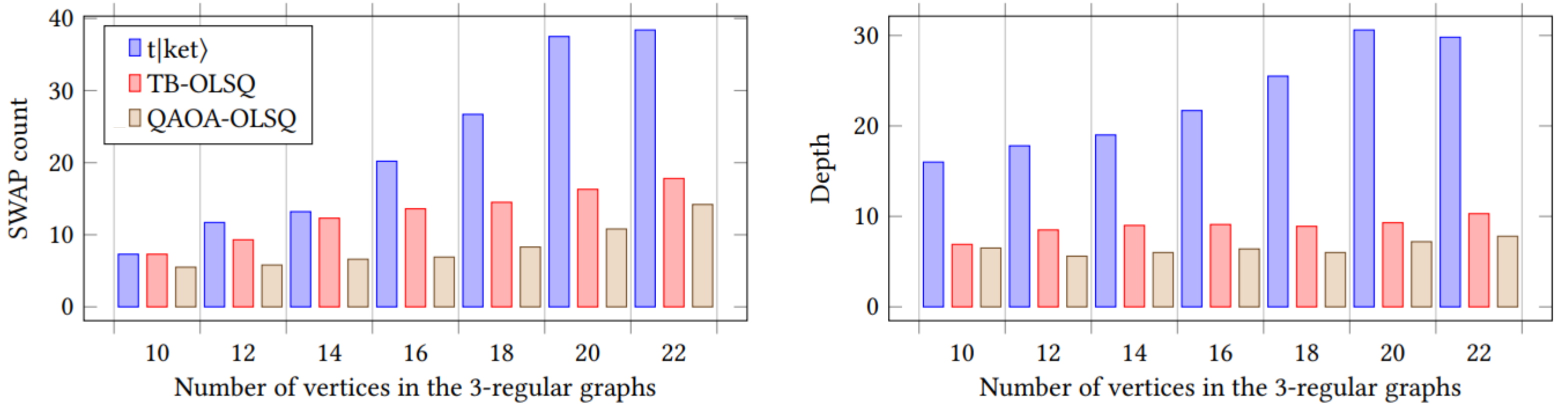


**Figure 9.** Evaluation of QAOA-OLSQ

# Outline

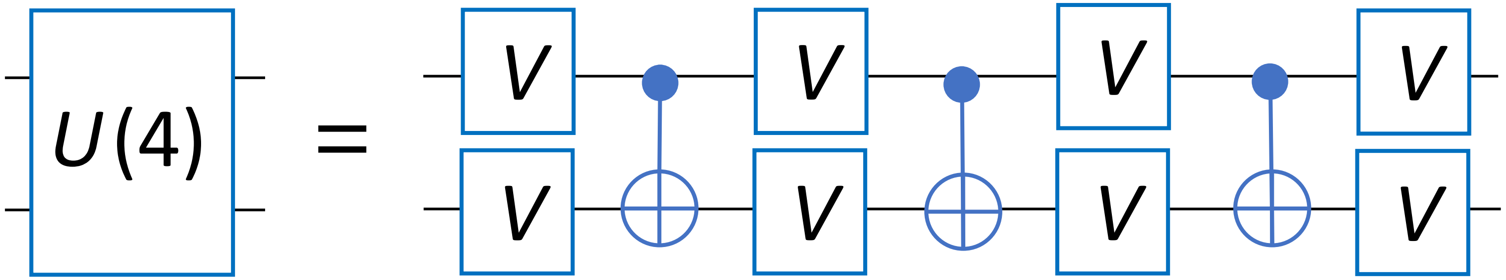
- Introduction
- Gap analysis for quantum compilation
- Optimal layout synthesis for quantum computing (OLSQ)
- <u>OLSQ with Gate Absorption</u>

# Programmable Single-Qubit Gate -> Programmable Two-Qubit Gate

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

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

- Native two-qubit gate: CX
- A programmable two-qubit gate can be configured to any matrix in $U(4)$
- KAK Decomposition [Vatan&Williams, PRA'04]: any $U(4)$ to 3 CX's and some $U(2)$

