



Article

A Transformation-Based Quantum Physical Synthesis Approach for Nearest-Neighbor Architectures

Golriz Hoseinimanesh  and Naser Mohammadzadeh * 

Quantum Architectures and Computation Group (QACG), Department of Computer Engineering, Shahed University, Tehran 33191-18651, Iran; g.hoseinimanesh@shahed.ac.ir

* Correspondence: mohammadzadeh@shahed.ac.ir

Abstract: The physical synthesis concept for quantum circuits, the interaction between synthesis and physical design processes, was first introduced in our previous work. This concept inspires us to propose some techniques that can minimize the number of extra inserted SWAP operations required to run a circuit on a nearest-neighbor architecture. Minimizing the number of SWAP operations potentially decreases the latency and error probability of a quantum circuit. Focusing on this concept, we present a physical synthesis technique based on transformation rules to decrease the number of SWAP operations in nearest-neighbor architectures. After the qubits of a circuit are mapped onto the physical qubits provided by the target architecture, our procedure is fed by this mapping information. Our method uses the obtained placement and scheduling information to apply some transformation rules to the original netlist to decrease the number of extra SWAP gates required for running the circuit on the architecture. We follow two policies in applying a transformation rule, greedy and simulated-annealing-based policies. Simulation results show that the proposed technique decreases the average number of extra SWAP operations by about 20.6% and 24.1% based on greedy and simulated-annealing-based policies, respectively, compared with the best in the literature.



Citation: Hoseinimanesh, G.; Mohammadzadeh, N. A Transformation-Based Quantum Physical Synthesis Approach for Nearest-Neighbor Architectures. *Quantum Rep.* **2021**, *3*, 435–443. <https://doi.org/10.3390/quantum3030028>

Academic Editor: Lucas Lamata

Received: 24 June 2021

Accepted: 11 August 2021

Published: 15 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: quantum circuits; transformation rules; nearest-neighbor architectures; simulated annealing

1. Introduction

Quantum computing is part of computer science research that focuses on the development of computers based on quantum theory that examines the nature and behavior of matter [1]. The invention of quantum computers represents a considerable leap in the ability of computer processing [2]. A quantum computer gains extreme processing power by following the laws of quantum physics. This improvement is achieved through the ability to have several states and execute different commands using all possible permutations at a single time. There is a fundamental difference between classical computers and next-generation quantum computers. A classical computer performs preset commands based on classic physics rules, but a quantum computer is a device that uniquely identifies a physical phenomenon based on quantum mechanics to basically detect a new mode of information processing. In a typical computer, the information is fed into a series of code bits, and these bits are manipulated by the Boolean logic gates that are applied serially to obtain the final result. In a quantum computer, instead of using transistors and conventional computer circuits, atoms and other fine particles are used to process information. An atom can act as a quantum bit of memory on a computer, and transfer of information from one location to another can also be achieved by optical fiber [3,4].

Physical design is one of the main processes of the quantum circuit design. The other process is synthesis. The synthesis and physical design processes were traditionally performed separately because the integration of two processes into one monolithic process makes the complexity of design process unmanageable [5]. However, without interaction between the physical design and the synthesis processes, the generated layout was not

good. Addressing this issue, the physical synthesis concept [5], the interaction between synthesis and physical design processes, was introduced in [6] for quantum circuits. The physical synthesis modifies the netlist or layout considering the layout information to improve the objectives (e.g., latency) or meet the design constraints.

In most physical platforms, performing quantum gates on non-adjacent qubits is error prone and sometimes is hindered by the target technology [7]. Hence, quantum gates are restricted to be performed on neighboring qubits. To perform a gate, a communication channel of SWAP gates is needed to be constructed if the qubits in the physical environment are not adjacent. These additional SWAP gates increase the latency and error probability of the original quantum circuit [8]. Therefore, the fewer SWAP gates are added, the faster a quantum circuit is executed. This problem is an NP-hard problem [9]. Several heuristic approaches have been already proposed that tried to map qubits of a circuit on nearest-neighbor architectures in a way that the number of SWAP gates were minimized [10–13]. Focusing on this problem, we propose a transformation-based physical synthesis technique to decrease the number of SWAP gates. This procedure takes the mapping information and uses transformation rules to substitute some parts of the netlist so that the circuit needs fewer SWAP gates to be inserted.

The rest of the paper is organized as follows: The Section 2 overviews the prior works. Section 3 contains the main idea of the transformation method as well as our explanation of our strategy for applying this method. Sections 4 and 5 show the experimental results and conclusion, respectively.

