Fault-Tolerant Reversible Circuits

Behrooz Parhami

Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA
parhami@ece.ucsb.edu

Abstract

Reversible hardware computation, that is, performing logic signal transformations in a way that allows the original input signals to be recovered from the produced outputs, is helpful in diverse areas such as quantum computing, low-power design, nanotechnology, optical information processing, and bioinformatics. We propose a paradigm for performing such reversible computations in a manner that renders a wide class of circuit faults readily detectable at the circuit's outputs. More specifically, we introduce a class of reversible logic gates (consisting of the well-known Fredkin gate and a newly defined Feynman double-gate) for which the parity of the outputs matches that of the inputs. Such parity-preserving reversible gates, when used with an arbitrary synthesis strategy for reversible logic circuits, allow any fault that affects no more than a single logic signal to be detectable at the circuit's primary outputs. We show the applicability of our design strategy by demonstrating how the well-known, and very useful, Toffoli gate can be synthesized from parity-preserving gates and apply the results to the design of a binary full-adder circuit, which is a versatile and widely used element in digital arithmetic processing.

1. Introduction

Conventional logic is not reversible. For example, knowing the values of both $A \vee B$ and $A \wedge B$ is inadequate for deducing the values of the inputs A and B. Reversible logic, which allows the reproduction of the circuit's inputs from observed outputs [Benn73], [Haye06], finds applications in quantum computing [Rief00], low-power design, nanotechnology, optical information processing, and bioinformatics. Elaborating on the relevance of reversible logic to low-power circuits, we note that the loss of 1 bit of information dissipates at least kT ln 2 joules of energy, where $k = 1.3806505 \times 10^{-23}\ \mathrm{m^2\,kg\,s^{-2}\,K^{-1}}$ (joule/kelvin) is the Boltzmann constant and T is the operating temperature [Land61]. Thus, the only way to avoid power waste, and hence to curtail the production of heat, is to build circuits from reversible elements.

Commonly used reversible elements include the Toffoli gate, TG [Toff80], and the Fredkin gate, FRG [Fred82], depicted in Figs. 1a and 1b. Owing to advantages of simplicity and universality, these two gate types have been studied extensively. There are also design methodologies and tools that incorporate them separately or in combination with each other [Masl05]. A generalized, k-way, Toffoli gate has $k + 1$ inputs: k control inputs, that are copied to the first k outputs, and one other input that is complemented if all control inputs are 1s and is directly copied to the last output otherwise. The Feynman gate, FG [Feyn85], also known as controlled NOT, is a special Toffoli gate with only one control input (Fig. 1c). The Peres gate, combining TG and FG (Fig. 1d), and several other types of reversible gates, have also been used.

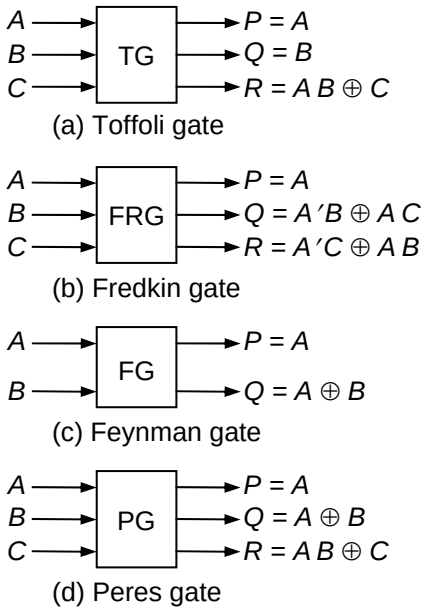


Fig. 1. Some well-known reversible logic gates.

