Quantifying Irreversible Information Loss in Digital Circuits

ISMO K. HÄNNINEN, CRAIG S. LENT, and GREGORY L. SNIDER,
University of Notre Dame

Heat generation limits the performance of state-of-the-art integrated circuits, originating from the wasteful static CMOS operating principle. Near-term solutions like adiabatic charging for energy recovery and limiting friction-type heat sources provide considerable improvement. However, these methods do not address the ultimate thermodynamic necessity to expel energy related to information loss in the computing process. In emerging beyond-CMOS technologies, this bit erasure heat alone can overwhelm the cooling capacity and set the limits of the computing performance. Therefore, logical information loss is becoming an important factor for digital circuit design, and tools have to be developed for analysis and optimization. This article presents a framework for estimating the amount of information loss in complex logic circuits, demonstrating the method by modeling the irreversible bit erasures in a standard binary adder structure. Binary addition is one of the most often used and highly optimized digital designs, and we estimate the erasure bounds for components on various levels of design abstraction, showing that the actual logic gate implementations have orders of magnitude higher loss than the addition operation itself would require. The method and the results can be used to optimize circuits for a higher degree of logical reversibility and energy conservation.

Categories and Subject Descriptors: B.6.3 [**Logic Design**]: Design Aids—*Optimization, Simulation, Switching Theory*; B.2.2 [**Arithmetic and Logic Structures**]: Performance Analysis and Design Aids; B.7.1 [**Integrated Circuits**]: Types and Design Styles—*Advanced technologies, Algorithms implemented in hardware*; H.1.1 [**Models and Principles**]: Systems and Information Theory—*Information theory*; F.1.m [**Computation by Abstract Devices**]: Miscellaneous

General Terms: Theory, Design, Performance

Additional Key Words and Phrases: Irreversible logic, information loss, reversible computation, heat generation, energy, power estimation, adiabatic CMOS, beyond-CMOS, quantum-dot cellular automata, QCA, emerging devices

ACM Reference Format:
Ismo K. Hänninen, Craig S. Lent, and Gregory L. Snider, 2014. Quantifying irreversible information loss in digital circuits. ACM J. Emerg. Technol. Comput. Syst. 11, 2, Article 10 (October 2014), 17 pages.
DOI: http://dx.doi.org/10.1145/2629523

1. INTRODUCTION

Heat generation limits the computing performance and the range of available applications of integrated circuit technology, and indeed, the most important design concern of modern ICs is the related electrical power [ITRS 2012]. The problem of heat dissipation dominates the continued evolution of CMOS processors in this era of "dark silicon," in which to avoid overheating, whole sections of a chip may need to be powered down—a dramatic inability to use the carefully fabricated resources. The steady rise in microprocessor clock speeds has nearly halted because of power issues; scaled CMOS transistors could individually be switching at much faster rates, but because they are

This work was supported by the Academy of Finland under research grant 132869 and by the Finnish Foundation for Technology Promotion.
Contact author's (I. K. Hänninen) email: ismo.hanninen@nd.edu.

DOI: http://dx.doi.org/10.1145/2629523

so densely packed on a modern chip, the additional heat created by running faster would be overwhelming.

As presently used, CMOS transistors switch in a way which is maximally wasteful—transferring essentially all the energy of a stored bit into heat at every rapid-state transition. The loss can be greatly reduced by utilizing adiabatic CMOS, which can recover most of the signal energy [Starosel'skii 2001]. There is also considerable search underway for the "next switch," a different binary device that could perform general-purpose computing but would generate far less heat. Potential emerging devices like quantum-dot cellular automata (QCA) have inherent capability to propagate and reuse the signal energy instead of expelling it as heat [Lent and Tougaw 1997].

Every conceivable switching device loses some energy due to *friction-like* physical phenomena, but these losses can be made as small as desired, by utilizing predicted technology improvements and purposefully lowering the relative switching speed. Another loss component, which has always been present, but has been insignificant in traditional static CMOS, is the energy loss related to *information erasure* [Keyes and Landauer 1970]. According to Landauer's Principle, the erasure of a logical bit always mirrors a physical change in entropy, which causes at least a *minimum constant quantum of energy* to be expelled as heat. In (quasi-)adiabatic CMOS, the bit erasures alone can be a major source of heat, while in the extremely energy-efficient beyond-CMOS technologies, the bit erasures can generate enough heat to overwhelm the cooling capacity, and consequently, set the performance limit.

Energy-efficient computer arithmetic is critical for low-power systems, including the underlying layers of all computing systems in the future, while the most important and common arithmetic operation is the binary addition. Standard adders [Koren 2002] are based on irreversible logic, which leads necessarily to energy loss originating from the bit erasures. In this article, we construct abstract models for quantifying the amount of information loss in a standard ripple carry adder structure. The number of average bit erasures in actual implementations is significantly higher than the minimum of the operation itself, which indicates potential for optimization for reversibility. The main contributions of this work can be summarized as follows.

—To the best of our knowledge, this work is the first that tracks the information loss properties of complex digital circuits in detail.
—A method is developed to systematically abstract the information loss in computing operations, refining the model step by step to the level of actual logic circuits, revealing the difference between the lower bounds and the actual expected losses.
—The most important arithmetic operation, binary addition, is characterized regarding the information loss properties, and the losses in a standard adder structure and the circuit implementations are compared to this ultimate bound.

This article is organized as follows. Section 2 discusses the connection between logical and physical reversibility, how to quantify the information loss, and the assumptions of our modeling approach, while Section 3 discusses the theoretical monolithic addition operation. The ripple carry adder structure and an ideal standard gate abstraction are analyzed in Section 4, and selected practical gate-level implementations are analyzed in Section 5. Section 6 presents the statistical results on the information loss properties, while the conclusion follows in Section 7.

2. PRELIMINARIES

Traditional circuit design does not explicitly consider heat generation related to information loss, but the ongoing strive for energy-efficient technologies will eventually lead to large-scale utilization of circuits capable of energy recovery and energy reuse. Eventually, all computing circuits will belong to this category, regardless of the specific

implementation technology, and they will have to account for the information loss, which is fundamental in nature. But how good could the circuits and devices possibly be? Can a computer operate in principle without needing to generate heat? Of course, a device in the real world operating at a useful speed will always have some losses, but is there some necessary connection between computation and heat generation that requires a minimum amount of heat dissipation per bit? Understanding the link between physics and information is important for both guiding the search for a new basic binary device and for creating new ways of employing CMOS transistors that are more energy conscious. A pre-requisite for insightful design is the development of figures of merit for quantifying irreversible information loss in the computing hardware.

2.1. Link between Logical and Physical Reversibility

The key concepts are *logical reversibility* and *physical reversibility*. Logical reversibility is a property of a logical operation that maps a set of input bits onto a set of output bits. If the inputs can be deduced from the outputs, the mapping is logically reversible. Physical reversibility is a consequence of the fact that for an isolated system, the microscopic laws of physics are time reversible. As a consequence, given the final physical state of a system, one could, in principle, solve the equations of motion backwards in time and deduce the initial state of the system. Implementing a logical computation with a physical system involves mapping the physical initial and final states onto computational states. To perform a computation on a particular set of inputs, one prepares the system in the appropriate initial physical state and then lets the system evolve under physical law. The output is then read by subsequently measuring the final physical state, identifying it with the corresponding computational output. There exists a fundamental duality between the computation and the physical implementation.

