VLSI architecture, synthesis & technology

UCLA

ML-QLS: Multilevel Quantum Layout Synthesis

Wan-Hsuan Lin, PhD, UCLA

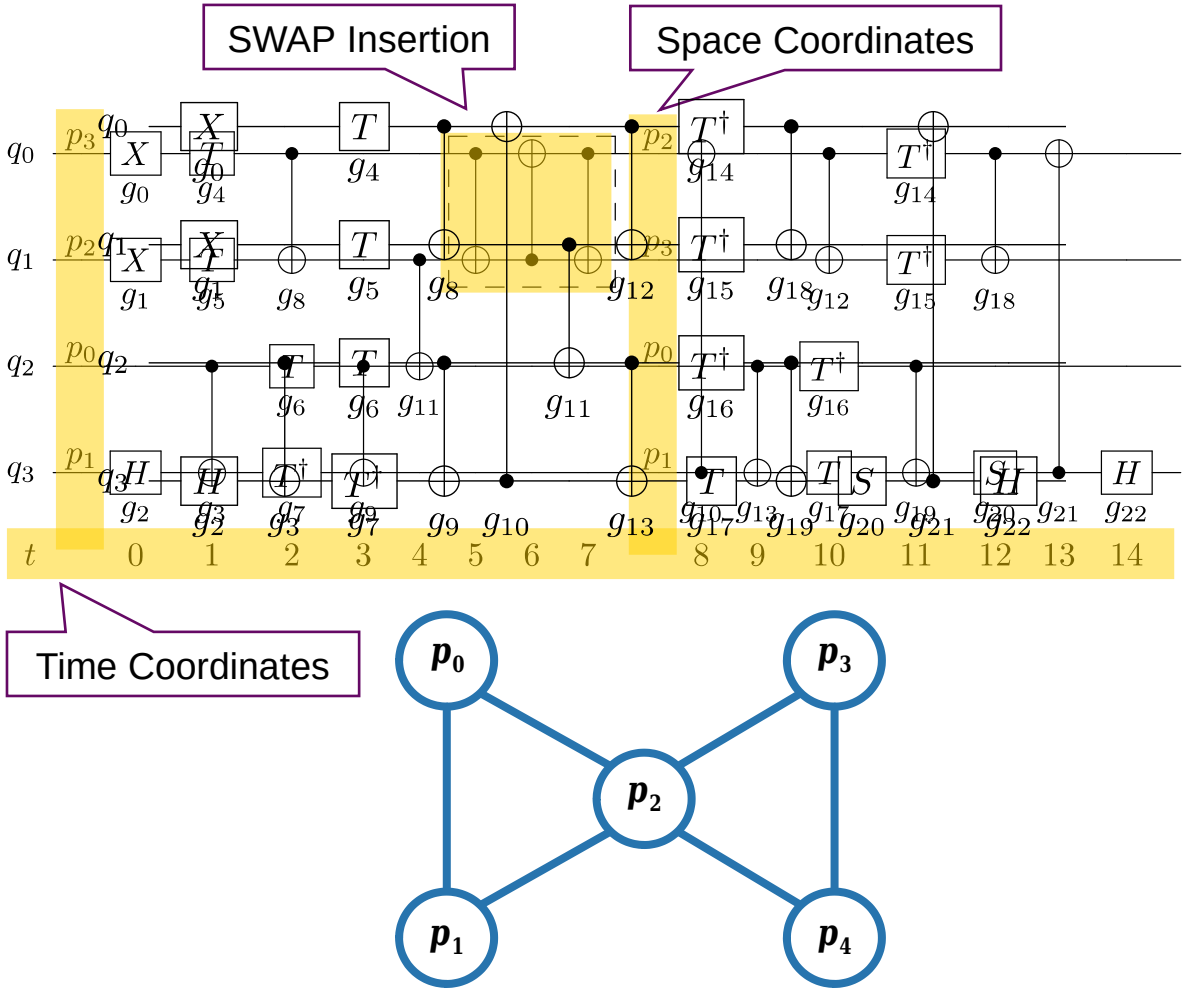
Advisor: Jason Cong UCLA

Overview

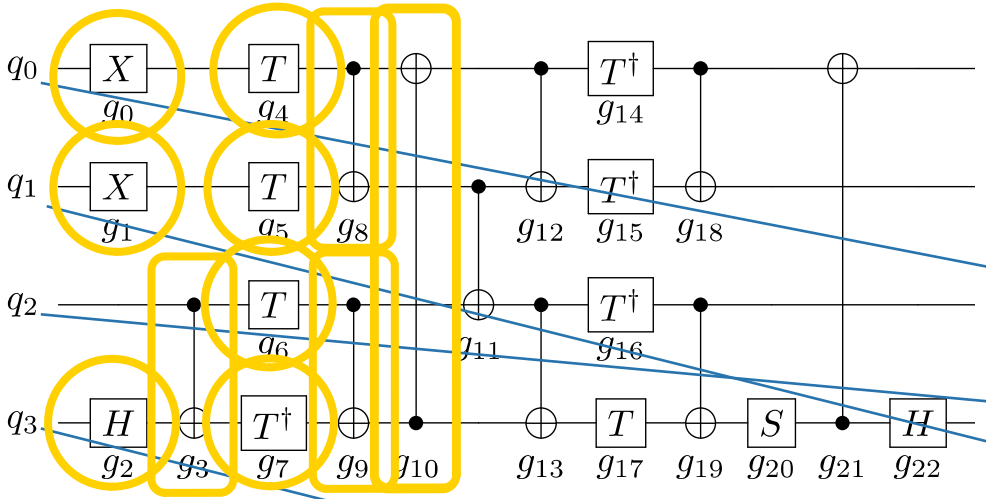
- Quantum Layout Synthesis
- Introduction to Multilevel Framework
- ML-QLS: Multilevel Layout Synthesis
- Future Directions

Quantum Layout Synthesis (QLS)

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



Quantum Layout Synthesis (QLS)



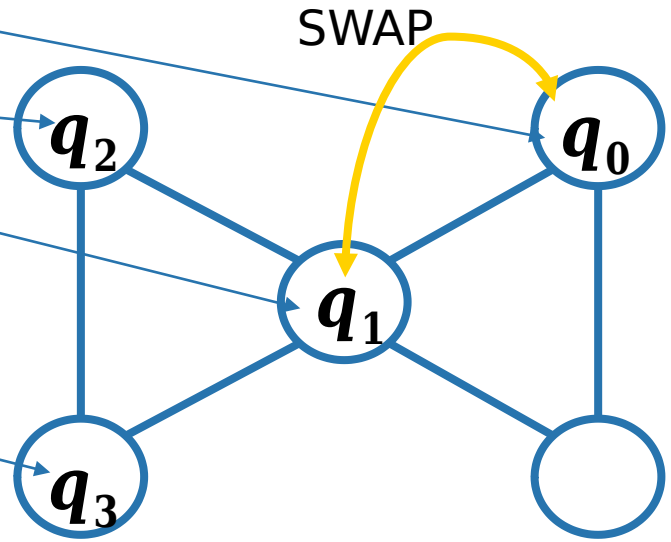
Input Circuit of Adder

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

CX on a pair of adjacent qubits, OK.

CX on a pair of non-adjacent qubits!

Insert SWAP gate to change the mapping



Coupling Graph of IBM QX 2

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

Previous works

- QLS has been proven to be NP-Hard
- Heuristic approach:
  - Efficient but suffer from quality degradation when problem size increase
- Exact approach: solver-based method
  - Poor scalability