Any reversible gate has the same number of input and output lines, and it implements a permutation from input values to output values. For example, a 3-input, 3-output reversible gate may permute input values 0, 1, 2, 3, 4, 5, 6, 7 to output values 0, 1, 4, 6, 2, 3, 5, 7, corresponding to a permutation definable by its cycle structure (2 4)(3 6 5). Neither feedback nor fan-out is allowed in reversible logic. Consequently, making a reversible circuit robust or fault-tolerant is much more difficult than a conventional logic circuit. For example,

the commonly used method of duplication and result checking via comparison implies the additional overhead of accommodating input fan-out, to feed the two copies of the circuit, and output fan-out, to allow comparison, besides requiring a reversible comparator.

In this paper, we present a fault detection method based on parity-preserving, reversible logic gates. The advantage of our method is that it is implemented at the gate level. Thus, once the required gates have been designed and an appropriate synthesis framework (either adapted from existing methods or custom-built for the new gates) has been established, fault-tolerant implementation requires no extra expenditure in design or verification effort. We note that additional, higher-level fault tolerance mechanisms can be built on top of our gate-level approach in order to increase the level of coverage beyond single-point deviations.

2. Parity-Preserving Reversible Gates

Parity checking is one of the oldest, as well as one of the most widely used, methods for error detection in digital systems. Its most common use is for detecting errors in the storage or transmission of information, primarily because most arithmetic and other processing functions do not preserve the parity of the data. There have been attempts at performing arithmetic operations on specially encoded operands in a way that parity checking becomes applicable [Parh02], [Parh02a], but such methods need further development and are not in widespread use.

With reversible logic, the use of standard methods of error detection presents some problems, given the requirement for fan-out and the associated increase in "garbage bits," that is, extra bits that are produced to maintain the reversibility property, without themselves being a necessary part of the computation itself or of its final results. If computation is performed in such a way that the parity of the input data persists throughout the computation (and, thus, at the output), no intermediate checking would be required. Such results can be forwarded to subsequent modules on the data path, while they are checked in a manner that is off the computation's critical path, and thus not subject to stringent performance or reversibility requirements. Any erroneous result tends to propagate through the downstream modules without a danger of corrupting additional information in the absence of multiple compensating faults.

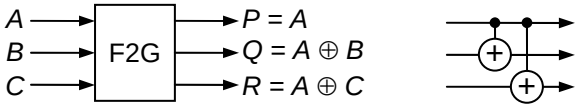
Given that reversible gates tend to have the same number of input and output lines, a sufficient requirement for parity preservation in the course of a reversible computation with such gates is that each gate be parity-preserving; i.e., have the same parity for input and output lines. Our objective is to show that such parity-preserving reversible gates exist and that they can be used to generate viable (competitive in terms of cost and latency) designs.

We first note that of the gates depicted in Fig. 1, only the Fredkin gate is parity-preserving. This is readily verified by comparing the input parity $A \oplus B \oplus C$ (or $A \oplus B$ for the Feynman gate) to the output parity $P \oplus Q \oplus R$ (or $P \oplus Q$). The Feynman gate is quite useful, but, unfortunately, it is inadequate for the synthesis of efficient reversible circuits. Given that synthesis methods with the Toffoli gate, using Fredkin gates to assist in optimizing cost or performance, are quite advanced, we are motivated to look for additional reversible gates that would lead to similarly efficient designs. In this search, the following impossibility result rules out a fundamental role for 2-input, 2-output gates.

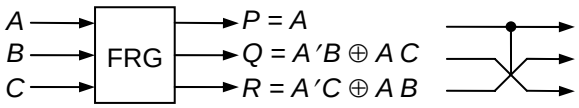
Theorem 1: Other than a gate that complements both inputs (unconditionally) to produce its outputs, no 2-input, 2-output reversible gate can be parity preserving.