The logical reversibility of a computation and the physical reversibility of a system which implements it are closely related. An isolated physical system can only implement a logically reversible computation. This follows simply from the fact that we could use the reversible laws of physics to deduce the logical input bits from the output state. Therefore, if we want to implement a logically irreversible computation with a physical system, the system cannot be isolated—it must be coupled to the environment. A logically irreversible operation such as AND or ERASE involves the loss of information; from knowing the output, one cannot determine the input. Where did the information go? It was present in the initial physical state of the system but is not available in the final physical state. The information must have been transferred from the system into the many untraceable degrees of freedom in the environment and is now irrecoverably lost in the complexity of that motion. Of course, if we could enlarge our system description to include all the relevant environmental degrees of freedom—all the positions of all the atoms in the substrate and chip holder for example—then we could deduce from the much larger final state what the input state must have been. But we cannot, and so we label the transfer of information and the associated energy from the system to the environment "heat generation." The development of statistical mechanics showed that thermodynamics is just mechanics applied in the context of our insurmountable ignorance and lack of information regarding the detailed motion of the very large systems.

Keyes and Landauer argued that the loss of information from the physical system that implements a computation results in a fundamental lower bound on how much heat is generated by a computation [1970]. This result follows from a thermodynamic (Boltzmann) entropy argument. One bit can be in two states, so the associated entropy is $S = k_B \ln 2$ Erasing an unknown bit, changing either 0 or 1 to a NULL state, means

there is a transfer of this entropy to the environment with associate free energy:

$$\Delta E = TS = k_B T \ ln\ 2. \tag{1}$$

Thus a physical implementation of a bit erasure or any logical operation that loses 1 bit of information must necessarily dissipate an amount of heat greater than or equal to ΔE Note that at room temperature, this is about 3zJ, nearly three orders of magnitudes lower than the end-of-the-roadmap CMOS transistor switching energy.

The fundamental lower limit is even lower than that. Bennett subsequently showed how any logically irreversible computation could be embedded within a logically reversible computation [1973]. Therefore, even logically irreversible operations, by embedding them in a larger computation, can be implemented with physically reversible processes. As a practical matter, this is often costly in terms of layout complexity, because all intermediate results have to be preserved. Nevertheless the fundamental lower limit for heat generation in a computation is 0.

In CMOS, a bit is represented by charge stored on a capacitor—either a transistor gate or a specifically fabricated storage capacitor. If the information is erased by just discharging the capacitor to ground, the maximum amount of heat is dissipated. But if, as is often the case, there is a copy of the bit elsewhere in the circuit, it can be used to erase the information with very little heat dissipation. Dissipation of as little as $\Delta E = 0.01 k_B T$ as recently been demonstrated experimentally [Boechler et al. 2010; Orlov et al. 2012]. Practical trade-offs in speed and circuit complexity of adiabatic CMOS have by now a substantial literature [Younis and Knight Jr. 1993; Starosel skii 2001]. The key to furthering this development is exploring erasure-aware alternatives to conventional approaches, because at molecular densities, the power dissipation constraints are even more formidable, and developing adiabatic and information-conserving switching for the nanoscale is even more crucial [Timler and Lent 2003; Lent et al. 2006]. To this end, we outline in the following a systematic method for quantifying the amount of irreversible information loss in digital circuits.

2.2. Quantifying Logical Information Loss

In order to evaluate the performance of existing and emerging computing hardware regarding the conservation of information, we must develop methods to quantify the amount of information loss related to various steps of the computing process. To this goal, we utilize Shannon's information theory and define the amount of information content in the inputs and outputs of an abstract *computing component*, based on the probabilities of each occurring *logic case*, which is defined as an instance of specific signal values. The difference in self-information is a measure of loss in this particular step, and with the specific logic input, and we can utilize another indicator, the difference in information entropy between input and output "ports" of the component as an average loss metric for the whole input space.

Self-information $I(x_i)$ also known as surprisal, measures the information content, that is, the specific entropy contribution of an individual *input logic case* x_i of a computing component:

$$I(x_i) = log_2\left(\frac{1}{P(x_i)}\right) = -log_2(P(x_i)), \tag{2}$$

where $P(x_i)$ is the probability of the occurrence of the input case x_i. Similarly, the self-information $I(y_j)$ of an individual *output logic case* y_j of a component is

$$I(y_j) = log_2\left(\frac{1}{P(y_j)}\right) = -log_2(P(y_j)), \tag{3}$$

Table I. Information Transformation in Binary AND Gate with Equiprobable Inputs

INPUT				OUTPUT			SELF-INFORMATION		
Case	Prob.	Bits		Case	Prob.	Bit	Input	Output	Difference
x_i	$P(x_i)$	a	b	y_j	$P(y_j)$	a AND b	$I(x_i)$ (bits)	$I(y_j)$ (bits)	$\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/4	0	0	y_1	3/4	0	2	0.415	1.585 bits (= $\log_2 3$ bits)
x_2		0	1						
x_3		1	0						
x_4		1	1	y_2	1/4	1		2	0 bits

where $P(y_j)$ is the probability of the occurrence of the output case y_j. The difference in self-information between each possible combination of input-output pairs is

$$\Delta I_{i,j} = I(x_i) - I(y_j). \tag{4}$$

Assuming equiprobable input, we can write Eq. (4) in terms of just the number n_j of input cases x_i mapped to the specific output logic case y_j:

$$\Delta I_j = log_2(n_j). \tag{5}$$

As an example, consider a two-input binary AND gate, which has one transition where the difference of information content is zero bits and three transitions where the difference is $\log_2 3 \approx 1.585$ bits, with the assumption that the inputs are uniformly distributed. Table I gives the self-information in each input and output case and the difference for each possible transition.

We can consider also the average information content through the input port of the computing component. This is known as information entropy $H(X)$ and defined as the expectation value of the self-information:

$$H(X) = \mathbb{E}_X[I(x_i)] = -\Sigma_{x_i \in X} P(x_i) log_2(P(x_i)), \tag{6}$$

where X is the set of all input logic cases. This probability-weighted average of the self-information measures the amount of uncertainty associated with the input. Similarly, the information entropy $H(Y)$ of the output port is

$$H(Y) = \mathbb{E}_Y[I(y_j)] = -\Sigma_{y_j \in Y} P(y_j) log_2(P(y_j)), \tag{7}$$

where Y is the set of all output logic cases. The difference in information entropy between the input and output port is

$$\Delta H_{X,Y} = H(X) - H(Y), \tag{8}$$

which represents the average information loss in this particular computation. In the AND gate example, equiprobable inputs give the maximized input entropy $H(X) = 2$ bits and the corresponding output entropy $H(Y) \approx 0.811$ bits, resulting in $\Delta H_{X,Y} = 1.189$ bits. In other words, the input signal set of the AND gate is capable of removing 2 bits of uncertainty, while the output signal is 1.189 bits less capable, on average. Once the AND operation has been performed and the inputs forgotten, we have only the information of the output signal available, which is 1.189 bits less than the inputs typically carry. This corresponds to an entropy decrease of $S = 1.189 k_B \ln 2$ and the related energy loss of $\Delta E = TS = 1.189 k_B T \ln 2 \approx 3.3$zJ at room temperature.

