Logical-Depth-Oriented Reversible Logic Synthesis

Mona Arabzadeh, Morteza Saheb Zamani, Mehdi Sedighi and Mehdi Saeedi

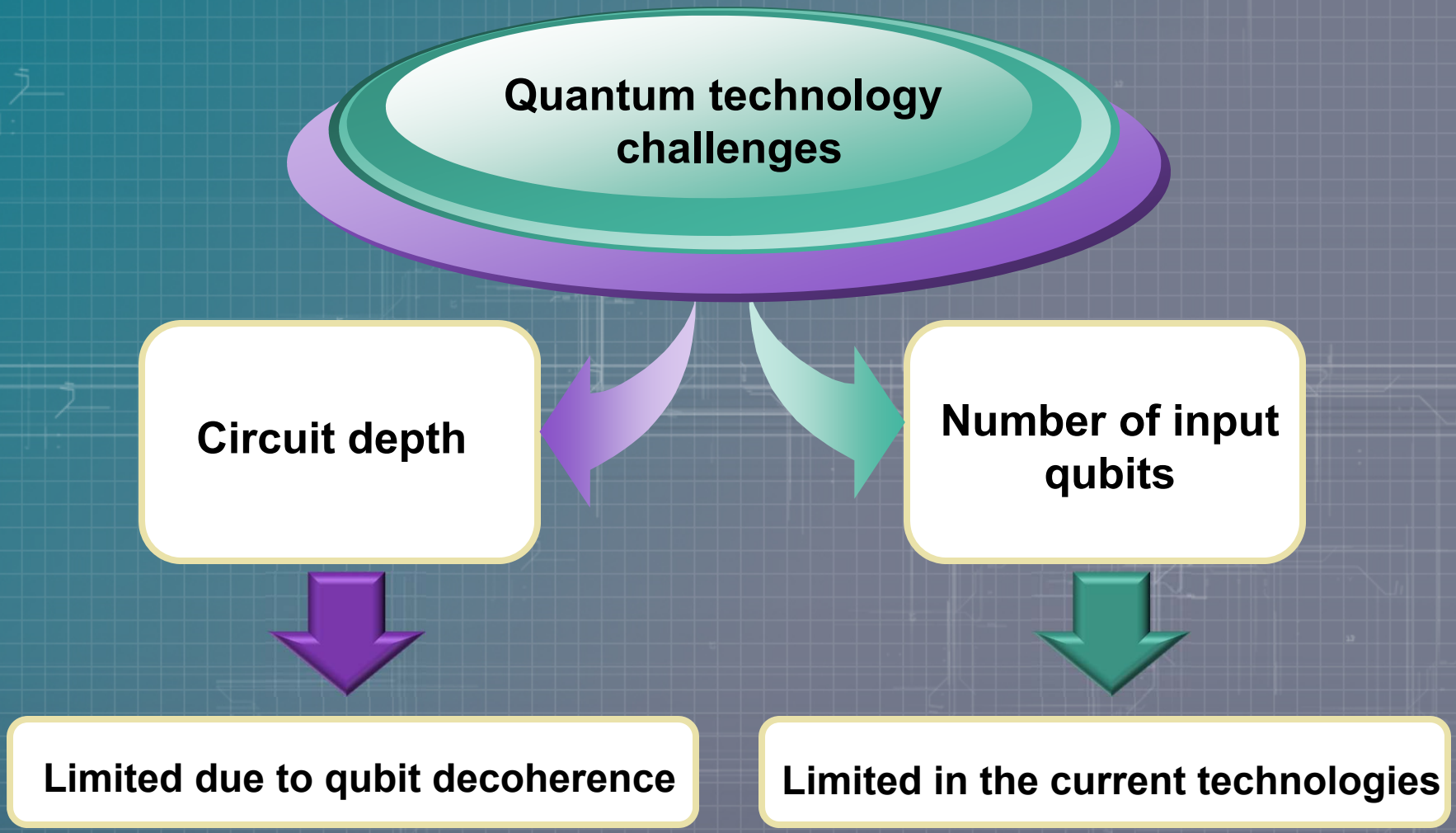
Email: {m.arabzadeh, szamani, msedighi, msaeedi}@ aut.ac.ir

Quantum Design Automation Lab
Computer Engineering and Information Technology Department

Amirkabir University of Technology
Tehran, Iran

Contents

Introduction



Introduction

Time Complexity

Depth:

A parameter to consider the execution time of a circuit: ***Number of Logical levels***

Gates that can be performed in parallel consider as one *Logical Level*

Space Complexity

Ancillae:

Helper qubits which are added to the circuit to reduce ***Circuit Depth***

"Parallel quantum computation and quantum codes", *SIAM Journal on computing, 2001* [2]

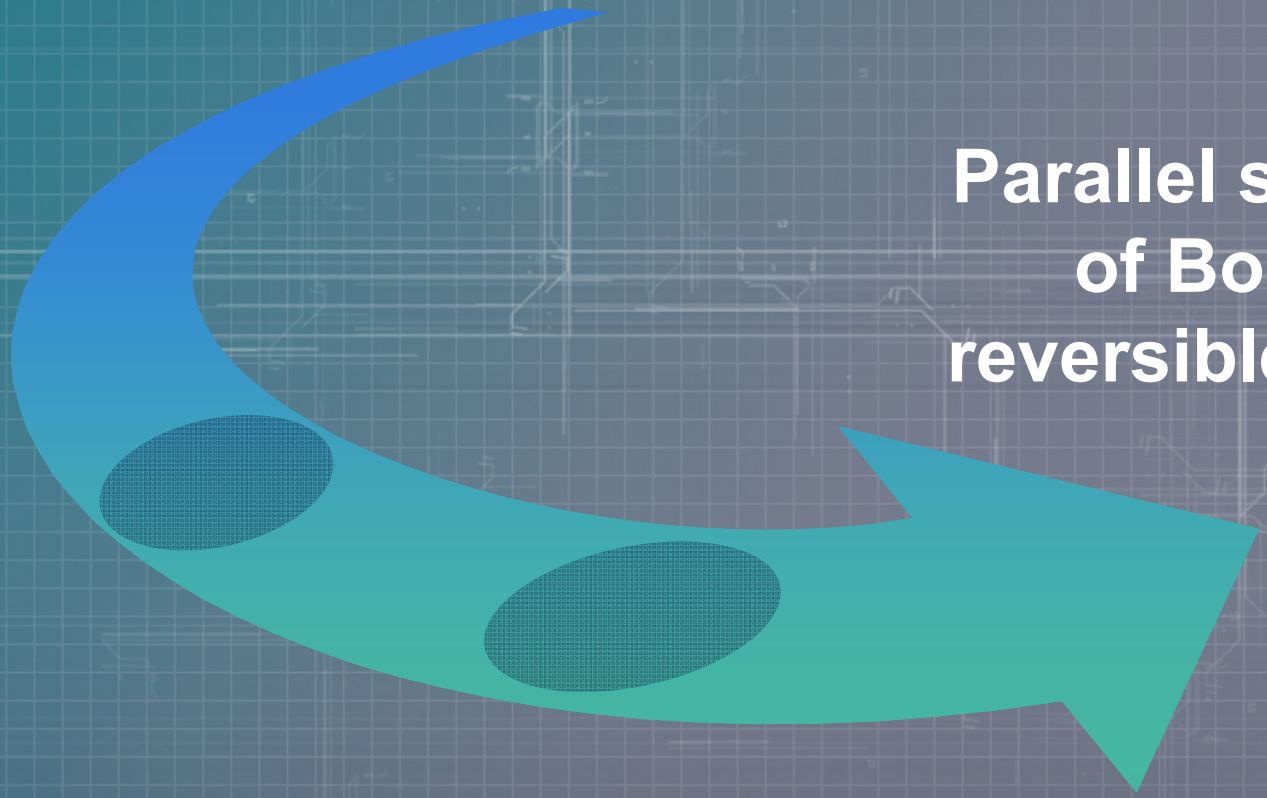
NC
Efficient parallel class in classical computation

