

Reversible Circuit Synthesis Using Evolutionary Algorithms



T. N. Sasamal, H. M. Gaur, A. K. Singh and A. Mohan

Abstract With the unprecedented growth in VLSI technology in recent years, managing power dissipation has become a challenging task for many researchers. In this aspect, reversible logic emerges as one of the basis of future lossless computing system that promises zero energy dissipation, meanwhile classical physics cannot survive due to constant scaling of transistors and the exponential growth of transistor density in integrated circuits. It has applications in various domain such as low power VLSI, fault-tolerant designs, quantum computing, nanotechnology, DNA computing, optical computing, cryptography, and informatics. There are many existing works for the synthesis of reversible logic circuits; some are exact methods while others based on heuristic approaches. In this survey, we review a range of evolutionary computation approaches to the problem of optimal synthesis of reversible Logic—GA (Genetic Algorithm) based, PSO (Particle Swarm Optimization) based, ACO (Ant Colony Optimization)-based circuits where aim is to obtain a near-optimal solution by efficiently exploring the entire search space. This study provides an algorithmic review with comparative study on metaheuristic-based reversible logic synthesis methods proposed in existing literatures. Comparison of experimental results based on large number of benchmark circuits conform that evolutionary algorithms-based technique enables optimal or near-optimal solutions with lesser synthesis time.

T. N. Sasamal (✉)

Department of Electronics & Communication Engineering,
NIT Kurukshetra, Kurukshetra, India
e-mail: tnsasamal.ece@nitkkr.ac.in

H. M. Gaur

Department of Electronics & Communication Engineering, ABES Institute of Technology,
Ghaziabad (Delhi NCR), India
e-mail: leoharimohan84@gmail.com

A. K. Singh

Department of Computer Applications, NIT Kurukshetra, Kurukshetra, India
e-mail: ashutosh@nitkkr.ac.in

A. Mohan

Department of Electronics Engineering, IIT BHU, Varanasi, India
e-mail: profanandmohan@gmail.com

© Springer Nature Singapore Pte Ltd. 2020

A. K. Singh et al. (eds.), *Design and Testing of Reversible Logic*, Lecture Notes in Electrical Engineering 577, https://doi.org/10.1007/978-981-13-8821-7_7

1 Introduction

In the current scenario with increasing complexity in VLSI circuits; managing power dissipation is an important issue in digital circuit design. With higher levels of integration and increasing scaling; Moore's law seems to be valid yet, but in traditional (irreversible) technologies heat produced by each IC doubles [44]. Lossless computing offers an alternative, where logical operations do not yield information loss called reversible operation [36, 69]. Reversible logic realizes n -input n -output functions where a bijective relation exists between input and output vector. In a reversible logic, every input pattern can be uniquely recovered from its output pattern, so no information is lost during computation. Reversible logic circuits take care of heat loss due to information erase. Thus reversibility will become an inherent property that will help to broaden low power design [66] and quantum computation [23, 46] horizon, and also have applications to fault-tolerant designs, nanotechnology, DNA computing, optical computing, cryptography, and informatics [36, 37, 46]. Work by Landauer [22] showed that, regardless of the underlying technology, Conventional logic circuits dissipate heat in an order of $kT \ln 2$ joules for every bit of information that is lost, where k is the Boltzmann constant and T is the operating temperature. Since today's computing devices are usually built of elementary gates like AND, OR, NAND, etc., they are subject to this principle and, hence, dissipate this amount of power in each computational step.

Synthesis of reversible logic circuits differs from the conventional one in many ways. Firstly, in reversible circuit, only once every output will be taken so that no fan-out should exist. Secondly, every input pattern there should have a distinctive output pattern. In last, an acyclic circuit must be as a result. The reversible gate performs the permutation of only input functions and synthesizes the reversible functions. If a reversible gate contains k inputs and k outputs, then it is addressed as a $k \times k$ reversible gate. The reversible gates are only included in reversible-based layouts. In reversible designs, the input lines which has constants are known as constant inputs and the outputs which are not used as primary outputs are known as garbage outputs. An optimal design is used to keep minimal number of constant inputs and garbages. Traditional Boolean logic synthesis approaches like Karnaugh Map, Quine–McCluskey, etc. are not allowed to apply directly to synthesize a reversible-based design due to the parameters like Fan-outs, feedback from output gates to input gates are not allowed, number of inputs equal to the numbers of outputs, existence of ancilla inputs and garbage outputs, etc. So implementation only could be possible in the form of cascading of reversible gates.