Proof: There are 24 reversible 2-input, 2-output gates, corresponding to the 24 possible permutations of the 4 input values 0-3. Of these 24 permutations, only 4 are parity-preserving: even-parity inputs 00 and 11 must be mapped to 00 and 11; odd-parity inputs 01 and 10 to 01 and 10. Two of these are the uninteresting identity and exchange gates that simply forward the inputs to the outputs. The remaining two are defined by permutations having the cycle structures (0, 3) and (0, 3)(1, 2), both of which complement the two inputs, with the first one also exchanging them. [QED]

Guided by Theorem 1, which limits reversible 2-input, 2-output gates to two rather unintereting cases, we next look into 3-input, 3-output gates. We have already observed that the Fredkin gate is parity-preserving. Are there other parity-preserving, reversible 3-input, 3-output gates? The answer to this question is affirmative, as evidenced by the gate shown in Fig. 2a (in both block-diagram form and symbolic form), which is a Feynman gate with an additional input and one more output. We call the gate depicted in Fig. 2a the "Fynman double-gate," given that the extra input and output, along with the control input A define a second controlled NOT operation. Note that, like the Feynman gate and the Fredkin gate, the Fynman double-gate is its own inverse.



(a) Feynman double-gate



(b) Fredkin gate

Fig. 2. Parity-preserving reversible gates, satisfying $A \oplus B \oplus C = P \oplus Q \oplus R$: Block diagrams and symbolic forms.

A natural question at this point is whether there exist other 3-input, 3-output parity-preserving reversible gates beyond the two depicted in Fig. 2. We must perform an exhaustive search to answer this question, but meanwhile have the following preliminary result that indicates such gates, if they exist, would not be of the kind that lend themselves to simple and efficient design procedures.

Theorem 2: Up to the permutation of inputs and outputs and (conditional) complementation of both outputs Q and R, no 3-input, 3-output reversible logic gate, other than the two depicted in Fig. 2, can be parity preserving if its first output equals its first input ($P = A$).

Proof: As shown in Table I, when $P = A$, the rest of the truth table for a reversible 3-input, 3-output gate can be filled out in only 16 possible ways, if $\{A, B, C\}$ and $\{P, Q, R\}$ are to have the same parity. The output functions Q and R are shown in Table II for these 16 cases. It is easily seen that the resulting gate is not useful in four of these cases, is idential to F2G or a variant (F2G-var) in which both Q and R are complemented in four additional cases, and is identical to FRG or FRG-var in the remaining eight cases. [QED]

Table I. Potential truth tables for a 3-input, 3-output, parity-preserving gate with $P = A$.

A	B	C	$P = A$	Q	R
0	0	0	0	a	a
0	0	1	0	b	b'
0	1	0	0	b'	b
0	1	1	0	a'	a'
1	0	0	1	c	c
1	0	1	1	d	d'
1	1	0	1	d'	d
1	1	1	1	c'	c'

Table II. The 16 possible ways of assigning values to the entries a, b, c, and d in Table I.

a	b	c	d	$Q(A, B, C)$	$R(A, B, C)$	Comment
0	0	0	0	B	C	Not useful
0	0	0	1	$A'B \vee AC$	$A'C \vee AB$	FRG
0	0	1	0	$A'B \vee AC'$	$A'C \vee AB'$	FRG-var
0	0	1	1	$A \oplus B$	$A \oplus C$	F2G
0	1	0	0	$A'C \vee AB$	$A'B \vee AC$	FRG′
0	1	0	1	C	B	Not useful
0	1	1	0	$A \oplus C$	$A \oplus B$	F2G
0	1	1	1	$A'C \vee AB'$	$A'B \vee AC'$	FRG-var
1	0	0	0	$A'C' \vee AB$	$A'B' \vee AC$	FRG-var
1	0	0	1	$(A \oplus C)'$	$(A \oplus B)'$	F2G-var
1	0	1	0	C'	B'	Not useful
1	0	1	1	$A'C' \vee AB'$	$A'B' \vee AC'$	FRG-var
1	1	0	0	$(A \oplus B)'$	$(A \oplus C)'$	F2G-var
1	1	0	1	$A'B' \vee AC$	$A'C' \vee AB$	FRG-var
1	1	1	0	$A'B' \vee AC'$	$A'C' \vee AB'$	FRG-var
1	1	1	1	B'	C'	Not useful

Note that in all the reversible logic gates that have been proposed in the literature to date, the condition $P = A$ holds. Any reversible logic gate that does not satisfy $P = A$ corresponds to another one that does. In order to see this, we let $P = f(A, B, C)$, $Q = g(A, B, C)$, and $R = h(A, B, C)$. Setting $T = A \oplus f(A, B, C)$, a modified logic gate with the three outputs $P = T \oplus f(A, B, C) = A$, $Q = T \oplus g(A, B, C)$, and $R = T \oplus h(A, B, C)$ essentially performs the same computation as the original gate, in the sense that one gate can replace the other with minimal redesign effort. This observation leads to the following result.