*Sabre: Li et al. ASPLOS’19

Our Work

First step: Push the scalability of exact tool

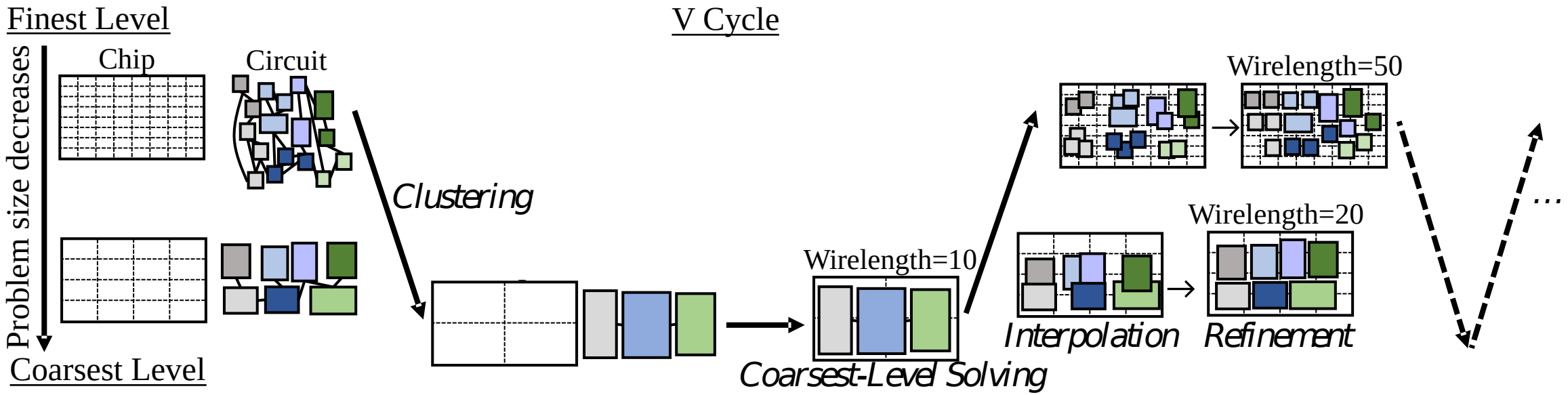
- OLSQ2 [DAC'23]: Improve constraint formulation, encoding, and optimization techniques
  - Show thousands of speedup compared to OLSQ [ICCAD'20]

Second step: A hybrid method combining heuristic and exact methods

- ML-QLS: Improve heuristic method by providing more global optimization information from exact method through multilevel framework
  - Effectiveness: 53% SWAP reduction compared to the leading heuristic tool
  - Scalability: Solve instances with hundreds of qubits

Multilevel framework comes into rescue!

- Shown its effectiveness in solving large-scale problem in VLSI design
  - E.g., graph partitioning, placement



Example of multilevel circuit placement

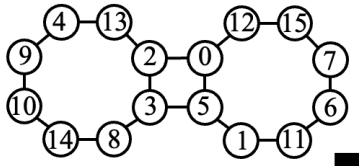
Challenges

Challenge 1: Hidden cost within cluster
Cost increase due to the overlook cost within a cluster

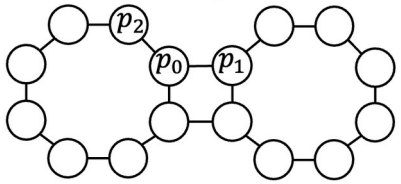
Solution: Compression rate=2 to ensure zero-cost execution within a cluster

A QUEKO circuit
Partial gate list:
$g_1(3,8)$
$g_2(2,13)$
$g_3(1,5)$
$g_4(0,2)$
$g_5(0,12)$
$g_6(3,8)$

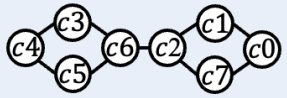
Connectivity graph



Rigetti Aspen-4

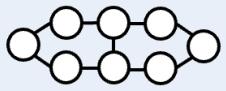


Circuit Clustering

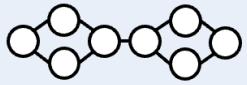


partial gate list:
$g_1(6,5)$
$g_2(6,3)$
$g_3(2,6)$
$g_3(7,2)$
$g_4(2,1)$
$g_5(6,5)$

Device Clustering
Option 1

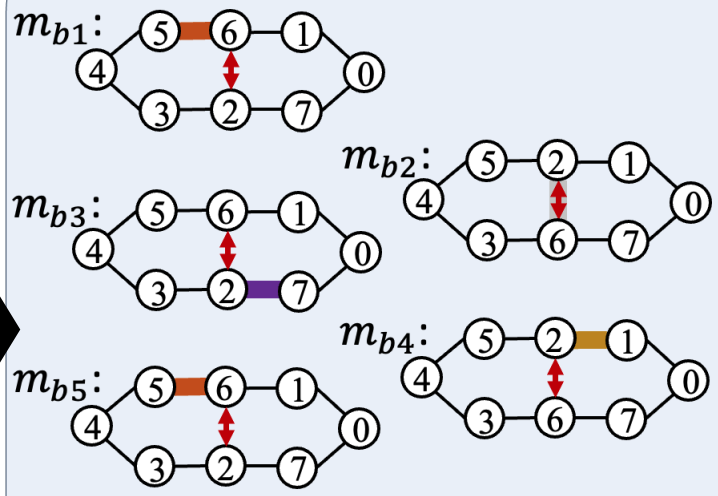


Option 2

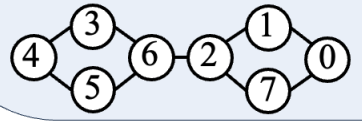


Coarsest-Level Solving

Option 1: Solution with four SWAPs



Option 2: Solution without SWAP



Challenge 2: Difficulty in Device Clustering
a. Distinct topology between finer and coarser graph
b. Define different solution spaces due to the change of relative qubit distance

