An Introduction to Reversible Circuits

Igor L. Markov, The University of Michigan
1301 Beal Ave., Department of EECS, Ann Arbor, MI 48109-2122
imarkov@eecs.umich.edu

Abstract

A reversible computation does not change the information content of its input, and is a permutation of input bit-strings. Reversible logic has been studied since 1980, following a thermodynamic argument that every lost or duplicated bit causes energy loss [2]. When a computational system erases a bit of information, it must dissipate $(\ln 2)kT$ energy, where k is the Boltzmann's constant and T is the temperature.[1]

Toffoli shows [12] that any finite mapping can be computed in three steps: (i) padding with zeros, (ii) permutation of bit-strings, and (iii) projection of some bit-strings onto others. Steps (i) and (iii) can be made particularly simple, and this technique can be viewed as performing an arbitrary computation by pre- and post-processing a reversible computation. To compactly represent permutations of bit-strings by logic circuits, Toffoli uses elementary reversible gates. Every reversible gate has as many inputs as outputs, and an output pin of a gate must drive exactly one input pin of another gate. Common elementary gates pass all of their inputs $x_1,\ldots,x_k$ to outputs, except for one input y, where the return value is $y \oplus (x_1 x_2 \ldots x_k)$. The Controlled-NOT (CNOT) gate corresponds to $k = 1$, and the Toffoli gate (T) to $k = 2$.

In a combinational (acyclic) reversible circuit, every circuit input can be traced to a circuit output, and such circuits are drawn by connecting horizontal "signal lines" with gates represented by vertical lines. Following Feynman, controlling inputs $x_1,\ldots,x_k$ are indicated by $\bullet$ and controlled inputs y are indicated by $\oplus$. Inverters (N gates) are shown with disconnected $\oplus$. Different circuits can implement the same computation, e.g., the four-gate circuit in Figure 1 can be simplified to one gate. Three non-cancelling CNOTS on two lines implement a wire-swap.[2] Such circuit identities can be proven by exhaustive simulation or by more subtle algebraic arguments.

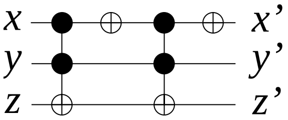

Figure 1: A 3×3 reversible circuit with two Toffoli gates and two inverters. The circuit performs the same computation as one Controlled-NOT gate on lines y and z.

x	y	z	x'	y'	z'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Figure 2: Truth table for the circuit in Figure 1. Observe that the value of input x does not affect the result. Therefore, the line x can be viewed as one bit of temporary storage.

Synthesis formulations for combinational reversible circuits may ask for a circuit that implements a given permutation, fully-specified by a table as in Figure 2 [7] or using the cycle notation [11]. Another popular formulation [6] specifies, for every input, only the most significant bit of the output, leaving other bits as don't-cares. The latter kind of synthesis can be reduced to the two-level XOR-sum decomposition [13]. Some circuits, in addition to their inputs, use several temporary-storage lines (ancilla bits).[3] The circuit in Figure 1 whose truth table is shown in Figure 2 can be used as an illustration. Toffoli points out [12] that a circuit implementing an odd permutation must use at least one line of temporary storage, under certain fairly general conditions. A more recent work by Shende et al [11] shows that (i) any even permutation can be implemented without temporary storage, and (ii) any odd permutation can be implemented with just one line of temporary storage. An optimal synthesis procedure is described in [11], based on dynamic programming. That work also contributes (i) circuit identities that can be viewed as reversible versions of De Morgan's rules, and (ii) a circuit decomposition (TCTN) in which gates of the same type are collected in groups — Toffoli, followed by CNOT, followed by Toffoli, followed by inverters. Three of those groups are relatively small, and for two of them (C and N) the synthesis task can be solved fairly efficiently. The authors of [6] propose a heuristic that simplifies a given reversible circuit by performing a series of local transformations.

Other interesting EDA problems can be formulated for reversible circuits, e.g., fault testing [9] and equivalence checking. However, empirical evaluation may be unconvincing until circuit benchmarks are available. Such benchmarks may have to wait until reversible EDA tools appear because large hand-crafted circuit examples are difficult to procure and may be super-optimized or untypical.

In the 1990s, implementation proposals were put forward at MIT for CMOS circuits whose energy consumption approaches zero as computation time increases to the infinity [14]. Recently, unrelated reversible CMOS adders have been built at Universiteit Ghent, Belgium [4]. Yet irreversibility seems an unlikely concern for commercial VLSI circuits in the next five years, thanks to other, more significant energy-dissipation mechanisms.