QNC
Quantum analog of the class NC

Logarithmic depth circuits by adding *ancillae*

Boolean Reversible circuits

Applications in quantum computing



Parallel synthesis of Boolean reversible circuits

Basic Concepts

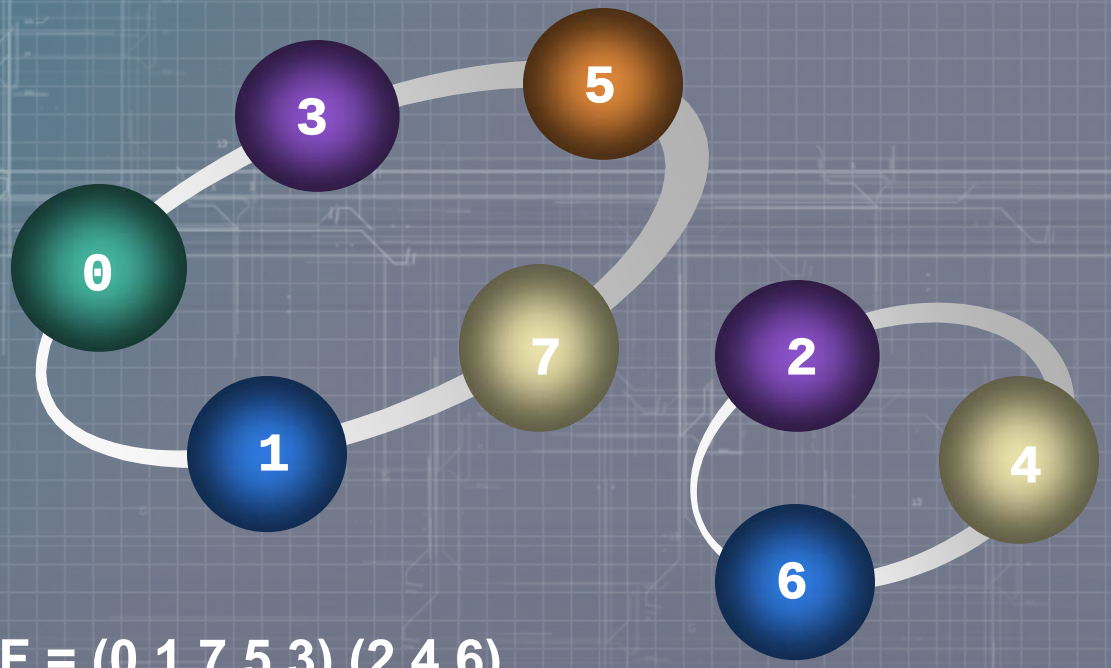
- *Boolean reversible function*
  - *n-input, n-output,*
  - *Unique output assignment*
  - *Example: a 3-input, 3-output*
    - *function: 2,7,0,1,6,3,4,5*

a_1	a_2	a_3		f_1	f_2	f_3	
0	0	0	**0**	0	1	0	**2**
0	0	1	**1**	1	1	1	**7**
0	1	0	**2**	0	0	0	**0**
0	1	1	**3**	0	0	1	**1**
1	0	0	**4**	1	1	0	**6**
1	0	1	**5**	0	1	1	**3**
1	1	0	**6**	1	0	0	**4**
1	1	1	**7**	1	0	1	**5**

Basic Concepts

- ***Permutation function:***
  - *Every permutation function can be written uniquely, except for the order, as a product of disjoint cycles*

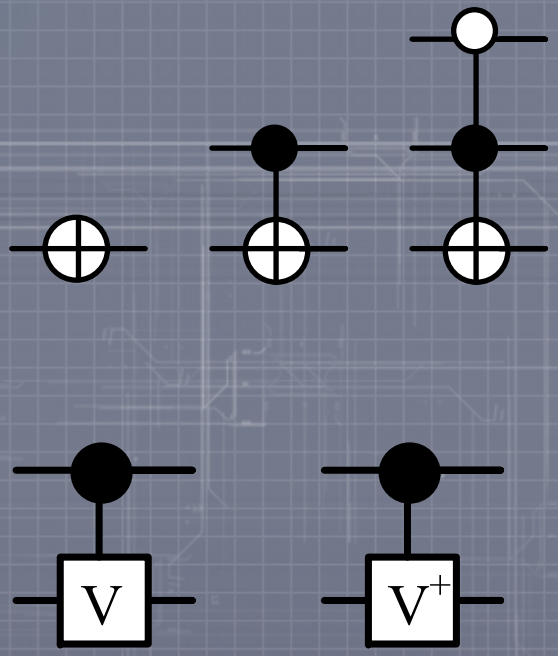
a_1	a_2	a_3		f_1	f_2	f_3	
0	0	0	**0**	0	0	1	**1**
0	0	1	**1**	1	1	1	**7**
0	1	0	**2**	1	0	0	**4**
0	1	1	**3**	0	0	0	**0**
1	0	0	**4**	1	1	0	**6**
1	0	1	**5**	0	1	1	**3**
1	1	0	**6**	0	1	0	**2**
1	1	1	**7**	1	0	1	**5**



F = (0,1,7,5,3) (2,4,6)

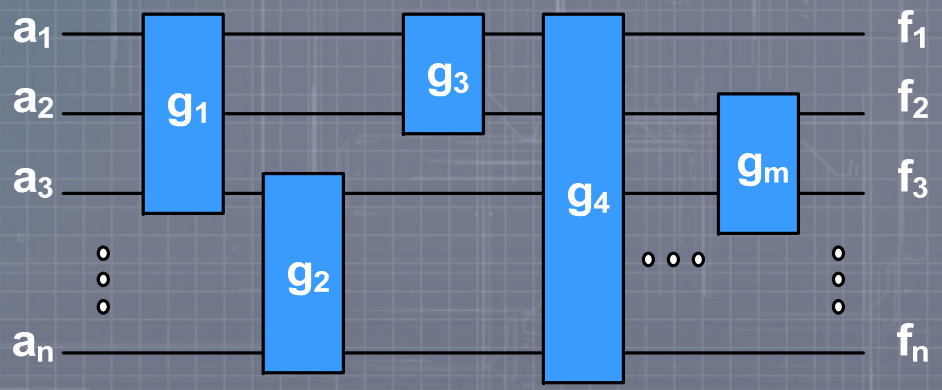
Basic Concepts

- *Reversible gate*
- *Various reversible gates*
  - *C^mNOT gates*
    - *NOT, CNOT, C^2NOT (Toffoli), …*
      - *Positive controls*
      - *Negative controls*
  - *Controlled-V*
  - *Controlled-V+*



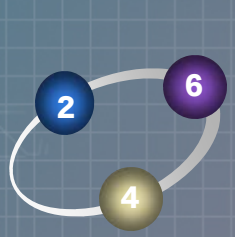
Basic Concepts

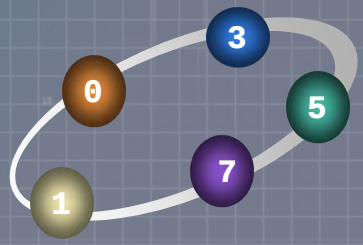
- *Elementary gates:*
  - *NOT, CNOT, controlled-V, and controlled-V+*
- *Quantum cost:*
  - *The number of elementary gates required for simulating a given gate*
- *Reversible circuit:*
  - *A set of reversible gates*