Solution: Circuit-Informed Clustering

Clustering

Based on a QLS solution:

Step 1: Generate coarser program qubits

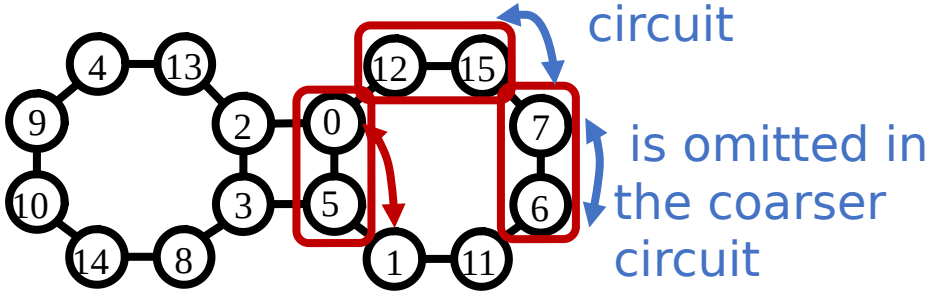
- Compute affinity between qubits
  - Every two-qubit gate increases affinity by one
- Iterative cluster two qubits 1) with the highest affinity and 2) mapped to adjacent physical qubits

Step 2: Generate coarser physical qubits based on program qubit clustering

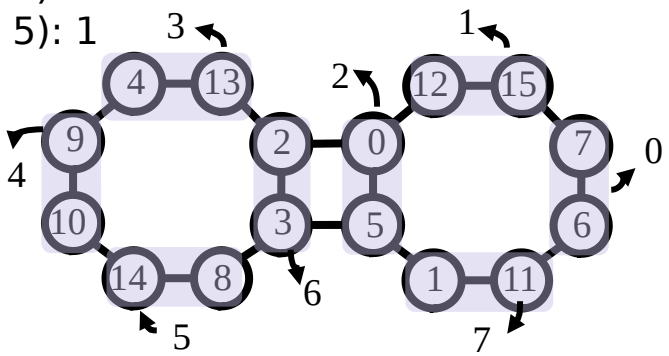
- If and are clustered, then and form a coarser physical qubits

Step 3: Generate coarser gates

Step 4: Generate coarser edges



Affinity between qubits:
(6, 7): 6
(12, 15): 5
(0, 1): 2
(0, 5): 1
…



Challenges

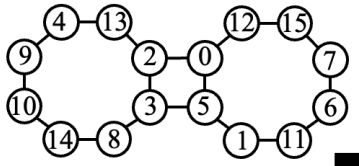
Challenge 1: Hidden cost within cluster
Cost increase due to the overlook cost within a cluster

Solution: Compression rate=2 to ensure zero-cost execution within a cluster

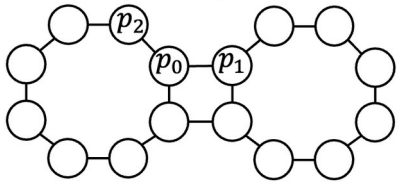
Challenge 3: Discrete solution space
a. Unable to do interpolation
b. Difficult to predict how optimizations will propagate across level

A QUEKO circuit
Partial gate list:
$g_1(3,8)$
$g_2(2,13)$
$g_3(1,5)$
$g_4(0,2)$
$g_5(0,12)$
$g_6(3,8)$

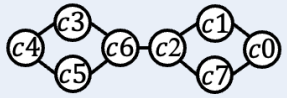
Connectivity graph



Rigetti Aspen-4

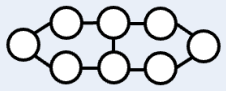


Circuit Clustering

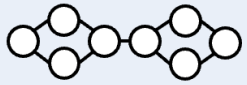


partial gate list:
$g_1(6,5)$
$g_2(6,3)$
$g_3(2,6)$
$g_3(7,2)$
$g_4(2,1)$
$g_5(6,5)$