[1] Ralph Merke from Xerox estimates that for T = 300 Kelvins (room temperature), this energy is about $2.9e-21$ joules, i.e., roughly the kinetic energy of a single air molecule at room temperature.

[2] This identity is illustrated in Figure 2 of the paper "Scalable Simplification of Reversible Circuits" by Shende et al. in this IWLS handout. Conceptually, the identity is similar to exchanging the contents of two registers without additional memory by performing three XOR operations [10].

[3] In applications one may also have input lines with constant 0 or 1.

An entirely independent *raison d'être* for reversible logic is due to quantum circuits [8], which are already being used in commercial products for secure communication and in experimental quantum computers [5] at IBM, Los Alamos, NIST and many universities. Quantum gates [1] and quantum circuits must be reversible but can handle "superposition" states such as the "half-zero/half-one" state (mathematically, quantum gates and computations are described by complex unitary matrices). Surprisingly, circuit composition rules for the two types of circuits are essentially the same. Because of this, non-quantum reversible logic embeds into quantum logic (the same graphical notation is used) — a quantum version of an existing gate can be applied to a complex linear combination of any or all possible inputs. This simply applies the non-quantum version to every component of the linear combination. For example, the matrix describing a quantum inverter is $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ because it exchanges the basis states zero and one. In contrast, the Hadamard gate $1/\sqrt{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ is not derived from a conventional reversible gate and is therefore purely quantum. Finally, since any conventional computation can be performed reversibly (with an overhead), such a computation can also be embedded into the quantum domain [8].

Consider a quantum gate that maps zero-one states into zero-one states, i.e., cannot create superposition states (good examples are given by permutation matrices, such as the inverter matrix above). In terms of circuit identities and circuit synthesis, such a gate behaves exactly like a non-quantum reversible gate. Therefore, if we can reduce non-quantum reversible circuits, we may be able to reduce some quantum circuits as well, e.g., those that implement classical algorithms as subroutines. However, greater savings may be achievable by manipulating purely quantum gates [3].

References

[1] A. Barenco et al., "Elementary Gates For Quantum Computation," *Physical Review A*, 52, 1995, pp. 3457-3467.

[2] C. Bennett and R. Landauer, "The Fundamental Limits of Computation," *Sci. American*, July 1985, pp. 48-56.

[3] S. S. Bullock and I. L. Markov, "An Arbitrary Two-Qubit Computation in 23 Gates," *DAC 2003*, to appear in *Physical Review A*. `http://arxiv.org/abs/quant-ph/0211002`

[4] B. Desoete and A. De Vos, " A Reversible Carry-look-ahead Adder Using Control Gates," *Integration, the VLSI Journal*, vol. 33 (2002) pp. 89-104.

[5] R. Feynman, "Quantum Mechanical Computers," *Optics News*, **11**, 1985, pp. 11-20.

[6] K. Iwama et al., "Transformation Rules For Designing CNOT-based Quantum Circuits," *DAC 2002*, pp. 419-425.

[7] P. Kerntopf, "A Comparison of Logical Efficiency of Reversible and Conventional Gates," *IWLS 2000*, pp. 261-269.

[8] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press, 2000.

[9] K. N. Patel, J. P. Hayes and I. L. Markov, "Fault Testing for Reversible Circuits," *VTS 2003*, pp. 410-416.

[10] T. Silke, "PROBLEM: Register Swap," December 1995, `http://www.mathematik.uni-bielefeld.de/˜silke/PROBLEMS/bit_swap`

[11] V. V. Shende, A. K. Prasad, I. L. Markov and J. P. Hayes, "Synthesis of Reversible Logic Circuits", IEEE Trans. on CAD, vol 22(6), June 2003, p. 714-726 (preliminary version in Proc. ICCAD 2002, pp. 353-360).

[12] T. Toffoli, "Reversible Computing," *Tech. Memo MIT/LCS/TM-151*, MIT Lab for CS, '80.

[13] B. Travaglione et al. "ROM-based computation: Quantum Versus Classical," Quantum Information and Computation, Vol. 2, No. 4 (2002). `http://arxiv.org/abs/quant-ph/0109016`

[14] S. Younis and T. Knight, "Asymptotically Zero Energy Split-Level Charge Recovery Logic," *Workshop on Low Power Design*, 1994.