The reduction in the logical capability to remove uncertainty can be interpreted as information loss regarding the path of evolution of the computing system. According to Landauer's Principle, each lost bit has an inevitable energy cost, generating at least about 3zJ of heat to the environment at room temperature. While the amount of information loss heat per bit is in the region of 3×10^{-21} J/bit, thepredicted logic densities of molecular electronics can reach 10^{12} gates/cm^2 and the

inherent switching frequencies 10^{12} Hz, which translates into maximum heat power $P_{\text{information}} = 10^{-21}\,\text{J/bit} \times 10^{12}\,\text{bit/cm}^2 \times 10^{12}\,\text{Hz} = 10^{3}\,\text{W/cm}^2 = 1000\,\text{W/cm}^2$, even with each gate losing only a single bit. This is outside the limits of our cooling capacity, and therefore we must analyze the information loss and find ways to reduce it. The probabilistic framework presented here will be utilized to track the information loss in computer arithmetic algorithms, hardware structures, and logic gate implementations. Specifically, we evaluate the standard approach to binary addition on several levels of abstraction in the following sections.

2.3. Simulating the Information Loss of Binary Addition Operation

Binary addition is one of the most important arithmetic operations, and it is inherently irreversible. However, typical adder implementations perform much more erasures than is theoretically required. The results in this article are based on a MATLAB simulation of the monolithic theoretical operation and the standard ripple carry adder structure, modeled to include accumulated uncertainty of the logical state trajectory during the computation. Exhaustive simulation through all values of both operand words is performed for several word lengths, and 2D maps of information loss are acquired and statistics calculated.

An important assumption in the following analysis is that the overall distribution of the logical input, the top-level binary input operand words, is uniform. The equiprobable distribution is not exactly correct for many applications, but we chose this to analyze the information transformation of the computing hardware itself, without the disturbance induced by the probabilistics of a specific application. We also treat each of the computing components as an individual, whose simulated response is only dependent on the specific input value to that component. The component cannot utilize the state of the other components in the system to infer additional information, and the component does not vary assumptions about the distribution of its own input bits. The component independently assumes that all of its own input logic cases are equiprobable. This division of the system into separate components can be justified similarly as in the Hamiltonian computer models where each logical operation involves only a small number of physical entities [Peres 1985].

Another assumption concerns the role of circuit fanouts on different levels. While a physical fanout can cause irreversible information loss, this depends very strongly on the specific implementation and could be avoided. Therefore, we decide to regard the fanouts as information-conserving entities in the logical-level analysis.

Based on the truth tables and transformation characteristics of idealized computing components on different levels of abstraction, as illustrated in Figure 1, we proceed from theoretical monolithic addition towards gate-level adder implementations. The following models A and B are fully theoretical idealizations, while models C, D, and E each use a different logic gate set. The components in each model receive the normal routed signals from each other, but are otherwise independent and do not use other information to limit their input space. Note that we can distinguish between global and local reversibility/irreversibility: the monolithic addition operation (model A) has a certain amount of irreversible information loss, which can be considered global and affects also the lower level models, which implement the addition. However, the lower level models have their own local irreversibility, specific to each abstraction level. Local irreversibility is not upper-bounded by the global irreversibility; even if the top-level global operation were fully reversible, the local implementation could be irreversible and could lose any amount of logical information [Peres 1985]. Based on our study, irreversibility on a higher level seems to set a lower bound for the average lower-level local irreversibility.

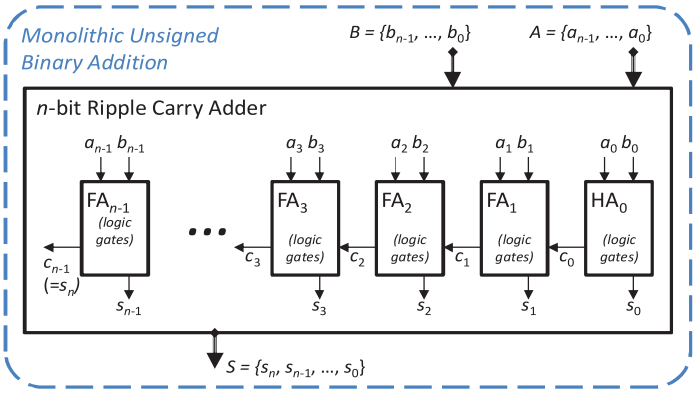


Fig. 1. Hierarchy of models for binary addition, from highest abstraction level to lowest. Model A: theoretical monolithic operation; model B: ripple carry adder structure with ideal HA/FA components; models C, D, E: components implemented with logic gates.

3. MONOLITHIC BINARY ADDITION (MODEL A)

Binary addition, one of the most important arithmetic operations, is inherently irreversible. The number of inherent bit erasures is important, since according to Landauer's Principle, each erasure generates at least the fundamental amount of heat, regardless of the implementation approach [Keyes and Landauer 1970]. As per Eq. (1), this is at room temperature $\Delta E = 3$ zJ, and usually much more is dissipated. The monolithic model presented here provides a lower bound for the average bit erasures in any adder implementation.

The highest abstraction level model considers the addition operation as a theoretical mapping between the input operand and output result spaces, as conceptualized in Figure 2(a), without taking into account any implementation details. The addition of two unsigned n bit operands A and B results in at most an $(n+1)$-bit result S, with the largest number of distinct operand pairs, $2n$ specific cases, compressed into the single result value $2n-1$, as illustrated in Figure 2(b), corresponding to the worst-case information loss of n bits [Hänninen and Takala 2010a]. The monolithic operation information loss for each operand pair in 8-bit addition is shown in Figure 2(c), where the loss ranges between 0 to 8 bits and the average is about 7 bits, provided equiprobable distribution of the input operand values. The topography is clearly shaped according to the base-2 logarithm of the frequency of occurrence of the result values, as expected.

4. ADDER STRUCTURE AND HIGH-LEVEL GATE SET

Standard adders based on binary logic [Koren 2002] perform much more bit erasures than is theoretically required by the addition operation itself, which will become apparent in the following comparison. Selecting a specific technology and a specific structure can only raise the number of expected erasures compared to the monolithic theoretical operation. From here onwards, we base all of the following models on the standard ripple carry adder structure, which we first construct using fully idealized subcomponents, consequently to be divided into basic logic gates. However, the gates are not necessarily tied to a specific implementation technology.

4.1. Ripple Carry Adder (Model B)

The addition operation is typically implemented by the standard ripple carry adder (RCA) constructed using half-adder (HA) and full-adder (FA) components [Koren 2002],