Device Clustering
Option 1

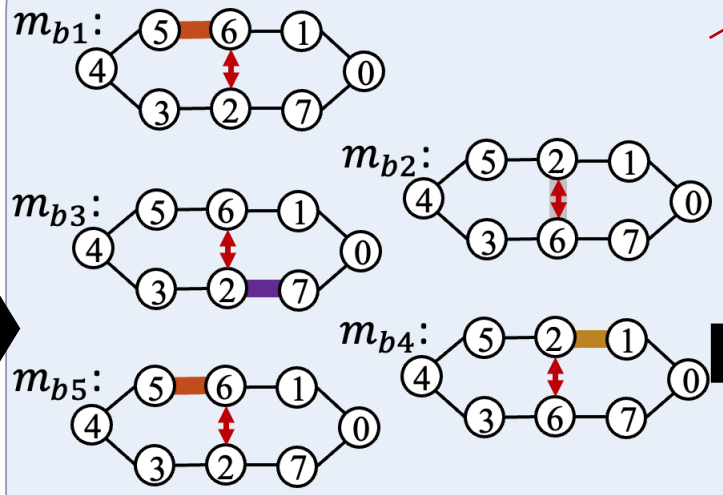


Option 2

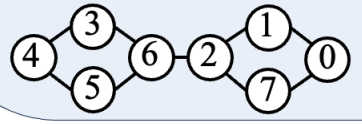


Coarsest-Level Solving

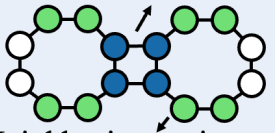
Option 1: Solution with four SWAPs



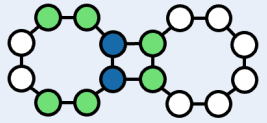
Option 2: Solution without SWAP



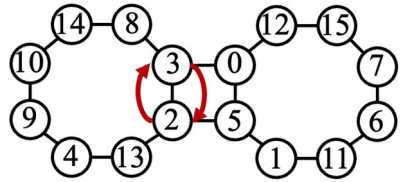
Refinement Example:
Qubit Region for $q_2 \in q_{c6}$
Region where q_{c6} is mapped