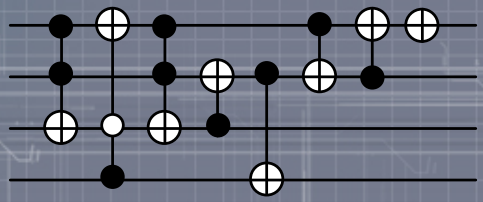
Reversible circuits: *Synthesis*

High-level Description

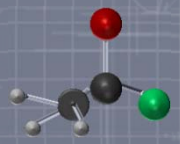




Gate-level circuits



Physical Implementation



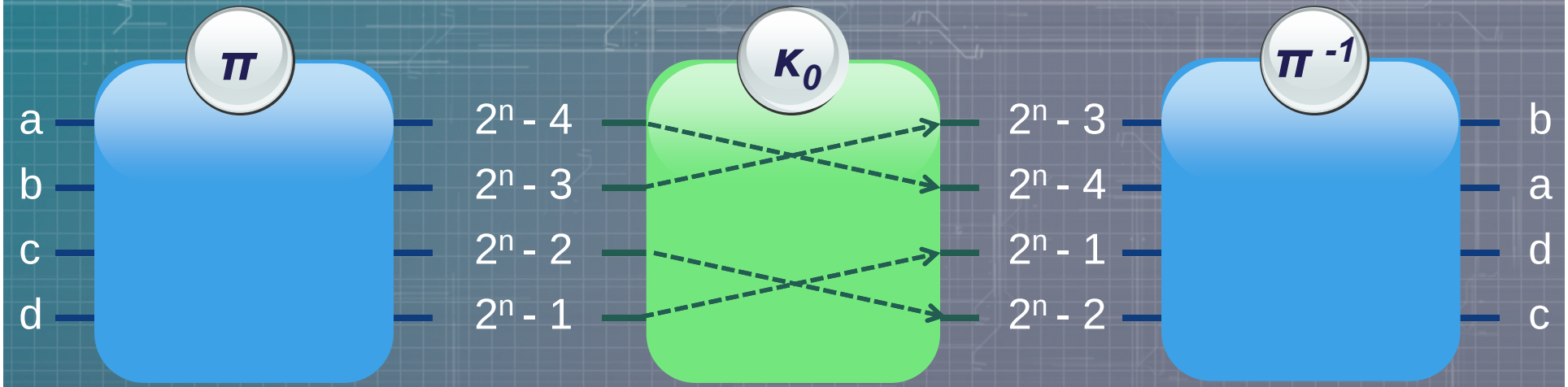
Previous work:

"Synthesis of Reversible Logic Circuits", *TCAD, 2003.* [12]

- **Decomposes every cycle with length larger than two to a set of pairs of disjoint *cycles with length two* :*Transposition***

$$(b_0, b_1, b_2, \ldots, b_k) = (b_0, b_1)(b_{k-1}, b_k)(b_0, b_2, b_3, \ldots, b_{k-1})$$

- **Synthesizes each disjoint transposition pair *(a, b)(c, d) using $\pi K_0 \pi^{-1}$ circuit***

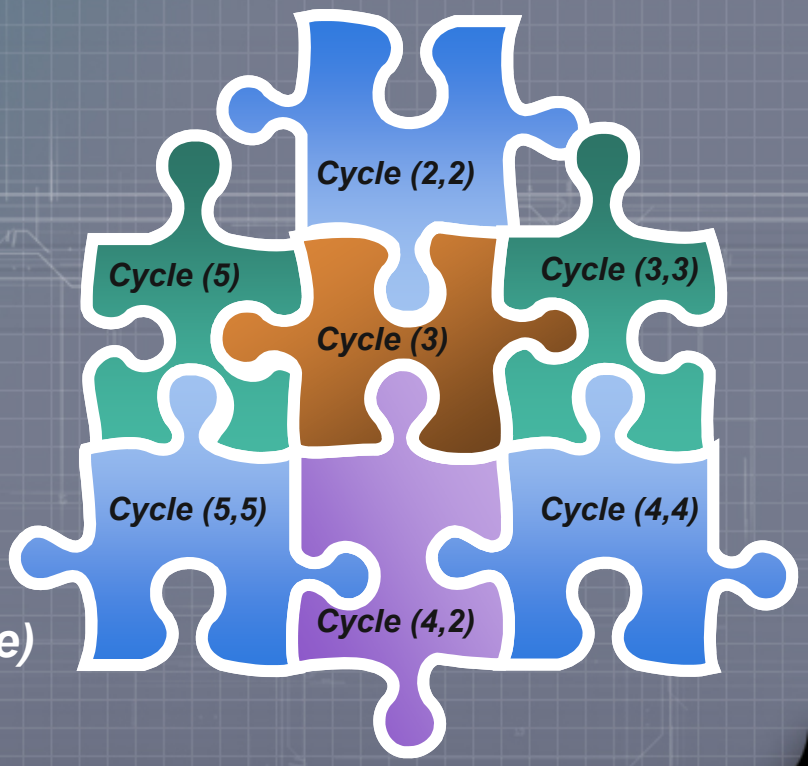


Previous work:

"Reversible circuit synthesis using a cycle-based approach", *JETC, 2010.* [9]

- *k-cycle-based synthesis algorithm: set of cycles of length less than 6*
- *The k-cycle method consists of two main parts:*
  - *Synthesis of Seven Building Blocks (Elementary Cycles):*
    - *a pair of 2-cycles*
    - *a single 3-cycle*
    - *a pair of 3-cycles*
    - *a single 5-cycle*
    - *a pair of 5-cycles*
    - *a pair of 4-cycles*
    - *a single 2-cycle (4-cycle)*

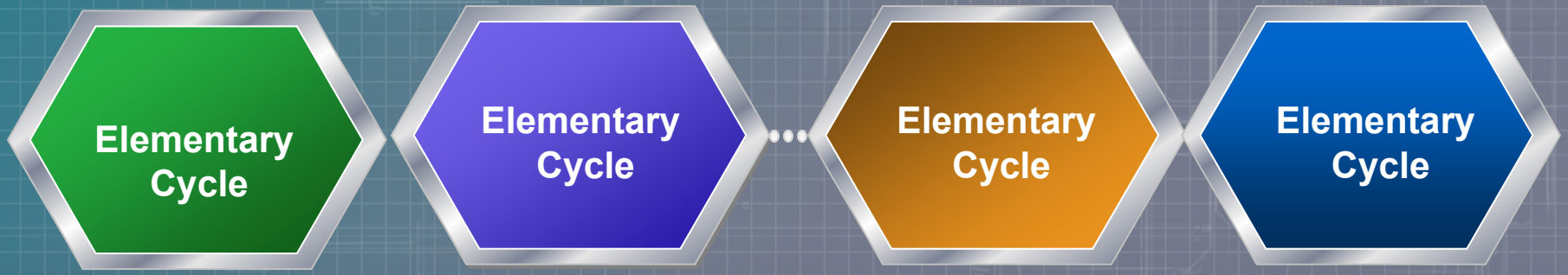
    followed by a single 4-cycle (2-cycle)



Previous work:

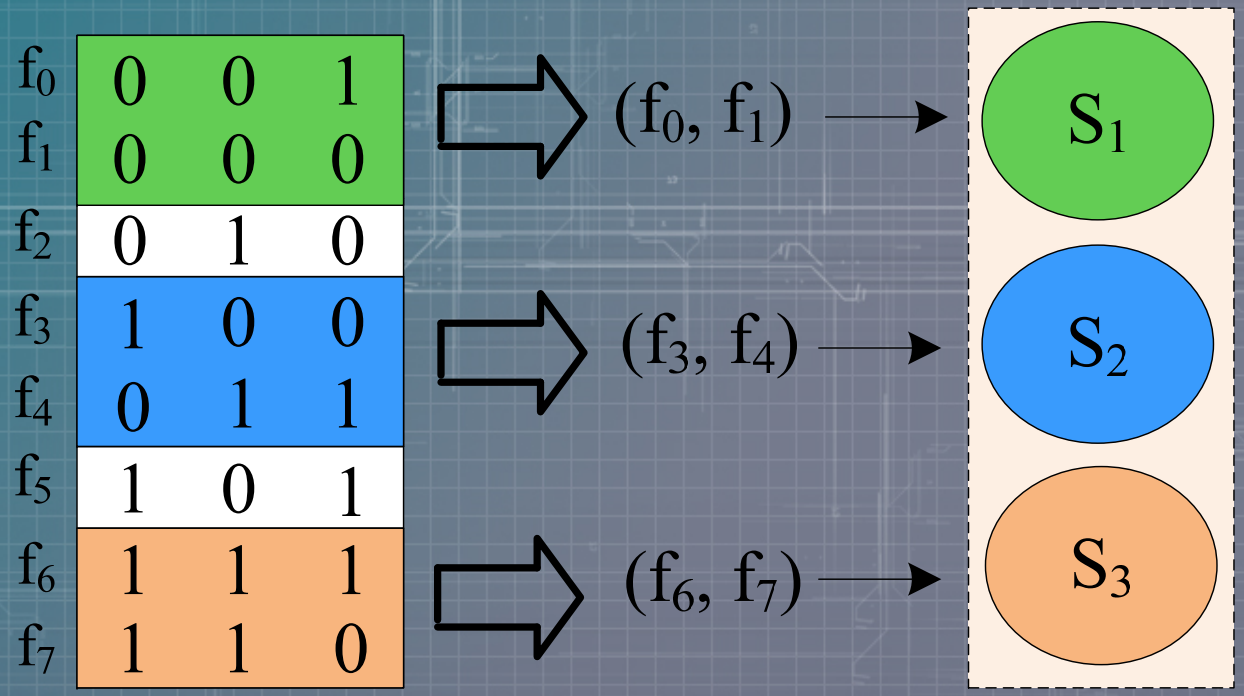
"Reversible circuit synthesis using a cycle-based approach", *JETC, 2010.* [9]

- ***k-cycle-based synthesis algorithm: set of cycles of length less than 6***
- ***The k-cycle method consists of two main parts:***
  - ***Decomposition:***
    - **A given k-cycle should be decomposed into a set of elementary cycles**
    - **It was shown that an arbitrary permutation can be decomposed into a set of elementary cycles:**

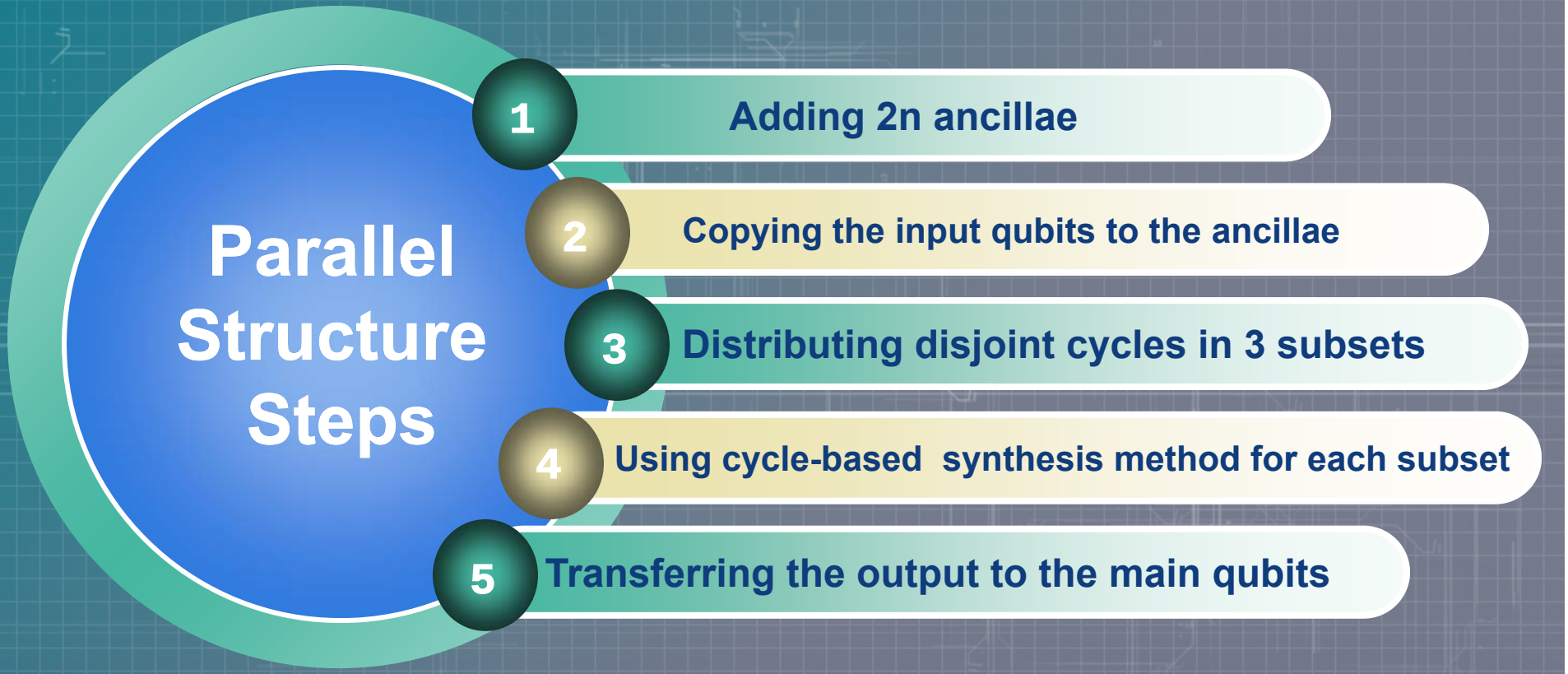


Parallel Cycle-Based Synthesis Method

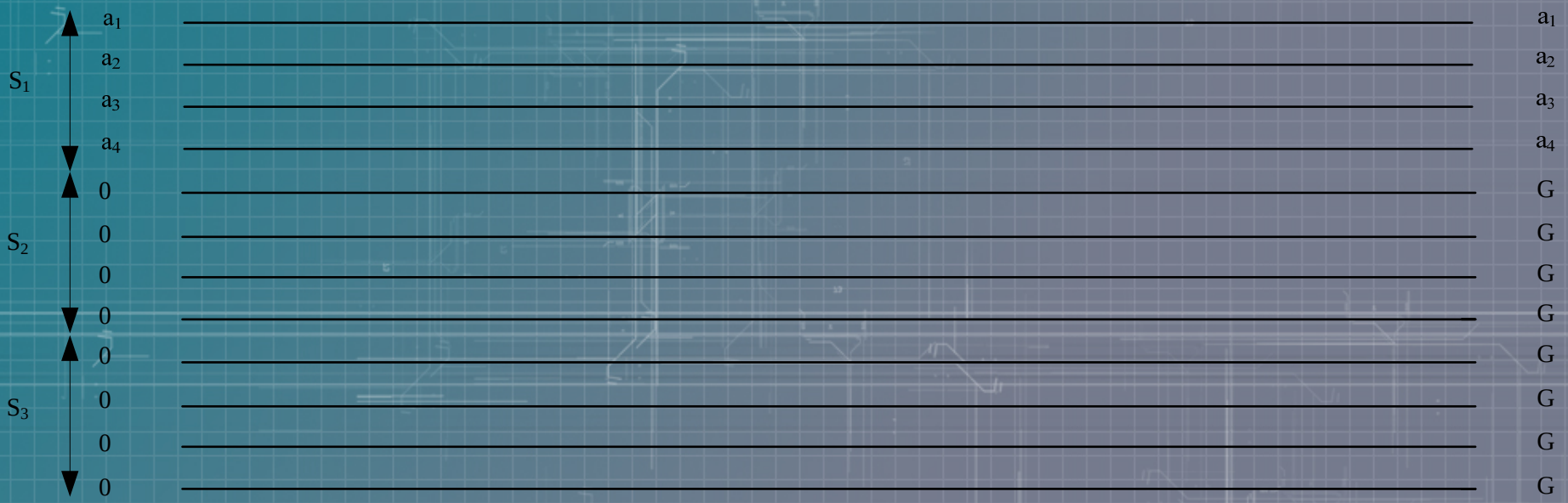
- **Goal:** ***To find distinct subsets in input specification***
  - ***To perform them in parallel***



