

# Qubit Mapping for Trapped-Ion Systems Using Satisfiability Modulo Theories

---

Wei-Hsiang Tseng, Yao-Wen Chang, and Jie-Hong Roland Jiang

Graduate Institute of Electronics Engineering  
National Taiwan University



臺灣大學

# Outline

---



- Introduction to Trapped-Ion Systems



- Qubit Mapping Problem



- Proposed Approach



- Experimental Results



- Conclusions

# Outline

---



- Introduction to Trapped-Ion Systems



- Qubit Mapping Problem



- Proposed Approach



- Experimental Results



- Conclusions

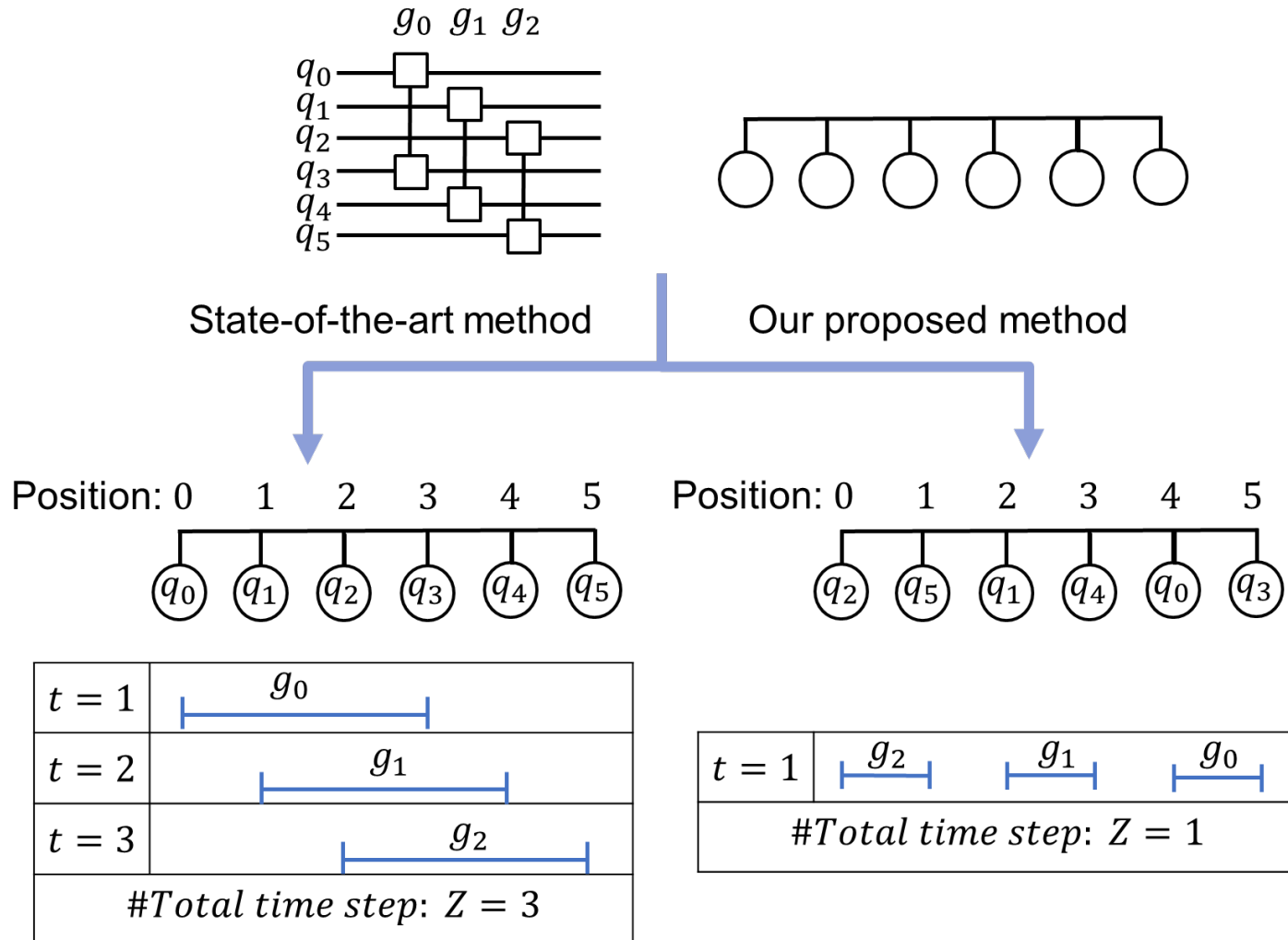
# Quantum Computing

---

- Quantum computing
  - Substantial speedup on several classes of problems that are considered intractable in classical computing
  - Example: integer factorization (Shor's algorithm), unstructured database search (Grover's algorithm)
- Superconducting systems
  - Offer promising capabilities for realizing large-scale quantum processors with improved coherence and gate fidelity
  - Limit the connectivity bottleneck on these hardware architectures
- Trapped-ion systems
  - Have a relatively long coherence time and the ion qubits in an ion array are fully coupled

# Motivation

- Not much work presents a mapping algorithm for 1D-array trapped-ion systems
  - Many studies focus on qubit mapping algorithms for Superconducting systems [Wu *et al.*, ICCAD22]



# Outline

---



- Introduction to Trapped-Ion Systems



- **Qubit Mapping Problem**



- Proposed Approach



- Experimental Results



- Conclusions

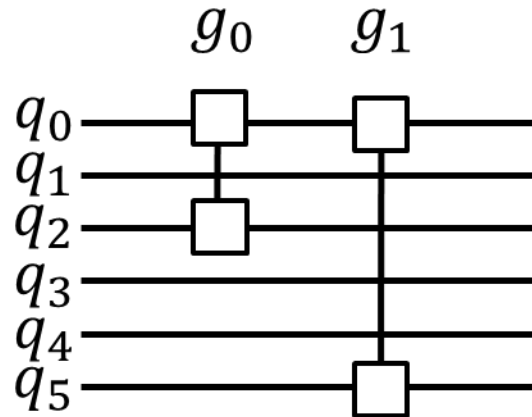
# Trapped-Ion Architecture

---

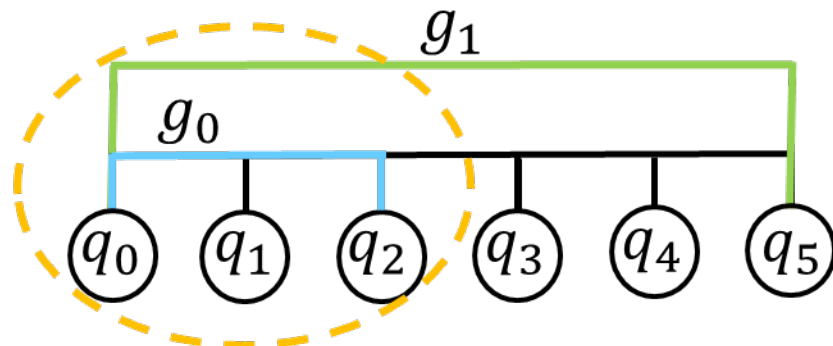
- Cirac-Zoller gate
  - First entangles two ions
  - Has significant limitations because it requires the ions to remain in the motional ground state
- Geometric-phase gate
  - Is insensitive to the initial ion motional state because the geometric-phase gate is achieved by applying specific phase transformations
  - Cannot be applied to all qubits
- Mølmer-Sørensen (MS) gate
  - Can be applied to ions not cooled to the motional ground state
  - Its motional states had to be disentangled for all values after the gate
  - Must consider the crosstalk effect

# Coupling Constraint Graph

- Propose a new coupling constraint graph to avoid the crosstalk effect
  - Based on the Mølmer-Sørensen gate and its related hardware architecture
  - Maintain fidelity and facilitate the internal and motional disentanglement

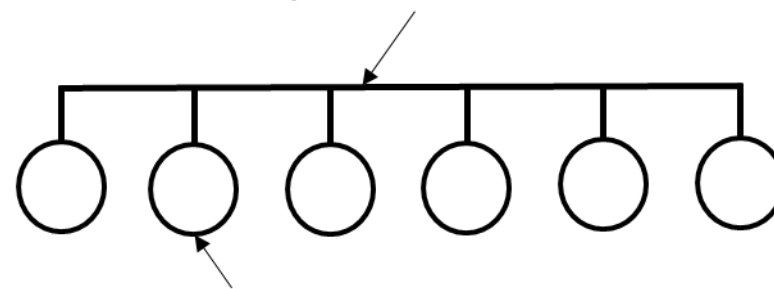


Crosstalk

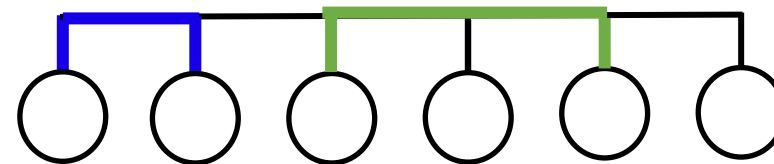


Overlap between  $g_0$  and  $g_1$

Channel (multiple-qubit net)



