Synthesis and Fundamental Energy Analysis of Fault-Tolerant CMOS Circuits

İlke Ercan
Electric-Electronics Engineering
Department,
Boğaziçi University
İstanbul, Turkey
ilke.ercan@boun.edu.tr

Ömercan Susam, Mustafa Altun
Electronics and Communication
Engineering Department
Istanbul Technical University
Istanbul, Turkey
{susam, altunmus}@itu.edu.tr

M. Hüsrev Cılasun
Radar, Electronic Warfare and
Intelligence Systems Division
Aselsan A.Ş.,
Ankara, Turkey
mcilasun@aselsan.com.tr

Abstract— In this study, we perform a physical-information-theoretic analysis to obtain fundamental energy dissipation bounds for fault-tolerant reversible CMOS circuits we synthesize using Hamming codes. We show that the approach we had initially developed to calculate theoretical efficiency limitations of emerging electronic paradigms can also be applied to CMOS technology base and can provide feedback to improve circuit design and performance. We illustrate our physical-information-theoretic methodology via applications to circuits that we synthesized using Hamming codes that result in detection of up to (d-1) bit errors and correction of up to (d-1)/2 bit errors where d represents the minimum Hamming distance between any pair of bit patterns. The fundamental lower bounds on energy dissipation are calculated for a one-bit *reversible* full adder and for *irreversible* full adders with block-code-, dual modular redundancy (DMR)- and triple modular redundancy (TMR)-based CMOS circuits. Our results reflect the fundamental difference in energy limitations across these circuits and provide insights into improved design strategies.

Keywords—Fundamental energy bounds; reversible circuits; fault tolerance; error detection and correction

I. INTRODUCTION

There are numerous sources of inefficiencies in computing circuits that interfere with improved performance of conventional circuits and realization of novel technology proposals. The practical sources of inefficiencies are inherently unpredictable; however, we can predict fundamental limits as they are based on the generic physical requirements for implementation of a computational strategy. In our previous work, we proposed a methodology to obtain fundamental efficiency limits for complex computing structures based on fundamental physical description of dynamics of information as it is processed by computing machines and calculate fundamental energy dissipation bounds based on this description for a given technology base [1, 2]. The bounds we obtain provide insights into fundamental performance projections of post-CMOS paradigms and challenges ahead in emerging electronic technologies [3]. In this work, we apply our methodology to conventional CMOS technology via circuits we developed using a novel fault-tolerant synthesis approach based on Hamming codes.

Motivated by the shortcomings of the existing approaches, we developed a novel technique to synthesize reversible fault-tolerant circuits, which exploits Hamming codes [4] and simultaneously performs fault detection and correction. The idea of using Hamming codes in fault tolerance of reversible circuits is previously introduced in [5, 6]. However, these studies focus on a constrained set of circuits for encoding and decoding purposes rather than presenting a generic method for converting any reversible circuit to a fault-tolerant one. In this paper, we propose a mapping to a block code domain where a fixed minimum Hamming distance between any pair of regular bit patterns is satisfied.

In implementing our physical-information-theoretic methodology to the fault-tolerant circuits we synthesize, we develop a common ground in comparing fundamental efficiency limitations of reversible and irreversible CMOS circuits. Another significant aspect of employing reversible circuits is to go beyond the convention of implementing reversible computation in post-CMOS paradigms and contribute to the limited reversible CMOS circuit applications. In principle, reversible circuits allow for dissipation-free computation by retaining the input data in the circuit throughout the computation promising 'zero power consumption' after Landauer's principle [7], dissipative costs do, however, arise in more practical scenarios, including large circuits with independent individual stages to enable pipelining, where input data is erased from each stage at the end of computation. The physical-information-theoretic methodology we developed captures these contributions and allows us to obtain a paradigm-dependent fundamental limit.

This paper is organized as follows: In the next section, we introduce the technique developed to synthesize fault-tolerant CMOS circuits by using block code based Hamming coding. In the following section, we provide an overview of the foundations of our physical-information-theoretic methodology and present its application to the circuits we synthesize. In the last section, we conclude by our final remarks and future directions of our work.