2. Related Work

This section is divided into two parts. In the first part, the works that use transformation rules for logic synthesis or post-synthesis optimization are mentioned. In the second part, the research done on the physical synthesis in quantum circuits is stated.

The work performed in [14] presented the idea of local transformation of reversible circuits. While the main purpose of this work was not post-synthesis optimization, its idea was extended by other researchers to improve circuit costs. The authors defined a canonical form for circuits in the NCT library and introduced a complete set of rules to transform any NCT-constructible circuit into its canonical form, which may or may not be compact. Shende et al. [15] proposed a new rule for simplification of reversible circuits in the NCT library. The concept of applying a rule set was extended in [16], where the authors introduced several transformation rules based on a set of predefined patterns called templates. In [17], template matching with up to six gates was used in post-synthesis optimization. Similarly, the Toffoli–Fredkin templates were explored in [18,19]. Toffoli templates were expanded in [20,21] by the addition of all templates of size 7 (five templates) and a set of templates of size 9 (four templates). Maslov et al. [22] used templates and rules to simplify quantum circuits, such as a 10-gate quantum network for a 3-qubit full adder. Lu et al. [23] proposed equivalent circuits. Equations were introduced in [23] to simplify the quantum circuit as much as possible. Saeedi et al. [24] extended the templates to work with up to three SWAP gates. Arabzadeh et al. [25] proposed a set of simplification rules in terms of positively and negatively controlled Toffoli gates. An optimization in [26] used a window to select potential subcircuits first. Abdessaied et al. [27] used Boolean satisfiability for template matching. In the work performed in [28], a systematic method of generating all templates with a given number of lines was presented. Bandyopadhyay et al. [29] proposed a post-synthesis optimization technique for reversible circuits based on newly defined templates. In their work, templates are applied on a specific order over the input circuit and exhaustively search through the circuit for possible replacements.

The physical synthesis concept in quantum circuits was introduced in [6] for the first time and some physical synthesis techniques proposed for ion trap technology in the next papers [30–33].

Most of the above-mentioned papers are at the circuit level. However, our focus is on the physical design level after the physical mapping is done in nearest-neighbor architectures.

3. Transformation-Based Physical Synthesis

The physical synthesis modifies the netlist or layout considering the layout information to improve the objectives (e.g., latency or error probability) or meet the design constraints. The proposed scheme in this paper uses transformation rules to achieve the desired improvements. These templates are composed of gates implementable in nearest-neighbor technologies [34]. To apply those to the circuit, we present a flow. In the rest of this section, we first introduce transformation rules and then present the flow.

Transformation rules consist of two equivalent sequences of gates. The first sequence of gates is matched with a section of the circuit for simplification, and it is replaced with its equivalent one when a match is found. Figure 1 shows our transformation rules used in our approach. These rules are easily verifiable by checking their equivalent matrices.

