

Reversible Circuit : Opportunities and Challenges

Brajesh Kumar Sharma

Abstract— After decades of continuous improvements and shrinking feature sizes, the development of conventional computing technologies faces enormous challenges. In particular, power dissipation in today's computer chips becomes crucial. Reversible computation is a promising alternative to these technologies, where power dissipation can be reduced or even eliminated. Furthermore, reversible logic builds the basis for quantum computation – a completely new way of processing which enables to solve certain problems exponentially faster compared to conventional methods. However, the design of reversible and quantum circuits is significantly different. Thus, new methods e.g. for synthesis, optimization, or verification are needed. This paper provides a brief introduction into reversible circuits and their respective design methods that have been proposed within the last years.

Index Terms— shrinking feature, faces enormous, synthesis, optimization.

I. INTRODUCTION

Power dissipation and therewith heat generation is a serious problem for today's computer chips. A significant part of energy dissipation is due to the non-ideal behavior of transistors and materials. Here, higher levels of integration and new fabrication processes reduced the heat generation in the last decade.

However, a more fundamental reason for power dissipation arises from the observations made by Landauer already in 1961 [1]. Landauer proved that using conventional (irreversible) logic, gate operations always lead to energy dissipation regardless of the underlying technology. More precisely, exactly $k \cdot T \cdot \log 2$ Joule of energy are dissipated for each "lost" bit of information during an irreversible operation (where k is the Boltzmann constant and T is the temperature). While this amount of power currently does not sound significant, it may become crucial additionally considering that (1) today millions of operations are performed in some seconds (i.e. increasing processor frequency multiplies this amount) and (2) more and more operations are performed with smaller and smaller transistor sizes (i.e. in a smaller area).

In contrast, Bennett showed that energy dissipation can be reduced or even eliminated if computation becomes information-lossless [2]. This does not hold for

Brajesh Kumar Sharma, Department of Computer Science, Hindustan Institute of Technology and Management, Agra (U.P) India.

conventional circuits (e.g. already the simple AND operation transforms two input bits into a single output bit, i.e. one bit is lost leading to the above mentioned power dissipation). But, reversible circuits, i.e. circuits where all

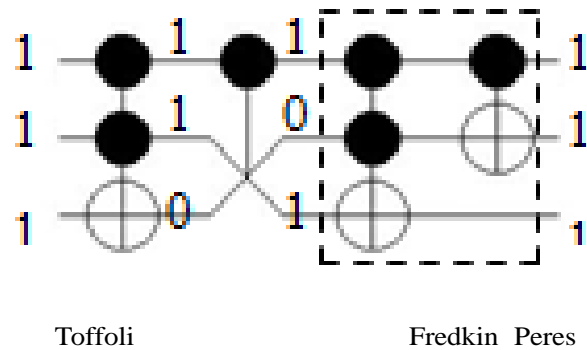


Fig. 1. Reversible gates

II. REVERSIBLE CIRCUITS

Reversible logic realizes n -input n -output functions that map each possible input vector to a unique output vector – in other words bijections are realized. In reversible circuits, fan-out and feedback are not allowed [4]. As a consequence, reversible circuits are cascades of reversible gates.

Let $X := \{x_1, \dots, x_n\}$ be the set of Boolean variables. Then, a reversible gate has the form $g(C, T)$, where $C = \{x_{i1}, \dots, x_{ik}\} \subset X$ is the set of *control lines* and $T = \{x_{j1}, \dots, x_{jl}\} \subset X$ with $C \cap T = \emptyset$ is the set of *target lines*. In the past, multiple control Toffoli [5], multiple control Fredkin [6], and Peres [7] gates have been established:

- A *multiple control Toffoli gate* (TOF) has a single target line x_j . The gate maps $(x_1, x_2, \dots, x_j, \dots, x_n)$ to $(x_1, x_2, \dots, x_{i1} x_{i2} \dots x_{ik} \oplus x_j, \dots, x_n)$, i.e., the value of the target line is inverted if all control lines evaluate to 1.
- A *multiple control Fredkin gate* has two target lines x_{j1} and x_{j2} . The values of the target lines are interchanged if all control lines evaluate to 1.
- A *Peres gate* has one control line x_i and two target lines x_{j1} and x_{j2} . The gate maps $(x_1, x_2, \dots, x_{j1}, \dots, x_{j2}, \dots, x_n)$ to

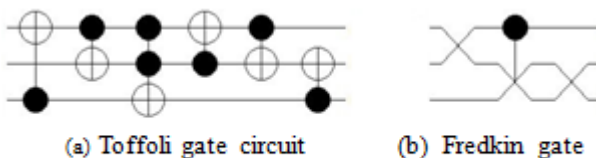
$(x_1, x_2, \dots, x_i \oplus x_{j1}, \dots, x_i x_{j1} \oplus x_{j2}, \dots, x_n)$, i.e., the Peres gate is a cascade of two Toffoli gates (with two and one control lines, respectively).