Classification of synthesis algorithms is shown in Fig. 1; where we have taken the milestone works in this area. All the existing reversible logic synthesis approaches that have been proposed previously can be divided into two major groups: (a) exact approach which produces the optimal solutions, suffered from huge computation time; and (b) heuristic approaches, on the other side which provides near-optimal solutions in short computation time. They can be described in different represen-

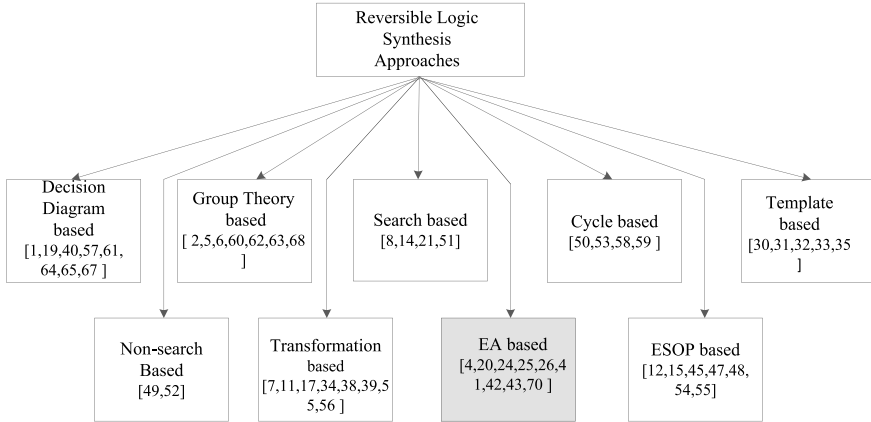


Fig. 1 Classification of synthesis algorithms

tations like BDDs [65], positive polarity Reed–Muller expansion [14], truth-tables [38], matrix representations, permutations [59].

Some of these approaches perform well on smaller designs, but fails when the input count increases, either in requirement of huge computation time and memory, or failure in terms of reaching to the solution. Although having encouraging progress in the field of reversible logic synthesis, still search of best possible synthesis solution remains an open challenge for the researchers. Number of results show the practicability and extend of synthesis are generally not sufficient [25]. So, EA-based algorithm can be used to bolster the potential and efficiency of reversible-based synthesis methods.

In this survey, we present a comparative study of several metaheuristic approaches, algorithms, benchmarks, and future aspects emphasizes to the realization of reversible logic designs. This chapter is organized in following way: Sect. 2 preliminaries are introduced. Section 3 outlines evolutionary approaches. Section 4 includes algorithmic details. Section 5 presents comparison of available benchmarks and discussion which manifest the feasibility of different approaches. Finally, Sect. 6 summarizes this work.

2 GA-Based Synthesis Algorithm

Genetic Algorithm (GA) is based on heuristic search approach and optimization tool for inheriting the method of natural evolution. A GA has the ability for evolving a solution, by exploring the search space with evolutionary heuristics, in same way of genetic information transference known from the Nature [13]. GA related to the broader area of evolutionary algorithms that produces optimal solutions, which are inspired from Darwin’s theory of natural evolution, such as fitness function, selec-

tion, mutation, and crossover. GA uses an initial population, where each individual within this population is a possible solution for the problem. In GA, an initial population evolves in direction of optimal solutions. GA starts with randomly generated population of individuals and occurs in generations. Each individual of this population evolved in various steps (mutation, crossover or repeated) to produce new generation of individuals for getting a better solution. The fitness function of every individual is evaluated in every generation. Selection considered the fitness function for selecting the individuals from the current population and modified to generate a next population. This new generation is then used in upcoming iterations of the process. Usually, the process terminates after reaching maximum generations or best fitness level has been achieved for the population.

In [70], the authors proposed an array model of reversible- based designs. This paper used universal gates such as wire gate, NOT gate, Toffoli gate and Feynman gates for configuring reversible- based functions. The model consists of cascade of gate-level reversible-based units in a $n \times m$ rectangular array fashion without the feedback and the fan-out as shown in Fig. 2, where n represents maximum input or output wire counts, m represents maximum number of cascade gate-level reversible-based units. Each chromosome generated by the cascade of T types of designable reversible functions and can be encoded by $E = (n \cdot A + n \cdot (p_{max} + q_{max}) \cdot B)$ bits, where $A = \lceil \log_2 T \rceil$ bits, $B = \lceil \log_2 n \rceil$ bits, p_{max} and q_{max} are the maximum controlled wire bit-count and controlling wire bits for all designable function respectively. New individuals are generated by BSAGA [70]. A pre-bit priority mechanism has been considered to ignore multiplexing error, which happens due to multiplexing between controlling wires and controlled wires.

In literature [27], GA algorithm is used to synthesize an exclusive-or sum-of-product (ESOP)-based structure. This method emphasizes the importance of well-designed encoding method and how it helps in fast convergence of GA. The fitness