Physical qubit

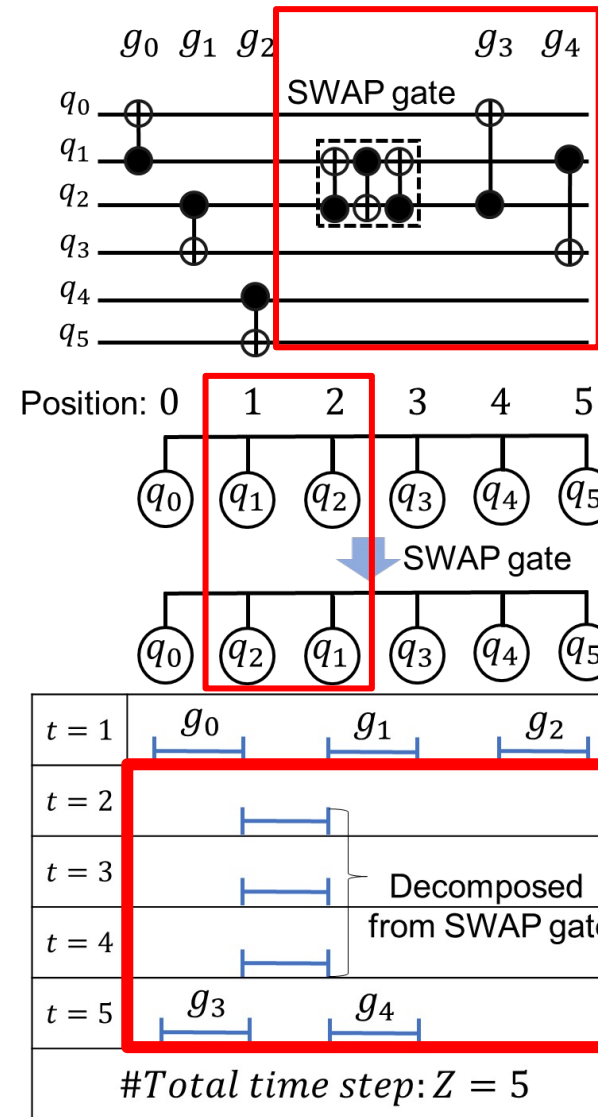
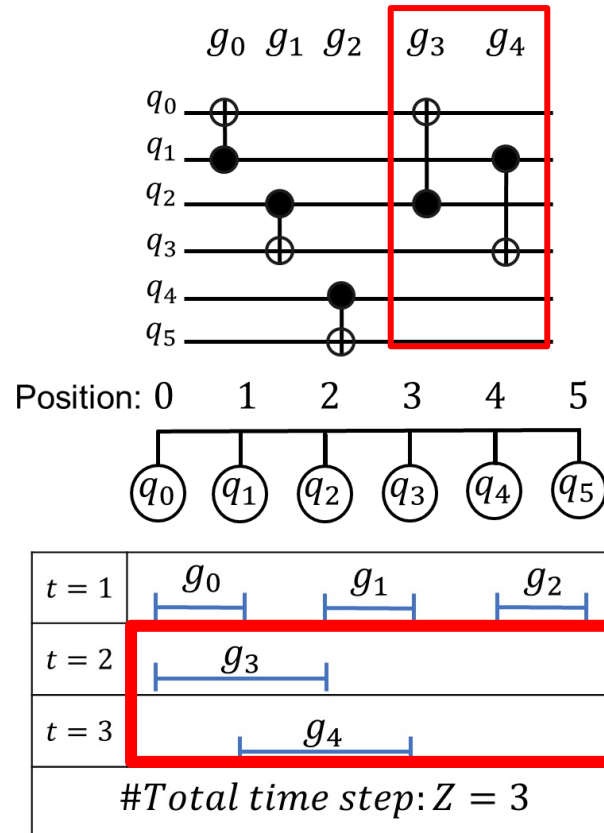


Avoid two gates at overlapping positions



# SWAP Gates

- Are unnecessary in trapped-ion systems, focusing on higher execution costs



Add two additional time steps

# Problem Formulation

---

- Input
  - A quantum circuit synthesized from a quantum algorithm
  - A hardware architecture
- Output
  - A mapping solution
- Objectives
  - Minimize the number of total time steps
- Constraints
  - Coupling constraint graph (to avoid the crosstalk effect)