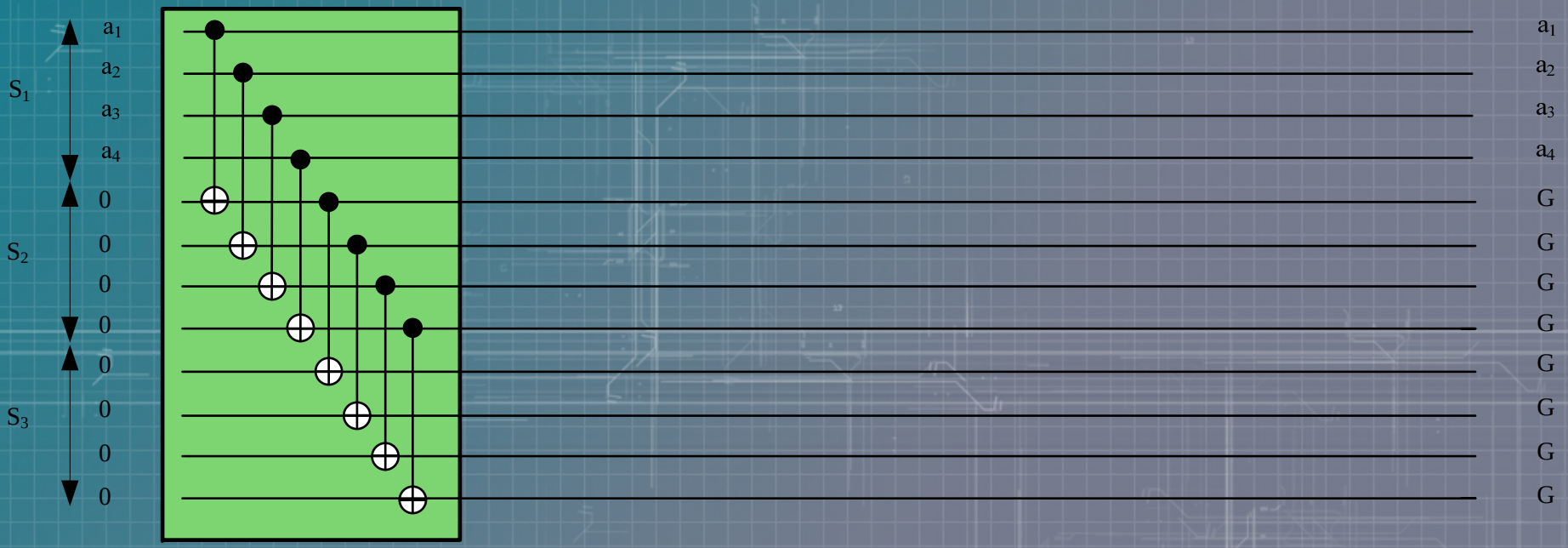
Parallel Structure



1 Adding 2n ancillae

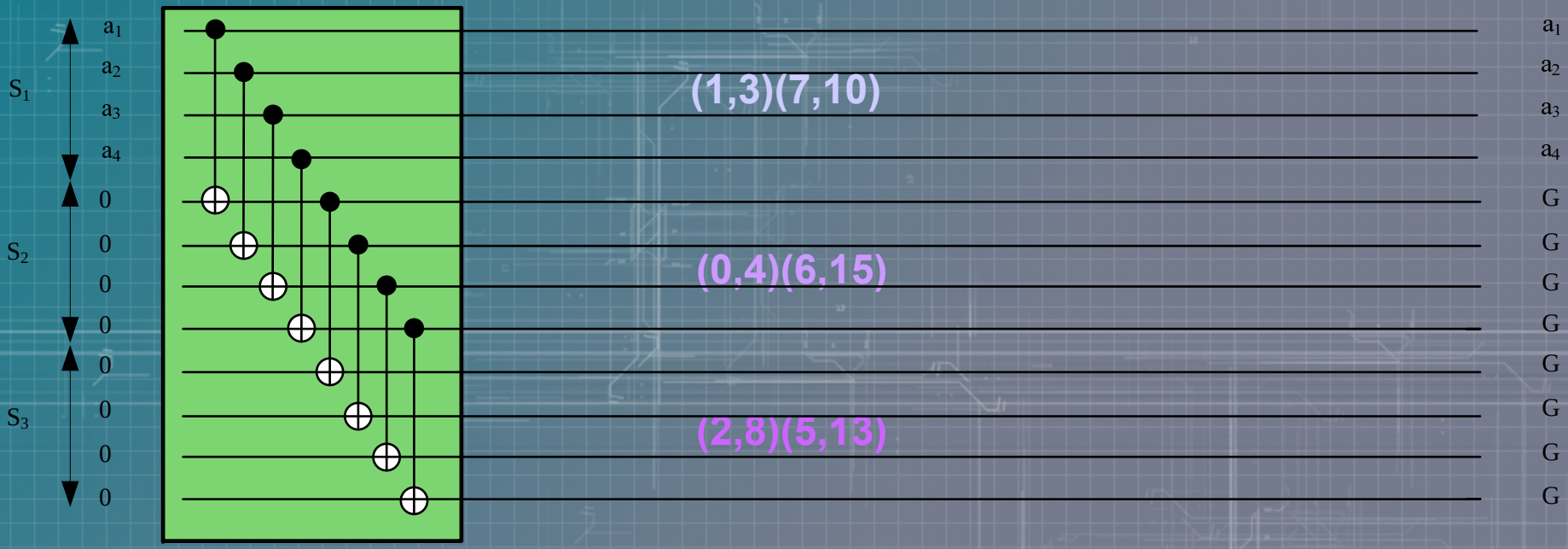


2 Copying the input qubits to the ancillae



The Preparation Circuit

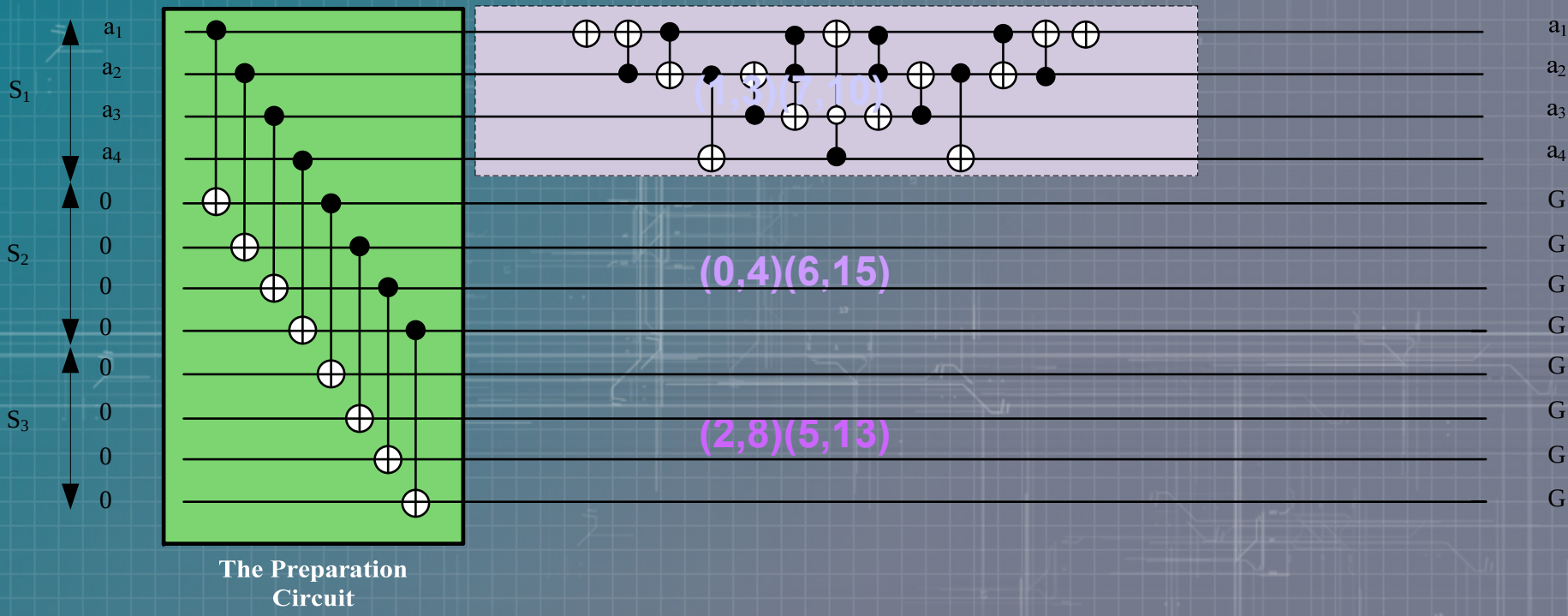
3 Distributing disjoint cycles in 3 subsets