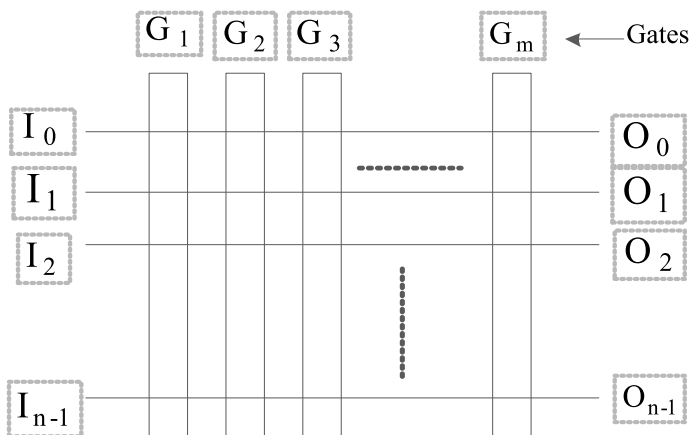


Fig. 2 Array model

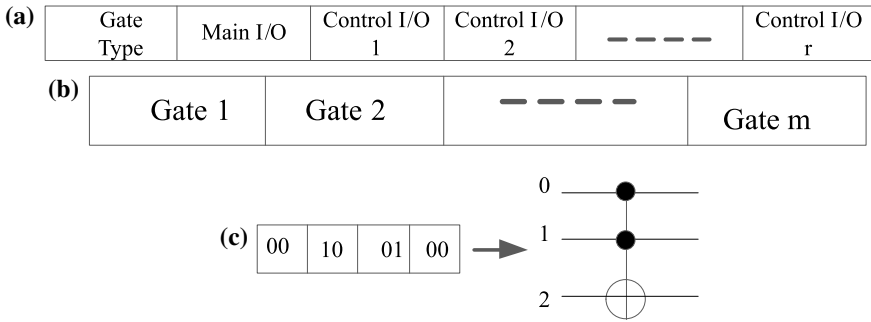


Fig. 3 a Coding of a generalized reversible gate. b Chromosome of a circuit with m gates. c Toffoli gate coding

function can handle any number of inputs and outputs, and suitable for incompletely specified functions. In this work stochastic universal sampling operator is used for selection of individuals because of its population diversity.

In literature [41, 42], aiming at the genetic algorithm (GA) to optimize a given specification. The method also synthesizes incompletely-defined functions. This optimization works more efficiently if the truth table of the given specification contain don't-care conditions and don't-care outputs (garbage outputs). In this chapter, the music line style is used for schematic of design, and a new coding method is proposed to encode a generalized $n \times n$ circuit with m gates, where n is maximum number of parallel input/output lines, and m represents maximum number of columns or gates placed on the parallel lines.

As shown in Fig. 3a some fields are associated with each gate. The first field shows the type of gate (If T types of configurable reversible logic gates available then this field need maximum $\lceil \log_2 T \rceil$ bits). The second field indicates the position of its main input or output. The line number of the main output given by this code, in number range from 0 to $n - 1$. The r fields give information about the position of the gate inputs. For instance a 3×3 Toffoli gate shown in Fig. 3c is encoded with 01, 10, 00, 01 (01 is considered as code of Toffoli gate). Figure 3b shows a chromosome, which represents a reversible circuit that contains m gates.

An improved ESOP-based realization of reversible function using genetic algorithm given in [10]. In this Pseudo Kronecker Expressions (PSDKROs) are used for very compact representation and the given algorithm is effective for functions having variables greater than 20.

Manna et al. [28] introduced a searching algorithm based on GA for realizing reversible layouts. This algorithm generates Toffoli gates network for realization of a reversible structure.

Table 1 Comparisons of figure of metrics based on GA

Function name	GA [4]			[70]
	GC	QC	TC	GC
[0, 1, 2, 3, 4, 5, 6, 7] 3-17	6	14	56	6
[0, 1, 2, 3, 4, 5, 6, 7] ham3	5	9	48	5
ham7	–	–	–	25
4mod5	5	55	56	5
graycode5	–	–	–	5
graycode6	5	5	40	–
rd32	–	–	–	4
rd53	–	–	–	12
[1, 0, 3, 2, 5, 7, 4, 6] rand_3_9	4	16	48	–
[7, 0, 1, 2, 3, 4, 5, 6] rand_3_1	3	7	24	–
[0, 1, 2, 3, 4, 6, 5, 7] rand_3_2	3	15	48	–
[0, 1, 2, 4, 3, 5, 6, 7] rand_3_3	5	17	64	–
[1, 2, 3, 4, 5, 6, 7, 0] rand_3_4	3	7	24	–
[3, 6, 2, 5, 7, 1, 0, 4] rand_3_5	7	19	72	–
[1, 2, 7, 5, 6, 3, 0, 4] rand_3_6	7	15	64	–
[4, 3, 0, 2, 7, 5, 6, 1] rand_3_7	6	10	48	–
[7, 5, 2, 4, 6, 1, 0, 3] rand_3_8	9	21	80	–
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0] rand_4_3	6	26	64	–
1-bit adder	4	12	48	–

In [4] author's proposed a similar algorithm, which is valid for any gate library of reversible-based logic. So, this chapter encoded generalized Toffoli gates to represent solution. This algorithm is suitable up to 4 or 5 variables and fails to provide solution for larger circuits. To avoid this, factor-based permutation cycles are used. Table 1 shows various functions name and the gate count for each implementation.