Theorem 3: Up to the permutation of inputs and outputs, (conditional) complementation of both outputs Q and R, and XORing of every output with the same function T of the inputs A, B, and C, no 3-input, 3-output reversible logic gate, other than the two depicted in Fig. 2, is parity preserving.

3. Parity-Preserving Design Process

Results on synthesis of parity-preserving reversible logic circuits for various functions will be reported in the future. In this preliminary report, we demonstrate that a versatile arithmetic building elements, the full-adder circuit, can be built to be parity-preserving. This is significant in itself, given that many arithmetic functions can be synthesized from full-adders and other simple circuits.

For our first design, we modify a known full-adder circuit based on Fredkin gates (also using one Feynman gate), depicted in Fig. 3. This design is not parity-preserving, given that after the two gates in the first stage on the left, the inputs A, B, 0, C, 1, 0 are transformed to A, B, B, C, C', C, with the latter set's parity then preserved through the remaining two circuit stages. The input and output parities are thus related by $p_{out} = p_{in} \oplus B$. A simple way to restore the output parity to that of the input, without affecting the circuit's functionality, is to pass the ouputs B and G through another Feynman gate, thereby transforming them to B and $G \oplus B$.

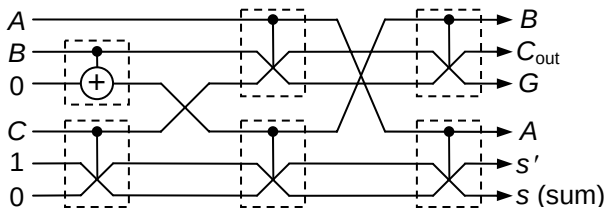


Fig. 3. Binary full-adder built of Fredkin gates. The single Feynman gate fans out B.

We next show how the functionality of a Toffoli gate can be synthesized by a parity-preserving reversible circuit. This will allow us to use existing designs for full-adders and other arithmetic circuits [Vedr96], [Masl05] via simple substitution,

with some postoptimization used to reduce the number of garbage bits. Figure 4 depicts a relatively simple full-adder circuit containing Peres gates (each composed of a Toffoli and a Feynman gate) as an example. Given that synthesis methods using Toffoli gates are widely available, this serves as evidence that our proposed parity-preserving design strategy would be viable.

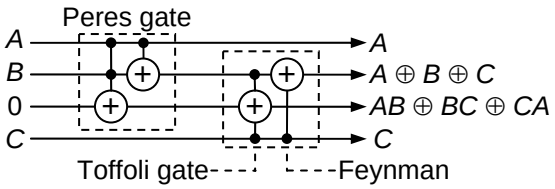


Fig. 4. Binary full-adder built of 2 Peres gates.

Figure 5 shows how one FRG and two F2Gs can be used to synthesize a Toffoli gate. Extra inputs and outputs (garbage outputs) have been indicated with dashed lines. Note that if we do not insist on reproducing all three Toffoli gate outputs exactly (e.g., by foregoing the $Q = B$ output), a less complex design becomes possible (Fig. 6), which can replace a Toffoli gate in most contexts. At present, we are investigating such alternative designs and their effect on synthesis procedures.