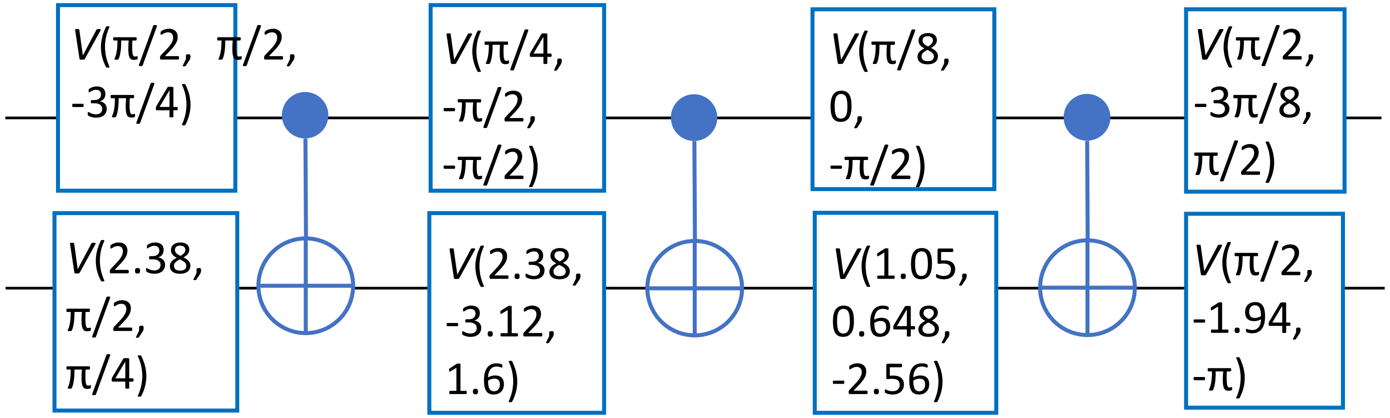


# Quantum Programs as Lists of Programmable Two-Qubit Gates

- Driving applications
  - Chemistry simulation [Kivlichan et al., PRL’18]
  - Quantum Approximate Optimization Algorithm (QAOA) [Farhi et al., arXiv’14, Harrigan et al., NatPhys’21]
  - Quantum neural networks (QCNN) [Cong et al., NatPhys’19]
- …

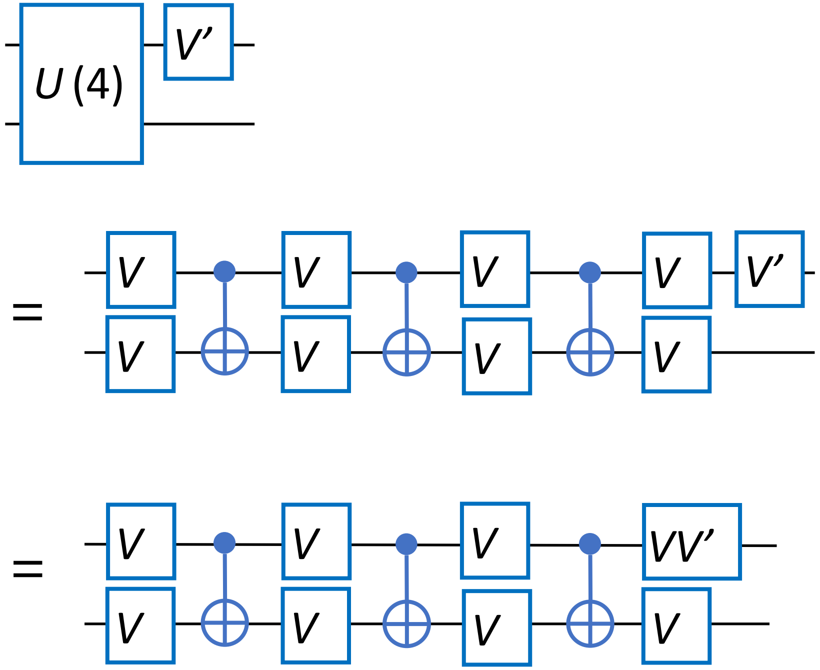
$$\mathrm{fSim}(\theta,\phi)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & -i\sin\theta & 0\\ 0 & -i\sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & e^{-i\phi}\end{bmatrix}$$