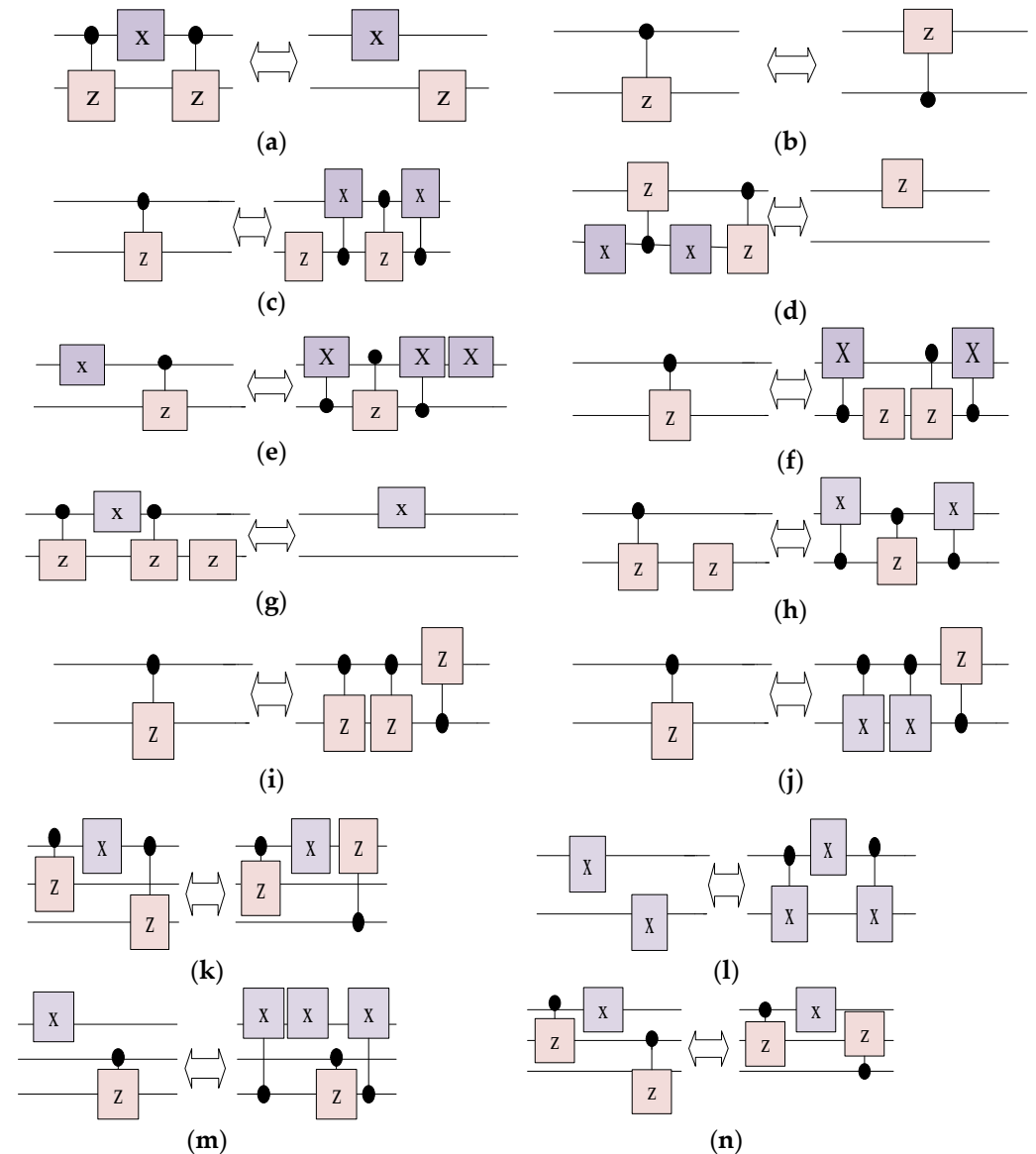


Figure 1. Transformation rules used in this paper.

Figure 2 shows the proposed flow for applying the transformation approach. The flow uses an optimized gate-level netlist as an input and generates a scheduled mapping. We use the mapping approach proposed in [11] to map the input netlist onto the architecture.

