2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS) | 979-8-3503-4805-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAECIS58353.2023.10170496

Synthesis of Reversible Logic Circuits-A Technical Review

Bhaskar Das
Dept. of Computer Science (Emerging Technologies)
Hyderabad Institute of Technology and Management
Hyderabad, India
bhaskardas2006@gmail.com

Vinutha N
Dept. of Computer Science
Dayananda Sagar University
Bengaluru, India
vinutha1v@gmail.com

Nadiya Zafar
Dept. of Computer Science (Emerging Technologies)
Hyderabad Institute of Technology and Management
Hyderabad, India
nadiyazafar5@gmail.com

Moghekar Rajeshwar
Dept. of Computer Science (Emerging Technologies)
Hyderabad Institute of Technology and Management
Hyderabad, India
rajeshwarm.cse@hitam.org

***Abstract*—Traditional technologies such as CMOS show their limitations with an expanded growth of packing density. The usage of recent modern technologies is needed to address the traditional challenges. In modern technologies, managing the loss of energy is also a primary issue in the digital logic design for both manufacturer and as well as for the customer. Thus, Reversible logic is considered to be one of the efficient solutions to overcome such problems. The applications of reversible logic are in a wide range of domains - Digital Signal Processing, Optical Computing, Cryptography, and CMOS design with low power consumption. Thus, this paper discusses optimal-cost synthesis problems related to reversible logic circuits. The synthesis problem aims to obtain a network of gates with minimum cost that realize a given function by assuming the same cost for each gate, and each gate's cost reflects the actual cost for its implementation. The discussion of various synthesis algorithms and a comparative study of them are presented by giving their essential features and limitations.**

Keywords— BDD; ESOP; Logic Gate Synthesis; Reversible Logic.

I. INTRODUCTION

In recent years, Logic Gate synthesis varies from state-of-the-art technologies; thus, the researchers have shown their focus on utilizing reversible functions. The paper presents ALU, synthesized from reversible logic gates with Fredkin, Toffoli, and Peres Gate, [1] as a replacement for OR and AND gate for every bit of ALU. It is observed from the synthesized ALU that reversible logic reduces 34% of area and a delay of 48.91%. The authors of paper [2] shows that "RL-VGG-19" and "RL-VGG-16" architecture with reversible logic circuits obtain 16.48% and 18.08% decrement in the total power loss in comparison with the classical VGG-19 and VGG-16 architecture, respectively.

With this motivation, the exploration of compact and fast computing in logic circuits raises the questionnaire to the researchers about the drawback of quick and compact computing. However, without consideration of the access methods, a minimum amount of heat generation is always associated with information processing. As an advancement of VLSI technologies, new manufacturing techniques have rendered heat loss in the previous years. Complexity of heat dissipation has increased more in Integrated-Circuits (IC). As per Landauer's principle [3], loss in energy has resulted due to the introduction of irreversibility in information processing. This problem can be addressed with the use of reversible computing that leads to lossless information.

Synthesis is to identify a network of gates with minimum cost. It realizes a given function by assuming the same cost for each gate, and the cost of a gate reflects its actual cost in its implementation. Synthesis implies reversible circuits should possess a reversible specification and thereby, its implementation can be done optimally or heuristically. In the mid-1990s, researchers focus gradually increased on the synthesis of a circuit using reversible logic with the development of powerful quantum algorithms.

A reversible gate computes a bijective function. This means that there are precisely as many output bits as input bits. The output bits can be generated from input bits and as well as the input can also be generated from output bits. In other words, if a n-input, n-output logic circuit with complete specification assign each input value to every distinct output value and the inverse is also true, then it is considered reversible. Applications of Reversible logic in several technologies are digital signal processing, optical computing, cryptography, and designing of low-power consuming CMOS.

The conventional logic gates like OR, XOR or AND applied in digital design is irreversible except for the NOT gate. The utilization of reversible gates performs the design of a Reversible circuit. Thus, conventional logic gates are not sufficient for their synthesis. Over the past years, various gates have been proposed. Among the various proposed gates, the controlled-not (CNOT) gates are proposed by Feynman [4], Toffoli [5], and Fredkin [6] and all these gates are shown in Fig. 1. The synthesized reversible circuit should also be devoid of fan-out and feedback. This restriction conforms to the design requirement of quantum networks.