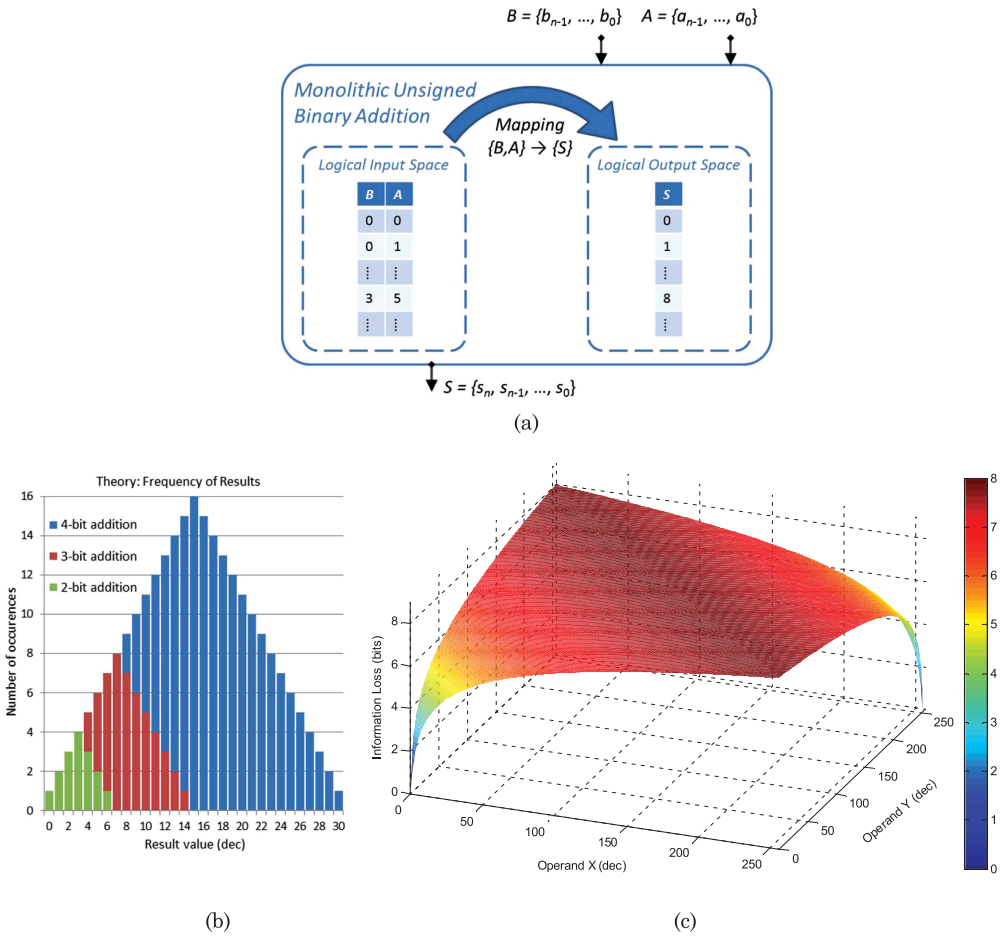


Fig. 2. Monolithic binary addition; (a) conceptual model, (b) frequency of addition results in 2-, 3-, and 4-bit operation, and (c) simulated information loss in 8-bit unsigned addition for each possible operand pair.

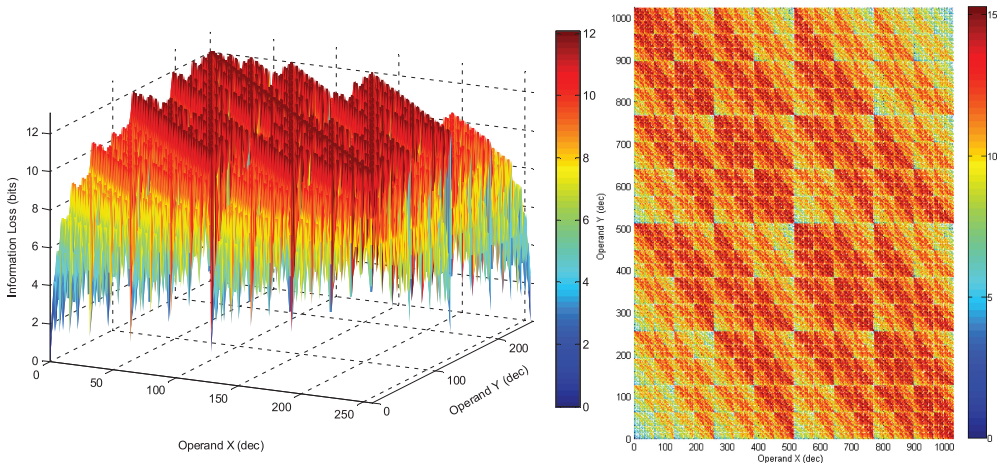
Table II. Truth Table and Transformation Characteristics of Half Adder

INPUT				OUTPUT				SELF-INFORMATION		
Case	Prob.	Bits		Case	Prob.	Bit		Input	Output	Difference
x_i	$P(x_i)$	a	b	y_j	$P(y_j)$	c_{out}	S	$I(x_i)$ (bits)	$I(y_j)$ (bits)	$\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/4	0	0	y_1	1/4	0	0	2	2	0 bits
x_2		0	1	y_2	2/4	0	1		1	1 bits
x_3		1	0							
x_4		1	1	y_3	1/4	1	1		2	0 bits

as depicted in the inner part of Figure 1. Considering these components ideal and completely characterized by the truth tables and transformation characteristics defined in Tables II and III, a lower bound can be formed for the average bit erasures in the standard RCA structure. Exact loss versus. the operand space of an 8-bit unit is shown in Figure 3(a), ranging from 0 to 12 bits with an average of about 9 bits, assuming evenly distributed inputs. Loss of a 10-bit unit shown in Figure 3(b) has a range from 0 to 15 bits.

Table III. Truth Table and Transformation Characteristics of Full Adder

INPUT					OUTPUT				SELF-INFORMATION		
Case	Prob.	Bits			Case	Prob.	Bits		Input	Output	Difference
x_i	$P(x_i)$	a	b	c_{in}	y_j	$P(y_j)$	c_{out}	s	$I(x_i)$ (bits)	$I(y_j)$ (bits)	$\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/8	0	0	0	y_1	1/8	0	0	3	3	0 bits
x_2		0	0	1	y_2	3/8	0	1		1.415	1.585 bits ($= \log_2 3$ bits)
x_3		0	1	0							
x_4		1	0	0							
x_5		0	1	1	y_3	3/8	1	0		1.415	1.585 bits ($= \log_2 3$ bits)
x_6		1	0	1							
x_7		1	1	0	y_4	1/8	1	1		3	0 bits
x_8		1	1	1							



(a) (b)

Fig. 3. Ripple carry adder structure with ideal HA/FA components, simulated information loss vs. each possible operand pair in (a) 8-bit unsigned addition, and (b) 10-bit unsigned addition.

The topography versus the operand axes still shows traces of the curvature of the theoretical operation, but the dominant feature is the overlaid repeating self-similar pattern, originating from the discrete implementation using ideal HA/FA components. It is interesting to note that even though the average loss is higher than the loss in the monolithic operation, as expected, there are also minima which are significantly lower than the monolithic average. This is a surprise and indicates that the RCA structure has some pathological properties of information transformation, which enable some operand pairs to propagate through the structure conserving relatively large part of the initial logical information. Similar information loss patterns are present in adders of all larger operand word lengths.

4.2. Ripple Carry Adder on {XOR,AND,OR} Gate Set (Model C)

The adder model of Figure 4(a) can be subdivided closer to an actual circuit implementation by constructing the half- and full-adder components using a set of standard gates {XOR, AND, OR}, based on the structures in Figures 4(b) and 4(c). The ideal truth Tables IV, V, and VI produce a very flat information loss throughout the operand pair space of an RCA, resulting an average loss in the 8-bit unit of about 38 bits per addition, illustrated in Figure 5(a). A topography map of the loss in a 10-bit unit is shown in Figure 5(b).

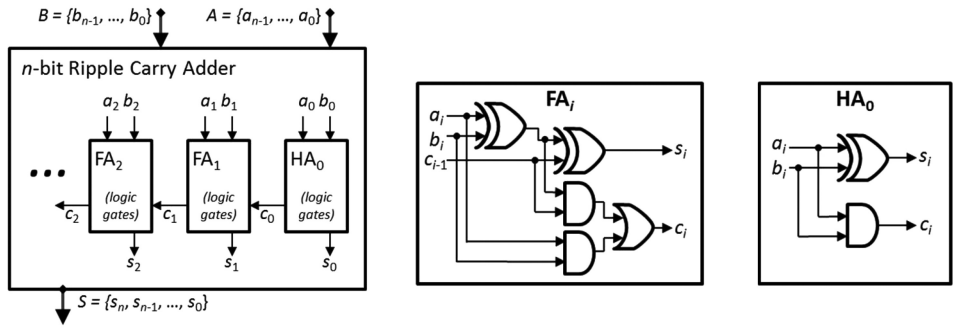


Fig. 4. (a) Composing RCA using a set of standard gates {XOR, AND, OR}, (b) full adder, and (c) half adder.