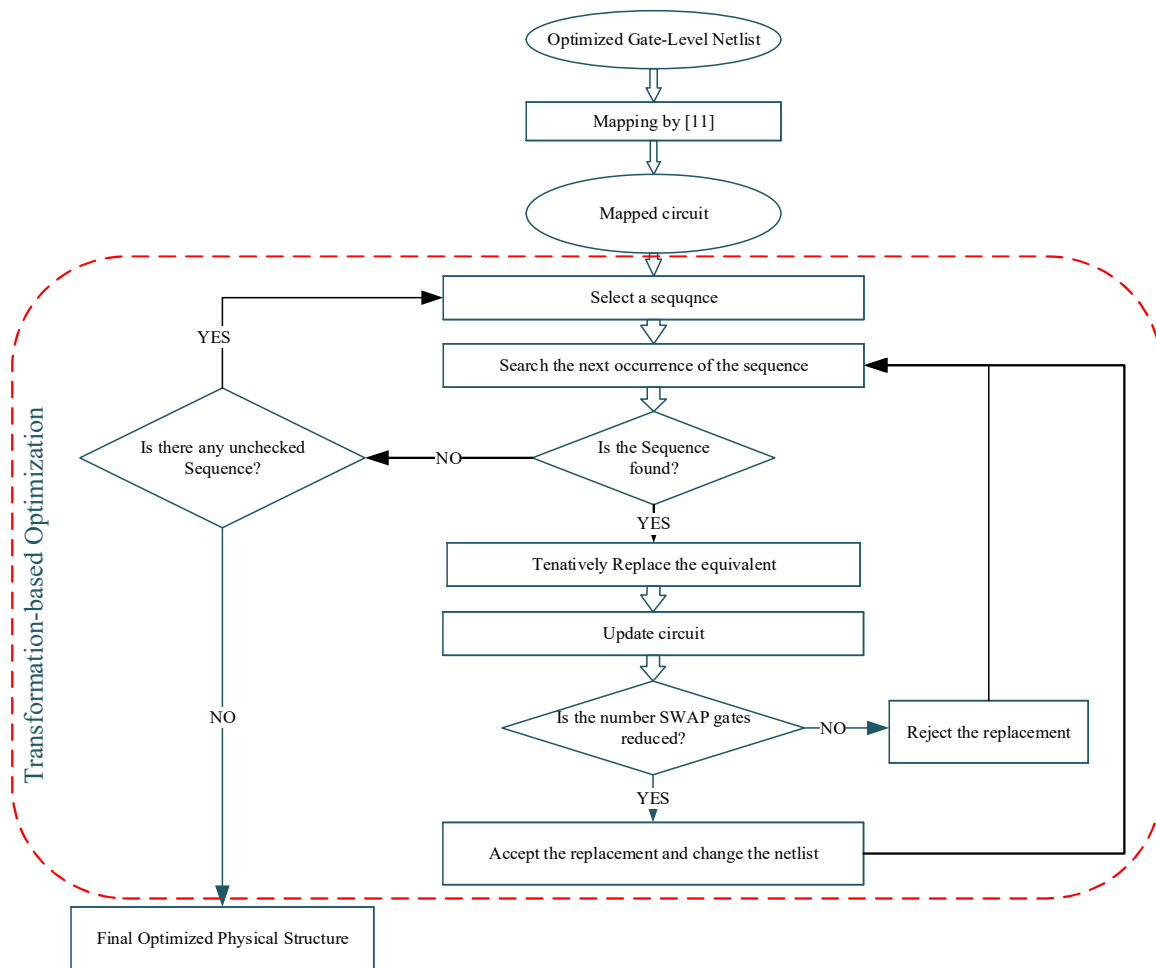


Figure 2. The proposed flow for applying transformations.

After building the mapped circuit, our optimization loop is started. Our greedy policy for transformation is as follows. One sequence is searched in the input netlist, and the equivalent sequence is tentatively replaced into the location that it is found. Then, the number of SWAP gates is calculated. If the number of SWAP gates is improved, this sequence replacement is accepted. Otherwise, it is rejected. When the search of the first sequence is completed, the search of the next sequence with the same mechanism follows. The optimization loop continues until all sequences are examined.

An Example

In this section, an example is given to illustrate our transformation-based physical synthesis approach. Figure 3a,b shows a template and a quantum circuit operating on q0, q1, q2, and q3, respectively. It can be easily verified by matrix multiplication that the two circuits shown in Figure 3a are functionally equal. This circuit has 6 gates and 4 qubits. If the initial locations of the qubits on the lattice are as in Figure 3c, four SWAP gates are needed, as shown in Figure 3d. However, when the two-gate template is replaced by its equivalent one, the circuit is transformed into one shown in Figure 3e that needs two SWAP gates, as shown in Figure 3f.