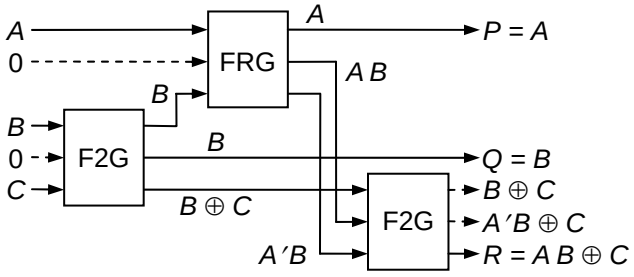


Fig. 5. Synthesizing a TG from one FRG and two F2Gs, with parity preservation.

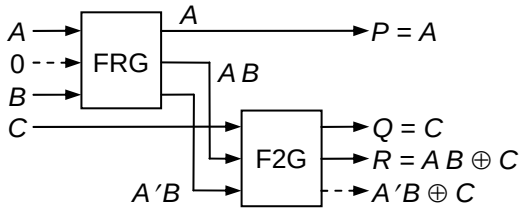


Fig. 6. Synthesizing a TG-like element from one FRG and one F2Gs.

4. Conclusion

We have demonstrated the feasibility of the parity-preserving approach to design of reversible logic circuits. This approach provides a way of incorporating fault tolerance into reversible circuits without much extra design effort and with modest hardware overhead, a goal that has proven difficult to achieve thus far. The small number of reports in the literature have been based on fairly ad hoc methods or have involved excessive overhead in circuit complexity (see, e.g., [Shor96], [Pres98], [Boyk05]). It is hoped that parity preservation, by itself and in combination with other methods, proves useful for ensuring the robustness of reversible logic circuits in their various application domains.

References

[Benn73] C. H. Bennett, "Logical Reversibility of Computation," *IBM J. Research and Development*, Vol. 17, pp. 525-532, November 1973.

[Boyk05] P. O. Boykin and V. P. Roychowdhury, "Reversible Fault-Tolerant Logic," *Proc. Int'l Conf. Dependable Systems and Networks*, pp. 444-453, 2005.

[Feyn85] R. Feynman, "Quantum Mechanical Computers," *Optics News*, Vol. 11, pp. 11–20, 1985.

[Fred82] E. Fredkin and T. Toffoli, "Conservative logic," *Int'l J. Theoretical Physics*, Vol. 21, pp. 219–253, 1982.

[Haye06] B. Hayes, "Reverse Engineering," *American Scientist*, Vol. 94, No. 2, pp. 107-111, March-April 2006.

[Land61] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM J. Research and Development*, Vol. 5, No. 3, pp. 183-191, July 1961.

[Masl05] D. Maslov, G. W. Dueck, and D. M. Miller, "Synthesis of Fredkin-Toffoli Reversible Networks," *IEEE Trans. VLSI Systems*, Vol. 13, No. 6, pp. 765-769, June 2005.

[Parh02] B. Parhami, "Parity-Preserving Transformations in Computer Arithmetic," *Proc. SPIE Conf. Advanced Signal Processing Algorithms, Architectures, and Implementations XII*, July 2002, pp. 403-411.

[Parh02a] B. Parhami, "An Approach to the Design of Parity-Checked Arithmetic Circuits," *Proc. 36th Asilomar Conf. Signals, Systems, and Computers*, November 2002, pp. 1084-1088.

[Pres98] J. Preskill, "Reliable Quantum Computers," *Proc. Royal Society*, Vol. 454, pp. 385-410, 1998.

[Rief00] E. Rieffel and W. Polak, "An Introduction to Quantum Computing for Non-Physicists," *ACM Computing Surveys*, Vol. 32, No. 3, pp. 300-335, September 2000.

[Shor96] P. Shor, "Fault-Tolerant Quantum Computation," *Proc. 37th Symp. Foundations of Computer Science*, 1996.

[Toff80] T. Toffoli, "Reversible Computing," Tech memo MIT/LCS/TM-151, MIT Lab for Comp. Sci, 1980.

[Vedr96] V. Vedral, A. Barenco, and A. Ekert, "Quantum Networks for Elementary Arithmetic Operations," *Physical Review A*, Vol. 54, pp. 147-153, July 1996.