# Outline

---



- Introduction to Trapped-Ion Systems



- Qubit Mapping Problem



- **Proposed Approach**



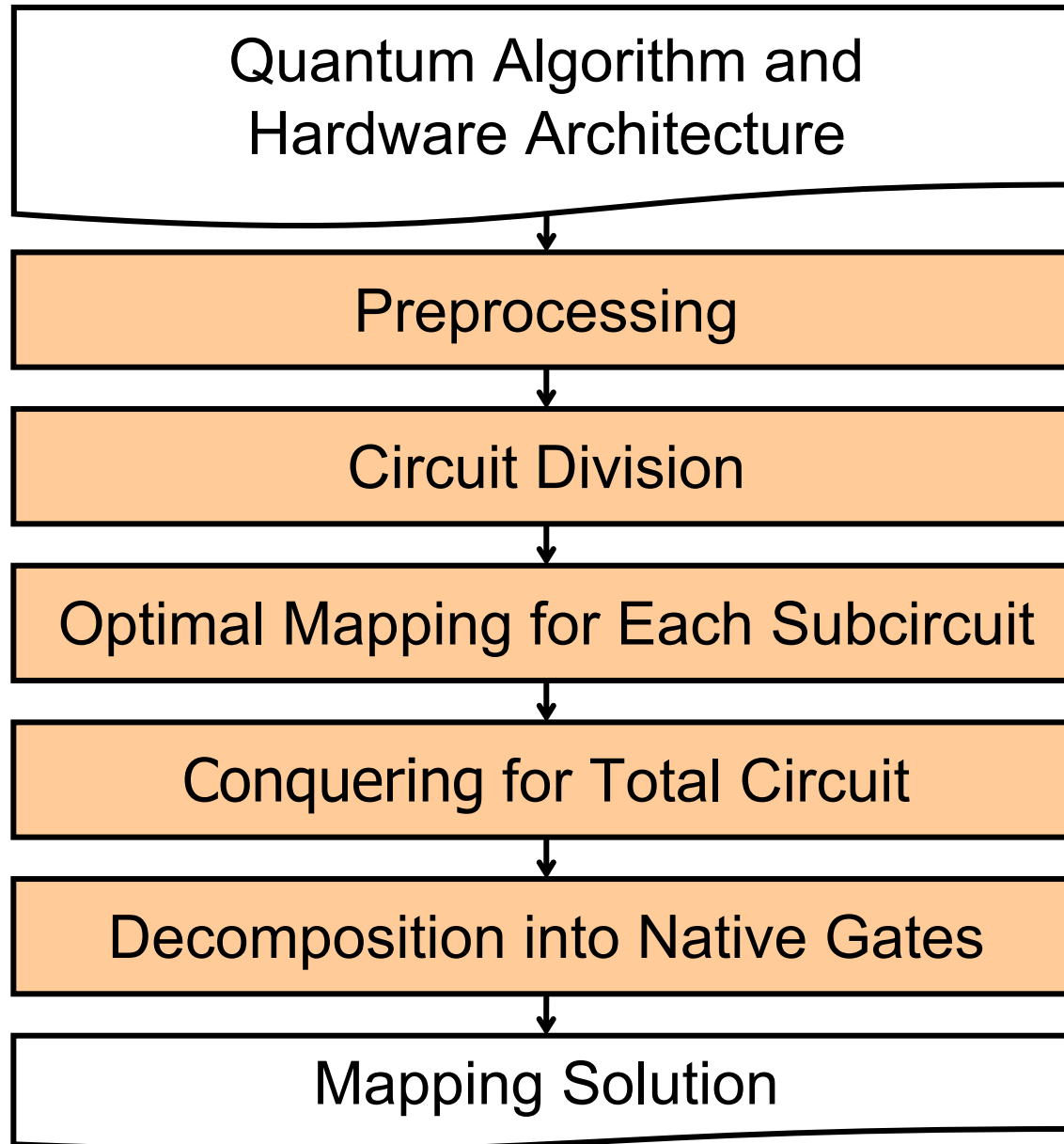
- Experimental Results



- Conclusions

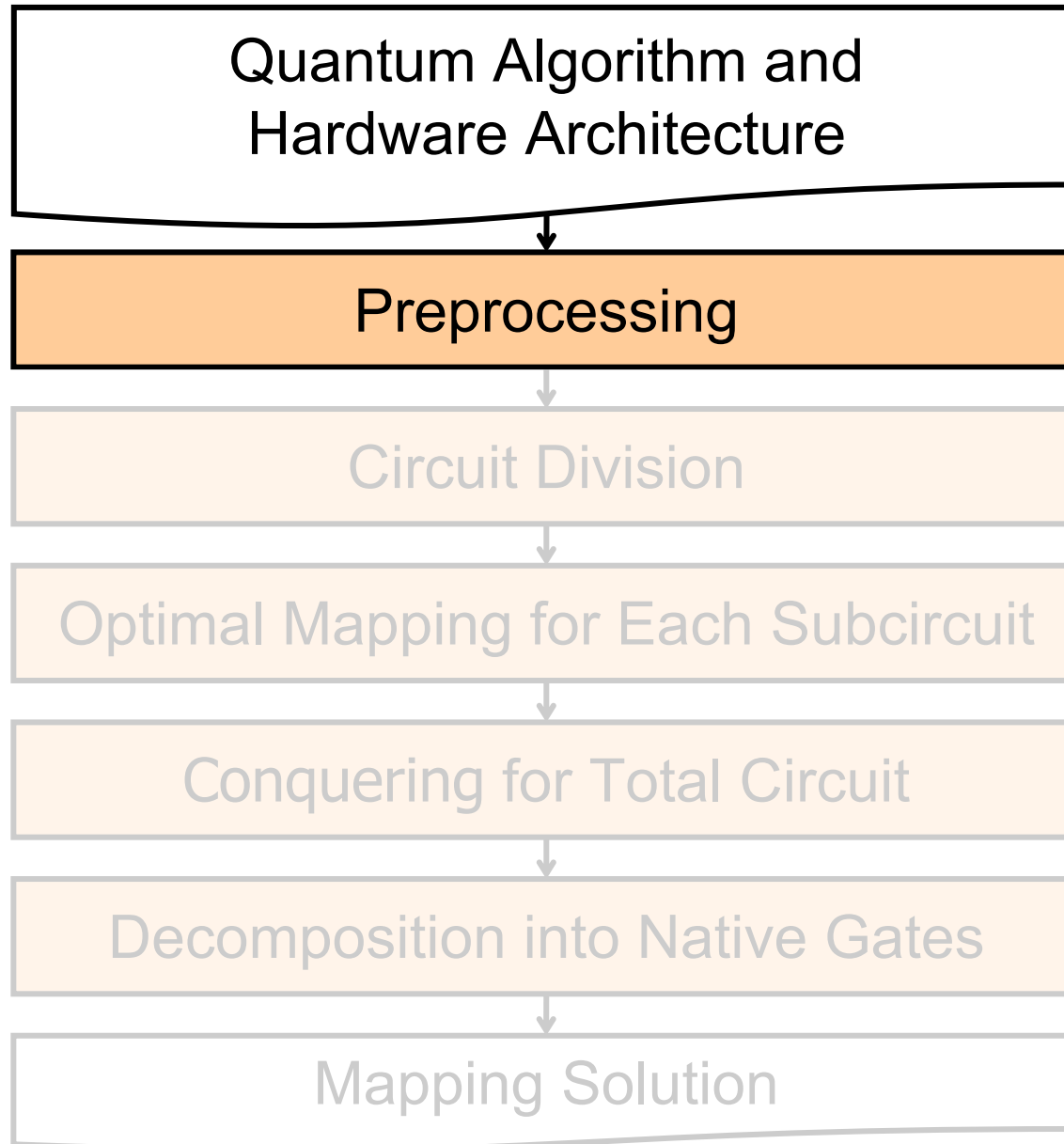
# Algorithm Flow

---



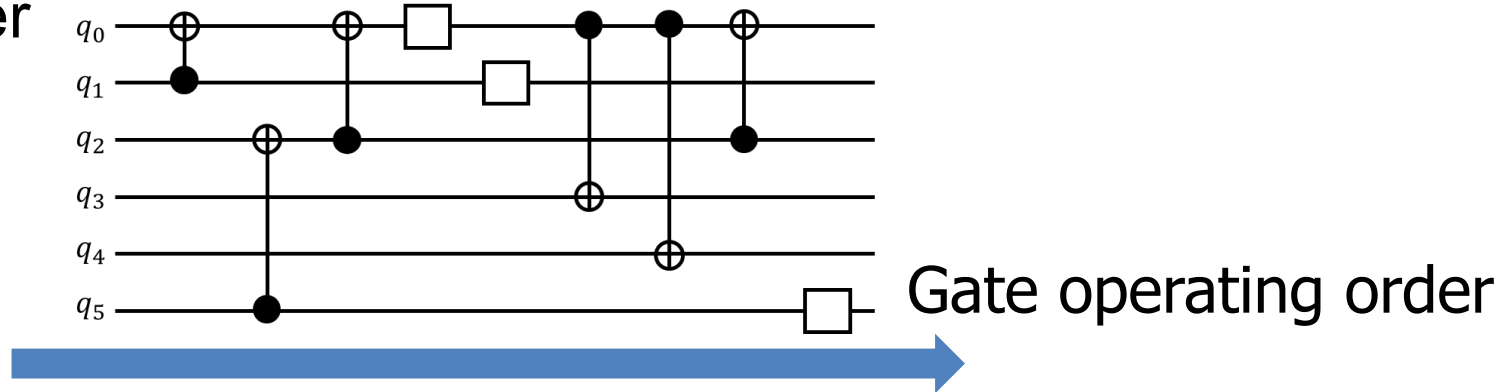
# Algorithm Flow

---

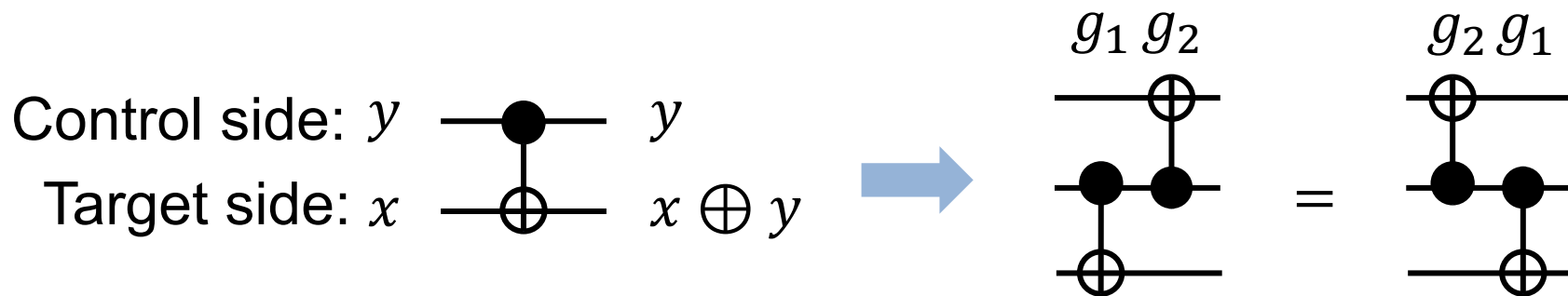


# Preprocessing

- Much work uses the left to right of gates on the circuit as their gate operating order

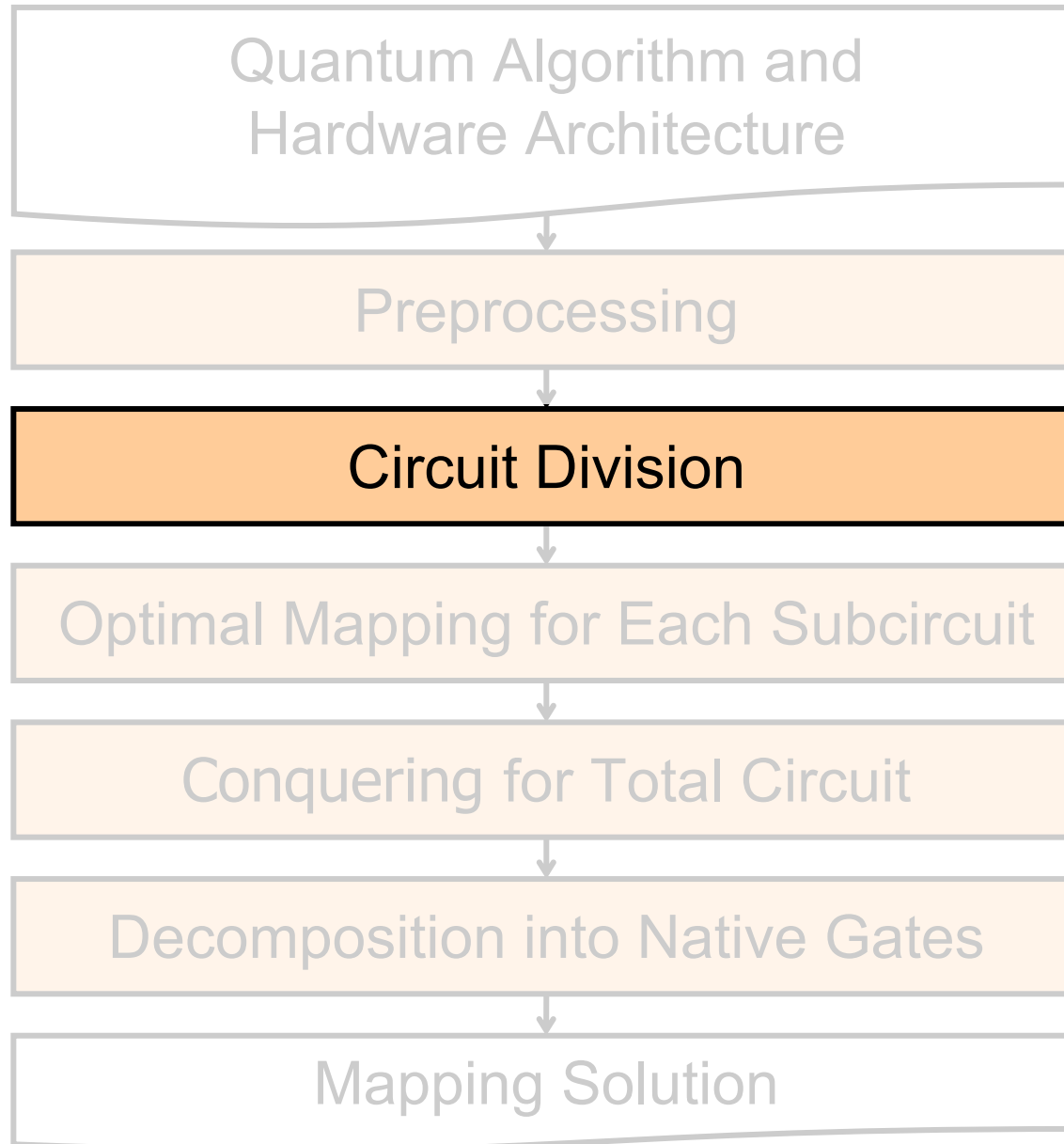


- Decompose the multiple-qubit gates into the CNOT gates instead of a native gate circuit in this stage
  - Consider the commutativity of CNOT gates on the control side in the following stages