II. FAULT-TOLERANT REVERSIBLE CIRCUIT SYNTHESIS

A conventional Boolean function always carries a one bit information (0 or 1) that is independent of the number of input

bits, whereas a reversible Boolean function carries information with using the same number of input and output bits. For reversible functions, each input bit combination results in a unique output bit combination; the reverse of this is also true because of the reversibility. A reversible function can be realized by a reversible circuit consisting of reversible gates. In this study, we mainly use the NCT (**N**OT, **C**NOT, **T**offoli) gate library. We also use mEXOR and tCNOT gates, where a mEXOR gate is equivalent of a Toffoli gate network such that each gate has the same control bit with different targets, and a tCNOT gate is equivalent of a CNOT gate network such that each gate has the same control bit with different targets. Further definition of the gates (**T** and **C** stand for target and control, respectively) is given in [8].

A. Synthesis

In information theory, Hamming codes are widely used for accurate transmission of the data. In mathematical terms, Hamming codes are described with an injective notation as $C: \Sigma^k \rightarrow \Sigma^n$. Here, Σ is the alphabet. Size of $|\Sigma|$ is the number of possible letters in the alphabet. If the code is binary, then $|\Sigma| = 2$. Superscript integers k (message length) and n (block length) are the dimensions of input and output alphabets, respectively. The encoding rate R is defined as the ratio k/n. Error detecting/correcting characteristics of block codes are defined by the distance parameter d, which is equal to the minimum Hamming distance between the elements of Σ^n. Once there is a b bits of corruption in a single codeword, the initial codeword can be recovered if $b < d/2$. Additionally, it is still possible to detect an error if $b < d$.

To tolerate an error with Hamming codes, one can use a decoder at the output to be modeled as a maximum likelihood receiver. The receiver tries to minimize the error, that can be defined as Hamming distance d of the received codeword $C_r C_r$ over the alphabet Σ^k. The problem can be expressed as follows: arg min$_i$ $d(C_r, \Sigma_i^k)$. Given a reversible circuit with k bits, we need to synthesize a fault-tolerant n-bit version of the circuit with selecting the value of d that guarantees detection of up to $(d$-1$)$ bit errors and correction of up to $(d$-1$)/2$ bit errors at the output. Here, the encoding rate $R = k/n$ is an important efficiency parameter; the larger R is, the less extra bit lines (comparatively) we use for tolerance. The calculations we performed using simple brute-force algorithm shows that R is positively correlated with k. The reader is referred to [9] for details of block code generation techniques. We present an expression for $d = 3$ below which is achieved by making a slight modification of the Hamming's limit given in [10].

$$R = \lfloor n\text{-log}_2(n+1)\rfloor\ /\ n. \tag{1}$$

In this study, d=3 is chosen as it provides 1-bit correction and 2-bit detection and includes both DMR (1-bit detect) and TMR (1-bit correct).

B. CMOS Implementation

Our algorithm results in a reversible circuit consisting of CNOT and Toffoli gates. To show feasibility and practicability of our approach, we show how to convert our reversible circuits into CMOS gate based realizations. Consider a conventional NAND gate. Since three input combinations are mapped to a
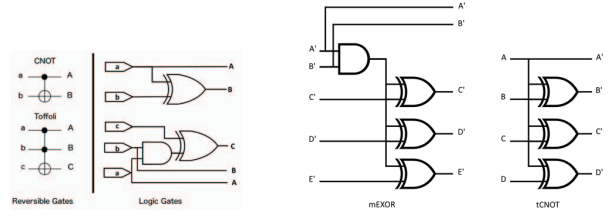

Fig. 1. CNOT and Toffoli gates with proposed scheme for their CMOS reversible soft fault-aware logic implementations (left), and realization of mEXOR (1 AND + 3 XOR) and tCNOT (3 XOR) gates (right).
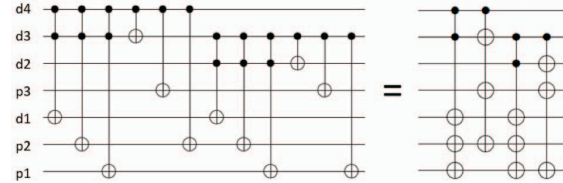

Fig. 2. The proposed synthesis results of the reversible fault-tolerant full adder after block code mapping with Hamming distance of 3.
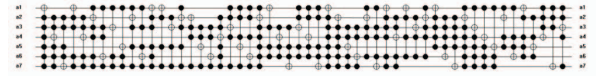

Fig. 3. Approximate transformation based synthesis result of the full adder after block code mapping with Hamming distance of 3.