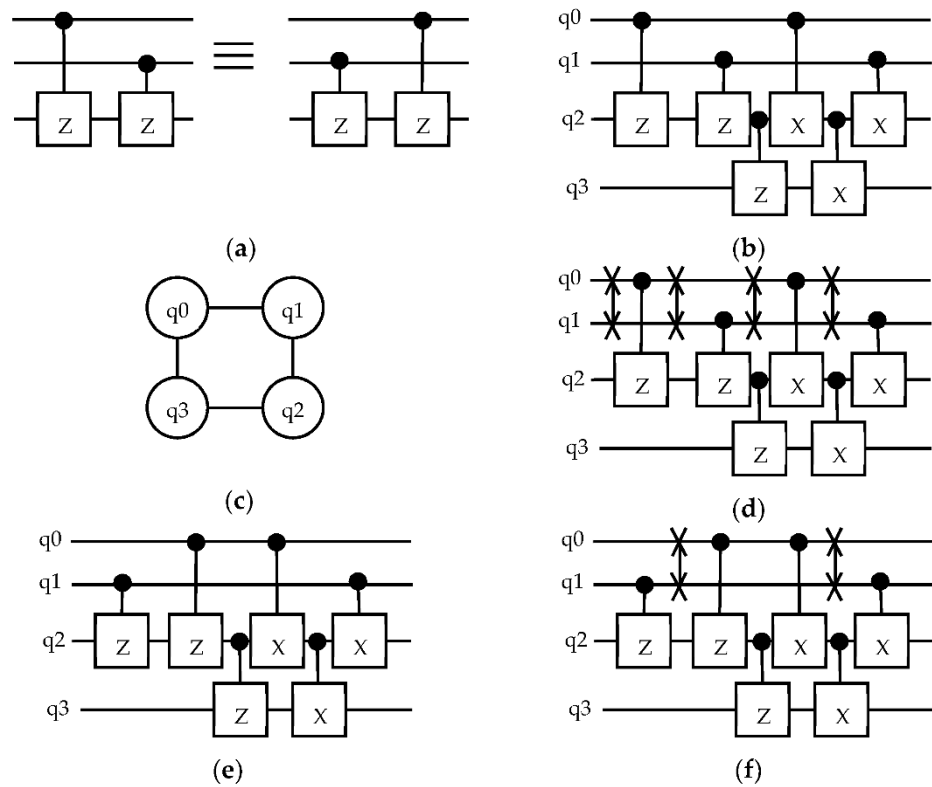


Figure 3. (a) A quantum transformation rule, (b) a quantum circuit, (c) initial locations of qubits, (d) the original circuit including SWAP gates, (e) the transformed circuit, and (f) the transformed circuit including SWAP gates.

4. Experimental Results

To evaluate the proposed approach, it was applied to the benchmark circuits from [34]. In this paper, we targeted the number of SWAP gates as the objective function and reduced it to minimize the latency and error probability of the circuits. Although our approach is applicable to all kinds of nearest-neighbor architectures, we applied it to the 2D square lattice topology to compare our approach with the previous one [11].

Table 1 shows the number of SWAP gates and the run time of the benchmark circuits resulted from prior physical design flow [11] and our physical design flow enhanced by the template-matching physical synthesis technique. The number of SWAP gates of circuits obtained by the prior physical design flow and ours are shown in the columns “Prior Physical Design Flow” and “Our Physical Design Flow”, respectively. The column “Improvement” shows that the improvement of the number of SWAP gates resulted from the physical synthesis approach proposed in this paper. As can be seen, a considerable improvement of 20.6% (on average) was achieved in the number of SWAP gates of the benchmarks. The columns “Prior Physical Design Flow” and “Our Physical Design Flow” under “Run Time” show the run time of the prior physical design flow and the run time of our optimization technique, respectively. The last column includes the runtime overhead imposed by our optimization approach.

Table 1. The results achieved by prior physical design flow and ours for the attempted benchmarks ¹.

#	Benchmarks	Qubits	Gates	Number of SWAP Gates			Run Time (ms) ²		
				Prior Physical Design Flow [11]	Our Physical Design Flow	Improvement (%)	Prior Physical Design Flow [11]	Our Physical Design Flow	Overhead (%)
1	Adder-8	18	381	55	50	9.1	237	250	5.2
2	Adder-16	34	757	107	93	13.1	289	306	5.6
3	Adder-24	50	1133	147	126	14.3	290	310	6.5
4	Adder-32	66	1509	193	160	17.1	295	317	6.9
5	Adder-64	130	3013	385	304	21.0	2640	2850	7.4
6	Qft7	7	104	15	13	13.3	12	12	0.0
7	Qft8	8	135	16	13	18.8	15	15	0.0
8	Qft9	9	170	25	20	20.0	18	19	5.3
9	Qft10	10	209	28	21	25.0	26	28	7.1
10	Grover6	7	3314	195	160	17.9	3020	3150	4.1
11	Grover7	8	10,672	1326	1002	24.4	7540	8108	7.0
12	Grover8	9	29,454	1508	1024	32.1	12,780	14,002	8.7
13	Grover9	10	80,480	8415	4928	41.4	24,987	28,050	10.9
Average					20.6			5.7	

¹ All results of this section are obtained on a Core i3 with 6 gigabyte of memory. ² As calculated by the Rational Quantify suite.

