



Available online at <https://int-scientific-journals.com>

International Scientific Journals



IJEISR (2019) Vol.3–no.2

<https://int-scientific-journals.com/ijeisr>

Reversible Computing: An Introduction

Federica Eftimiadi

Student of Electronics, Information and Bioengineering (Elettronica, Informazione e Bioingegneria), Polytechnic University, Milan., Italy

Email: federica.eftimiadi@polimi.it

Enrico Pugini Trimigliozzi

Faculty of Electronics, Information and Bioengineering (Elettronica, Informazione e Bioingegneria), Polytechnic University, Milan., Italy

Email: ep.trimigliozzi@polimi.it

Abstract

Reversible computing is a paradigm where computing models are defined so that they reflect physical reversibility, one of the fundamental microscopic physical property of Nature. Also, it is one of the basic microscopic physical laws of nature. Reversible computing refers to the computation that could always be reversed to recover its earlier state. It is based on reversible physics, which implies that we can never truly erase information in a computer. Reversible computing is very difficult and its engineering hurdles are enormous. This paper provides a brief introduction to reversible computing. With these constraints, one can still satisfactorily deal with both functional and structural aspects of computing processes; at the same time, one attains a closer correspondence between the behavior of abstract computing systems and the microscopic physical laws (which are presumed to be strictly reversible) that underlay any implementation of such systems

Doi: [10.31219/osf.io/z9a23](https://doi.org/10.31219/osf.io/z9a23)

© 2019 The Authors. Published by ISJ.

This is an open access article under the CC BY 4.0 license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Composition Rule, reversible computing, reverse computation, quantum computing, adiabatic computing.

Introduction

The world of computing is in transition. Conventional computing models such as Turing machines and random access machines destroy information at each computation step. As chips become smaller and faster, they dissipate or waste energy. Reverse computing has been found to provide more efficient forms of computation. Reverse computing means computation that takes place in steps that are time reversible. If a reversible computation yields a series of outputs, then the inverse computation on these outputs should reproduce the original states [3]. This idea of reversible computing goes to the very heart of thermodynamics and information theory. It is the only possible way to keep improving the cost and energy efficiency of computers far into the future. The idea is to get rid of the immense energy dissipation and heat generation caused by the irreversibility of conventional computing processes. Reversible computing needs to be adiabatic, as nearly as possible [1,4]. That essentially means that there is no heat flow between subsystems and the system is thermodynamically reversible [8].

Reversible Computing Systems

The idea of reversible computing came from physicist Rolf Landauer of IBM, who published a paper in 1961 titled “Irreversibility and Heat Generation in the Computing Process.” According to Landauer, in order for a computational process to be physically reversible, it must also be logically reversible. This means that the most fundamental laws of physics are reversible [15]. Landauer’s idea has since been experimentally confirmed. The reversibility of physics implies that we can never truly erase information in a computer. Reversible computing did not receive much attention in the past because it is hard to implement [2,5].

Reversible computing obeys the von Neumann-Landauer principle which implies that ordinary irreversible logic operations incur a fundamental minimum energy cost [16]. It difficult for practical technology to progress far beyond current levels of energy efficiency if reversible computing principles are not applied [9,11]. Implementing reversible systems requires estimating its cost and judging its limits. In these systems computation is often carried out in a different manner from conventional (or irreversible) computing systems [7,16].

Reversible computing operations can be carried out by idealized lossless electronic circuits that used inductors and capacitors, with no resistors damping the flow of energy. Unfortunately, these idealized electronic implementations cannot be built in practice [3,6].

Applications

There are some applications in which reverse computation is natural. Examples include quantum computer simulation, queuing network simulation, and the low-power CMOS (complementary metal-oxide-semiconductor). Quantum computing is closely related to reverse computing and has a lot in common with reversible computing [8]. Molecular quantum-dot cellular automata (QCA) is a practical implementation of reversible computing. It is a promising nanotechnology that offers significant improvements such as power dissipation over CMOS. It is possible to build reversible logic circuits using QCA [1,13]. Reverse computation is well suited for the optimistic simulation of queuing network models [5,9].

Typical models of reversible computing include reversible Turing machines, reversible cellular automata, reversible logic elements, and reversible physical models [7,11]. A reversible computer promises to reduce energy consumption and thereby becomes very useful in laptops, mobile devices, and other battery operated devices which run on limited-life batteries. It would require new hardware designs, software, development tools, programming languages, and compilers [8,14]. Reversible machines will require special reversible programs and several reversible programming languages have been proposed. Calculations can be performed by reversible machines made of only reversible primitives [4].