single logical '1' value, the information regarding a possible fault at one of the inputs can be lost. Same problem occurs in NOR, OR, and AND gates. This is indeed related to don't care conditions. On the other hand, NOT, XOR, and XNOR gates satisfy the awareness of an input fault. However, they do not form a universal set. We use AND and XOR gates as shown in Fig. 1 to realize CNOT and Toffoli gates. For CNOT realization, we directly use an XOR gate. For Toffoli realization, along with an XOR gate we use an AND gate. Here, a probable information lost for the AND gate is eliminated by using its inputs as also outputs. Using NOT gates in the realization of an AND gate, e.g., one NAND plus one inverter, would also satisfy fault-awareness since NOT gates do not have a don't-care condition. As a result, no internal bit flipping disappears; any possible bit flipping occurring in inputs and internal nodes is directly reflected to the outputs.

C. A Fault-Tolerant Adder

A reversible full adder is mapped to the block code domain with Hamming distance $d = 3$. The optimum implementation of a full adder in terms of the number of gates using the NCT library in [11] is used as an input of the algorithm. Our NCT library based synthesis result is given on the left side of Fig. 3. Parity bits and their related data sets are respectively given as $\{p1, p2, p3\} = \{\{d1, d2, d4\}, \{d1, d3, d4\}, \{d2, d3, d4\}\}$. Note that the first gate on the left side in Fig. 4 results in two new gates (second and third gates in Fig. 2) since its target bit "d1" is included in the data set of p1 and p2. On the right side of the Fig. 2, the circuit is compressed by converting toffoli gates into mEXOR and tCNOT gates for simplification. Fig. 3 shows a reversible full adder that is obtained using an approximate transformation based synthesis technique [12] in block code

domain with Hamming distance $d = 3$. Comparing the circuits in Fig. 2 and 3, we see that our synthesis technique overwhelms the transformation based one in terms of the circuit size. CMOS implementation of the circuit on the left side of Fig. 2 can be directly achieved by using the implementations that is previously given in Fig. 1. In a similar fashion, CMOS implementation of the circuit on the right side of Fig. 2 is obtained by using the implementations in Fig. 1 (right).

III. FUNDAMENTAL ENERGY ANALYSIS

A. Methodology

The first step in our physical-information-theoretic methodology is the construction of a functional *abstraction* of information flow in the computing structure, which captures the essential features of the circuit structure and control. This involves creating both a *physical abstraction* of the computing structure and the *process abstractions* of the information dynamics. The physical abstraction is performed by modeling the computing structure and its environment as a globally closed and isolated universe which enables us to assume unitarily evolution via Schrödinger's equation. The physical abstraction of a transistor-based circuit, treated as a composite quantum system can be found in detail in our previous work [1]. In this study, the information-processing artifact A consists of our reversible full-adder circuit, as well as the source, S, and drain, D. The S and D are nominally regarded as idealized Fermi gases at temperature T with associated chemical potentials μ_{SS} and μ_{DD}, respectively, with $\Delta\mu = \mu_{SS} - \mu_{DD} = qV_{DD}$, where q is the electric charge. The bath B is the phonon gas in the underlying substrate, which is in direct thermal contact with the CMOS circuit, and is nominally in a thermal state at temperature T. The greater environment includes subsystems that drive B toward thermal equilibrium and supply the energy and particles required to maintain the nominal populations of S and D and a chemical potential difference qV_{DD} when these subsystems are driven from their nominal states during computation. The adder circuit exchanges particles with S and D, and heat with the bath B as it processes input data held in register (referent) R. The greater environment provides the energy, particles, and heat removal, which enable circuit operation, and everything else required to thermodynamically isolate the global system. The process abstraction of the full adder circuits we employed in this study is straight forward as the output is obtained in a single step computational cycle with all the logic elements getting activated simultaneously and no intermediate information erasure taking place. However, even in reversible circuits, information is irreversibly lost every time a new input is loaded in the circuit. In this reversible full-adder, each input is introduced without a prior reset, therefore information is erased partially by overwriting.