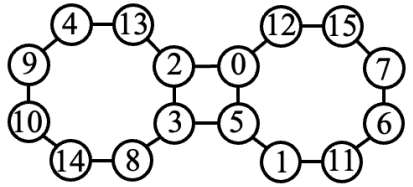
Neighboring region



Solution with two SWAP



SWAP-free solution

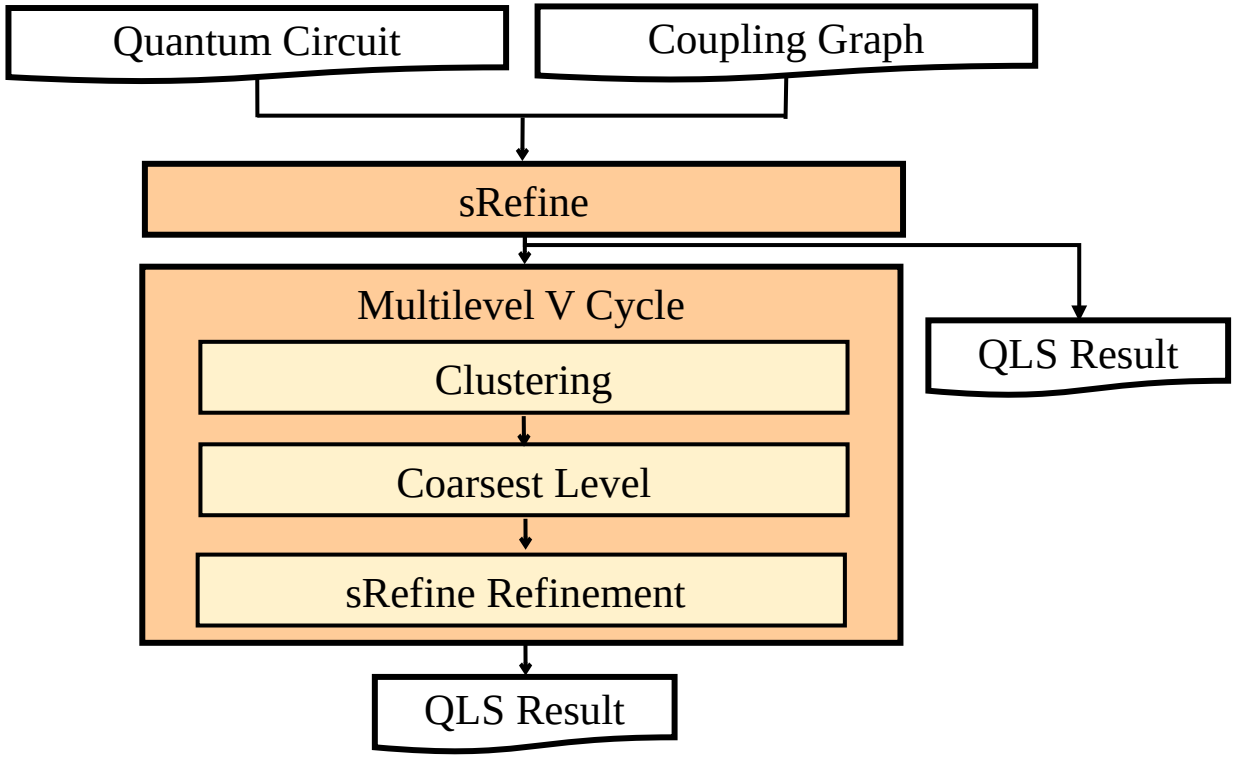


Challenge 2: Difficulty in Device Clustering
a. Distinct topology between finer and coarser graph
b. Define different solution spaces due to the change of relative qubit distance

Solution: Circuit-Informed Clustering

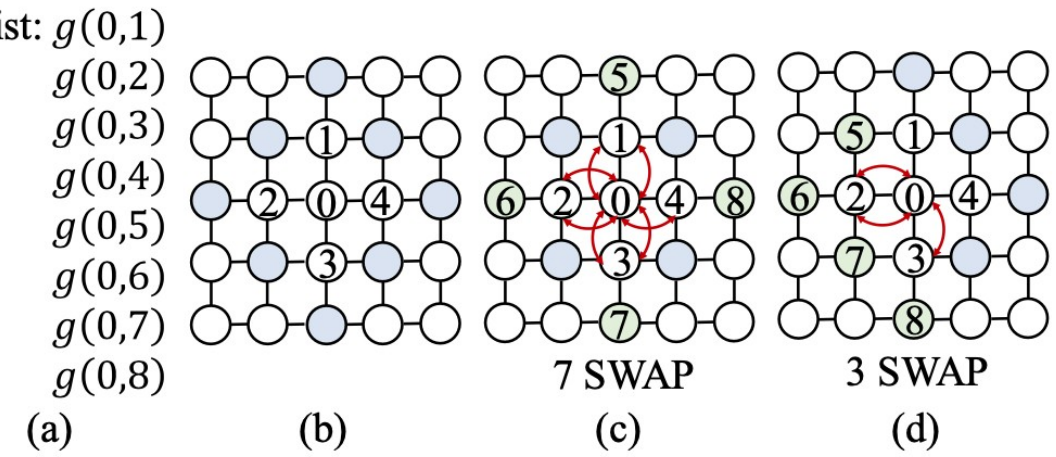
ML-QLS

- ***sRefine***:
  - Scalable heuristic tool to provide a high-quality QLS result as the guidance of clustering
  - Serve as an effective refinement operation
- ***Coarsest-level solving***: TB-OLSQ2