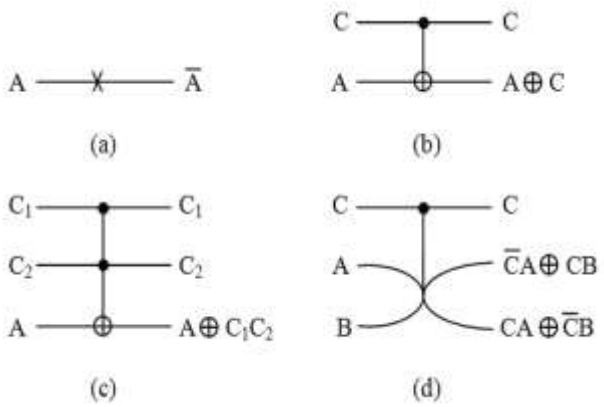

Fig. 1. (a) NOT gate (b) Feynman gate (c) Toffoli gate (d) Fredkin gate

This paper discusses the most popular techniques of reversible functions, algorithms, problems, and advancements toward synthesizing of reversible circuits. The paper is presented as follows. Well-known representation techniques of reversible functions are presented in Section II. We outline reversible synthesis costs in Section III. Algorithmic details are shown in Sections IV. Next, a comparative study of some well-known algorithms is presented in Section V. Finally, we conclude by stating some challenges of reversible circuit synthesis in Section VI.

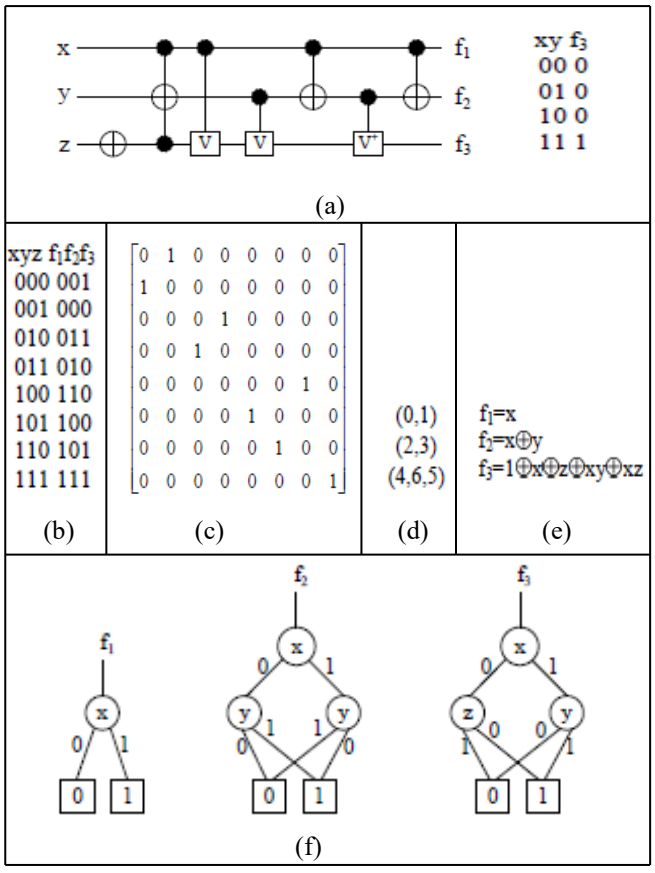

Fig. 2. (a) A sample of reversible specification with its irreversible truth table (b) reversible truth table (c) matrix representation (d) cycle form (e) PPRM (f) ROBDD

This particular circuit changes the value (x,y,z) to (z,y,x). The truth table of this circuit is as follows:

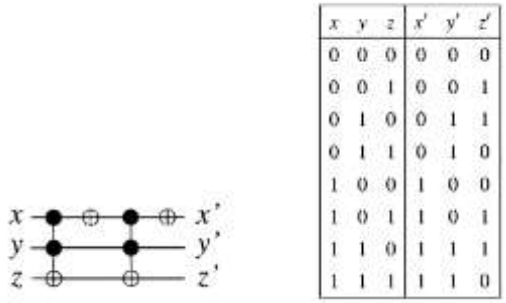

Fig. 3. (a) Reversible gate with two T gates and two N gates and (b) truth table for it.[21]