The Preparation Circuit

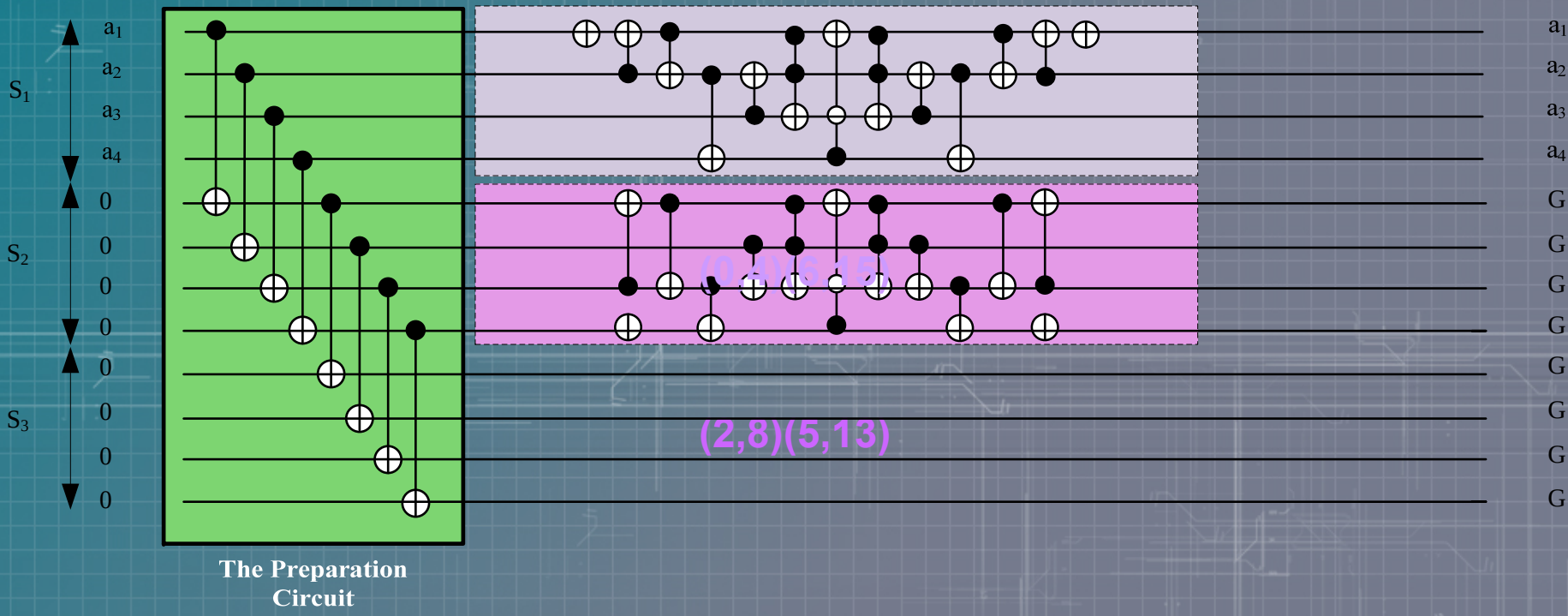
$$F = (1,3)(7,10)(0,4)(6,15)(2,8)(5,13)$$

4 Using cycle-based synthesis method for each subset



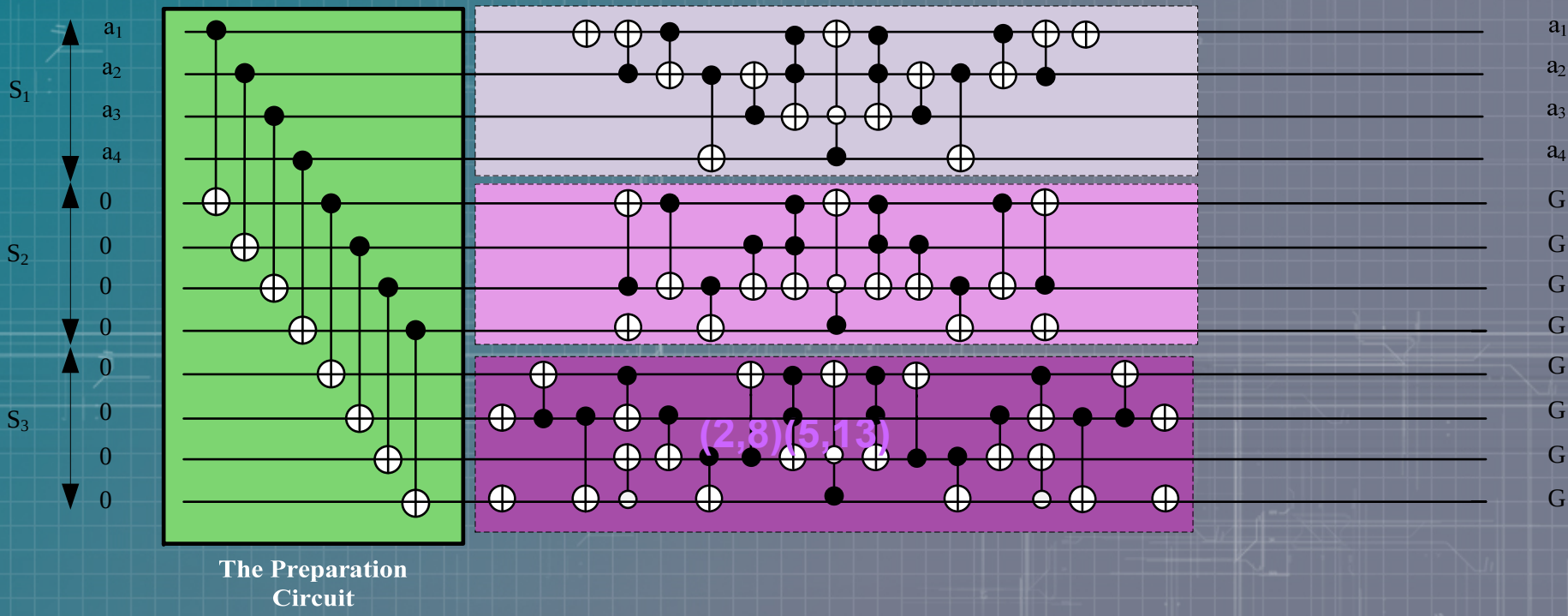
$$F = (1,3)(7,10)(0,4)(6,15)(2,8)(5,13)$$