# Algorithm Flow

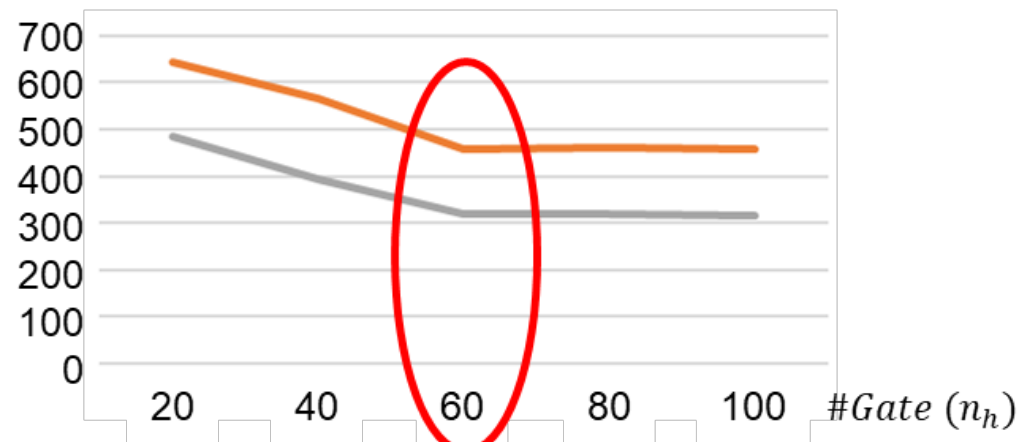
---



# Circuit Division

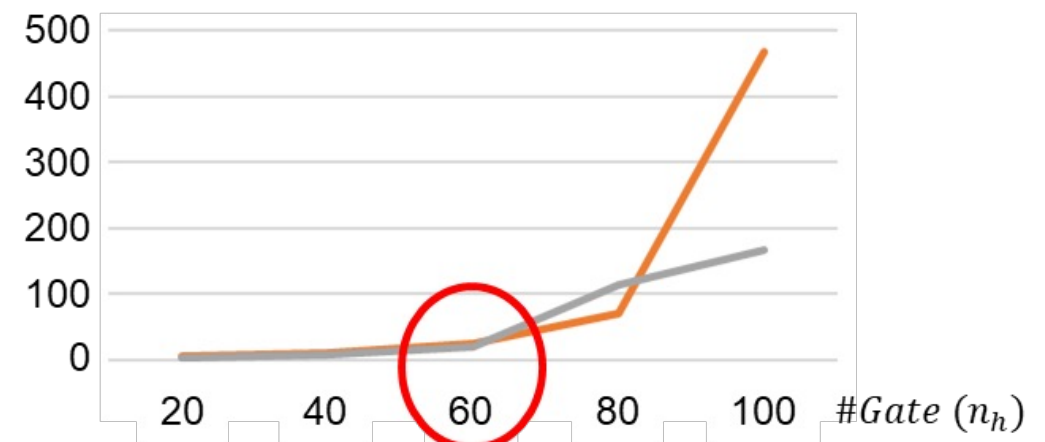
- We introduce the parameter  $n_h$  to determine the necessity of circuit division
- If the number of two-qubit gates  $n_g$  in the decomposed circuit exceeds  $n_h$ 
  - The circuit is divided into  $\lceil n_g/n_h \rceil$  subcircuits
- Choosing an appropriate value for  $n_h$  involves a trade-off between solution quality and computation time

#Total time step (Z)



— 20QBT\_queko\_100\_0

Time (s)

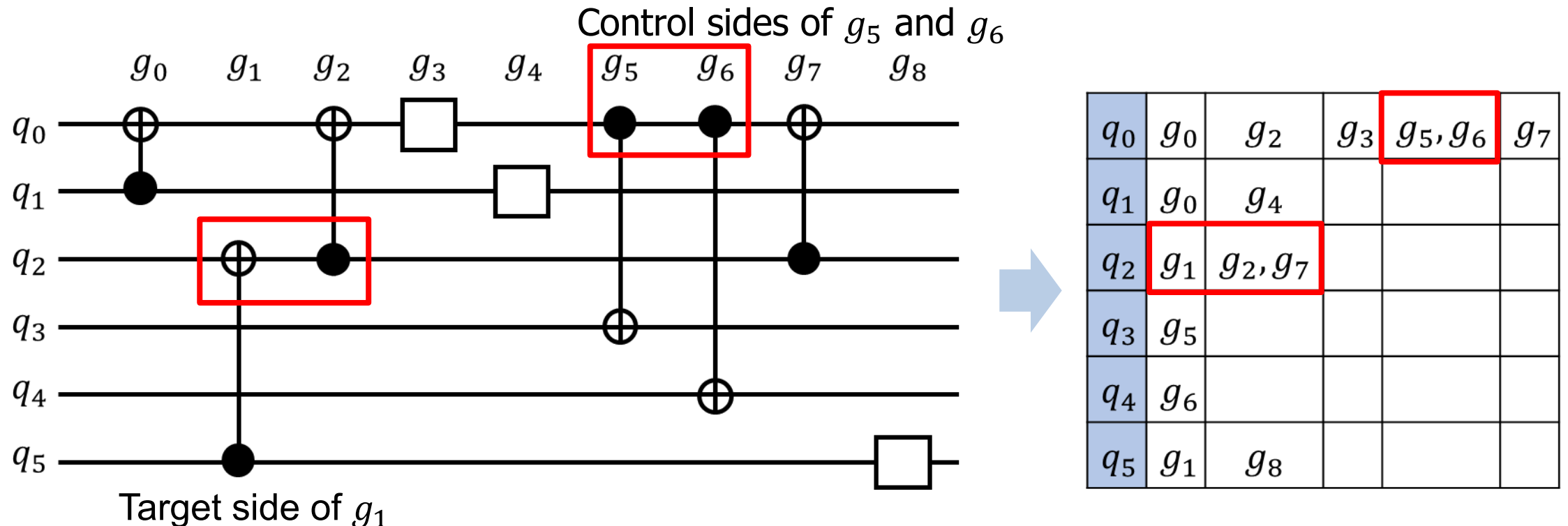


— 16QBT\_queko\_100\_0



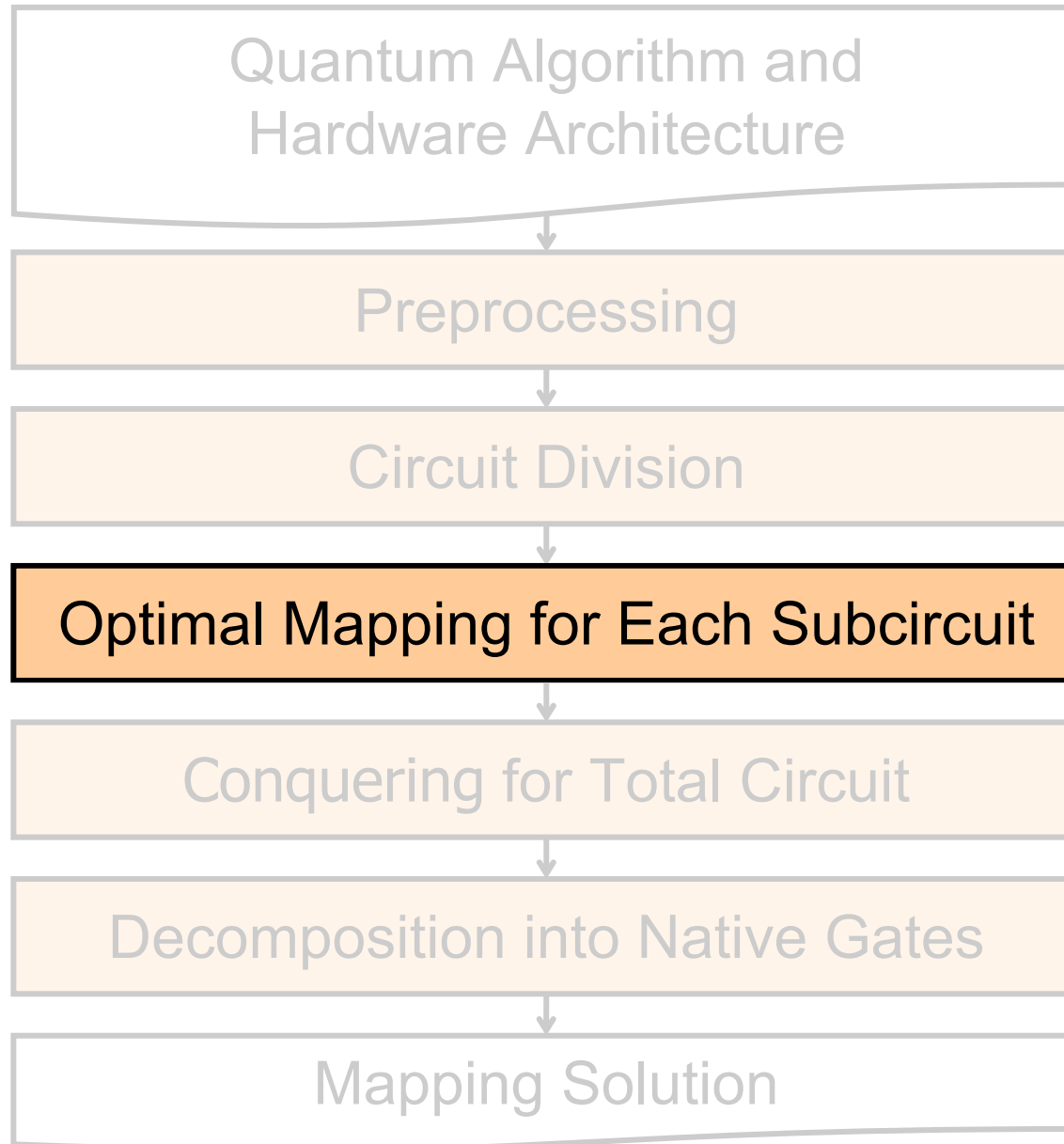
# Gate Dependency Table Generation

- Focus on the operation location at the target side
- Consider the operation location of each single-qubit gate