Heuristic Algorithm Analysis

As stated before, we followed a greedy approach to accept or reject one substitution. In other words, the substitutions increasing the number of SWAP gates were rejected. To examine the impact of applying another heuristic on the result, we used simulated annealing (SA) heuristics [35] in accepting or rejecting substitutions. Table 2 shows the results of using the heuristic. The column “Our Approach Based on SA” under “Number of SWAP Gates” shows the number of SWAP gates obtained by our physical design flow when we substitute the simulated annealing heuristic for our greedy approach. The columns “Our Approach Based on Greedy” and “Our Approach Based on SA” under “Run Time” show the run times of our physical design flow using the simulated annealing approach and the greedy approach, respectively. The column “SA/Greedy Ratio” under “Number of SWAP Gates” contains the ratio of the number of SWAP gates obtained by simulated annealing to that achieved by our greedy approach. The column “Improvement” shows that the improvement of the number of SWAP gates resulted from the physical synthesis approach based on SA. The column “Overhead” includes the runtime overhead imposed by the SA approach. The last column includes the ratio of the run time of the flow based on the simulated annealing approach to that based on our greedy approach. It can be observed from the table that simulated annealing provided slightly better results than the greedy approach in most cases. However, on average, the run time of simulated annealing is almost 6.49 times longer. This observation might suggest that while various heuristics may provide slightly different results, it is the execution time that varies the most among them. In other words, it appears that the execution time is the determining factor in choosing among the heuristic approaches. Based on this, we chose the greedy approach for the remainder of this paper.

Figure 4 depicts the behavior of the number of SWAP gates obtained by the two approaches in accepting or rejecting a substitution. Although the improvement obtained by our approach depends on the structure of a circuit, as the number of gates increases, more templates can be potentially found in the circuit. Therefore, as the figure implies, the improvement increases with increasing the number of gates.

Table 2. The number of SWAP gates of the benchmark circuits achieved by using simulated annealing heuristics compared with our greedy approach in accepting or rejecting a substitution.

#	Benchmarks	Number of SWAP Gates				Run Time (ms)			
		Our Approach Based on Greedy	Our Approach Based on SA	Improvement (%)	SA/Greedy Ratio	Our Approach Based on Greedy	Our Approach Based on SA	Overhead (%)	SA/Greedy Ratio
1	Adder-8	50	50	0.0	1	250	580	132.0	2.3
2	Adder-16	93	91	2.2	1.0	306	600	96.1	2.0
3	Adder-24	126	123	2.4	1.0	310	780	151.6	2.5
4	Adder-32	160	155	3.1	1.0	317	900	183.9	2.8
5	Adder-64	304	295	3.0	1.0	2850	20,950	635.1	7.4
6	Qft7	13	13	0.0	1.0	12	20	66.7	1.7
7	Qft8	13	12	7.7	1.0	15	30	100.0	2.0
8	Qft9	20	18	10.0	0.9	19	46	142.1	2.4
9	Qft10	21	19	9.5	0.9	28	150	435.7	5.4
10	Grover6	160	155	3.1	0.9	3150	30,270	861.0	9.6
11	Grover7	1002	906	9.6	1.0	8108	100,704	1142.0	12.4
12	Grover8	1024	995	2.8	0.9	14,002	193,509	1282.0	13.8
13	Grover9	4928	4656	5.5	1.0	28,050	563,282	1908.1	20.1
Average				4.5	1.0			548.9	6.49

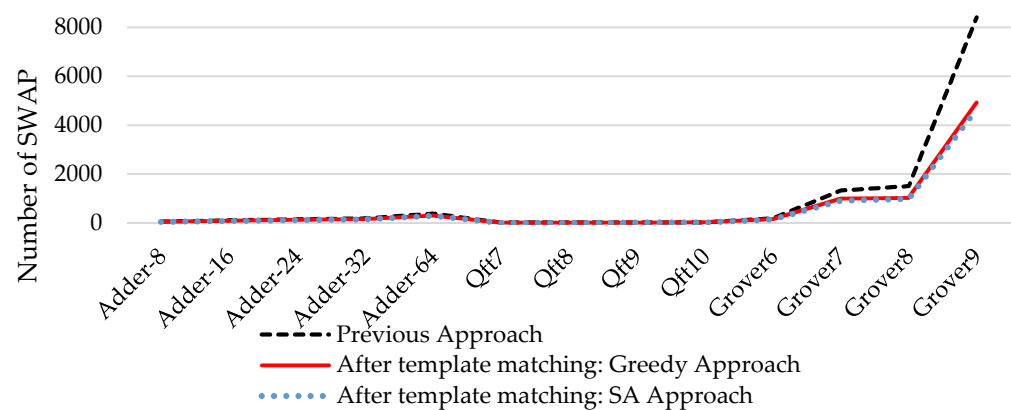


Figure 4. The behavior of the number of SWAP gates obtained by using two different approaches in accepting or rejecting substitutions.