Table IV. Truth Table and Transformation Characteristics of XOR Gate

INPUT				OUTPUT			SELF-INFORMATION		
Case	Prob.	Bits		Case	Prob.	Bit	Input	Output	Difference
x_i	$P(x_i)$	a	b	y_j	$P(y_i)$	a XOR b	$I(x_i)$ (bits)	$I(y_j)$ (bits)	$\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/4	0	0	y_1	2/4	0	2	1	1 bit
x_4		1	1						
x_2		0	1	y_2	2/4	1		1	1 bit
x_3		1	0						

Table V. Truth Table and Transformation Characteristics of AND Gate

INPUT				OUTPUT			SELF-INFORMATION		
Case	Prob.	Bits		Case	Prob.	Bit	Input	Output	Difference
x_i	$P(x_i)$	a	b	y_j	$P(y_i)$	a AND b	$I(x_i)$ (bits)	$I(y_j)$ (bits)	$\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/4	0	0	y_1	3/4	0	2	0.415	1.585 bits ($= \log_2 3$ bits)
x_2		0	1						
x_3		1	0	y_2	1/4	1		2	0 bits
x_4		1	1						

Table VI. Truth Table and Transformation Characteristics of OR Gate

INPUT				OUTPUT			SELF-INFORMATION		
Case	Prob.	Bits		Case	Prob.	Bit	Input	Output	Difference
x_i	$P(x_i)$	a	b	y_j	$P(y_i)$	a OR b	$I(x_i)$ (bits)	$I(y_j)$ (bits)	$\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/4	0	0	y_1	1/4	0	2	2	0 bits
x_2		0	1						
x_3		1	0	y_2	3/4	1		0.415	1.585 bits ($= \log_2 3$ bits)
x_4		1	1						

The nearly flat topography completely hides any traces of the monolithic addition and the RCA structure patterns. This presumably originates from the duality of AND and OR gates canceling each other out in the utilized configuration, while the XOR gate has fully balanced logical state compression. The gate set turns out to be relatively efficient in logical information conservation when compared to the other gate-level adders. However, we are not aware of adequately robust direct physical implementations of this set, and the gates have to be further divided into subgates with higher information loss.

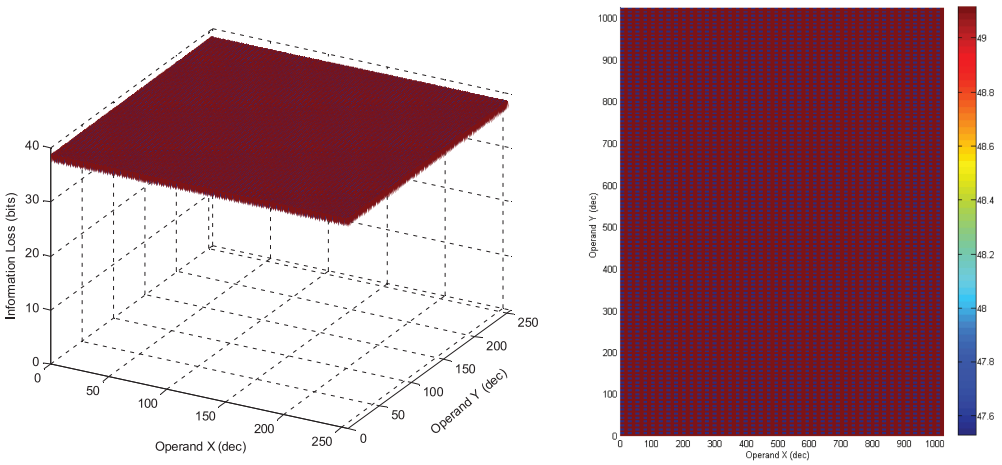


Fig. 5. RCA structure with {XOR,AND,OR} gate set-based HA/FA components, simulated information loss vs. each possible operand pair in (a) 8-bit unsigned addition, and (b) 10-bit unsigned addition.

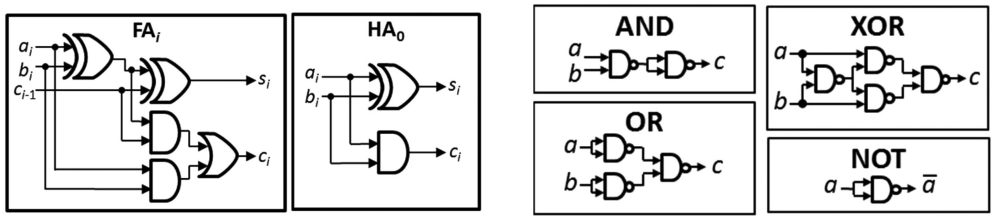


Fig. 6. (a) Composing RCA: (a) HA/FAs structured using gate set {XOR, AND, OR}, and (b) implemented using only NAND gates.

5. ADDER IMPLEMENTATIONS

Practical digital circuits, including the binary adders, are based on logic gates with existing and realistic physical implementations available. In traditional complementary transistor circuits, the unit gate of choice is the two-input NAND, which can be implemented very efficiently in CMOS and adiabatic CMOS. In several post-transistor technologies, including quantum-dot cellular automata (QCA) paradigm, the main logic component is the three-input majority voter gate (MG).

5.1. Ripple Carry Adder on Ideal NAND Gates (Model D)

Low-level implementation of the ripple carry adder has the half-adder and full-adder components constructed using two-input NAND-gates, typical in standard digital circuits [Koren 2002]. The HAs/FAs in Figure 6(a) are structured by the abstract gate set {XOR,AND,OR} of the previous abstraction level, but implemented using only ideal NAND-gates, as illustrated in Figure 6(b), with the truth table in Table VII. The NAND performs at worst a 3-to-1 logical state mapping between input and output cases. Information loss for an 8-bit ripple carry adder unit is shown in Figure 7, with an average of about 112 bits lost. Information loss in a 10-bit unit is presented in Figures 8 and 9.

It is very interesting to note that the topography versus operand axes shows again the traces of the curvature of the monolithic theoretical operation and the self-similar

Table VII. Truth Table and Transformation Characteristics of NAND Gate

INPUT				OUTPUT			SELF-INFORMATION		
Case	Prob.	Bits		Case	Prob.	Bit	Input	Output	Difference
x_i	$P(x_i)$	a	b	y_j	$P(y_i)$	a NAND b	$I(x_i)$ (bits)	$I(y_j)$ (bits)	$\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/4	0	0	y_1	3/4	1	2	0.415	1.585 bits ($= \log_2 3$ bits)
x_2		0	1						
x_3		1	0	y_2	1/4	0		2	0 bits
x_4		1	1						

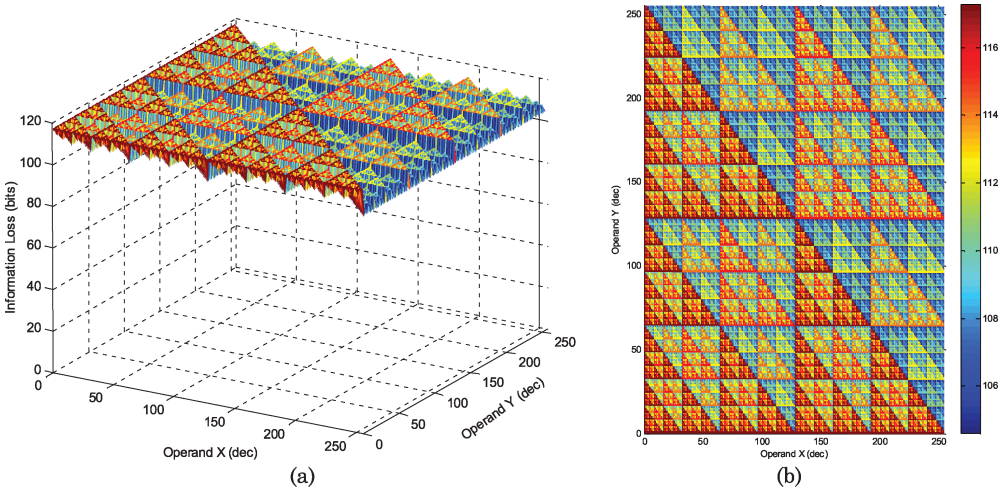


Fig. 7. RCA structure with ideal NAND gate-based components, simulated information loss vs. each possible operand pair in 8-bit unsigned addition. (a) 3D view, and (b) topography map.