4 Using cycle-based synthesis method for each subset



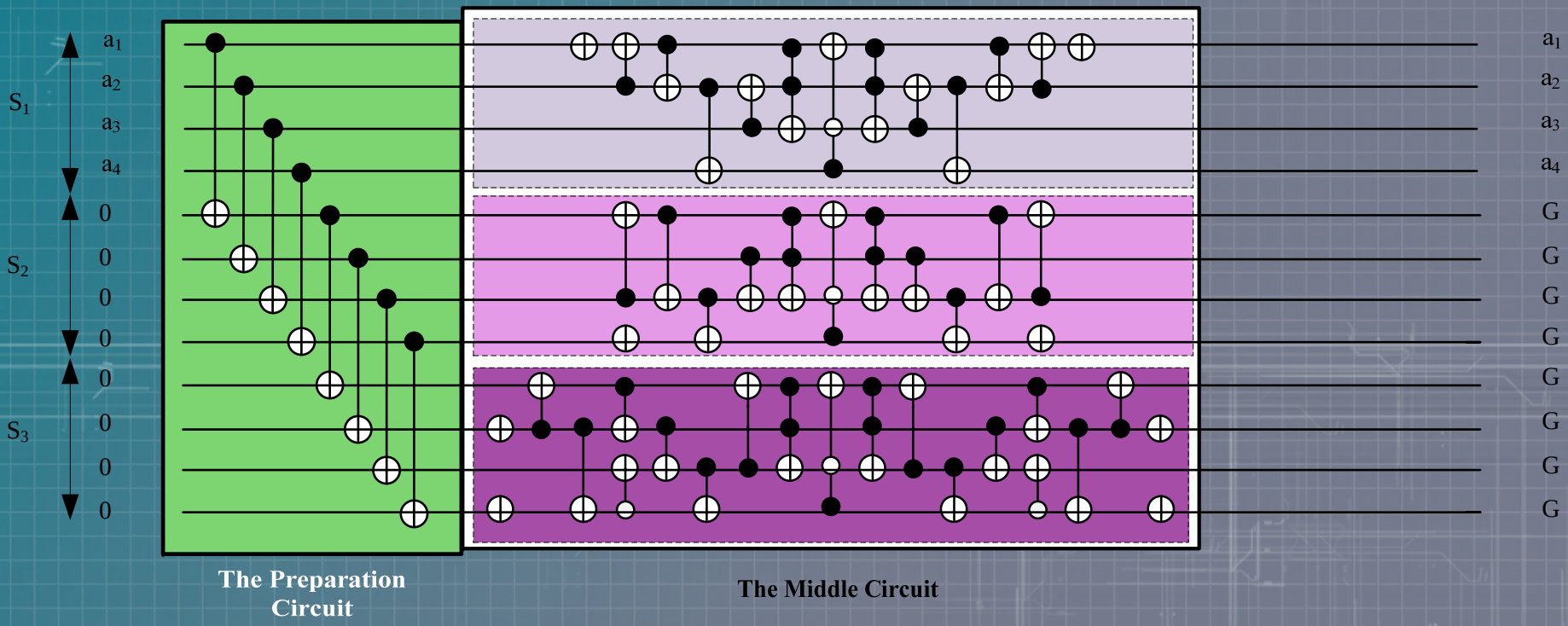
$$F = (1,3)(7,10)(0,4)(6,15)(2,8)(5,13)$$

4 Using cycle-based synthesis method for each subset



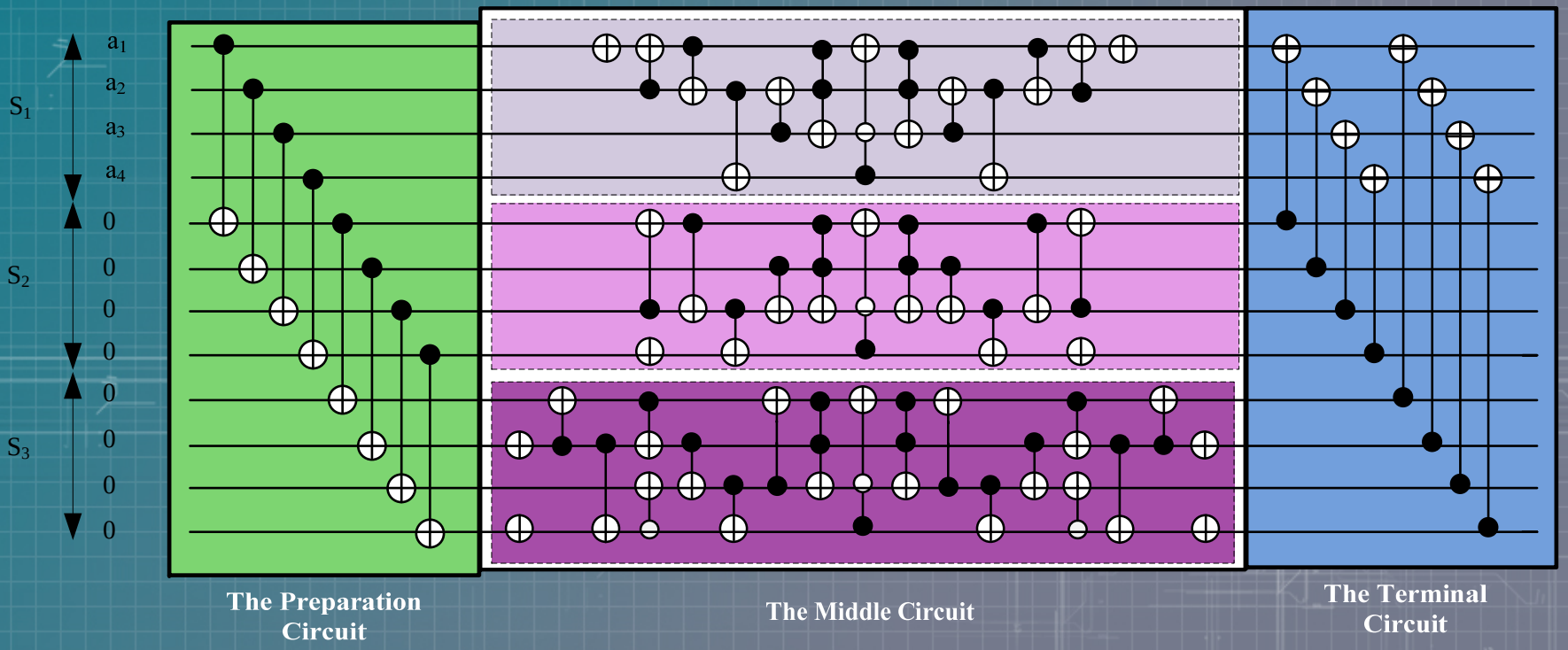
$$F = (1,3)(7,10)(0,4)(6,15)(2,8)(5,13)$$