5. Conclusions

The idea behind this paper is to present equivalent models with the same level number and to use them in the physical synthesis of quantum circuits. Physical synthesis involves making local changes in the netlist to improve design criteria, including the delay of quantum circuits. In this paper, a number of templates are proposed, and by substituting these templates in different benchmark circuits, an improvement is presented in the number of SWAP gates. To put the proposed templates into the experiment, nearest-neighbor architectures are selected as our substrate architecture. The results show that the template-matching approach improves the number of SWAP gates up to 41%.

To evaluate our greedy approach in deciding a substitution, we compared it with the SA approach. The results showed that the SA method improves the number of SWAP gates only marginally, while its run time is almost 6.49 times longer.

Author Contributions: Conceptualization, N.M.; methodology, N.M.; software, G.H.; validation, N.M.; formal analysis, N.M.; investigation, N.M.; resources, G.H.; data curation, G.H.; writing—original draft preparation, N.M.; writing—review and editing, N.M.; visualization, G.H.; supervision, N.M.; project administration, N.M.; funding acquisition, N.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nielsen, M.A.; Chuang, I. *Quantum Computation and Quantum Information*; American Association of Physics Teachers: College Park, MD, USA, 2002.
2. Monroe, D. *Quantum Leap*; ACM: New York, NY, USA, 2018.
3. Ladd, T.D.; Jelezko, F.; Laflamme, R.; Nakamura, Y.; Monroe, C.; O'Brien, J.L. Quantum computers. *Nature* **2010**, *464*, 45–53. [[CrossRef](#)] [[PubMed](#)]
4. Sargaran, S.; Mohammadzadeh, N. Saqip: A scalable architecture for quantum information processors. *ACM Trans. Archit. Code Optim. (TACO)* **2019**, *16*, 1–21. [[CrossRef](#)]
5. Alpert, C.J.; Chu, C.; Villarrubia, P.G. Physical synthesis comes of age. In Proceedings of the International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, 4–8 November 2007.
6. Mohammadzadeh, N.; Sedighi, M.; Zamani, M.S. Quantum physical synthesis: Improving physical design by netlist modifications. *Microelectron. J.* **2010**, *41*, 219–230. [[CrossRef](#)]
7. Meter, R.; Oskin, M. Architectural implications of quantum computing technologies. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* **2006**, *2*, 31–63. [[CrossRef](#)]
8. Lye, A.; Wille, R.; Drechsler, R. Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits. In Proceedings of the 20th Asia and South Pacific Design Automation Conference, Chiba, Japan, 19–22 January 2015.
9. Wille, R.; Lye, A.; Drechsler, R. Optimal SWAP gate insertion for nearest neighbor quantum circuits. In Proceedings of the 19th Asia and South Pacific Design Automation Conference (ASP-DAC), Singapore, 20–23 January 2014.
10. Lin, C.-C.; Sur-Kolay, S.; Jha, N.K. PAQCS: Physical Design-Aware Fault-Tolerant Quantum Circuit Synthesis. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2014**, *23*, 1221–1234. [[CrossRef](#)]
11. Farghadan, A.; Mohammadzadeh, N. Quantum circuit physical design flow for 2D nearest-neighbor architectures. *Int. J. Circuit Theory Appl.* **2017**, *45*, 989–1000. [[CrossRef](#)]
12. Maslov, D.; Falconer, S.M.; Mosca, M. Quantum circuit placement. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2008**, *27*, 752–763. [[CrossRef](#)]
13. Shafaei, A.; Saeedi, M.; Pedram, M. Qubit placement to minimize communication overhead in 2D quantum architectures. In Proceedings of the 19th Asia and South Pacific Design Automation Conference (ASP-DAC), Singapore, 20–23 January 2014.
14. Iwama, K.; Kambayashi, Y.; Yamashita, S. Transformation rules for designing CNOT-based quantum circuits. In Proceedings of the 39th Annual Design Automation Conference, New Orleans, LA, USA, 10–14 June 2002.
15. Shende, V.V.; Prasad, A.K.; Patel, K.N.; Markov, I.L.; Hayes, J.P. Scalable simplification of reversible circuits. In Proceedings of the 12th International Workshop on Logic and Synthesis (IWLS'03), Laguna Beach, CA, USA, 28–30 May 2003.
16. Miller, D.M.; Maslov, D.; Dueck, G.W. A transformation based algorithm for reversible logic synthesis. In Proceedings of the 40th Annual Design Automation Conference, Anaheim, CA, USA, 2–6 June 2003.
17. Maslov, D.; Dueck, G.W.; Miller, D.M. Toffoli network synthesis with templates. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2005**, *24*, 807–817. [[CrossRef](#)]
18. Maslov, D.; Dueck, G.W.; Miller, D.M. Fredkin: Toffoli templates for reversible logic synthesis. In Proceedings of the ICCAD-2003. International Conference on Computer Aided Design (IEEE Cat. No.03CH37486), San Jose, CA, USA, 9–13 November 2003.
19. Maslov, D.; Dueck, G.W.; Miller, D.M. Synthesis of Fredkin-Toffoli reversible networks. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2005**, *13*, 765–769. [[CrossRef](#)]
20. Maslov, D.; Dueck, G.W.; Miller, D.M. Simplification of Toffoli networks via templates. In Proceedings of the 16th Symposium on Integrated Circuits and Systems Design, SBCCI 2003, Sao Paulo, Brazil, 8–11 September 2003.
21. Maslov, D.; Dueck, G.W.; Miller, D.M. Techniques for the synthesis of reversible Toffoli networks. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* **2007**, *12*, 42. [[CrossRef](#)]
22. Maslov, D.; Dueck, G.W.; Miller, D.M.; Negrevergne, C. Quantum circuit simplification and level compaction. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2008**, *27*, 436–444. [[CrossRef](#)]
23. Lu, L.; Li, R.; Guo, L.; Fu, Q. Maximal entanglement entanglement-assisted quantum codes constructed from linear codes. *Quantum Inf. Process.* **2014**, *14*, 165–182. [[CrossRef](#)]
24. Saeedi, M.; Wille, R.; Drechsler, R. Synthesis of quantum circuits for linear nearest neighbor architectures. *Quantum Inf. Process.* **2010**, *10*, 355–377. [[CrossRef](#)]