Benefits and Challenges

Reversible computing could offer dramatic improvements in energy efficiency. There is no limit to the efficiency of reversible computing. In other words, there is no limit on the amount of reversible computation that can be performed using a fixed amount of energy. For this reason, some researchers are looking to reversible computing to save energy.

Perhaps the main benefit for building technologies that implement reversible computing is that they offer what is predicted to be the only potential way to improve the computational energy efficiency of computers beyond the fundamental von Neumann-Landauer bound [5,9].

Although our understanding of how to design machines based on reversible logic has improved dramatically, some significant research challenges remain. No one has successfully built a properly reversible logic gate. Reversible computing has experienced slow progress. It may require a concerted effort on the part of the semiconductor industry, the computing industry, and government to make significant research progress in reversible computing [9,16].

Conclusion

Reversible computing is emerging as one of the most exciting dimensions in computing for the future. It aims at realizing reversible versions of conventional computations in order to reduce power consumption. Although the idea is still ahead of its time, some practical reversible computing technology might not be far away. Research on developing fast reversible circuits is still ongoing. For more information about reversible computing, one should consult books [8].

References

1. Bennett, C.H., (1973). Logical Reversibility of Computation," *IBM J. Res. Dev.* **6**, 525–532. [Google Scholar]
2. Fredkin, Edward, and Toffoli, T. (2007). "Conservative Logic," (in preparation). Some of the material of this paper is tentatively available in the form of unpublished notes from Prof. Fredkin's lectures, collected and organized by Bill Silver in a *6.895 Term Paper*, "Conservative Logic," and in the form of another *6.895 Term Paper*, "A Reversible Computer Using Conservative Logic," by Edward Barton, both at the MIT Dept. of Electr. Eng. Comp. Sci. [Google Scholar]
3. Kinoshita, Kozo, *et al.*, "On Magnetic Bubble Circuits," *IEEE Trans. Computers* **C-25**(1976), 247–253.[Google Scholar]

4. Landauer, Rolf, "Irreversibility and Heat Generation in the Computing Process," *IBM J.***5**(1961), 183–191.[Google Scholar]
5. Toffoli, Tommaso, "computation and Construction Universality of Reversible Cellular Automata," *J. Comput. Syst. Sci.***15** (1977), 213–231.[Google Scholar]
6. Toffoli, Tommaso, "Cellular Automata Mechanics" (Ph. D. Thesis), *Tech. Rep. no. 208*, Logic of Computers Group, Univ. of Michigan (1977).[Google Scholar]
7. Toffoli, Tommaso, "The Role of the Observer in Uniform Systems," *Applied General Systems Research* (ed. G. J. Klir), 395–400 (Plenum Press, 1978).Google Scholar]
8. Toffoli, Tommaso, (1979). "Bicontinuous Extensions of Invertible Combinatorial Functions," *Tech. Memo MIT/LCS/TM-124*, MIT Lab. for Comp. Sci. (to appear in *Math. Syst. Theory*).[Google Scholar]
9. Toffoli, Tommaso, (1980).["Reversible Computing," *Tech. Memo MIT/LCS/TM-151*, MIT Lab. for Comp. Sci. Google Scholar]
10. Van Rentergem, Y., De Vos, A. (2005). Optimal design of a reversible full adder. *International Journal of Unconventional Computing* 1(4), 339–355. [Google Scholar]
11. Vedral, V., Barenco, A., Ekert, A. (1996). Quantum networks for elementary arithmetic operations. *Physical Review A* 54(1), 147–153.<http://www.ams.org/mathscinet-getitem?mr=1398399> [CrossRef] [Google Scholar]
12. Vieri, C.J. (1999). Reversible Computer Engineering and Architecture. Ph.D. thesis, EECS Department, Massachusetts Institute of Technology. Google Scholar]
13. Wille, R., Drechsler, R. (2010). Towards a Design Flow for Reversible Logic. Springer Science [Google Scholar]
14. Wille, R., Offermann, S., Drechsler, R. (2010). SyReC: A programming language for synthesis of reversible circuits. In: Proceedings of the Forum on Specification & Design Languages, pp. 1–6. IET, Southhampton. [Google Scholar]
15. Yokoyama, T., Axelsen, H. B., & Glück, R. (2008, May). Principles of a reversible programming language. In Proceedings of the 5th conference on Computing frontiers (pp. 43-54). ACM. [Google Scholar]
16. Yokoyama, T., Glück, R. (2007). A reversible programming language and its invertible self-interpreter. In: Proceedings of Partial Evaluation and Program Manipulation, pp. 144–153. [Google Scholar]



This work is licensed under a Creative Commons Attribution 4.0 International License.