$\mathrm{fSim}(\frac{\pi}{4},\frac{\pi}{4})$ can be decomposed as

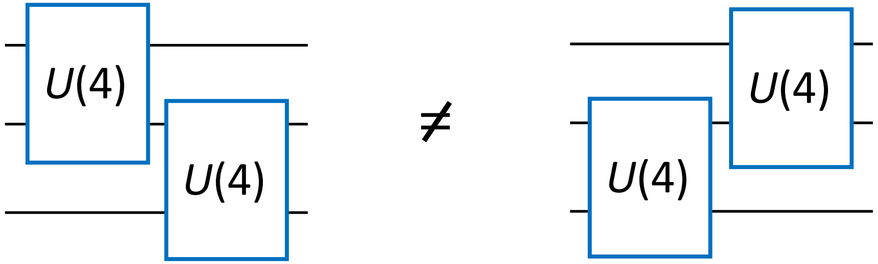


# Take Advantage of Gate Absorption and Commutation

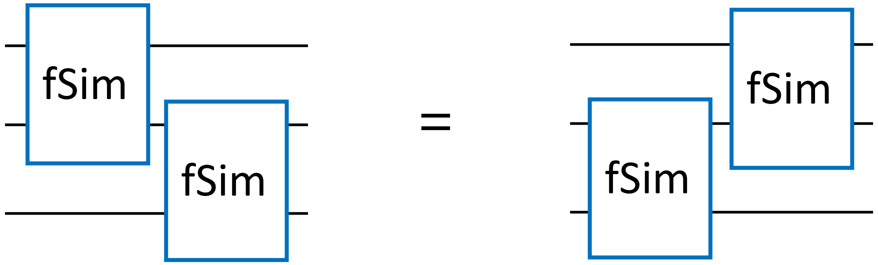
- Gate absorption



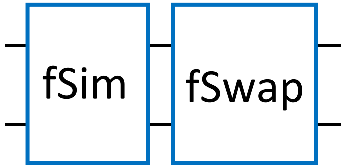
- Dependency: relative order of the gates

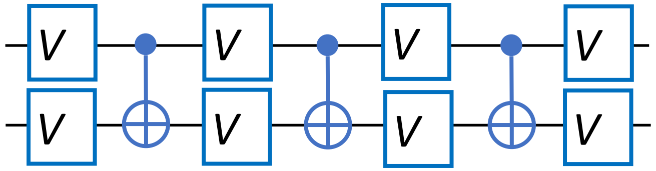


- Gate commutation:



# Two-Qubit Gate Absorption





$$\left\{\begin{array}{l}\mathrm{fSim}(\theta,\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -i\sin\theta & 0 \\ 0 & -i\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{bmatrix} \\[2em] \mathrm{fSwap} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\end{array}\right.$$