Fig. 1 shows a Toffoli gate, a Fredkin gate, and a Peres gate in a cascade. The control lines are denoted by ●, while the target lines are denoted by ⊕ (except for the Fredkin gate where a × is used instead). The annotated values demonstrate the computation of the respective gates. As shown, the calculation can be done in both directions, i.e. the calculation is reversible.

To measure the cost of a reversible circuit, different metrics are applied (sometimes depending on the addressed technology). In general, the number of circuit lines is an important criterion. In particular in the domain of quantum computation, a very prominent application of reversible circuits [4], the number of lines is crucial. Beyond that, the costs of the respective gates themselves are important, too. But simply counting the number of gates does not adequately reflect the effort to realize them. Hence, also the following, more technology depended metrics are applied:

TABLE I
QUANTUM COST FOR TOFFOLI AND FREDKIN GATES

NO. OF CONTROL LINES	QUANTUM COST	
	OF A TOFFOLI GATE	OF A FREDKIN GATE
0	1	3
1	1	7
2	5	15
3	13	28, if at least 2 lines are unconnected 31, otherwise
4	26, if at least 2 lines are unconnected 29, otherwise	40, if at least 3 lines are unconnected 54, if 1 or 2 lines are unconnected 63, otherwise
5	38, if at least 3 lines are unconnected 52, if 1 or 2 lines are unconnected 61, otherwise	52, if at least 4 lines are unconnected 82, if 1, 2 or 3 lines are unconnected 127, otherwise



circuit Fig. 2. Reversible circuits

Quantum cost denotes the effort needed to transform a reversible circuit to a quantum circuit. Table I shows the quantum cost for a selection of Toffoli and Fredkin gate configurations as introduced in [9] and further

optimized e.g. [10]. As can be seen, gates of larger size are considerably more expensive than gates of smaller size. The Peres gate represents a special case, since it has quantum cost of 4, while the realization with two Toffoli gates would imply costs of 6.

- Transistor cost denotes the effort needed, to realize a reversible circuit in CMOS according to [11]. The transistor cost of a reversible gate is $8 \cdot s$ where s is the number of control lines.

As an example, consider the circuits depicted in Fig. 2. The circuit composed of Toffoli gates (Fig. 2(a)) has a gate count of 6, quantum cost of 10, and transistor cost of 56. The circuit composed of Fredkin gates (Fig. 2(b)) has a gate count of 3, quantum cost of 13, and transistor cost of 8, respectively.

III. DESIGN METHODS

The model for reversible circuits as introduced in the last section already represents the basis for research in the domain of reversible circuit design. Nevertheless, reversible circuits have not been intensively studied by researchers before the year 2000. The main reason for that may lie in the fact that applications of reversible logic have been seen as “dreams of the future”. But, this changed as first physical realizations of reversible circuits emerged, e.g. in terms of quantum circuits [12] or in terms of reversible CMOS realizations with certain low-power properties [3]. Therewith, proofs of concept were available motivating reversible circuits as a promising research area.

As a consequence, in the last years researchers started

to develop new methods for the design of this kind of circuits. In the following, some of these methods are briefly reviewed.

A. Synthesis

Synthesis is the most important step while building complex circuits. Considering the conventional design flow, synthesis is carried out in several individual steps such as high-level synthesis, logic synthesis, mapping, and routing. To synthesize reversible logic, adjustments and extensions are needed. For example, further tasks such as embedding of irreversible functions must be added [13], [14]. Furthermore, throughout the whole flow, the restrictions caused by the reversibility (no fanout and feedback) and a completely new gate library must be considered as well.

Existing synthesis approaches addressing these issues

can be categorized as follows:

- 1) Boolean Synthesis Approaches for Small Functions:

These approaches can handle small Boolean functions, e.g. provided in terms of permutations [15], [16], truth tables [17], [18], [19], positive-polarity Reed-Muller expansion [20], or Reed-Muller spectra [21]. Besides that, also exact approaches exist that do not only generate a reversible circuit for a given function, but additionally ensure that the resulting circuit is composed of a minimal number of gates [22].