25. Arabzadeh, M.; Saeedi, M.; Zamani, M.S. Rule-based optimization of reversible circuits. In Proceedings of the 2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC), Taipei, Taiwan, 18–21 January 2010.
26. Soeken, M.; Wille, R.; Dueck, G.W.; Drechsler, R. Window optimization of reversible and quantum circuits. In Proceedings of the 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, Vienna, Austria, 14–16 April 2010.
27. Abdessaied, N.; Soeken, M.; Wille, R.; Drechsler, R. Exact template matching using Boolean satisfiability. In Proceedings of the 2013 IEEE 43rd International Symposium on Multiple-Valued Logic, Toyama, Japan, 22–24 May 2013.
28. Rahman, M.M.; Dueck, G.W. Properties of quantum templates. In *International Workshop on Reversible Computation*; Springer: Berlin/Heidelberg, Germany, 2012.
29. Bandyopadhyay, C.; Wille, R.; Drechsler, R.; Rahaman, H. Post Synthesis-Optimization of Reversible Circuit using Template Matching. In Proceedings of the 2020 24th International Symposium on VLSI Design and Test (VDATE), Bhubaneswar, India, 23–25 July 2020.
30. Mohammadzadeh, N.; Zamani, M.S.; Sedighi, M. Auxiliary qubit selection: A physical synthesis technique for quantum circuits. *Quantum Inf. Process.* **2010**, *10*, 139–154. [[CrossRef](#)]
31. Mohammadzadeh, N.; Zamani, M.S.; Sedighi, M. Improving latency of quantum circuits by gate exchanging. In Proceedings of the 2009 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools, Patras, Greece, 27–29 August 2009.
32. Mirkhani, Z.; Mohammadzadeh, N. Physical synthesis of quantum circuits using templates. *Quantum Inf. Process.* **2016**, *15*, 4117–4135. [[CrossRef](#)]
33. Mohammadzadeh, N.; Zamani, M.S.; Sedighi, M. Quantum circuit physical design methodology with emphasis on physical synthesis. *Quantum Inf. Process.* **2013**, *13*, 445–465. [[CrossRef](#)]
34. Lin, C.-C.; Chakrabarti, A.; Jha, N.K. FTQLS: Fault-tolerant quantum logic synthesis. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2013**, *22*, 1350–1363. [[CrossRef](#)]
35. Brooks, S.; Morgan, B.J.T. Optimization Using Simulated Annealing. *J. R. Stat. Soc. Ser. D* **1995**, *44*, 241. [[CrossRef](#)]