Once the abstraction step is completed we perform fundamental energy analysis on the idealized model of the physical circuit structure and information processing. Our analysis on the abstraction of the transistor-based circuits operated under single-shot computation scheme yields lower bounds on the total amount of energy dissipated into the bath for a single input as

$$\Delta E_{diss} \geq kT \ln(2)\Delta I + f q V_{DD}\Delta N, \qquad (2)$$

where, k is the Boltzmann constant, T is temperature, ΔI is the amount of information irreversibly lost, ΔN is the total number of electrons that flow in from source to drain in order to implement computation and f represents a fraction of energy invested in source and drain [1]. Here, the first term represents the amount of dissipation caused by irreversible information loss and the second term represents the additional unavoidable energy cost associated with particle supply required to maintain the computational "working substance" in transistor-based paradigms. This bound is calculated under the paradigmatic operation where the analysis solely involves dissipative costs that are required to perform computation. Note that this bound is higher than the Landauer's principle as our methodology captures the essential functional features of computational paradigm.

B. Application

We perform physical-information-theoretic analysis on the reversible full adder circuit, as well as its irreversible counterparts, and compare their fundamental energy limitations. One of the key points in the application of the methodology to the circuits we synthesize is the codebook used in the operation of our circuits. Here, the inputs are selected among a codebook composed of sixteen equiprobable codewords; the amount of information erased by overwriting at the end of the computation, averaged over all possible codewords, is half the number of input digits M, i.e. $\Delta I = 3.5$ bits for seven-digit inputs in the reversible full adder. The reversible full-adder circuit of interest is composed of two mEXOR and two tCNOT logic elements containing two AND and twelve XOR gates in total. In addition to the AND and XOR gates, we also used fourteen NOT gates in the reversible full-adder in order to obtain the inverse of certain signals. CMOS implementation of this logic circuit is, therefore, composed of 68 nMOS and 68 pMOS transistor gates. Therefore, the *average* amount of electrons transferred from source to drain to process a single input, in a circuit containing 2N transistor gates is (N+M)/2. Even if we make an extreme idealization and assume that each transistor gate can be switched using a single electron, this would mean that the reversible circuit needs 37.5 electrons, on average, to process a single input. Based on this calculation, the two term in the lower bound on the energy dissipated locally into the bath as a result of overwriting of an input, among the possible codewords, during computation in this reversible full-adder circuit implemented in CMOS are $3.5kT \ln 2$ and $f37.5qV_{DD}$ as given in Table 1. The contributions to the fundamental energy bounds show that dissipation-free computation is not possible for a reversible circuit, even under best-case assumption; in particular, unavoidable dissipative cost of irreversible information loss arises, as represented by the first term, due to operation and underlying strategy of the computing structure.

We also performed fundamental efficiency analysis on the irreversible counterparts of the full adder circuit. We consider

three full adder circuits: (a) Codeword-based, composed of 26 NAND gates containing 52 pMOS and 52 nMOS transistor gates, (b) DMR, composed of 18 NAND gates containing 36 nMOS and 36 pMOS transistor gates, and (c) TMR, composed of 27 NAND gates composed of 52 nMOS and 52 pMOS transistor gates. Our analyses have shown that the first term in the above inequality, representing the amount of dissipation caused by irreversible information loss, reduces to $1.5kT \ln 2$, on average, for the irreversible circuits due to the reduction in the input bits. The irreversible circuits contain fewer number of transistor gates, therefore, the second term in the lower bound on the energy dissipation, averaged over all inputs, changes as a listed in Table I. The fundamental efficiency analyses show that, lower bounds on the energy dissipation of the reversible and irreversible full adders are comparable and that there is only slight increase in the lower bound for the reversible circuit, which is a tolerable compromise given the advantages provided by the strategy.

TABLE I.   FUNDAMENTAL ANALYSIS OF THE FAULT-TOLERANT CMOS
FULL ADDER CIRCUITS

Contributions to the Fundamental Energy Dissipation Bound		
Full-Adder Circuit Type	*Information Erasure Cost*	*Unavoidable Charge Re-Supply Cost*
Reversible	$3.5kT \ln 2$	$f\,37.5qV_{DD}$
Irreversible Codeword-based	$1.5kT \ln 2$	$f\,27.5qV_{DD}$
Irreversible DMR	$1.5kT \ln 2$	$f\,19.5qV_{DD}$
Irreversible TMR	$1.5kT \ln 2$	$f\,27.5qV_{DD}$

Note-that the contributions to the fundamental energy bound excludes any cost that is deemed non-functional from a physical-information-theoretic perspective; e.g. parasitic capacitance is not considered, and each transistor gate assumed to be switched on or off with a single electron in an idealized scenario in order to obtain the lowest bound possible. Comparing the two terms constituting the bound, we see that the dominance of the particle-supply cost in the bound can be reduced but cannot be eliminated; the condition $V_{DD} \gg kT$ needs to be met in order to ensure reliable computation provided by maintaining the directional electron injection and extraction processes that enable proper circuit operation. Therefore, our calculations show that, in transistor-based-circuit, implemented in either post-CMOS or CMOS paradigm, the fundamental bound on maintaining the computational working substance far exceeds the cost of unavoidable cost of the irreversible logic operations even under the assumptions.