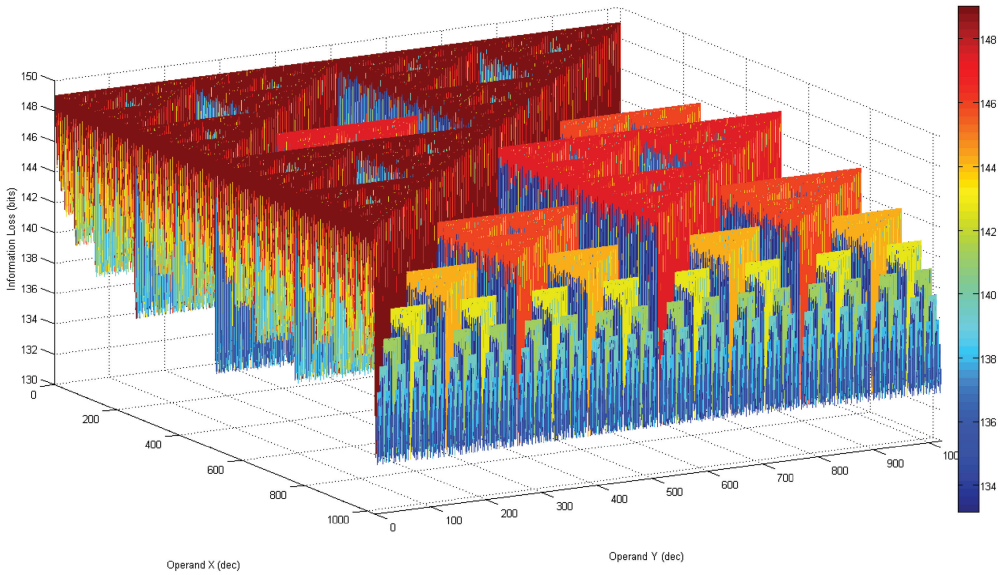


Fig. 8. RCA structure with ideal NAND gate-based components, simulated information loss vs. each possible operand pair in 10-bit unsigned addition, 3D view.

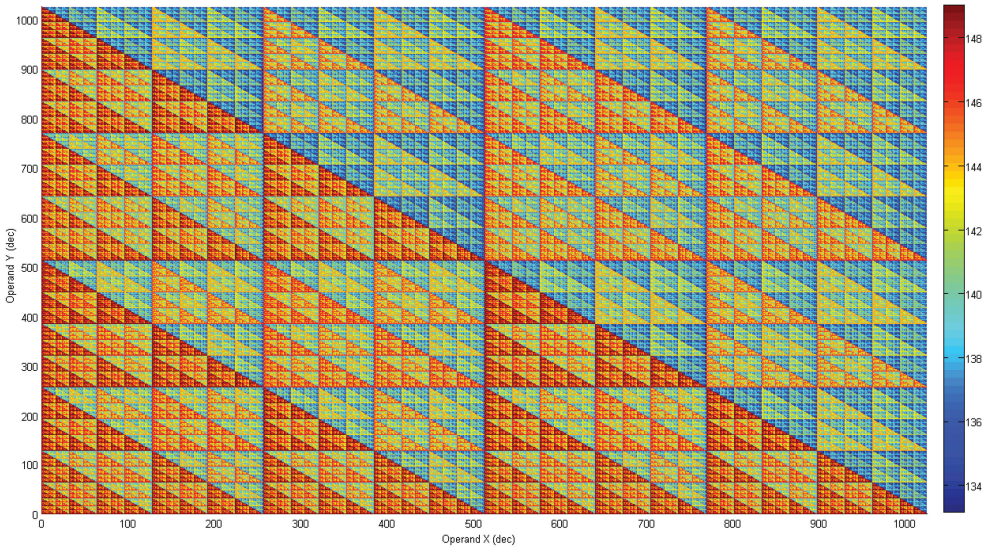


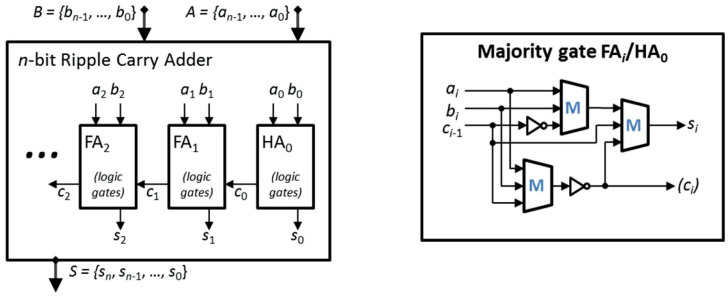
Fig. 9. RCA structure with ideal NAND gate-based components, simulated information loss vs. each possible operand pair in 10-bit unsigned addition, topography map.

overlay of the RCA structure patterns, even though they were completely absent from the structured {XOR,AND,OR} gate-level idealization. This effect originates from the NAND-based implementation of the AND and OR gates, which are not symmetrical anymore, from the viewpoint of information loss. The number of bit erasures in this implementation is an order of magnitude higher than in the previous, more coarse and abstract idealizations.

5.2. Ripple Carry Adder on Ideal Majority Voter Gates (Model E)

Another physically realistic implementation utilizes three-input majority voter gates (MG), which are available in several beyond-CMOS technologies, including quantum-dot cellular automata (QCA) paradigm. Majority gate-based implementation of the half/full-adders underlying the RCA structure is illustrated in Figure 10 [Hänninen and Takala 2010b], while the ideal MG has the truth table in Table VIII. The MG performs a fully balanced and uniform compression of the logical space, which has the consequence that the RCA erasure number is constant throughout the whole operand pair space. An 8-bit unit erases a constant 48 bits per addition operation, and a 10-bit unit erases a constant 60 bits per addition, as illustrated in Figure 11.

The flat topography of the 2D erasure chart indicates that the information loss is constant versus the operand axes and very competitive against the standard Boolean gate-based implementations. This suggests that the majority gate is a very suitable logic primitive for reversible computation, especially since the potential physical implementations are predicted to have inherent signal energy transmission from device to device [Timler and Lent 2003]. Based on our observations, the properties of balanced and uniform state compression have beneficial consequences on information loss on the module and structure level, and they can be simply identified in the truth tables of the basic gates. Therefore, other logic primitives, including the minority voter gate, may be as useful as the majority gate for reversible computing.