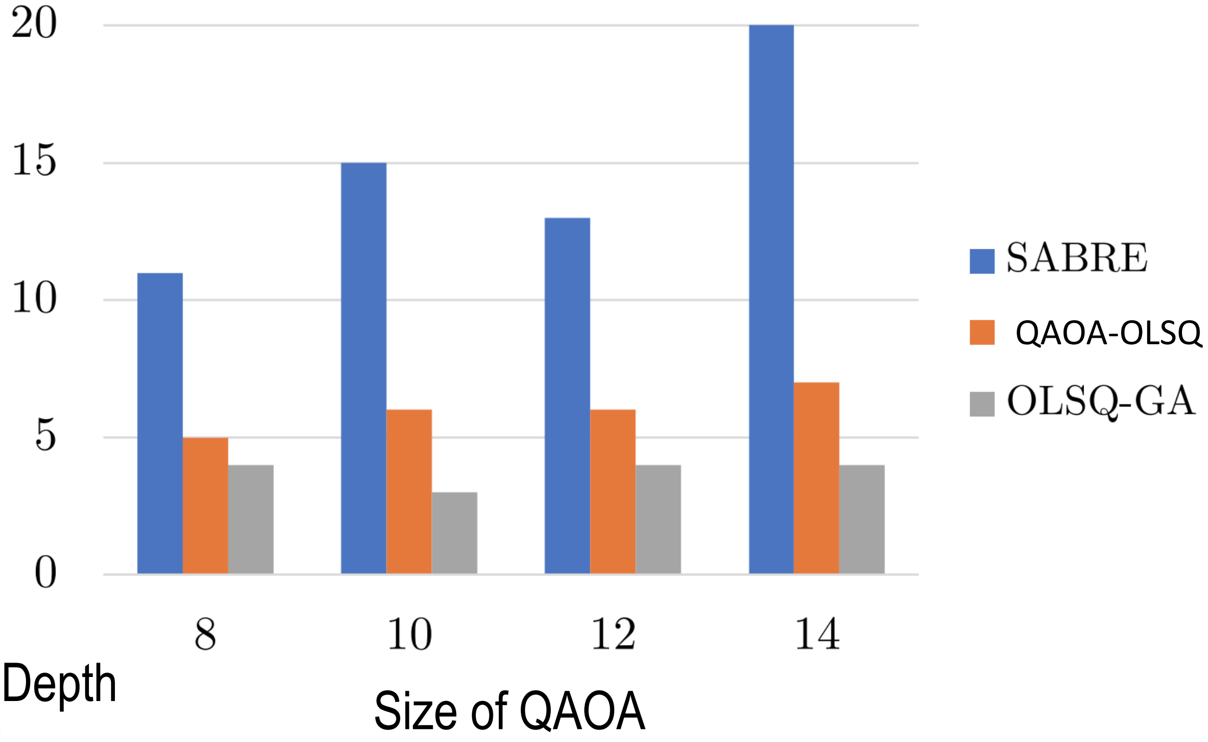
$$\mathrm{fSwap} \cdot \mathrm{fSim} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -i\sin\theta & \cos\theta & 0 \\ 0 & \cos\theta & -i\sin\theta & 0 \\ 0 & 0 & 0 & -e^{-i\phi} \end{bmatrix}$$

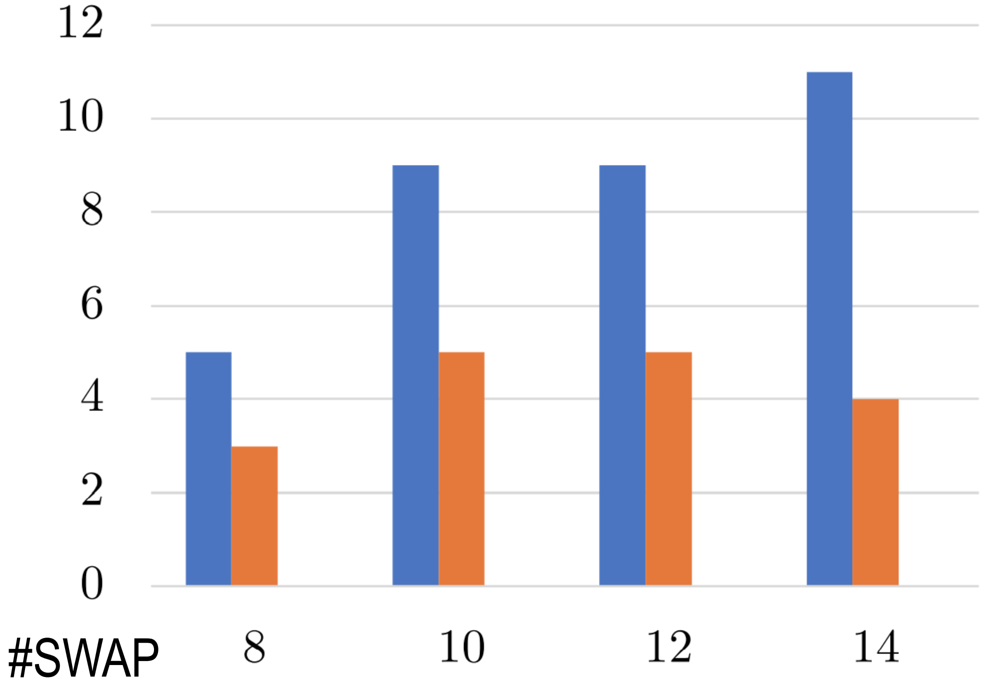
Formulation [Tan&Cong, ICCAD’21]:

- 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 $\alpha_e^t$ and $\sigma_e^t$

# QAOA Results with Gate Absorption (Depth and # Swap Gates)

- Similar QAOA instances of size 8 to 14 like in [Harrigan et al., NatPhys'21]
- SABRE [Li et al., ASPLOS'19]: leading heuristic mapper, recently adopted in Qiskit
- OLSQ-GA (considers commutation) reduced depth up to 80%, absorbed all the SWAPs

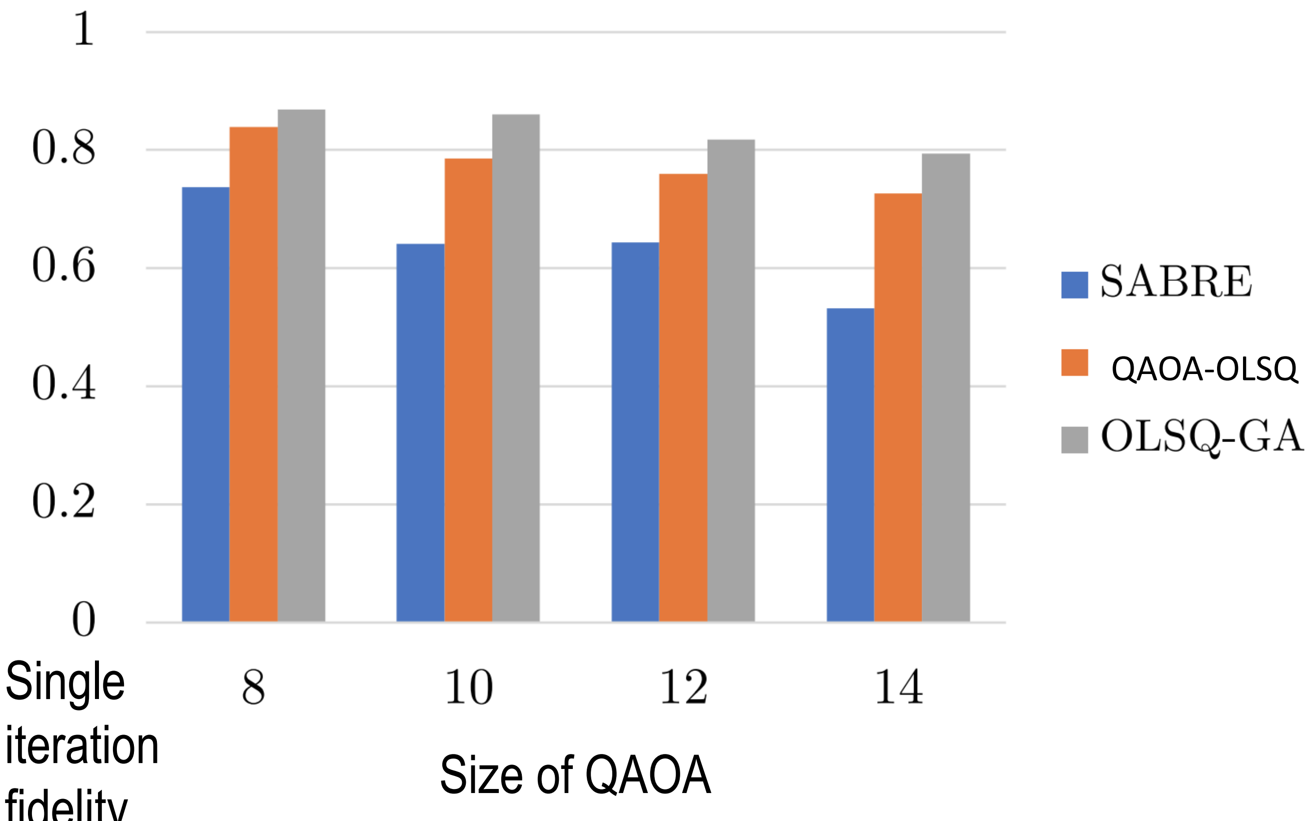




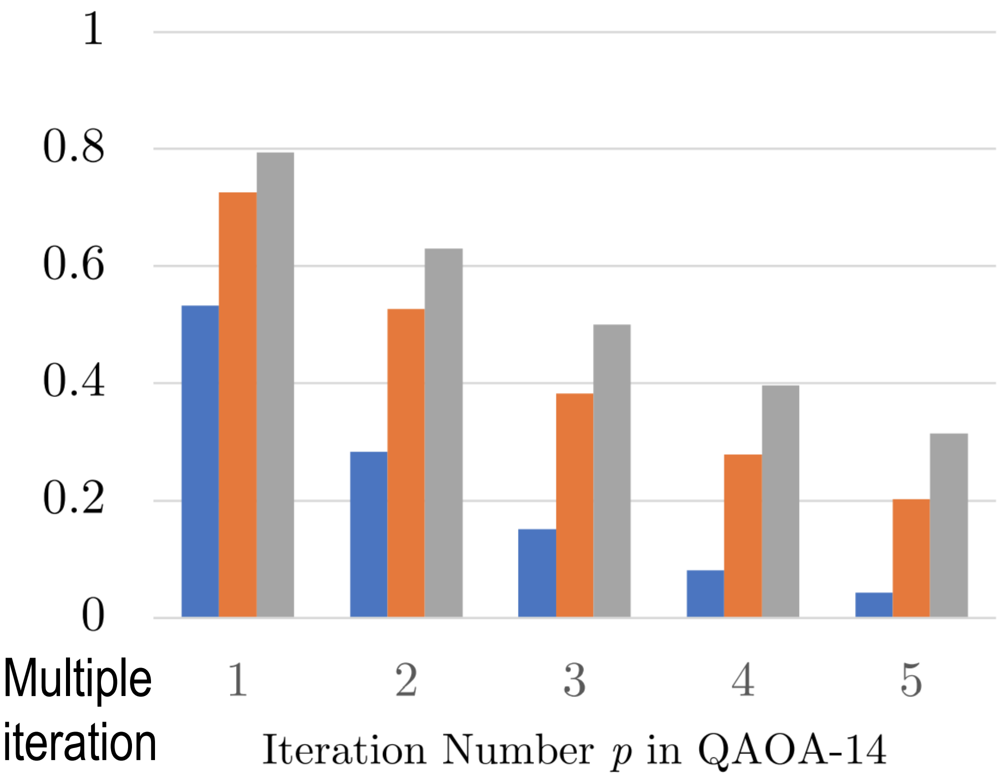
Size of QAOA (There are no gray bars since OLSQ-GA absorbed all the SWAPs)

# QAOA Results with Gate Absorption (Fidelity)

- Fidelity estimated with slightly optimistic parameters $T_0 = 50$ and $f_{U(4)} = 99\%$
- OLSQ-GA improves fidelity by up to 49% for 1 iteration, 636% for 5 iterations.



Single iteration fidelity



Multiple iteration fidelity

# Concluding Remarks

- **There are significant advances in quantum computing (QC) device technology**
- **There is a great need for better design automation or compilation tools for QC**
  - As measured by the QUEKO circuits
- **Optimization objectives for NISQ applications**
  - Circuit depth (decoherence time)
  - Overall fidelity
  - Scalability
- **OLSQ provides a framework for optimal solution for layout synthesis**
- **Further opportunities to combine layout synthesis with logic synthesis**
- **A lot of more opportunities for compilation/design automation for QC on novel architectures**