IV. CONCLUSIONS

The physical information-theoretic analyses performed on the CMOS implementation of the reversible and irreversible fault-tolerant circuits provide insights into fundamental efficiency limitations of these circuits. The increase in the fundamental lower bound for the reversible circuit, as compared to the irreversible counterparts, is found to be a tolerable compromise given the advantages provided by the reversible strategy. It is important to note that, we do not make any claims

regarding the achievability of these limits, given the practical limitations imposed on the CMOS technology. The manufacturing techniques proposed for post-CMOS paradigms allow aggressive scaling that can mean higher performance, density, and power efficiency that can go far beyond the performance of CMOS technology. For emerging technologies, the fundamental lower bounds provide insights into how far the efficiency of a given computing strategy can be improved in principle. These analyses provide a reference point regarding the physical possibility that specified performance targets can be met under best case assumptions regarding circuit fabrication and computational control; the dominating factors in this minimum cost may vary for different paradigms and must be studied separately.

The analysis of the CMOS implementation of full adder circuits we synthesize here do not only serve as a valuable application of our methodology to a well-known and familiar circuit structure but also presents us perspective on the limitations of the methodology and further steps we need to take towards using this approach to provide practical feedback to improve design strategies of emerging technologies. The next steps in our research involve expanding the theoretical framework of the methodology to include practical sources of inefficiencies and enable fundamental analyses at the processor level. As a potential application of such generalized approach we also aim to synthesize a reversible arithmetic logic unit, control and memory units of the processor, realize the processor in CMOS technology in gate and transistor levels, develop simulation, fabrication, and test of the processor, and test our expanded theoretical calculations against the practical results.

REFERENCES

[1] İ. Ercan and N. Anderson, "Heat Dissipation in Nanocomputing: Lower Bounds from Physical Information Theory," IEEE Transactions on Nanotechnology, vol. 12, no. 6, pp. 1047 1060, 2013.

[2] N. G. Anderson, İ. Ercan, and N. Ganesh, "Revealing Fundamental Efficiency Limits for Complex Computing Structures," Poster presented at the 4th IEEE Rebooting Computing Summit, December 2015.

[3] R. S. Williams and E. P. DeBenedictis, "OSTP Nanotechnology Inspired Grand Challenge," Proc. of IEEE Rebooting Computing, 2015.

[4] R.W. Hamming, "Error detecting and error correcting codes." Bell System technical journal 29.2, 1950, pp. 147-160.

[5] R.K. James, K.P. Jacob, and S. Sasi. "Fault tolerant error coding and detection using reversible gates," TENCON 2007-2007 IEEE Region 10 Conference. IEEE, 2007.

[6] M. Haghparast and K. Navi. "Novel reversible fault tolerant error coding and detection circuits," International Journal of Quantum Information 9.02, 2011, pp.723-738.

[7] C.H. Bennett, and R. Landauer. "The fundamental physical limits of computation," Scientific American, vol. 253.1, 1985, pp.48-56.

[8] O. Susam and M. Altun, "Fast Synthesis of Reversible Circuits using a Sorting Algorithm and Optimization," Journal of Multiple-Valued Logic and Soft Computing , vol. 28, issue. 5-6, pp. 1-23 , 2017.

[9] T.K. Moon, Error Correction Coding, New Jersey: John Wiley & Sons. 2005, ISBN 978-0-471-64800-0.

[10] R.W. Hamming, "Error detecting and error correcting codes," Bell System technical journal 29.2, 1950, pp. 147-160.

[11] R. Wille and D. Große, "Fast Exact Toffoli Network Synthesis of Reversible Logic," Int'l Conf. on CAD, 2007.

[12] R. Wille, O. Keszocze, S. Hillmich, M. Walter, and A. Garcia-Ortiz, "Synthesis of approximate coders for on-chip interconnects using reversible logic," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2016, pp. 1140-1143.