This is the realization circuit for if-else logic using reversible logic circuit. Fig. 4. [22]

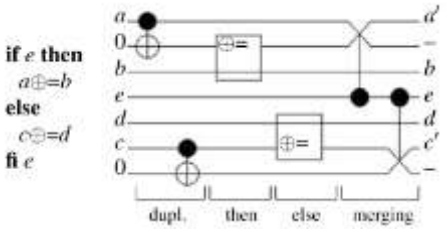

Fig.4. Realization circuit for if-else logic

REPRESENTATION MODELS FOR REVERSIBLE FUNCTION

The representation of reversible functions are performed using different approaches and their illustration are shown in Fig. 2.

Truth tables: This is the simplest method for describing of a "reversible function." This representation of size m is a truth table comprising m columns and 2^m rows.

Matrix illustrations: The representation of a reversible function f_i is performed by a 0-1 matrix with a single 1 in every row and within every column, ie., the non-zero element present in row i occur in column f(i).

Cycle expansion: With respect to the view of reversible function as a permutation, it is represented as the product of disjoint cycles.

Reed-Muller expansion: It is represented by the formula of algebraic form in Positive Polarity Reed-Muller (PPRM) expansion. Reed-Muller expansion utilizes the uncomplemented parameters. Its calculation is performed using the EXOR-Sum-of-Products (ESOP) with the replacement of the complement of an EXOR.

Decision Diagrams: Boolean reversible function is represented by a Binary Decision Diagram (BDD). BDD is defined as an acyclic-directed graph with the application of Shannon decomposition to every non-terminal nodes. Next, it is extended to Reduced Ordered BDDs (ROBDDs). ROBDDs form canonical representations of Boolean functions. A ROBDD construction is performed from BDD based on arranging the variables, combining corresponding sub-graphs, and removal of nodes having similar sub nodes.

II. Synthesis Costs

The main objective of the synthesis problem is to obtain the network of gates with minimum cost that realize a given function by assuming the same cost for each gate, and each gate's cost reflects the actual cost for its implementation. There are many parameters for reversible circuit design for determining circuits' complexity and performance.

Reversible gates Count (N): It is the representation of the number of reversible gates employed in a reversible circuit. If the number of gates is less, then the complexity of that circuit reduces and performance increases.

Number of Constant Inputs: It is referred to as a number of inputs to retain as constant as either 1 or 0 in the synthesis for the given logical function.

Quantum Cost (QC): Quantum Cost is termed as the circuit cost with regard to the primitive gate cost. The calculation is based on the known number of basic reversible logic gates that are needed to produce logic gate circuit. The circuit performance is high if the circuit QC is less.

Counts of Garbage Outputs (GO): GO refers to the unused outputs present in a reversible logic circuit. Reversibility can be obtained by adding the garbage outputs and these obtained garbage outputs should be less in number for developing high-performance circuits.

Depth: It is termed as the highest number of elementary gates in any of the paths considered between inputs and outputs of a designed circuit. A decrease in circuit depth aids in the reduction of circuit latency when any of the subsets of gates are concurrently invoked.

III. Related Works

Heuristic algorithms were proposed because many of the circuits that are used in real time were too big in size for optimal synthesis and also non-linear. Choosing an appropriate model to represent the reversible functions is important in the development of efficient synthesis algorithms. In concise, every model favors a particular category of reversible functions. Synthesis algorithms can be created after the detection of simpler cases and decompose reversible functions into a series of simpler functions for the considered model.

A. Transformation based Synthesis:

The transformation-based approach is proposed in [7]. It is an experiment over a defined reversible function (RF) with a truth table, which recognizes the reconstruction of the function which is deployed over the resultant so the input and resultant pattern will remain the same. After this transformation, a function can be changed into gate cascades. If it is Toffoli gates then the cascade also comprises of Toffoli gates.

The paper [8] describes the enhancement of transformation-based approach. In that case, the bi directional method applies transformations to input and as well as to an output of the truth table. This results to obtain minimum gates count for the chosen circuit.