3 PSO-Based Synthesis Algorithm

PSO is a stochastically optimization method introduced by Kennedy and Eberhart [18], inspired from the social behavior of creatures such as bird flocking or fish shoaling. In PSO method, particles shows the solution, wander through a multidimensional search space, at every instance each particle modify its position based upon its own experience, and as per the experience of a neighboring particles, maximizing the usage of best positions confronted by itself and its neighboring particles. The method has basic attempt to merge local and global searching techniques in order to detect best feasible solutions.

Table 2 Comparisons of figure of metrics based on PSO

Function name	PSO [3]			[28]
	GC	QC	TC	GC
[1, 0, 3, 2, 5, 7, 4, 6] rand_3_9	5	9	40	4
[7, 0, 1, 2, 3, 4, 5, 6] rand_3_1	3	7	24	3
[0, 1, 2, 3, 4, 6, 5, 7] rand_3_2	3	15	48	3
[0, 1, 2, 4, 3, 5, 6, 7] rand_3_3	5	9	48	5
[1, 2, 3, 4, 5, 6, 7, 0] rand_3_4	3	7	24	4
[3, 6, 2, 5, 7, 1, 0, 4] rand_3_5	7	20	80	7
[1, 2, 7, 5, 6, 3, 0, 4] rand_3_6	6	14	56	7
[4, 3, 0, 2, 7, 5, 6, 1] rand_3_7	6	10	48	6
3_17	6	14	56	–
ham3	5	9	48	–
ex1	4	8	32	–
nth prime	4	8	40	–
miller	5	9	48	–
peres	2	6	24	–
fredkin	3	7	32	–
1bitadder	4	12	48	–
hwb4	12	24	120	–
4mod5(M)	5	55	72	–

Datta et al. [3] presented a PSO-based searching methods to realizing a reversible-based design. The method has ability to find near-optimal solution without searching the complete search space. Each particle in swarm creates a network structure represented as an array of generalized Toffoli gate. The array can be codified as a string of integer $[0 \text{ to } n \times 2^{n-1}]$. For a reversible design with n line, there are $n \times 2^{n-1}$ possible generalized Toffoli gates. For given specification f this algorithm generates N solutions each of k gates. At the initialization stage, each particle is initialized randomly. Fitness function accounts length of gates, mismatch, hamming distance between present and desire permutation for particles in the swarm. On each iteration the positions of particles changed using well- chosen random function. To accept the new positions for the next iteration, fitness function at new and old position is compared. In [28] similar synthesis algorithm is considered. Fitness function calculated as the ratio of number of matches and the length of the permutation. Table 2 shows various function name and the gate count for each implementation.

4 Ant Colony Optimization-Based Synthesis Algorithm

ACO algorithms are most efficient and widely used algorithm motivated by the foraging behavior of ant colonies [9]. Behavioral patterns exhibited by ants can be explored to find shortest paths in graphs and similar application. They establish a communication between individuals based on a chemical substance called pheromones, which they deposit and smell during their search of food from source to nest. This behavior can be used to develop an algorithm for the solution of optimization problems. The convergence of ACO depends on pheromone deposit on their forward and backward travel.

In literature [24], the author's introduced an ACO-based technique for reversible-based units to formulate the best-path search problem. The approach is capable to hand massive reversible operations and efficiently produces near-optimal design or optimal design having less number of gates. A generalized Toffoli gate library $\{TOF_k | k \leq n\}$ is proposed to implement an n -input reversible-based operation. Each Toffoli gates is designated as $g(\vec{c}, t)$, where c represents a vector of control bit out of n input bits, $\vec{c} = \{[c_1, c_2, \dots, c_n] | c_i \in \{0, 1, 2\}, i = 1, 2, \dots, n\}$. $c_i = 0$ indicates positive control bit i ; $c_i = 1$ represents negative control bit i , and $c_i = 2$ shows neither positive nor negative control line. t represents target bit $t \in \{1, 2, \dots, n\}$. Two graphs have been proposed. First a probabilistic state transition graph $G(V, E)$ that correlates the gate selection to help the ants during efficient path search process, which dynamically updated by pheromone levels and gate count after all the ants completed their tour. Second a weighted graph $G(C, A)$ called as an ant system graph (ASGraph) where $C = \{c_1, c_2, \dots, c_n\}$ is a finite set of elements (reversible operation). Set comprises of all the arcs (gates) linking the elements. So, the synthesis of reversible operation able to phrase like a minimization problem on an ASGraph, in which each arc weight w_{ij} is defined as the lowest cost of gates in a_{ij} .

Key steps in the algorithm:

- *Speculative model for gate selection* (DFS and BFS algorithm are used in local search to choose a gate $g(\vec{c}, t)$. The speculative model decides the target bit and the value of each control bit).
- *Initialization of τ_{pq}* (Ant takes decision based on set of pheromone values $\tau_{pq} = (\psi_{pq}, \phi_{pq})$, where τ_{pq} is the amount of pheromone for choosing bit t as target bit, and ϕ_{pq} is the amount of pheromone to set control bit i as c_i for target bit t if an ant moves from state p to state q . The updated pheromone values are stored in a hash table to optimize memory utilization).
- *Pheromone update* (The pheromone graph $G(V, E)$ get updated each time after all the ants completed their tour. Gates with less quantum cost get more pheromones than other) (Figs. 4a, b and 5).