The scalability of all these approaches is thereby limited. Usually only circuits for functions containing not more than 30 variables can be obtained with them. Hence, if larger functions should be synthesized, more compact function descriptions and, accordingly, other synthesis approaches have to be considered.

2) *Boolean Synthesis Approaches for Larger Functions:* Synthesis methods for larger functions make use of more compact function representations. In particular, approaches based on Exclusive Sum of Products [23] and approaches based on Binary Decision Diagrams [24] fall into this category. Here, the structure of the function representation is mapped to respective reversible sub-circuits. By cascading the resulting sub-circuits, the overall function is realized.

3) *Synthesis Approach based on a Programming Language:*

While the previous approaches rely on Boolean descriptions for the synthesis of reversible circuits, recently also a programming language has been proposed for this purpose [25]. This language, called *SyReC*, enables the design of more complex reversible systems

B. Optimization

After synthesis, the resulting circuits often are of high cost. In particular, dedicated technology-specific constraints are not considered by synthesis approaches. To address this, optimization methods have been introduced. In particular, the reduction of the quantum cost of given circuits has been considered [26], [18], [27], [28]. But also reducing the number of circuit lines [29] or optimization of further, more technology depended cost metrics [30] was subject of research activities.

C. Verification and Debugging

To ensure that the respective results (e.g. obtained by optimization) still represent the desired functionality, verification is applied. For this purpose, first verification approaches have been introduced [31], [32], [33], [34]. Moreover, even automatic debugging methods aimed at supporting the designer to detect the error in case of a failed verification are already available [35].

IV. CONCLUSIONS

In this paper, reversible circuits have been briefly reviewed. This kind of circuits has promising applications with respect to low-power design since here all computations are performed without information-loss (a necessary condition of circuits to have zero power dissipation). Furthermore, reversible circuits also find application in the domain of quantum computation. First physical realizations confirming the promising properties are already available.

Motivated by this, in the last years researchers started to develop new methods for the design of this kind of circuits. In this paper, some of the resulting methods have been briefly summarized. References for a more in-depth treatment have been provided in the respective sections. Furthermore, implementations of many of the reviewed approaches are available at RevKit [36].

However, despite the progress made in this area, research on reversible circuits is still in its infancy and further research is needed. Considering the existing design methods, many of the proposed approaches have limitations with respect to the size of the function to be synthesized or the circuit to be considered, respectively. Furthermore, circuits obtained by current synthesis approaches often are of significant costs. Finally, most of the approaches have been considered separately. So far, no integrated design flow for complex reversible circuits exists.

REFERENCES

- [1] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Dev.*, vol. 5, p. 183, 1961.
- [2] C. H. Bennett, "Logical reversibility of computation," *IBM J. Res. Dev.*, vol. 17, no. 6, pp. 525–532, 1973.
- [3] B. Desoete and A. D. Vos, "A reversible carry-look-ahead adder using control gates," *INTEGRATION, the VLSI Jour.*, vol. 33, no. 1-2, pp. 89–104, 2002.
- [4] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.
- [5] T. Toffoli, "Reversible computing," in *Automata, Languages and Programming*, W. de Bakker and J. van Leeuwen, Eds. Springer, 1980, p. 632, technical Memo MIT/LCS/TM-151, MIT Lab. for Comput. Sci.
- [6] E. F. Fredkin and T. Toffoli, "Conservative logic," *International Journal of Theoretical Physics*, vol. 21, no. 3/4, pp. 219–253, 1982.
- [7] A. Peres, "Reversible logic and quantum computers," *Phys. Rev. A*, no. 32, pp. 3266–3276, 1985.
- [8] R. Wille and R. Drechsler, *Towards a Design Flow for Reversible Logic*. Springer, 2010.
- [9] A. Barenco, C. H. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, "Elementary gates for quantum computation," *The American Physical Society*, vol. 52, pp. 3457–3467, 1995.
- [10] D. Maslov, C. Young, G. W. Dueck, and D. M. Miller, "Quantum circuit simplification using templates," in *Design, Automation and Test in Europe*, 2005, pp. 1208–1213.
- [11] M. K. Thomson and R. Glück, "Optimized reversible binary-coded decimal adders," *J. of Systems Architecture*, vol. 54, pp. 697–706, 2008.
- [12] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of

- Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, p. 883, 2001.
- [13] D. Maslov and G. W. Dueck, "Reversible cascades with minimal garbage," *IEEE Trans. on CAD*, vol. 23, no. 11, pp. 1497–1509, 2004.
- [14] D. M. Miller, R. Wille, and G. Dueck, "Synthesizing reversible circuits for irreversible functions," in *EUROMICRO Symp. on Digital System Design*, 2009, pp. 749–756.
- [15] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. on CAD*, vol. 22, no. 6, pp. 710–722, 2003.
- [16] M. Saeedi, M. S. Zamani, M. Sedighi, and Z. Sasanian, "Synthesis of reversible circuit using cycle-based approach," *J. Emerg. Technol. Comput. Syst.*, vol. 6, no. 4, 2010.
- [17] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in *Design Automation Conf.*, 2003, pp. 318–323.
- [18] D. Maslov, G. W. Dueck, and D. M. Miller, "Toffoli network synthesis with templates," *IEEE Trans. on CAD*, vol. 24, no. 6, pp. 807–817, 2005.
- [19] R. Wille, D. Große, G. Dueck, and R. Drechsler, "Reversible logic synthesis with output permutation," in *VLSI Design*, 2009, pp. 189–194.
- [20] P. Gupta, A. Agrawal, and N. K. Jha, "An algorithm for synthesis of reversible logic circuits," *IEEE Trans. on CAD*, vol. 25, no. 11, pp. 2317–2330, 2006.
- [21] D. Maslov, G. W. Dueck, and D. M. Miller, "Techniques for the synthesis of reversible Toffoli networks," *ACM Trans. on Design Automation of Electronic Systems*, vol. 12, no. 4, 2007.
- [22] D. Große, R. Wille, G. W. Dueck, and R. Drechsler, "Exact multiple control Toffoli network synthesis with SAT techniques," *IEEE Trans. on CAD*, vol. 28, no. 5, pp. 703–715, 2009.
- [23] K. Fazel, M. Thornton, and J. Rice, "ESOP-based Toffoli gate cascade generation," in *Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on*, 2007, pp. 206–209.
- [24] R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions," in *Design Automation Conf.*, 2009, pp. 270–275.
- [25] R. Wille, S. Offermann, and R. Drechsler, "SyReC: A programming language for synthesis of reversible circuits," in *Forum on Specification and Design Languages*, 2010, pp. 184–189.
- [26] K. Iwama, Y. Kambayashi, and S. Yamashita, "Transformation rules for designing CNOT-based quantum circuits," in *Design Automation Conf.*, 2002, pp. 419–424.
- [27] J. Zhong and J. Muzio, "Using crosspoint faults in simplifying Toffoli networks," in *IEEE North-East Workshop on Circuits and Systems*, 2006, pp. 129–132.
- [28] D. M. Miller, R. Wille, and R. Drechsler, "Reducing reversible circuit cost by adding lines," in *Int'l Symp. on Multi-Valued Logic*, 2010.
- [29] R. Wille, M. Soeken, and R. Drechsler, "Reducing the number of lines in reversible circuits," in *Design Automation Conf.*, 2010, pp. 647–652.
- [30] M. Saeedi, R. Wille, and R. Drechsler, "Synthesis of quantum circuits for linear nearest neighbor architectures," *Quantum Information Processing*, 2010.
- [31] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Checking equivalence of quantum circuits and states," in *Int'l Conf. on CAD*, 2007, pp. 69–74.
- [32] S. Gay, R. Nagarajan, and N. Papanikolaou, "QMC: A model checker for quantum systems," in *Computer Aided Verification*, 2008, pp. 543–547.
- [33] S.-A. Wang, C.-Y. Lu, I.-M. Tsai, and S.-Y. Kuo, "An XQDD-based verification method for quantum circuits," *IEICE Transactions*, vol. 91-A, no. 2, pp. 584–594, 2008.
- [34] R. Wille, D. Große, D. M. Miller, and R. Drechsler, "Equivalence checking of reversible circuits," in *Int'l Symp. on Multi-Valued Logic*, 2009, pp. 324–330.
- [35] R. Wille, D. Große, S. Frehse, G. W. Dueck, and R. Drechsler, "Debugging of Toffoli networks," in *Design, Automation and Test in Europe*, 2009, pp. 1284–1289.
- [36] M. Soeken, S. Frehse, R. Wille, and R. Drechsler, "RevKit: A toolkit for reversible circuit design," in *Workshop on Reversible Computation*, 2010, pp. 69–72, RevKit is available at <http://www.revkit.org>.