The circuits synthesized using bi-directional and transformation approaches are not optimal. The template is the sequence of two gates, and here both of the sequences give similar output, but the first one has greater gates count and/or more quantum cost in comparison to the second. In this mechanism, it searches for the matching of first sequence in the circuit. If at all, a match is identified, then the replacement of first sequence of the circuit is done by the second. There also exists a library of templates [8].

B. PPRM-based Synthesis:

In [9] an efficient reversible logic synthesis approach accepts a Positive Polarity Reed-Muller representation of a function. This technique produces a tree like structure having a root node which contains all the output as an expansion of tree, then the common factors for expression of output variables are recognized.

Then, every factor is put into the PPRM-expansions for generating the new node having modified expansions. If the generated expansion is unable to lessen the count of terms, then that new node is discarded. This algorithm is heuristic in nature and a priority queue is applied for the processing tree remarkably. Those substitutions which result in the best one is transformed into the Toffoli gates.

C. ESOP-based Synthesis:

In [10] a fundamental Exor Sum of Product based reversible logic synthesis algorithm uses the input of ESOP representation and Toffoli gates for designing a circuit. Here, there are two inputs (one is of +ve and other one is of -ve polarity) corresponding to every input x_k. There is one output line for every output f_j. Hence, initially the circuit is comprised of $2p+q$ lines of an empty cascade.

For each cube of every output, a Toffoli gate is cascaded. In other words, mapping of each cube to a Toffoli gate is performed for every output variable; where $f_j = 1$ for the target output line f_j, and the controls of the gate are the input lines. The input line x_k is connected to the positive control of the Toffoli gate if x_k possesses positive polarity in the cube. And, the input line x_k is in connection to negative control of the Toffoli gate if x_k possesses negative polarity in the cube. Hence, this technique modifies from the cube list of a function to a cascade of the Toffoli gates network.

In line optimization technique, count of gates are increased because of insertion of NOT gates. In order to decrease the number of inverter gate, rearrangement of cubes in their cube list might be applied. $O(n!)$ is the complexity for computing the optimal reordering. Here n shows the cubes count present in the cube list. The Brute Force technique is not suitable here because if larger the circuit, then a greater number of cubes exists.

For reordering of cubes, two metrics are proposed:

- alpha/beta cost metric [10] and,
- auto-correlation coefficient-based cost metric [11].

Both algorithms follow the divide-and-conquer paradigm. These are the three steps for this technique:

"Divide": First, the input value having the lowermost cost metric is identified. Assume x is the input value. Then, based on the value of x, division of cube list into two parts happens. The first one consists of cubes that contain don't care value and the positive polarity of x. The other one contains the negative polarity of x.

"Conquer": Recursively, both lists are arranged.

"Combine": In this phase, for generating the finally arranged list, these two ordered lists are recombined.

Maslov and Miller [12] proposed the negative-control Toffoli gate which one is used in [13] including "ESOP-based

circuits". This gate has negative and positive controls. The NOT gates are removed by utilizing this gate from the "ESOP-based circuits." Hence, the number of gates and quantum costs are also decreased automatically. The approach in [10], with line optimization is very similar to the synthesis approach. Where $f_j = 1$, a Toffoli gate is generated for every cube of every output in this approach. Toffoli gate positive control is in connection with input line x_k if it possesses the positive polarity. And, Toffoli gate negative control is in connection with input line x_k if it has negative polarity.

Earlier approaches gave identical Toffoli gate with different targets but similar controls when there are two or more outputs in the cube.

substitute the cubes into the other outputs with the help of CNOT and produce one "Toffoli gate" for every shared cubes. This "shared-cube synthesis" [14,15] increases the circuit quality. The algorithm works as follows:

First, find a pair that has a greater count of shared cubes and all pairs of examined outputs. Initially, the empty output line is the target line. A Toffoli gate is produced for every cubes by using target or control part of any Toffoli gate and empty output line.

Anyone output can be chosen as the target if both output lines are not empty. In the latter case, CNOT gate is introduced in the first to cancel the effect of gates that are on the line. A CNOT gate is added at the end for both cases for the transformation of shared functionality to another outputs. After that, from the cube list, these two outputs were removed. Similar approach will repeat for all sets of shared cubes.