Sarkar et al. [57] presented a modified version of classical Quine–McCluskey method under the guidance of ACO techniques has been proposed. Table 3 shows various function name and the gate count for each implementation.

Fig. 4 a State transition graph b weighted graph for reversible function

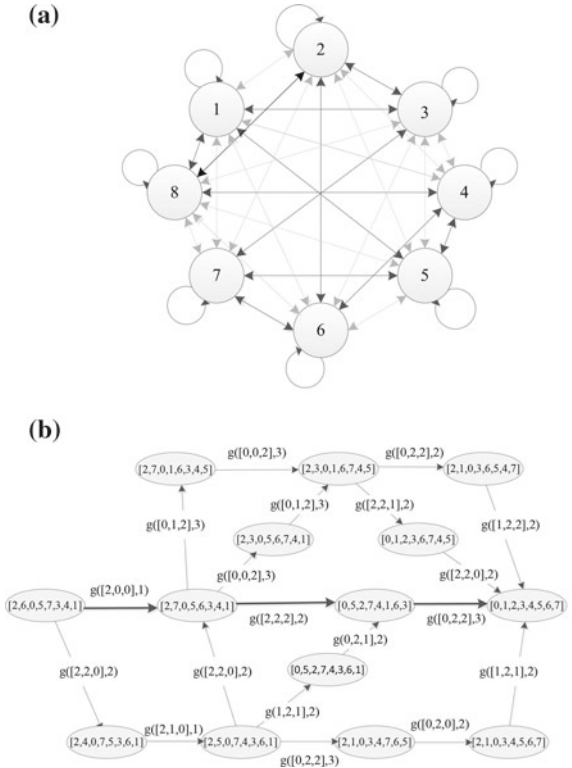


Fig. 5 Realization of reversible function [2, 6, 0, 5, 7, 3, 4, 1] using Toffoli gates

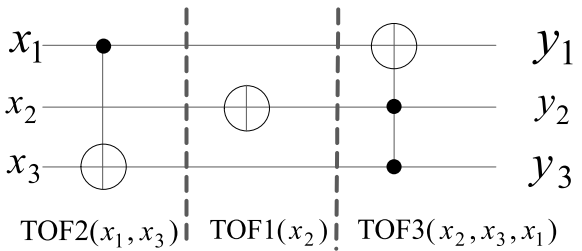


Table 3 Comparisons of figure of metrics based on ACO

Function name	GC	
	ACO based [24]	[57]
[1, 0, 3, 2, 5, 7, 4, 6] rand_3_9	3	5
[7, 0, 1, 2, 3, 4, 5, 6] rand_3_1	3	3
[0, 1, 2, 3, 4, 6, 5, 7] rand_3_2	3	5
[0, 1, 2, 4, 3, 5, 6, 7] rand_3_3	4	6
[0, 1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 14, 15] rand_4_2	7	10
[1, 2, 3, 4, 5, 6, 7, 0] rand_3_4	3	3
[1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 14, 15, 0] rand_4_3	3	4
[0, 7, 6, 9, 4, 11, 10, 13, 8, 15, 14, 1, 12, 3, 2, 5] rand_4_4	4	4
[3, 6, 2, 5, 7, 1, 0, 4] rand_3_5	6	8
[1, 2, 7, 5, 6, 3, 0, 4] rand_3_6	6	7
[4, 3, 0, 2, 7, 5, 6, 1] rand_3_7	5	7
[7, 5, 2, 4, 6, 1, 0, 3] rand_3_8	5	7
[6, 2, 14, 13, 3, 11, 10, 7, 0, 5, 8, 1, 15, 12, 4, 9] rand_4_5	11	14
[2, 9, 7, 13, 10, 4, 2, 14, 3, 0, 12, 6, 8, 15, 11, 1, 5]	11	–
[6, 4, 11, 0, 9, 8, 12, 2, 15, 5, 3, 7, 10, 13, 14, 1]	13	–
[13, 1, 14, 0, 9, 2, 15, 6, 12, 8, 11, 3, 4, 5, 7, 10] rand_4_1	10	–

5 Comparison and Discussion

To evaluate the effectiveness of EA-based synthesis algorithms, we have taken the best results available for each benchmark functions [14, 29, 52]. Table 4 shows the function name and the gate count for each implementation. We compare the best announced algorithm outputs for the NCT library. From Table 4 it is clear that the EA-based algorithm provides better results in terms of gate count (GC) in most of the cases. In other cases, synthesizing circuit using EA results with identical gates. For instance, rand_3_1, rand_3_2, which are implemented with identical gates.

6 Summary

In this paper, a brief algorithmic overview on different types of EA- based synthesis approach has been provided. Comparison with existing work shows EA-based method has better performance in GC, QC, and computational cost. In the last decade, study of reversible circuits and its synthesis methods has received significant attention and the results obtained are quite encouraging, still new efficient synthesis algorithm will remain an open challenge for the researchers.