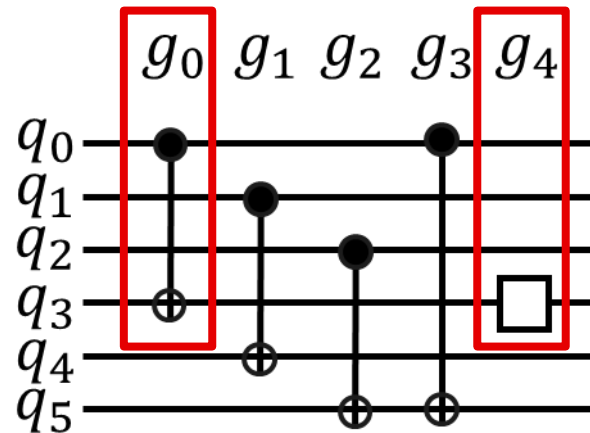
# Algorithm Flow

---



# Satisfiability Modulo Theories

- Can check whether the model is satisfiable
  - Parameterize the quantum circuit and hardware architecture
  - Use variables to give the model constraints for ultimately capturing the integral hardware architecture
- Parameterize the quantum circuit
  - Use an example with four two-qubit gates and a single-qubit gate



$$\begin{aligned}
 g_0^1 &= q_0, g_0^2 = q_3 \\
 g_1^1 &= q_1, g_1^2 = q_4 \\
 g_2^1 &= q_2, g_2^2 = q_5 \\
 g_3^1 &= q_0, g_3^2 = q_5 \\
 g_4^1 &= q_3, g_4^2 = q_3
 \end{aligned}$$

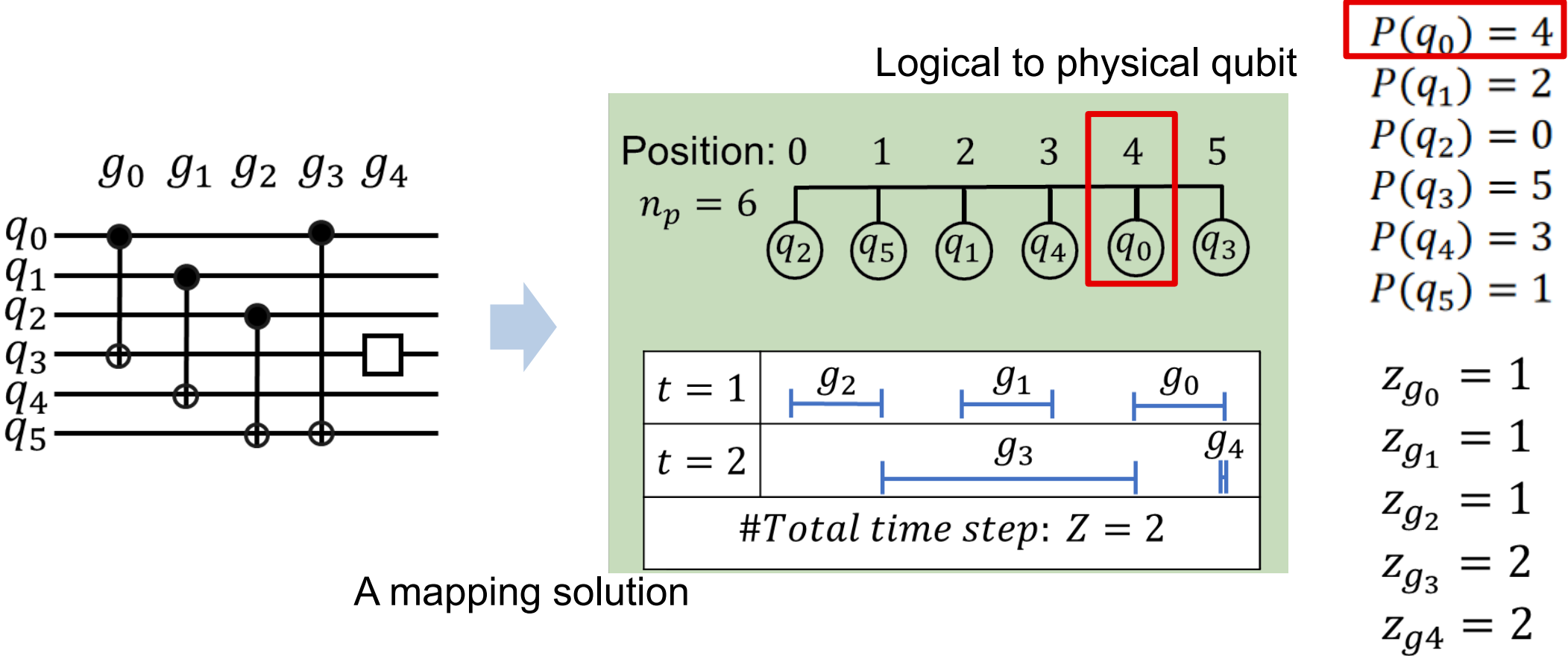
$q_i$ : the  $i$ -th logical qubit in  $Q$

$g_i$ : the  $i$ -th two-qubit gate

$g_i^k$ : the  $k$ -th logical qubit, operated by  $g_i$

# Parameterize Hardware Architecture

- Parametric quantum circuit mapping to the physical architecture



Goal: obtain the optimal mapping solution using SMT solver

## Four SMT Constraints: 1<sup>st</sup> and 2<sup>nd</sup> Constraints

---

1. One-to-one mapping constraint: restricts one-to-one mapping between logical and physical qubits

$$P(q_i) \neq P(q_j), \forall q_i, q_j \in Q, i \neq j, \forall P(q_i) < n_p$$

2. Operation order constraint: ensures that gates are operated in dependency order, thus maintaining the correctness of the circuit functionality

$$z_{g_i} < z_{g_j}, \forall g_i \mapsto g_j, \forall g_i, g_j \in D$$

$Q$ : a set of logical qubits

$q_i$ : the  $i$ -th logical qubit in  $Q$

$G$ : a set of two-qubit gates

$g_i$ : the  $i$ -th two-qubit gate

$n_p$ : the number of physical qubits

$P(q_i)$ : the position of the physical qubit, mapped by  $q_i$

$z_{g_i}$ : the time step executing  $g_i$

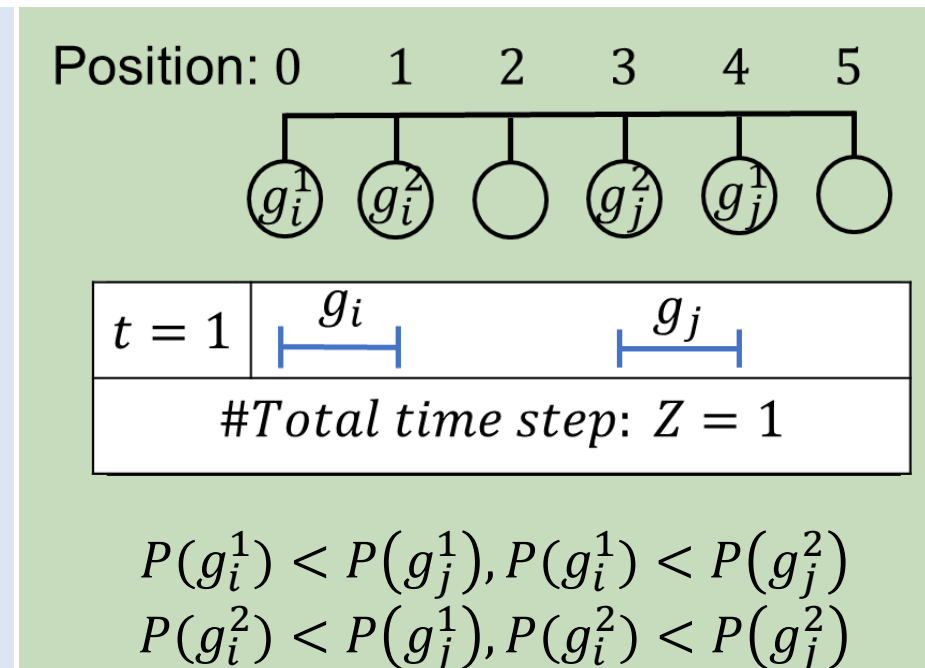
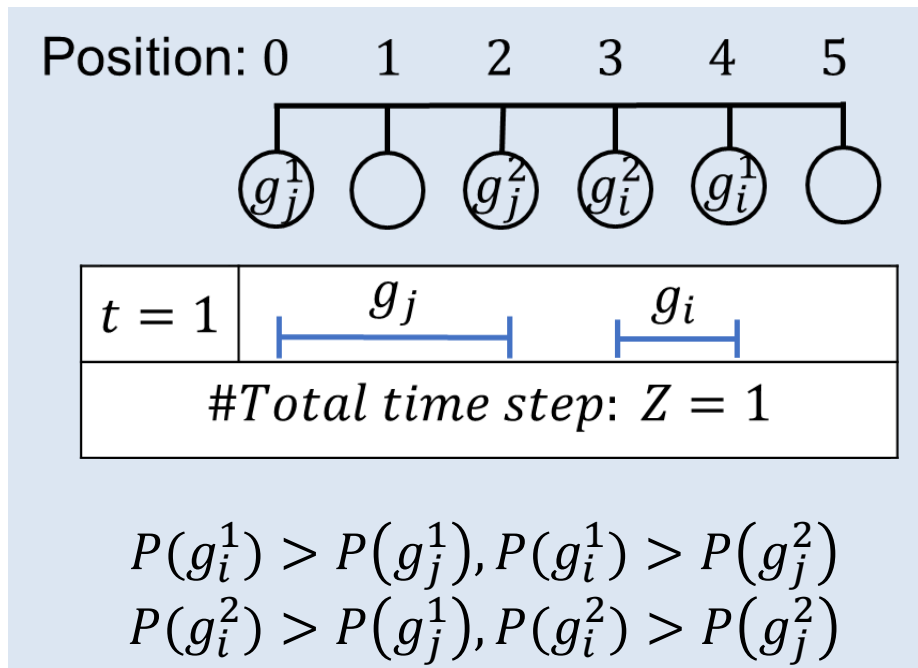
$D$ : a gate dependency table