D. Decision Diagram-based Synthesis:

BDD-based methods scale better than others. It uses a binary decision diagram for the improvement of sharing between control of reversible gates. However, they need ancilla qubits in large numbers and this is a valuable resource in the development of quantum computers. An efficient synthesis algorithm of Toffoli circuits is proposed by Wille and Drechsler [16] from functions defined in binary decision diagrams (BDDs).

A tool such as CUDD can be used for creating a BDD of the given function. very node in BDD can be changed into a "cascade of Toffoli gates". It is seen that fan-outs are caused due to BDD in the resultant circuit but fan-outs are not acceptable in a reversible circuit. So, some extra lines are added to the circuit to avoid fan-outs.

E. Exploiting coding techniques for logic synthesis

As the function is often non-reversible in traditional logic design, extra values are added to the function for the establishment of reversibility. That produces an overhead and also alters the scalability of the synthesis technique and thereby produces the circuits which are relatively complex.

Alwin Zulehner et al. [17] proposed to resolve this problem by using coding techniques in the synthesis of reversible logic circuits. They further gave an intermediate encoding in output patterns which require few numbers of additional inputs and as well as outputs. The proposed synthesis scheme, lets us to accomplish most of the synthesis on a lesser number of variables which are significant and utilizes many don't care values in the code. Experimental evaluations obtained improved scalability and lesser cost circuits.

IV. Comparative Study of Existing Synthesis Algorithms

Logic synthesis [18] of reversible circuits has achieved considerable interest with the recent advancement of quantum computation. It is noticed that the heuristic algorithms were proposed because most of the circuits used practically are more in number for optimal synthesis and non-linear. To develop effective synthesis algorithms, choosing an appropriate representation model for reversible functions is very important. By representing them concisely, every model favors the different reversible functions. In a model, developing synthesis algorithms requires simple cases and also the decomposition of reversible functions into simpler functions also plays important role. Here, Table I is given to show the comparative study of various, well-known synthesis algorithms by giving their features and limitations.

TABLE I. Comparative Study of Existing Synthesis Techniques

Methods	***Features***	***Limitations***
Shende et al. [7]	Fast, Heuristic, No garbage.	Dependency on Circuit
Miller et al. [8]	Heuristic, No garbage.	Dependency on Circuit
Gupta et al. [9]	Heuristic, No garbage.	Scalability is limited
Fazel et al. [10]	Fast, Heuristic	Dependency on Function
Maslov et al. [13]	Heuristic	Windowing strategy, More runtime
Wille et al. [16]	Heuristic, Fast, Scalable	Numerous ancilla, Local optimum.
Alwin Zulehner et al. [17]	Heuristic	Further improvement possible

V. Conclusions and Future Scope

Reversible logic circuits are studied with several different motivations. The various applications of Reversible logic circuits are low-power computing, the development of photonic circuits, quantum information processing, and bit-twiddles in computer graphics algorithms. From this point of view, we can conclude that all the synthesis algorithms have their own limitations, and to overcome those limitations some new algorithms have been proposed. Even though new algorithms are developed but still it has their own limitations. Thus, in the future, the following limitations can be addressed:

- To obtain a near-optimal circuit within a minimal time duration, a population-based search technique can be used when the search space is large.
- In the future, the researchers can focus on the investigation of trade-off of the ancilla lines with gate count that can be utilized in the synthesis techniques [19].
- In paper [20] describes the implementation of KFDD-based logic synthesis in reversible circuits. The results show the probability of occurrence of each combination possible.
- The space complexity to generate all Toffoli gates of input size n is $O(n*2^n)$, which is exponential in nature. For such cases, a dynamic generation approach can be developed for large circuits.
- The reversible circuits based on a dynamic programming algorithm are synthesized using symmetric functions.

The same information is utilized to obtain the generalized structure of the target function and the resulting circuits through using this technique are anticipated to be small in size and well scaled.

- Quantum Technology requires quantum gates, such as Hadamard gate (A special gate), Z, and X (analogy of Boolean NOT), whose specification cannot be taken into consideration, but those quantum gates can be considered in the Boolean logic.
- The biggest advantage using reversible logic circuit is function can be resolved with storage of one bit of memory.
- Permutation function, odd or even can be temporarily stored without consuming memory.
- Still, most of the circuits are not able to solve complex and large Boolean functions and this is still a limitation in the field of reversible logic circuit.
- Memory is also one of the limitations with these circuits.