Table 4 Benchmark comparison

Function name	Gate count					
	MOSAIC	MASLOV	PPRM	GA	PSO	ACO
	[52]	[29]	[14]	[4]	[16]	[24]
[1, 0, 3, 2, 5, 7, 4, 6] rand_3_9	4	–	4	4	5	3
[7, 0, 1, 2, 3, 4, 5, 6] rand_3_1	3	–	3	3	3	3
[0, 1, 2, 3, 4, 6, 5, 7] rand_3_2	3	–	3	3	3	3
[0, 1, 2, 4, 3, 5, 6, 7] rand_3_3	7	–	5	5	5	4
1 bit adder	–	–	–	4	4	4
4mod5	–	4	–	5	5	–
graycode6	–	–	–	5	–	–
graycode5	–	–	–	–	–	–
ham3	–	5	–	3	5	–
3_17	–	6	–	6	6	–
[0, 1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 14, 15] rand_4_2	9	–	7	–	–	7
[1,2,3,4,5,6,7,0] rand_3_4	3	–	3	3	3	3
[1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 14, 15, 0] rand_4_3	4	–	4	–	–	3
[0, 7, 6, 9, 4, 11, 10, 13, 8, 15, 14, 1, 12, 3, 2, 5] rand_4_4	3	–	4	–	–	4
[3, 6, 2, 5, 7, 1, 0, 4] rand_3_5	8	–	7	7	8	6
[1, 2, 7, 5, 6, 3, 0, 4] rand_3_6	8	–	7	7	6	6
[4, 3, 0, 2, 7, 5, 6, 1] rand_3_7	6	–	6	6	6	5
[7, 5, 2, 4, 6, 1, 0, 3] rand_3_8	6	–	7	9	–	5
[6, 2, 14, 13, 3, 11, 10, 7, 0, 5, 8, 1, 15, 12, 4, 9] rand_4_5	19	–	14	–	–	11
[2, 9, 7, 13, 10, 4, 2, 14, 3, 0, 12, 6, 8, 15, 11, 1, 5]	23	–	14	–	–	11
[6, 4, 11, 0, 9, 8, 12, 2, 15, 5, 3, 7, 10, 13, 14, 1]	21	–	17	–	–	13
[13, 1, 14, 0, 9, 2, 15, 6, 12, 8, 11, 3, 4, 5, 7, 10] rand_4_1	29	–	14	–	–	10

References

1. Abdollahi A, Pedram M (2006) Analysis and synthesis of quantum circuits by using quantum decision diagrams. In: Proceedings of design, automation and test in Europe (DATE), vol 1, pp 1–6
2. Al-Rabadi AN (2004) New classes of Kronecker-based reversible decision trees and their group theoretic representations. In: Proceedings of the international workshop on spectral methods and multirate signal processing (SMMSP), pp 233–243
3. Datta K, Sengupta I, Rahaman H (2012) Particle swarm optimization based circuit synthesis of reversible logic. In: Proceedings of international symposium on electronic system design (ISED), pp 226–230
4. Datta K, Sengupta I, Rahaman H (2012) Reversible circuit synthesis using a evolutionary algorithm. In: Proceedings of 25th international conference on computers and devices for communication (CODEC), pp 1–4