$g_i \mapsto g_j$ :  $g_i$  has to precede  $g_j$

## Four SMT Constraints: 3<sup>rd</sup> Constraint (Non-Overlap Case)

3. Crosstalk avoidance constraint: prevents the occurrence of severe crosstalk effects during gate operations

$$\bigwedge_{k,l \in [1,2]} P(g_i^k) > P(g_j^l) \parallel \bigwedge_{k,l \in [1,2]} P(g_i^k) < P(g_j^l) \parallel z_{g_i} \neq z_{g_j}, \forall g_i, g_j \in G.$$

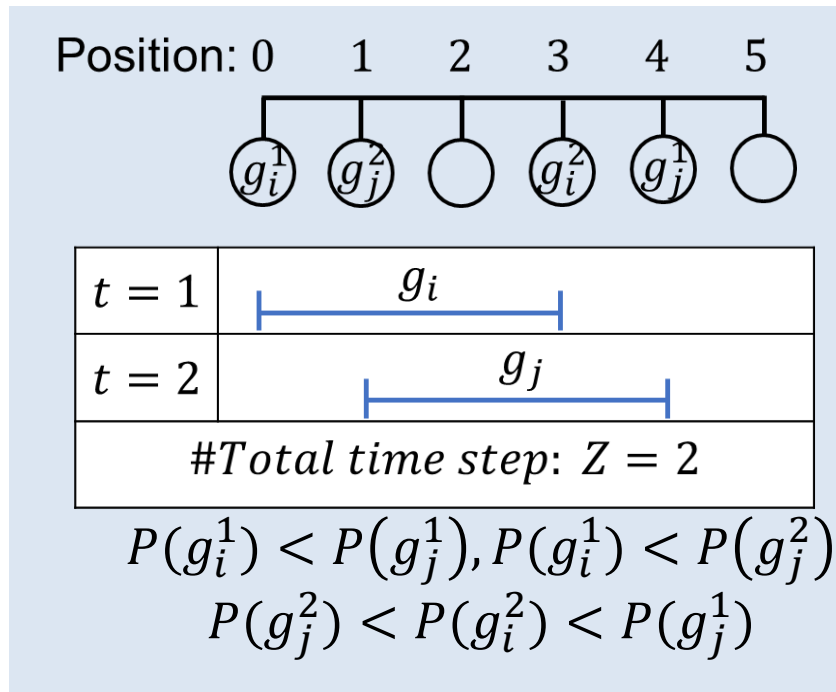


Both endpoints of  $g_i$  must be greater (less) than both endpoints of  $g_j$

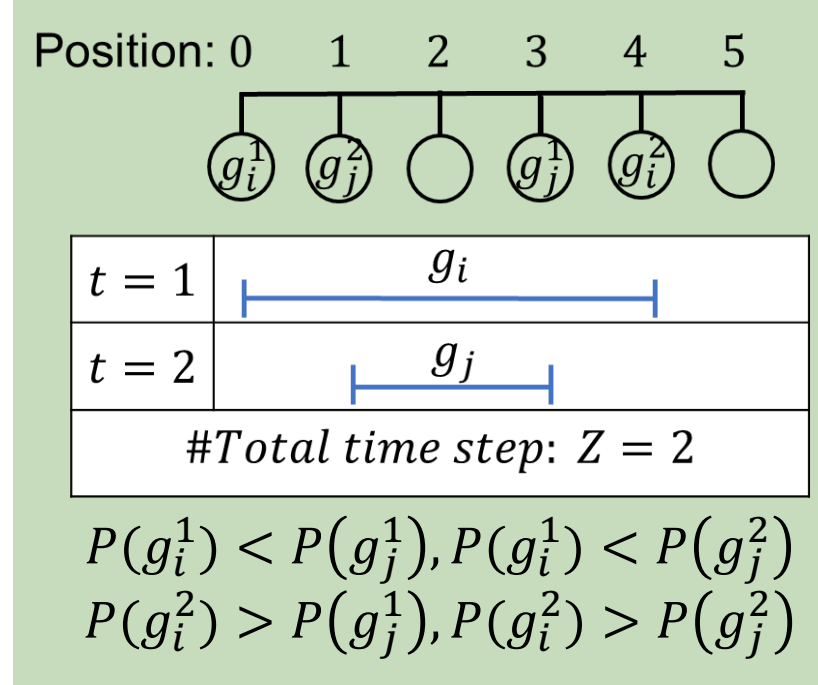
## Four SMT Constraints: 3<sup>rd</sup> Constraint (Overlap Case)

- If the operation positions of  $g_i$  and  $g_j$  overlap, they must be operated in different time steps

$$\bigwedge_{k,l \in [1,2]} P(g_i^k) > P(g_j^l) \parallel \bigwedge_{k,l \in [1,2]} P(g_i^k) < P(g_j^l) \parallel z_{g_i} \neq z_{g_j}, \forall g_i, g_j \in G.$$



One endpoint of  $g_i$  is outside  $g_j$   
The other endpoint is inside  $g_j$

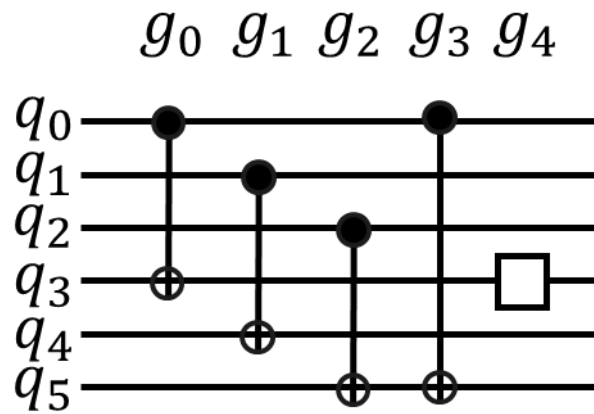


Both endpoints of  $g_j$  are inside  $g_i$

# Four SMT Constraints: 4<sup>th</sup> Constraint

- Total time step constraint: constraints all time steps of implementing gates smaller than the given  $Z$ 
  - If the model is satisfied, it represents the desired mapping solution
  - Gradually increase  $Z$  until a satisfying model is obtained
  - Can obtain the optimal mapping solution for a medium-scale circuit or a subcircuit