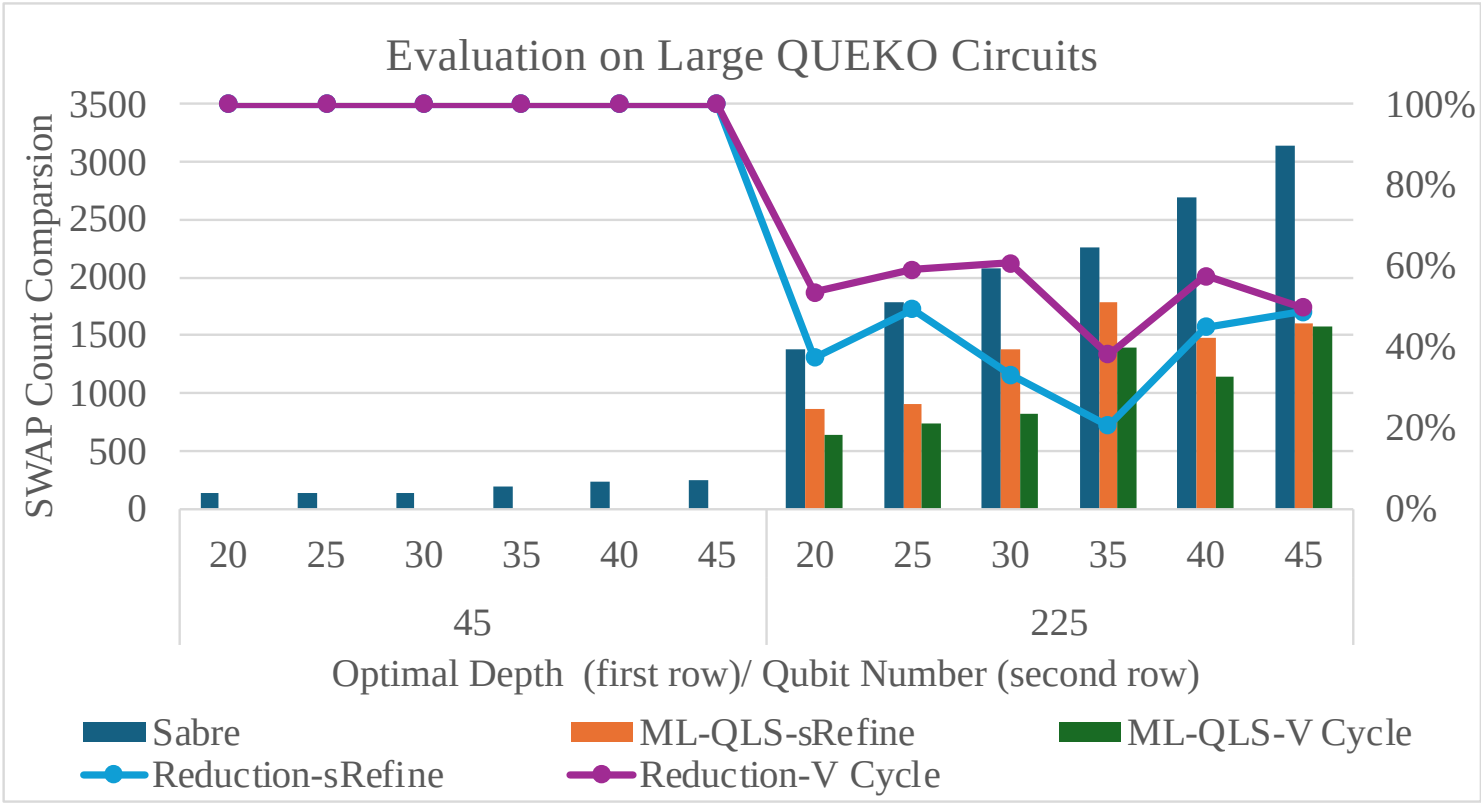
sRefine

- SA-based initial mapping
  - Cost integrates qubit distance cost of gates and qubit interaction cost
    - Qubit interaction cost: qubits that interact with the same qubit should locate near each other as well
- A*-Based SWAP Insertior