REFERENCES

[1] S. Nagaraj, B.Vamsi Krishna, Botta Chakradhar, Debanjan Sarkar, "Comparison of 32-bit ALU for Reversible Logic and Irreversible Logic", Innovations in Power and Advanced Computing Technologies, 2021.

[2] Bappaditya Dey, Kasem Khalil, Ashok Kumar, Magdy Bayoumi, "A Reversible-Logic based Architecture for VGGNet", 28th IEEE International Conference on Electronics, Circuits, and Systems, 2021.

[3] R. Landauer, "Irreversibility and heat generation in the computing process", IBM Journal of Research and Development, vol. 5, pp. 183-191, 1961.

[4] R. Feynman, "Quantum mechanical computers", In Optic News, vol. 11(2), pp. 11-20, 1985.

[5] T. Toffoli, "Reversible computing", Technical memo mit/lcs/tm-151, MIT Lab for Computer Science, February 1980.

[6] E. Fredkin and T. Toffoli, "Conservative logic", International Journal of Theoretical Physics, vol. 21, pp. 219-253, 1982.

[7] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22(6), pp. 710–722, 2003.

[8] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis", In Proceedings of the 40th annual Design Automation Conference (DAC), pp. 318–323, 2003.

[9] P. Gupta, A. Agrawal, and N. K. Jha, "An algorithm for synthesis of reversible logic circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25(11), pp. 2317–2330, 2006.

[10] K. Fazel, M. Thornton, and J. E. Rice, "ESOP-based Toffoli gate cascade generation", In Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM*)*, pp. 206–209, Victoria, BC, Canada, 22-24 Aug.2007.

[11] J. E. Rice and V. Suen, "Using autocorrelation coefficient-based cost functions in ESOP-based Toffoli gate cascade generation", In Proceedings of 23rd Canadian Conference on Electrical and Computer Engineering (CCECE), Calgary, Canada, May 2010.

[12] D. Maslov and D. M. Miller, "Comparison of the cost metrics through investigation of the relation between optimal NCV and optimal NCT three-qubit reversible circuits", IET Computers & Digital Techniques, vol. 1(2), pp. 98–104, 2007.

[13] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, "Quantum circuit simplification and level compaction", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27(3), pp. 436–444, 2008.

[14] Y. Sanaee and G. W. Dueck, "Generating Toffoli networks from ESOP expressions", In Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp. 715–719, Victoria, BC, Canada, 13-18 June 2009.

[15] N. M. Nayeem and Jacqueline E. Rice, "A Shared-cube Approach to ESOP-based Synthesis of Reversible Logic", Facta Universitatis, vol. 24(3), pp. 385-402, December 2011.

[16] R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions", In Proceedings of the 46th Annual Design Automation Conference (DAC), pp. 270–275, 2009.

[17] Zulehner, A., & Wille, R., Exploiting coding techniques for logic synthesis of reversible circuits. 2018 23rd Asia and South Pacific Design Automation Conference, 2018.

[18] N. M. Nayeem, "Synthesis and Testing of Reversible Toffoli Circuits", A Thesis of Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Alberta, Canada, 2011.

[19] Nesa Rani, P. M., & Lyngton Thangkhiew, P. A Review on Fundamentals of Ternary Reversible Logic Circuits. 2020 International Conference on Computational Performance Evaluation, 2020.

[20] Sangeetha, P., & Kumari, P., An Implementation of KFDD based Reversible Logic using IBM-Q Experience. 2020 7th International Conference on Computing for Sustainable Global Development, 2020.

[21] V. V. Shende, A. K. Prasad, I. L. Markov and J. P. Hayes, "Synthesis of reversible logic circuits," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, no. 6, pp. 710-722, June 2003, doi: 10.1109/TCAD.2003.811448.

[22] V. V. Shende, A. K. Prasad, I. L. Markov and J. P. Hayes, "Synthesis of reversible logic circuits," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, no. 6, pp. 710-722, June 2003, doi: 10.1109/TCAD.2003.811448.

[23] Wille, R., & Drechsler, R. (2010). Synthesis of Reversible Logic. Towards a Design Flow for Reversible Logic, 27–56. doi:10.1007/978-90-481-9579-4_3