$$\forall z_{g_i} \leq Z, \forall g_i \in G.$$

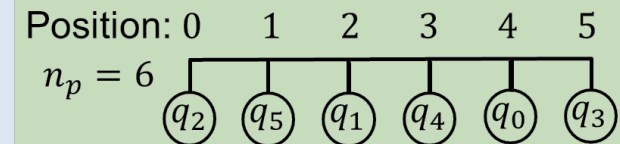


Gradually increase  $Z$

$Z = 1$

SMT solver:  
Unsat

$Z = 2$

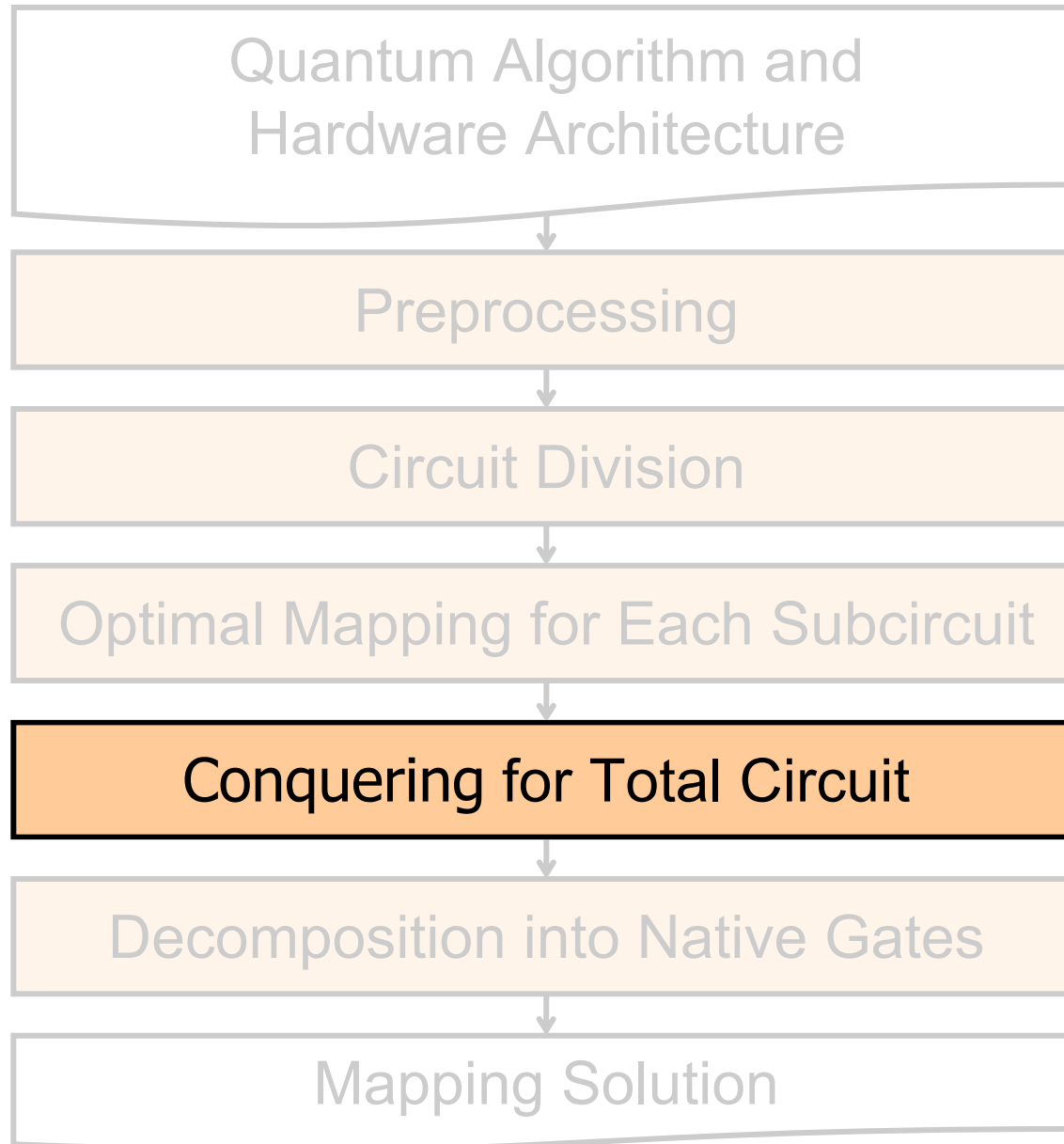


$t = 1$	$g_2$	$g_1$	$g_0$		
$t = 2$		$g_3$			$g_4$
#Total time step: $Z = 2$					



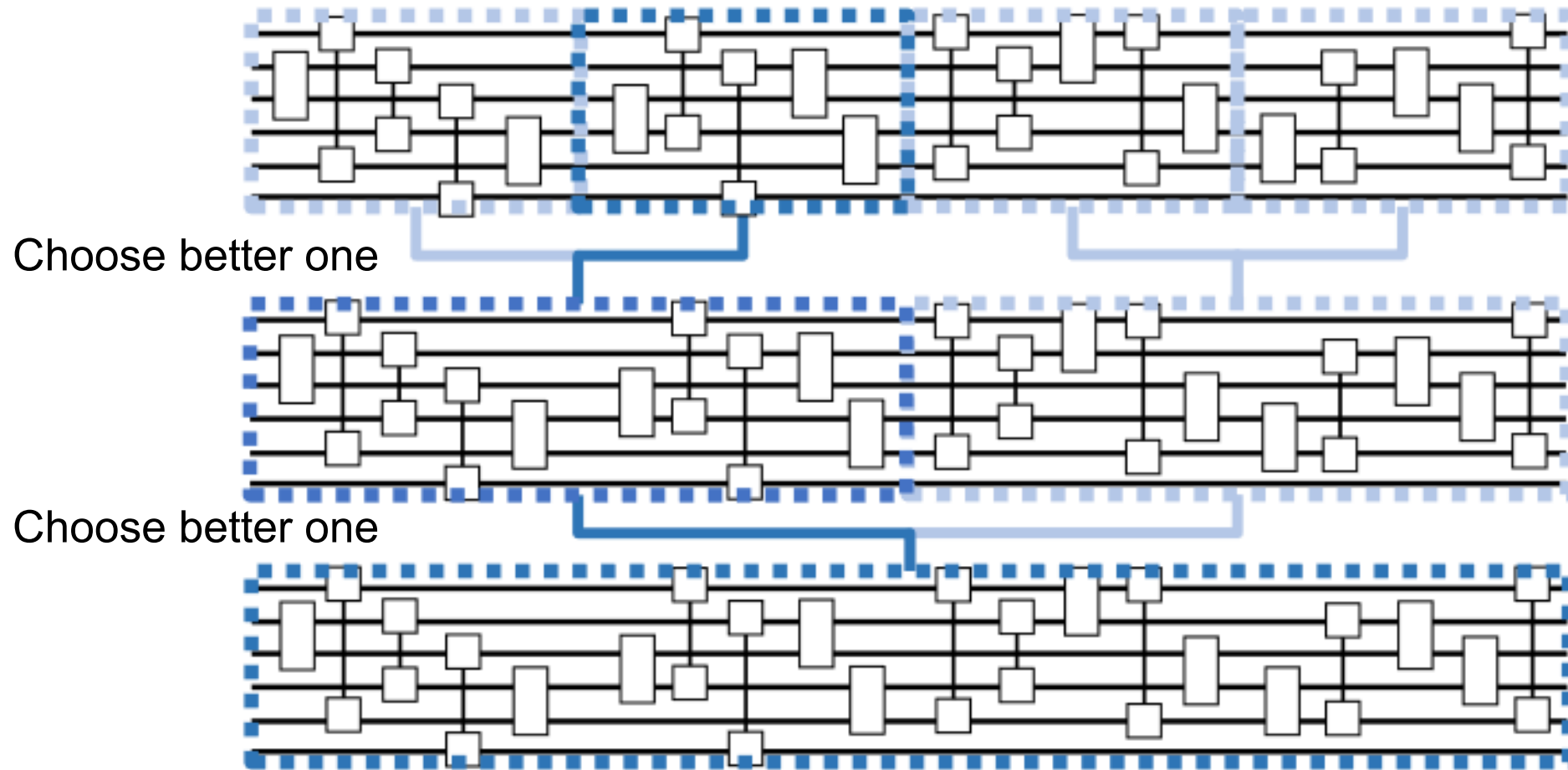
# Algorithm Flow

---



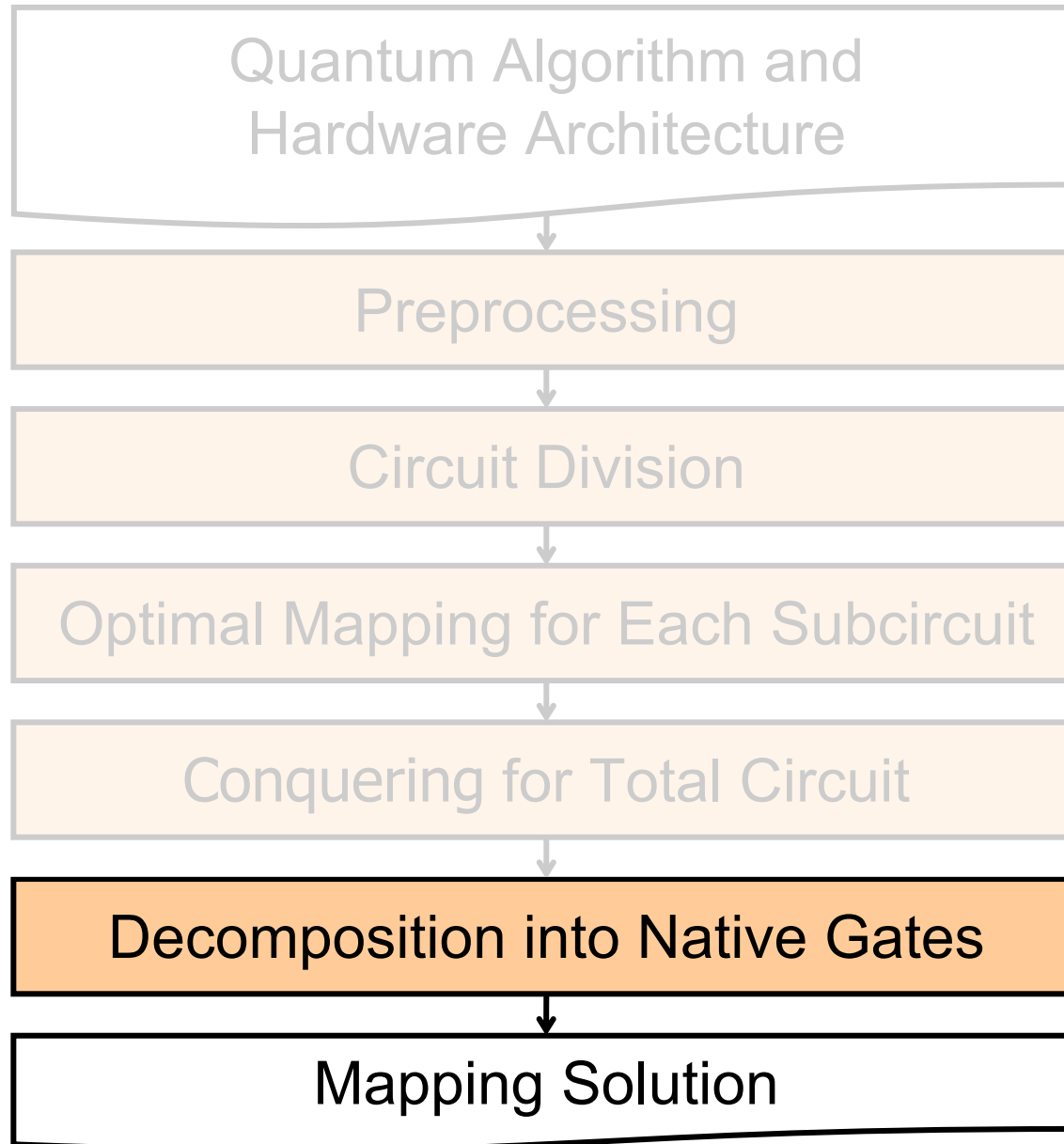
# Conquering for the Total Circuit

- After performing the SMT-based qubit mapping algorithm
  - Must conquer all results to reconstruct the entire circuit for the large-scale circuits