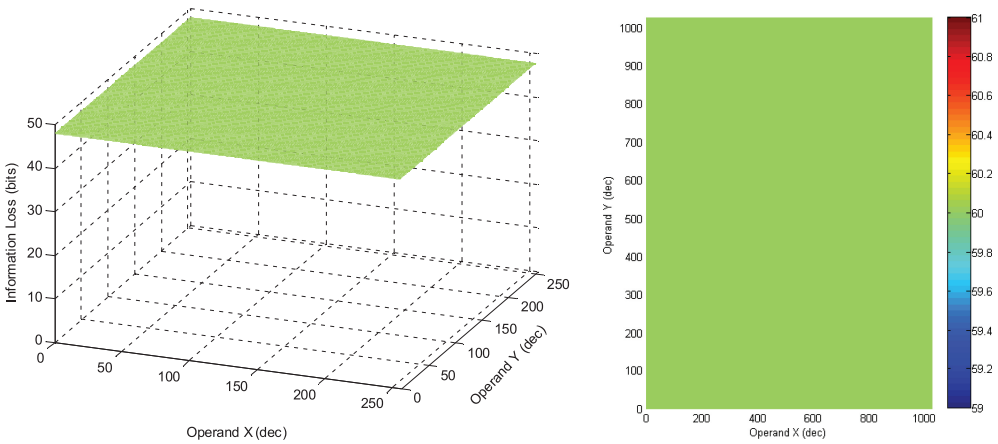


(a) (b)

Fig. 10. (a) Composing RCA, and (b) full/half adder using only majority voter gates.

Table VIII. Truth Table and Transformation Characteristics of Majority Voter Gate

INPUT					OUTPUT			SELF-INFORMATION		
Case x_i	Prob. $P(x_i)$	Bits a	b	c	Case y_j	Prob. $P(y_j)$	Bits (M a, b, c)	Input $I(x_i)$ (bits)	Output $I(y_j)$ (bits)	Difference $\Delta I_{i,j} = I(x_i) - I(y_j)$ (bits)
x_1	1/8	0	0	0	y_1	4/8	0	3	1	2 bits ($= \log_2 4$ bits)
x_2		0	0	1						
x_3		0	1	0						
x_4		1	0	0						
x_5		0	1	1	y_2	4/8	1		1	2 bits ($= \log_2 4$ bits)
x_6		1	0	1						
x_7		1	1	0						
x_8		1	1	1						



(a) (b)

Fig. 11. RCA structure with majority voter gate-based HA/FA components, simulated information loss vs. each possible operand pair in (a) 8-bit unsigned addition, and (b) 10-bit unsigned addition.

6. STATISTICS AND SCALING

The loss of information in the adders usually depends slightly on the specific operand pair, as visible in the topography maps in Figures 3, 5, and 7–9. The developed abstractions follow the statistics presented in Table IX.

Table IX. Statistics for 8-Bit Binary Addition, assuming Uniform Input Operand Distribution

ABSTRACTION LEVEL	INFORMATION LOSS STATISTICS (bits)			
	Min	*Avg*	*Max*	*Std. dev.*
A. Monolithic theoretical addition	0	7.3	8	1.5×10^{-14}
B. RCA, ideal HA/FA	0	8.8	12.1	1.5
C. RCA, ideal XOR, AND, OR	37.2	38.4	38.8	0.4
D. RCA, ideal NAND	104.6	111.7	117.3	2.4
E. RCA, ideal MG	48	48	48	0

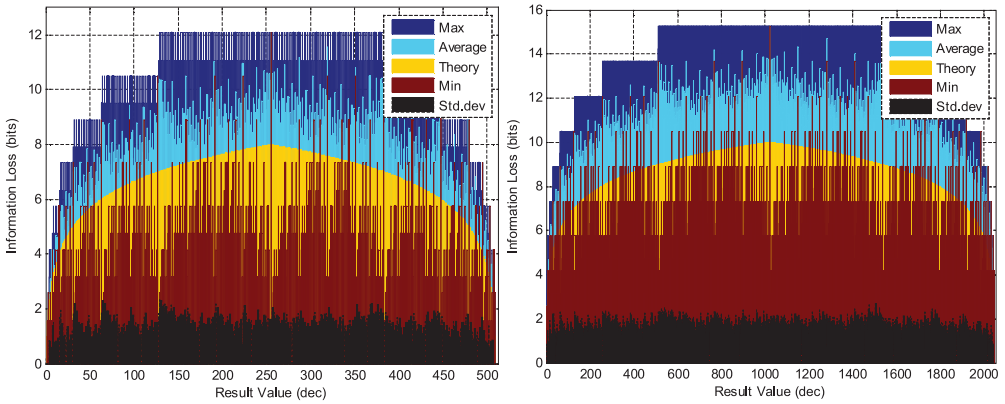


Fig. 12. Model B, ideal RCA: erasure statistics per result value with ideal HA/FA components, showing also theoretical monolithic operation in yellow, (left) 8-bit operand length, and (right) 10-bit operand length.

The ripple carry adder based on ideal half- and full-adder components (model B) is the only adder abstraction, where the number of bit erasures is significantly affected by the specific result value, ranging from 0 to 12 bits, as illustrated in Figure 3(a) for an 8-bit adder. The statistics of this model in Figure 12 show significant standard deviation of two bits around the average erasure per each result value. Although the average is above the erasures of the theoretical operation in Figure 13, there are minima below this. Apparently, some operand pairs for some specific result values can propagate through the ideal HA/FA-based RCA conserving logical information surpassing the theoretical operation, which is a surprise.

The other RCA models have nearly constant erasure rates versus the result values and operands. The largest deviation is about one bit, less than 1% of the average erasures, found in the ideal NAND-gate-based model (D) with the statistics versus result value shown in Figure 14. The XOR-, AND-, OR-set-based model (C) erases about constant 38 bits and the majority gates-based model (E) erases constant 48 bits per 8-bit addition, as illustrated versus result value in Figure 15.

The number of bit erasures depends also on the operand word length n of the adder. The scaling trends for the theoretical operation (model A) and the ideal HA/FA-based model (B) are linear versus n [Hänninen and Takala 2010a], while this relationship for the other models has yet not been explored. In summary, models A and B are linear, while model D behaves worse by an order of magnitude. Models C and E will likely scale better, but of these, only model E is likely to have a directly corresponding physical implementation, for example, a quantum-dot cellular automata majority gate.

7. CONCLUSIONS

We believe that eventually all computing circuits will have to utilize energy-recovery and reuse approaches to achieve the energy efficiency required for sustaining the

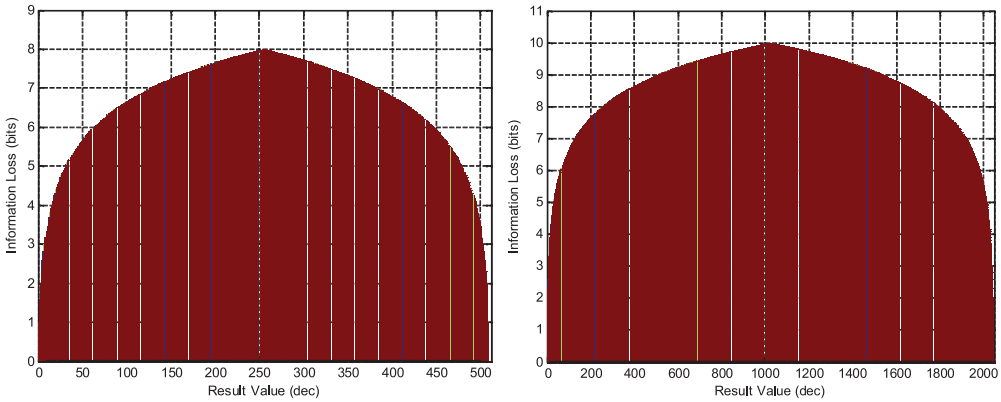


Fig. 13. Model A, theoretical monolithic addition operation: erasures per result value, (left) 8-bit operand length, and (right) 10-bit operand length.