5. De Vos A, Van Rentergem Y (2009) Multiple-valued reversible logic circuits. *J Mult-Valued Log Soft Comput* 15(5/6):489–505
6. De Vos A, Van Rentergem Y, De Keyser K (2006) The decomposition of an arbitrary reversible logic circuit. *J Phys A Math Gen* 39(18):5015–035
7. Dirac PAM (1939) A new notation for quantum mechanics. *Math Proc Camb Philos Soc* 144:416–418
8. Donald J, Jha NK (2008) Reversible logic synthesis with fredkin and peres gates. *J Emerg Technol Comput Syst* 4
9. Dorigo M (1995) Optimization, learning and natural algorithms, PhD thesis, Politecnico di Milano
10. Drechsler R, Finder A, Wille R (2011) Improving ESOP-based synthesis of reversible logic using evolutionary algorithms. In: *EvoApplications*, vol 2, pp 151–161
11. Dueck GW, Maslov R, Miller DM (2003) Transformation-based synthesis of networks of Toffoli/Fredkin gates. In: *Proceedings of the IEEE Canadian conference on electrical and computer engineering*, vol 1, pp 211–214
12. Fazel K, Thornton M, Rice JE (2007) ESOP-based Toffoli gate cascade generation. In: *Proceedings of the IEEE Pacic Rim conference on communications, computers and signal processing (PACRIM)*, pp 206–209
13. Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Professional, Boston
14. Gupta P, Agrawal A, Jha NK (2006) An algorithm for synthesis of reversible logic circuits. *IEEE Trans Comput-Aided Des Integr Circuits Syst* 25:2317–2330
15. Hamza Z, Dueck GW (2010) Near-optimal ordering of ESOP cubes for Toffoli networks. In: *Proceedings of the 2nd annual workshop on reversible computation (RC)*
16. Hung WNN, Song X, Yang G, Yang J, Perkowski M (2006) Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. *IEEE Trans Comput-Aided Des* 25(9):1652–1663
17. Iwama K, Kambayashi Y, Yamashita S (2002) Transformation rules for designing CNOT based quantum circuits. In: *Proceedings of the 39th design automation conference*, pp 419–424
18. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, IEEE Service Center, pp 1942–1948
19. Kerntopf P (2004) A new heuristic algorithm for reversible logic synthesis. In: *Proceedings of the design automation conference*, pp 834–837
20. Khan MHA, Perkowski M (2004) Genetic algorithm based synthesis of multi-output ternary functions using quantum cascade of generalized ternary gates. In: *Proceedings of the 2004 congress on evolutionary computation*
21. Kole D, Rahaman H, Das DK, Bhattacharya B (2010) Optimal reversible logic circuits synthesis based on a hybrid dfs-bfs technique. In: *International symposium on electronic system design (ISED 2010)*, pp 208–212
22. Landauer R (1961) Irreversibility and heat generation in the computing process. *IBM Res. Dev.* 5:183–191
23. Likharev KK (1982) Classical and quantum limitations on energy-consumption in computation. *Int J Theor Phys* 21:311–325
24. Li M, Zheng Y, Hsiao MS, Huang C (2010) Reversible logic synthesis through and colony optimization. In: *Proceedings of design automation test in Europe (DATE 2010)*, pp 208–212
25. Lukac M, Perkowski M, Gol H (2003) Evolutionary approach to quantum and reversible circuits synthesis. *Artif Intell Rev* 20(3–4):361–417
26. Lukac M, Perkowski M (2002) Evolving quantum circuits using genetic algorithm. In: *Proceedings of the 2002 NASA/DoD conference on evolvable hardware*
27. Lukac M, Pivtoraiko M, Mishchenko A, Perkowski M (2002) Automated synthesis of generalized reversible cascades using genetic algorithms. In: *Proceedings of the 5th international workshop on Boolean problems*, Freiberg, pp 33–45
28. Manna P, Kole DK, Rahaman H, Das DK, Bhattacharya BB (2012) Reversible logic circuit synthesis using genetic algorithm and particle swarm optimization. In: *Proceedings of international symposium on electronic system design*, pp 246–250

29. Maslov D (2005) Reversible logic synthesis benchmarks page. <http://www.cs.uvic.ca/~dmaslov/>
30. Maslov D, Dueck GW, Miller DM (2005) Toffoli network synthesis with templates. *IEEE Trans Comput-Aided Des Integr Circuits Syst (TCAD)* 24(6):807–817
31. Maslov D, Dueck GW, Miller DM (2003) Fredkin/Toffoli templates for reversible logic synthesis. In: *International conference on computer aided design (ICCAD)*, pp 256–261
32. Maslov D, Dueck GW, Miller DM (2003) Simplification of Toffoli networks via templates. In: *Proceedings of the 16th symposium on integrated circuits and system design*
33. Maslov D, Dueck GW, Miller DM (2003) Templates for Tffoli network synthesis. In: *Proceedings of the international workshop on logic synthesis*
34. Maslov D, Dueck GW, Miller DM (2005) Synthesis of Fredkin-Toffoli reversible networks. *IEEE Trans Very Large Scale Integr (VLSI) Syst* 13(6):765–769
35. Maslov D, Dueck GW, Miller DM (2007) Techniques for the synthesis of reversible Toffoli networks. *ACM Trans Des Autom Electron Syst (TODAES)* 12(4):42
36. Merkle RC (1993) Two types of mechanical reversible logic. *Nanotechnology* 4(2):114–131
37. Merkle RC (1993) Reversible electronic logic using switches. *Nanotechnology* 7:21–40
38. Miller DM, Maslov D, Dueck GW (2003) A transformation based algorithm for reversible logic synthesis. In *Proceedings of the design automation conference*, pp 318–323
39. Miller DM, Maslov D, Dueck GW (2003) A transformation based algorithm for reversible logic synthesis. In: *DAC '03: proceedings of the 40th conference on design automation*, pp 318–323. ACM
40. Miller DM, Thornton MA (2006) QMDD: a decision diagram structure for reversible and quantum circuits. In: *Proceedings of the IEEE international symposium on multiple-valued logic (ISMVL)*, p #30 on *Proceedings CDRom*
41. Mohamadi M, Eshghi M (2008) Heuristic methods to use don't cares in automated design of reversible and quantum logic circuits. *Quantum Inf Process J (Springer)* 7:175–192
42. Mohammadi M (2012) Efficient genetic based methods for optimizing the reversible and quantum logic circuits. *JACR* 3(3):85–96
43. Mohammadi M, Eshghi M (2009) On figures of merit in reversible and quantum logic designs. *Quantum Inf Process J (Springer)* 8(4):297–318
44. Moore GE (1965) Cramping more components onto integrated circuits. *Electronics* 38(8)
45. Nayeem NM, Rice JE (2011) Improved ESOP-based synthesis of reversible logic. In: *Proceedings of the 2011 Reed-Muller workshop*, pp 57–62
46. Nielsen M, Chuang I (2000) *Quantum computation and quantum information*. Cambridge University Press, Cambridge
47. Rice JE, Nayeem N (2011) Ordering techniques for ESOP-based Toffoli cascade generation. In: *Proceedings of the IEEE Pacific Rim conference on communications, computers and signal processing (PACRIM)*, pp 274–279
48. Rice JE, Suen V (2010) Using autocorrelation coefficients-based cost functions in ESOP based Toffoli gate cascade generation. In: *Proceedings of the 23rd IEEE Canadian conference on electrical and computer engineering (CCECE)*, pp 1–6
49. Saeedi M, Sedighi M, Zamani MS (2007) A novel synthesis algorithm for reversible circuits. In: *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp 65–68
50. Saeedi M, Sedighi M, Zamani MS (2010) A library-based synthesis methodology for reversible logic. *Microelectron J* 41(4):185–194
51. Saeedi M, Zamani MS, Sedighi M (2007) On the behaviour of substitution-based reversible circuit synthesis algorithms: investigation and improvement. In: *Proceedings ISVLSI*, pp 428–436
52. Saeedi M, Zamani MS, Sedighi M (2008) Moving forward: a nonsearch based synthesis method toward efficient CNOT-based quantum circuit synthesis algorithms. In: *ASPDAC*, pp 83–88
53. Saeedi M, Zamani MS, Sedighi M, Sasanian Z (2010) Reversible circuit synthesis using a cycle-based approach. *JETC* 6(4):13
54. Sanaee Y, Dueck GW (2009) Generating Toffoli networks from ESOP expressions. In: *Proceedings of the IEEE Pacific Rim conference on communications, computers and signal processing (PACRIM)*, pp 715–719