# Acknowledgements

***Supports from the Industrial Partners of the Center for Domain-Specific Computing (CDSC)***

# Reference

[Tan&Cong, TC’20] B. Tan and J. Cong, “Optimality Study of Existing Quantum Computing Layout Synthesis Tools,” *IEEE Transactions on Computers*, Jul. 2020, doi: 10.1109/TC.2020.3009140. arXiv:quant-ph/2002.09783

[Tan&Cong, ICCAD’20] 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

[Tan&Cong, ICCAD’21] 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, arXiv: 2109.06445

[Cong&Hwang, TCAD’01] Cong, J. and Yean-Yow Hwang. “Boolean Matching for LUT-Based Logic Blocks with Applications to Architecture Evaluation and Technology Mapping.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20, no. 9 (September 2001): 1077–90. https://doi.org/10.1109/43.945303.

[Li et al., FPGA’03] F. Li, D. Chen, L. He, and J. Cong, “Architecture Evaluation for Power-Efficient FPGAs,” in *Proceedings of the 2003 ACM/SIGDA Eleventh International Symposium on Field Programmable Gate Arrays*, New York, NY, USA, 2003, pp. 175–184, doi: 10.1145/611817.611844.

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

[Barends et al., Nature’14] 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.

J. Kelly, “A Preview of Bristlecone, Google’s New Quantum Processor,” Mar. 05, 2018. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html.

# Reference

[Gambetta et al., npjQI’17] 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.

[Jurcevic et al., arXiv’2008] P. Jurcevic *et al.*, “Demonstration of quantum volume 64 on a superconducting quantum computing system,” Aug. 2020.

[Chang et al., TCAD’04] 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.

[Arute et al., Nature’19] 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.

[Cowtan et al., TQC’19] 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.

[Cong et al., NatPhys’19] 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.

IBM, “IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation,” May 04, 2016. https://www-03.ibm.com/press/us/en/pressrelease/49661.wss.

D. McClure and J. Gambetta, “Quantum Computation center opens,” Sep. 18, 2019. https://www.ibm.com/blogs/research/2019/09/quantum-computation-center/.

# Reference

[Zulehner et al., DATE’18] 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.

[Wille et al., DAC’19] 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.

[Bhattacharjee et al., ICCAD’19] D. Bhattacharjee, A. A. Saki, M. Alam, A. Chattopadhyay, and S. Ghosh, “MUQUT: Multi-Constraint Quantum Circuit Mapping on NISQ Computers: Invited Paper,” in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Westminster, CO, USA, Nov. 2019, pp. 1–7, doi: 10.1109/ICCAD45719.2019.8942132.

[Murali et al., ISCA’19] 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.

[Farhi et al., arXiv’14] E. Farhi, J. Goldstone, and S. Gutmann, “A Quantum Approximate Optimization Algorithm,” *arXiv:1411.4028 [quant-ph]*, Nov. 2014, Accessed: Jan. 20, 2020. [Online]. Available: http://arxiv.org/abs/1411.4028.

[Vatan&Williams, PRA’04] 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.

[Zhang et al., ASPLOS’21] C. Zhang, A. B. Hayes, L. Qiu, Y. Jin, Y. Chen, and E. Z. Zhang, “Time-optimal Qubit mapping,” in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, Virtual USA, Apr. 2021, pp. 360–374. doi: 10.1145/3445814.3446706.

# Reference

[Harrigan et al., NatPhys’21] 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.

[Maslov et al., TCAD’08] 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.

[Shafaei et al., ASPDAC’14] Shafaei, Alireza, Mehdi Saeedi, and Massoud 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.

[Siraichi et al., CGO’18] Siraichi, Marcos Yukio, Vinícius Fernandes dos Santos, Sylvain Collange, and Fernando Magno Quintao 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.

[Kivlichan et al., PRL’18] 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.

[Li et al., ASPLOS’19] Li, Gushu, Yufei Ding, and Yuan 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.

J. Chow and J. Gambetta, “Quantum Takes Flight: Moving from Laboratory Demonstrations to Building Systems,” Jan. 08, 2020. https://www.ibm.com/blogs/research/2020/01/quantum-volume-32/.