4 Using cycle-based synthesis method for each subset



$$F = (1,3)(7,10)(0,4)(6,15)(2,8)(5,13)$$

5 Transferring the output to the main qubits



$$F = (1,3)(7,10)(0,4)(6,15)(2,8)(5,13)$$

Cycle-Based Synthesis

- ***Three main differences with [9]:***

Sequential-Cycle Structure

Cycle Operating Interval

Using Negative Controls

To improve

Quantum Cost

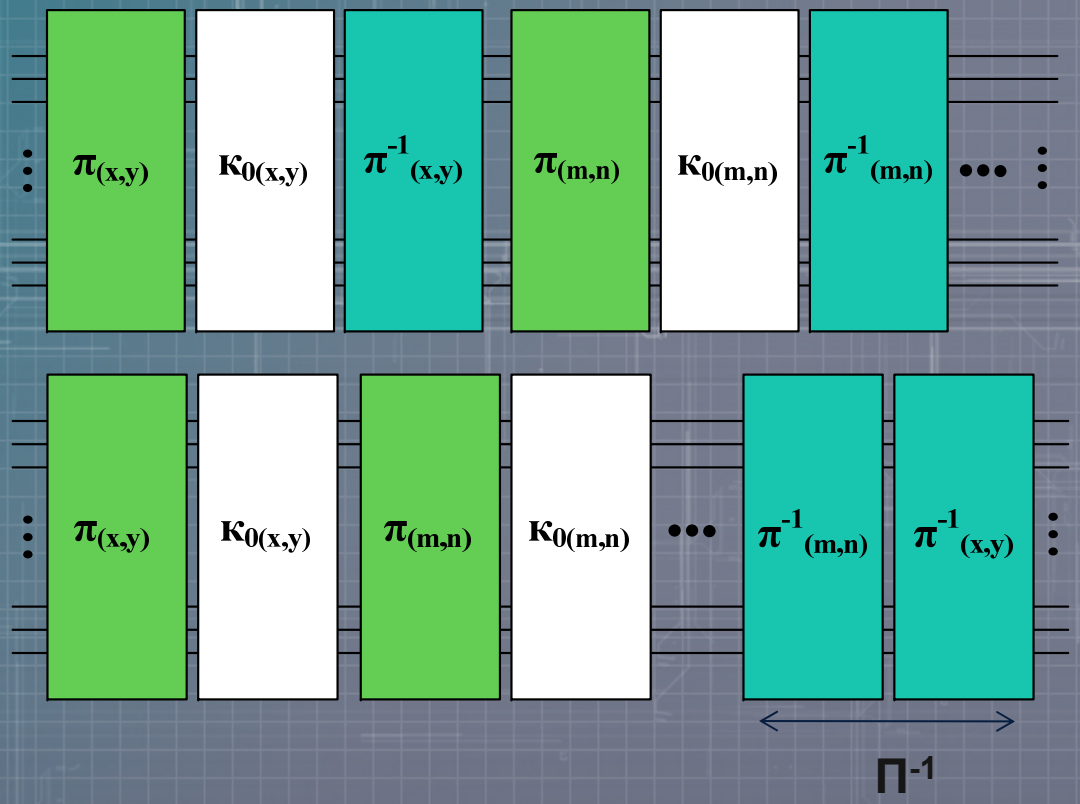
and

Depth

Cycle-Based Synthesis

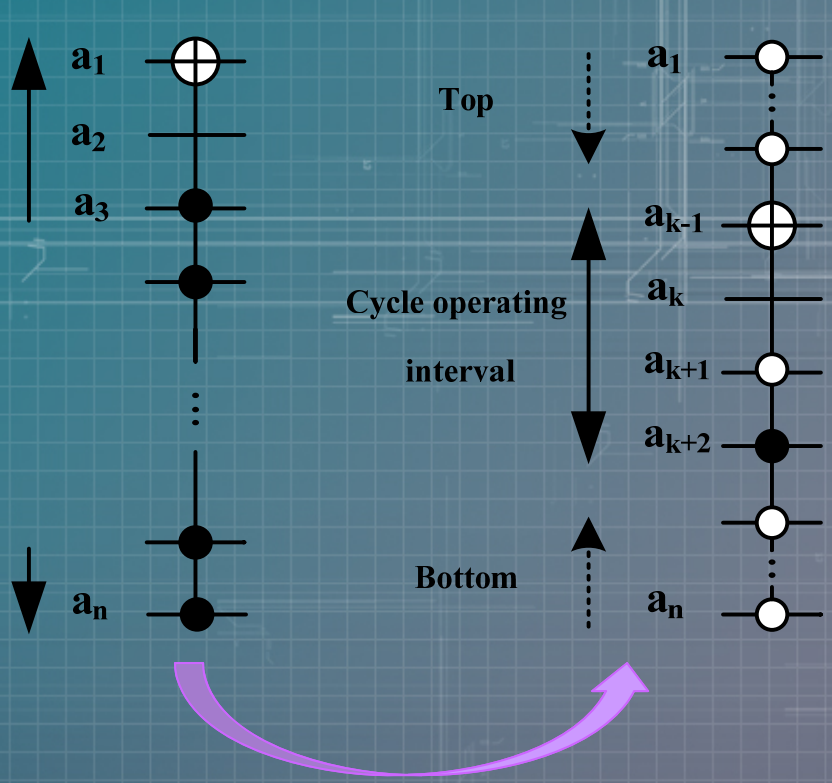
Sequential-Cycle Structure

Structure in [9], [12]

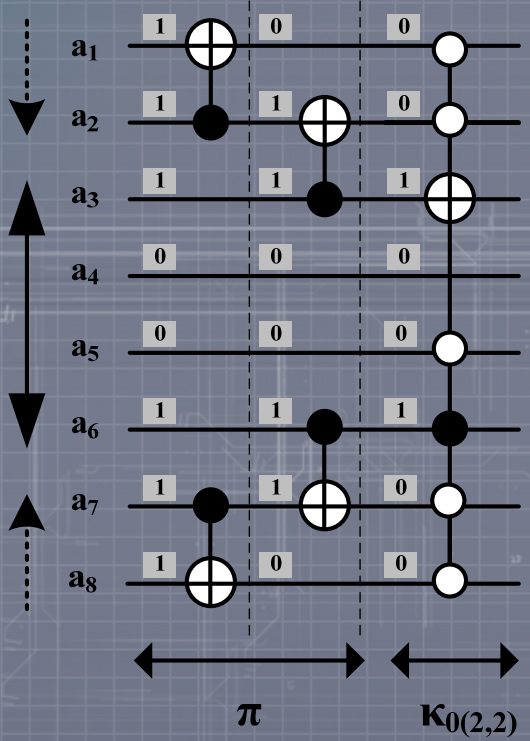


Cycle-Based Synthesis

Cycle Operating Interval



Example



Cycle-Based Synthesis

Using Negative Controls:

Elementary Cycle	Length	Total cost	Cost/Length	Cost/Length [9]
(2,2)	4	32n-66	8n-16.5	8.5n-16
(3)	3	30n-80	10n-26.6	10.7n-27.3
(3,3)	6	36n-60	6n-10	6.3n-15.3
(4,2)	6	48n-152	8n-25.3	8.3n-20.3
(4,4)	8	52n-136	6.5n-17	7n-15.7
(5)	5	58n-140	11.6n-28	12n-26
(5,5)	10	56n-92	5.6n-9.2	6.4n-5.4

Experimental Results

Benchmark	n	Garbage	QC	Depth	QC [9]	Depth [9]	Imp. QC	Imp. Depth
hwb8	8	16	7400	2316	6940	6205	-6.6	62.6
hwb9	9	18	15376	4800	16173	14312	4.9	66.4
hwb10	10	20	38388	11787	35618	31908	-7.7	63.0
hwb11	11	22	89434	27079	90745	81440	1.4	66.7
hwb12	12	24	208992	64727	198928	184210	-5.0	64.8
hwb13	13	26	431054	131166	436305	397147	1.2	66.9
nth_prime7	7	14	3108	1651	3172	2782	2.0	40.6
nth_prime9	9	18	17744	15202	17975	15767	1.2	3.5
nth_prime10	10	20	43026	14446	40301	35511	-6.7	59.3
nth_prime11	11	22	93548	40316	95433	85093	1.9	52.6
nth_prime12	12	24	217294	140507	208227	187006	-4.3	24.8
nth_prime13	13	26	469422	281129	474660	431644	1.1	34.8
Average							-1.3	50.5

Future Directions

- ***Working on the improvement of the resulting synthesized circuits***
  - ***By Better distribution algorithm***
    - *In terms of **depth** or **quantum cost***
  - ***By developing level compaction algorithms***

Conclusions

- *A Boolean reversible logic synthesis method with logical-depth optimization is proposed*
- *Cycle representation was chosen to distinct parallel subsets in input specification*
- *The cycle-based synthesis method equipped with depth-consideration was used*
- *The number of logical levels can be improved by up to 67% by adding 2n ancillae with less than 2% increase in the quantum cost on average*


Thanks for your attention!
Questions …