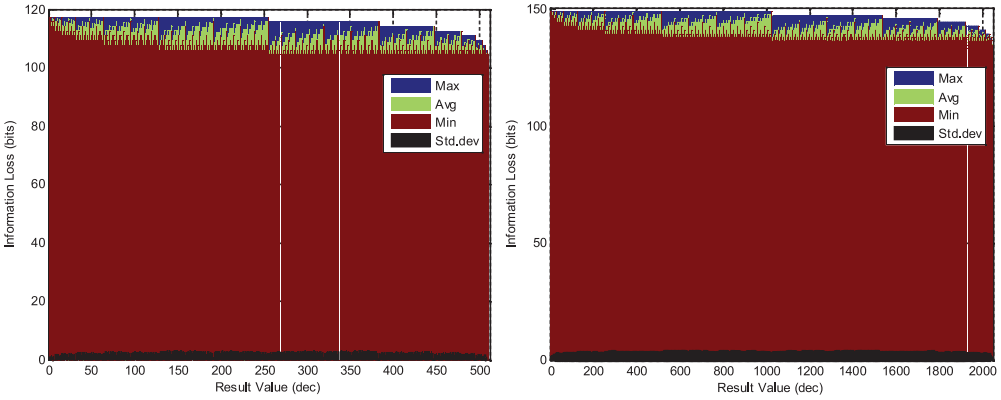


Fig. 14. Model D, RCA based on NAND gate only: erasure statistics per result value, (left) 8-bit operand length, and (right) 10-bit operand length.

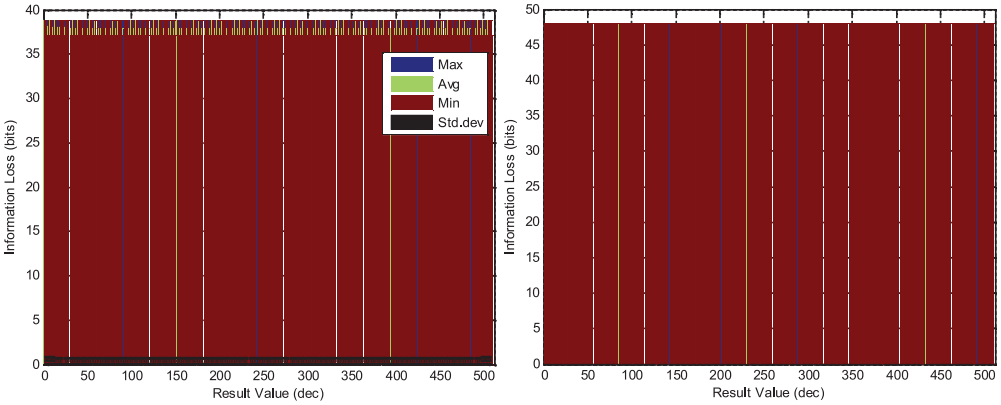


Fig. 15. Erasure statistics per result value with 8-bit operand length, (left) model C, ideal {XOR,AND,OR} gate set, and (right) model E, ideal majority voter gate, which has no deviation at all.

continuous growth of computing power in the future. While the physical friction-like loss effects can be limited by the foreseen technology improvements and techniques, including adiabatic charging, the energy loss related to the destroying of information can be limited only by avoiding the loss of logical information itself. A necessary prerequisite for practical chip design is the capability to assess the information loss properties of the underlying technology, circuit style, circuit structure, and the logical operation to be performed. In this work, we have analyzed the information transformations of relatively complex digital circuits in detail, developing a method to systematically abstract the information loss in logical operations. The most important arithmetic operation, binary addition, has been characterized in this regard, both to demonstrate our analysis method and to obtain understanding of the information loss in the addition operation itself and the adder implementations.

The binary addition has been modeled on several design abstraction levels and bounds for information erasure developed for optimizing the circuits from the perspective of this new goal, reducing the information loss. The gate-level implementations lose significantly more information than the theory suggests, while there are also surprising operand combinations which may actually dissipate less heat than the "monolithic" average value suggested by the theory. This kind of pathological phenomena and the found complex self-similar structure of the information-loss patterns remain a future research topic.

REFERENCES

Charles Bennett. 1973. Logical reversibility of computation. *IBM J. Res. Dev.* 17, 6 (1973), 525–532. DOI: http://dx.doi.org/10.1147/rd.176.0525

Graham P. Boechler, Jean M. Whitney, Craig S. Lent, Alexei O. Orlov, and Gregory L. Snider. 2010. Fundamental limits of energy dissipation in charge-based computing. *Appl. Phys. Lett.* 97, 10 (2010). DOI: http://dx.doi.org/10.1063/1.3484959

Ismo Hänninen and Jarmo Takala. 2010a. Irreversible bit erasures in binary adders. In *Proceedings of the IEEE Conference on Nanotechnology*. 223–226. DOI: http://dx.doi.org/10.1109/NANO.2010.5697744

Ismo Hänninen and Jarmo Takala. 2010b. Binary adders on quantum-dot cellular automata. *J. Sig. Proc. Syst.* 58, 1 (January 2010), 87–103. DOI: http://dx.doi.org/10.1007/s11265-008-0284-5

ITRS. International technology roadmap for semiconductors. (2012). http://www.itrs.net/Links/2012ITRS/Home2012.htm. (Last accessed September 2012.)

Robert.W. Keyes and Rolf W. Landauer. 1970. Minimal energy dissipation in logic. *IBM J. Res. Dev.* 14, 2 (1970), 152–157. DOI: http://dx.doi.org/10.1147/rd.142.0152

Israel Koren. 2002. *Computer Arithmetic Algorithms* (2nd. ed.). A. K. Peters, Ltd., Natick, MA.

Craig S. Lent and P. Douglas Tougaw. 1997. A device architecture for computing with quantum dots. *Proc. IEEE* 85, 4 (1997), 541–557. DOI: http://dx.doi.org/10.1109/5.573740

Craig S. Lent, Mo Liu, and Yuhui Lu. 2006. Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling. *Nanotechnol.* 17, 16 (2006), 4240–4251. DOI: http://dx.doi.org/10.1088/0957-4484/17/16/040

Alexei O. Orlov, Craig S. Lent, Cameron C. Thorpe, Graham P. Boechler, and Gregory L. Snider. 2012. Experimental test of Landauer's Principle at the sub-kBT level. *Jap. J. of Appl. Phys.* 51, 06FE10 (2012). DOI: http://dx.doi.org/10.1143/JJAP.51.06FE10

Asher Peres. 1985. Reversible logic and quantum computers. *Phys. Rev. A* 32, 6 (1985), 3266–3276.

Victor I. Starosel'skii. 2001. Adiabatic logic circuits: A review. *Russian Microelectron.* 31, 1 (2001), 37–58. DOI: http://dx.doi.org/10.1023/A:1013857006906

John Timler and Craig S. Lent. 2003. Maxwell's demon and quantum-dot cellular automata. *J. Appl. Phys.* 94, 2 (2003), 1050–1060. DOI: http://dx.doi.org/10.1063/1.1581350

Saed G. Younis and Thomas F. Knight Jr. 1993. Practical implementation of charge recovering asymptotically zero power CMOS. In *Proceedings of the Symposium on Research Integrated Systems*. 234–250

Received September 2013; revised December 2013; accepted April 2014