55. Sanaee Y, Dueck GW (2010) ESOP-based Toffoli network generation with transformations. In: Proceedings of the 40th IEEE international symposium on multiple-valued logic (ISMVL), pp 276–281
56. Sanaee Y, Dueck GW (2010) ESOP-based Tooli network generation with transformations. In: Proceedings of the 40th IEEE international symposium on multiple-valued logic (ISMVL), pp 276–281
57. Sarkar M, Ghosal P, Mohanty SP (2013) Reversible circuit synthesis using ACO and SA based Quine-McCluskey method. In: Proceedings of 56th international midwest symposium on circuits and systems (MWSCAS), pp 416–419
58. Sasanian Z, Saeedi M, Sedighi M, Zamani MS (2009) A cycle-based synthesis algorithm for reversible logic. In: Proceedings of Asia South Pacific design automation conference (ASP-DAC), pp 745–750
59. Shende VV, Prasad AK, Markov IL, Hayes JP (2003) Synthesis of reversible logic circuits. *IEEE Trans Comput Aided Des* 22(6):710–722
60. Storme L, Vos AD, Jacobs G (1999) Group theoretical aspects of reversible logic gates. *J Univers Comput Sci* 5(5):307–321
61. Thornton M, Miller DM, Goodman D (2006) A decision diagram package for reversible and quantum circuit simulation. In: Proceedings of the IEEE congress on evolutionary computation, pp 8597–8604
62. Van Rentergem Y, De Vos A, Storme L (2005) Implementing an arbitrary reversible logic gate. *J Phys A Math Gen* 38(16):3555–3577
63. Van Rentergem Y, De Vos A, De Keyser K (2006) Using group theory in reversible computing. In: Proceedings of the 2006 IEEE congress on evolutionary computation (CEC2006), pp 2397–2404
64. Wille Robert, Drechsler Rolf (2010) BDD-based synthesis of reversible logic. *Int J Appl Meta-heuristic Comput (IJAMC)* 1(4):25–41
65. Wille R, Drechsler R (2009) BDD-Based synthesis of reversible logic for large functions. In: Proceedings of the design automation conference, pp 270–275
66. Wille R, Drechsler R, Oswald C, Garcia-Ortiz A (2012) Automatic design of low power encoders using reversible circuit synthesis. In: Design, automation and test in Europe, pp 1036–1041
67. Wille R, Groe D, Miller DM, Drechsler R (2009) Equivalence checking of reversible circuits. In: Proceedings of the 39th international symposium on multiple-valued logic (IS-MVL), pp 324–330
68. Yang G, Song X, Hung WN, Xie F, Perkowski MA (2006) Group theory based synthesis of binary reversible circuits. *Lecture notes in computer science*, vol 3959. Springer, pp 365–374
69. Younis SG, Knight TF (1994) Asymptotically zero energy split-level charge recovery logic. In: Workshop low power design, pp 177–182
70. Zhang M, Zhao S, Wang X (2009) Automatic synthesis of reversible logic circuit based on genetic algorithm. In Proceedings of IEEE international conference on intelligent computing and intelligent systems (ICIS 2009), pp 542–546