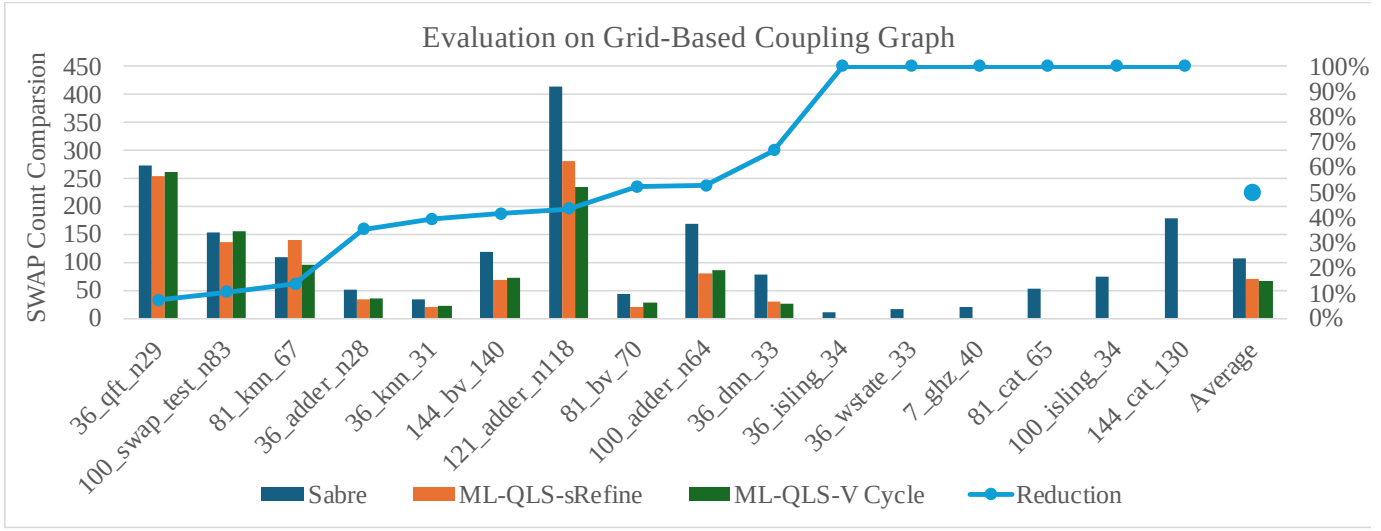
Evaluation on Known-Optimal Benchmark

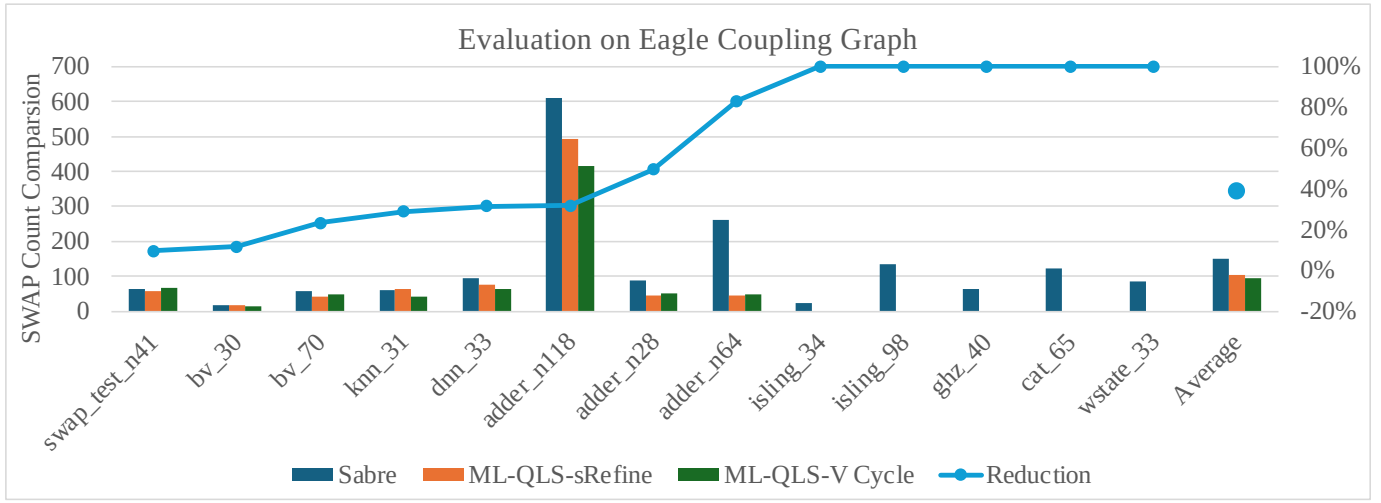
- 57% SWAP reduction compared to SABRE



Evaluation on General Circuits

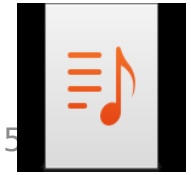
- 45% SWAP reduction compared to SABRE
- Scalability: take 10 min to solve a problem on average





Takeaways

- Propose ML-QLS, which is the first work applying multilevel framework to solve problems in quantum computing
  - Offer a clustering method to effectively construct problem approximation by generating device clustering via the guidance of circuit clustering
  - Propose a heuristic tool, sRefine, which achieves a notable performance improvement by integrating a novel qubit interaction cost
- ML-QLS is flexible: we can plug in any exact tool for coarsest-level solving and use other tools as refinement operations



Thank you for listening!

- OLSQ2

Github: 

Paper: 

Video: 

- ML-QLS

Github: 

Paper: 

Acknowledgement

- The authors would like to thank Bochen Tan, Jason Kimko, Yunong Shi, Eric Kessler, and Yaroslav Kharkov for useful discussion about quantum layout synthesis and valuable comments on the manuscript.
- This research is supported in part by the National Science Foundation Award 2313083 and Amazon under the Science Hub program.

Thank you!