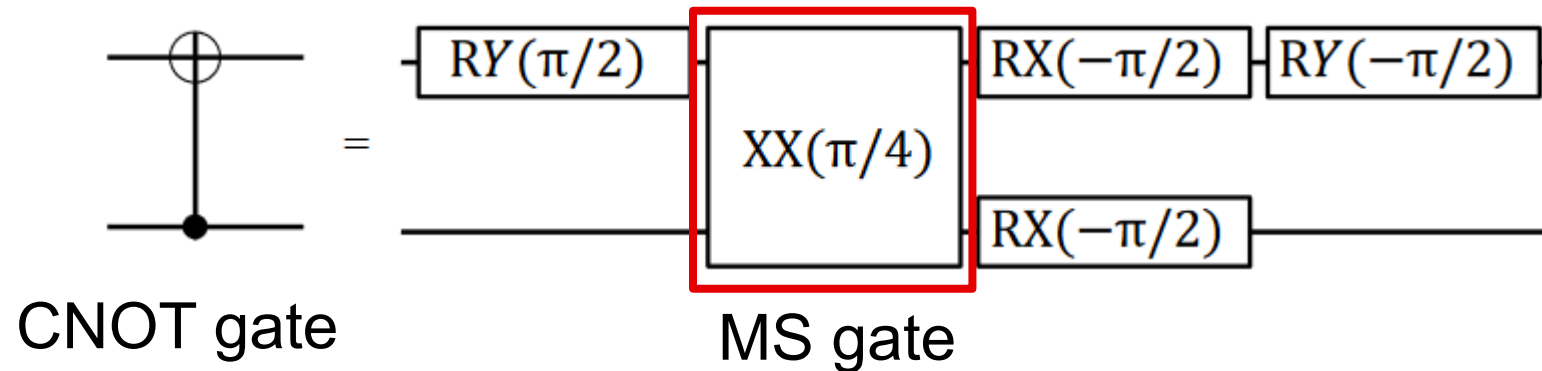
# Algorithm Flow

---



# Decomposition into Native Gates

- Convert the CNOT gates into native gates (MS gates) to enable the execution of the circuit composed of single-qubit gates and MS gates by the trapped-ion hardware architecture



# Outline

---



- Introduction to Trapped-Ion Systems



- Qubit Mapping Problem



- Proposed Approach



- **Experimental Results**



- Conclusions

# Experimental Settings

- Platform
  - C++/Linux workstation with 64-core 2.9 GHz AMD Ryzen CPU with 125 GB memory
- Benchmarks [Wu *et al.*, ICCAD22]

Benchmark	#Qubits	#Gate <sub>s</sub>	#Gate <sub>f</sub>
or	3	17	41
adder	4	23	63
qaoa5	5	22	54
Mod5mils_65	5	36	99
queko_05_0	16	37	97
queko_10_3	16	73	189
tof_4	7	55	143

Medium-scale benchmarks

Benchmark	#Qubits	#Gate <sub>s</sub>	#Gate <sub>f</sub>
queko_15_1	16	109	285
barenco_tof_4	7	72	208
tof_5	9	75	195
barenco_tof_5	9	104	304
mod_mult_55	9	91	251
vbe_adder_3	10	89	289
4gt13_92	5	66	186
rc_adder_6	14	140	424
16QBT_queko_100_0	16	1136	2416
16QBT_queko_100_1	16	1136	2416
16QBT_queko_900_0	16	10224	21744
16QBT_queko_900_1	16	10224	21744
20QBT_queko_100_0	20	1420	3020
20QBT_queko_100_1	20	1420	3020
20QBT_queko_500_0	20	7100	15100
20QBT_queko_500_1	20	7100	15100
54QBT_queko_05_0	54	192	408
54QBT_queko_05_1	54	192	408
54QBT_queko_900_0	54	34506	73386
54QBT_queko_900_1	54	34506	73386

Large-scale benchmarks

# Comparison: Medium-Scale Benchmarks

- Compare ours with SMT-based method [Wu *et al.*, ICCAD22]
  - **1.73X** total time steps overhead than ours

Benchmark	ICCAD22			Ours		
	Z	Ratio	Runtime (s)	Z	Ratio	Runtime (s)
or	24	1.14	0.17	21	1.00	0.07
adder	28	1.08	0.30	26	1.00	0.14
qaoa5	35	2.19	0.37	16	1.00	0.11
Mod5mils_65	53	1.33	0.99	40	1.00	6.17
queko_05_0	32	1.78	0.25	18	1.00	0.20
queko_10_3	68	2.83	1.60	24	1.00	4.09
tof_4	104	1.79	5.01	58	1.00	1.52
<b>Avg. Ratio</b>		<b>1.73</b>			<b>1.00</b>	

# Comparison: Large-Scale Benchmarks

- Compare ours with SMT-based method [Wu *et al.*, ICCAD22]
  - **1.82X** total time steps overhead than ours

Benchmark	ICCAD22			Ours		
	Z	Ratio	Runtime (s)	Z	Ratio	Runtime (s)
queko_15_1	99	2.11	1.50	47	1.00	1.68
barenco_tof_4	157	1.80	3.61	87	1.00	0.55
tof_5	141	2.35	3.15	60	1.00	0.33
barenco_tof_5	221	2.33	9.43	95	1.00	2.20
mod_mult_55	132	1.63	3.36	81	1.00	2.35
vbe_adder_3	143	1.59	4.28	90	1.00	2.08
4gt13_92	88	1.22	1.65	72	1.00	47.97
rc_adder_6	184	1.67	6.30	110	1.00	4.65

Benchmark	ICCAD22			Ours		
	Z	Ratio	Runtime (s)	Z	Ratio	Runtime (s)
16QBT_queko_100_0	712	2.24	11.79	318	1.00	18.54
16QBT_queko_100_1	708	2.07	11.90	342	1.00	16.39
16QBT_queko_900_0	6156	2.10	107.35	2926	1.00	166.76
16QBT_queko_900_1	5305	1.81	106.53	2953	1.00	150.46
20QBT_queko_100_0	769	1.68	12.51	457	1.00	27.89
20QBT_queko_100_1	856	1.72	12.88	499	1.00	27.30
20QBT_queko_500_0	3681	1.53	62.83	2403	1.00	225.30
20QBT_queko_500_1	3833	1.63	62.50	2377	1.00	259.04
54QBT_queko_05_0	99	1.98	1.03	50	1.00	9.81
54QBT_queko_05_1	116	1.93	1.02	60	1.00	9.19
54QBT_queko_900_0	18195	1.47	227.92	12375	1.00	1966.96
54QBT_queko_900_1	18035	1.56	227.01	11582	1.00	1931.58
<b>Avg. Ratio</b>		<b>1.82</b>			<b>1.00</b>	



# Comparison: w/o and w/ Divide-and-Conquer Approach

- The results can only be generated by including the divide-and-conquer method (runtime limit of 3600 seconds)
- Our divide-and-conquer method achieves a 30.56x speedup with only a 3% average loss in solution quality
  - Achieves a **245.41x** speedup on **twenty large-scale benchmarks** by the Penalized Average Runtime PAR-2 score

Benchmark	w/o divide-and-conquer approach				w/ divide-and-conquer approach			
	Z	Ratio	Runtime (s)	Ratio	Z	Ratio	Runtime (s)	Ratio
queko_15_1	45	0.96	48.95	29.14	47	1.00	1.68	1.00
barenco_tof_4	87	1.00	13.24	23.99	87	1.00	0.55	1.00
tof_5	58	0.97	3.87	11.72	60	1.00	0.33	1.00
barenco_tof_5	95	1.00	165.17	75.18	95	1.00	2.20	1.00
mod_mult_55	74	0.91	87.69	37.28	81	1.00	2.35	1.00
vbe_adder_3	90	1.00	12.61	6.08	90	1.00	2.08	1.00
<b>Avg. Ratio</b>		<b>0.97</b>		<b>30.56</b>		<b>1.00</b>		<b>1.00</b>

# Outline

---



- Introduction to Trapped-Ion Systems



- Qubit Mapping Problem



- Proposed Approach



- Experimental Results



- **Conclusions**

# Conclusions

---

- We propose a new coupling constraint graph with a multiple-qubit net based on the Mølmer-Sørensen gate and its related hardware architecture
  - Maintain the fidelity of the quantum circuit
  - Mitigate the occurrence of crosstalk effects
- We present an SMT-based qubit mapping algorithm to find an optimal qubit-mapping solution for medium-scale problems on the trapped-ion systems
- We present an effective divide-and-conquer method to scale our algorithm and maintain the quality of the SMT solutions for large-scale problems
- Experimental results have shown the effectiveness of our algorithm compared with the state-of-the-art work
  - Achieve an average 44% total time steps reduction for all benchmarks



**Thank You!**

National Taiwan University