Irreversible Bit Erasures in Binary Multipliers

Ismo Hänninen and Jarmo Takala
Department of Computer systems
Tampere University of Technology
Korkeakoulunkatu 1, Tampere, Finland
Email: [ismo.hanninen, jarmo.takala]@tut.fi

Craig S. Lent
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556, USA
Email: lent@nd.edu

***Abstract*—Nanocircuits will suffer from heat dissipation due to irreversible information erasure, which is a potential new limiting factor for the maximum operating frequencies of the future circuit technologies. This paper estimates the degree of information loss in the binary multiplier structures and demonstrates that the standard hardware approaches are sub-optimal, by several orders of magnitude in comparison with the determined theoretical limit of the multiplication operation. The hardware analysis is based on the arithmetic units proposed for implementation with quantum-dot cellular automata (QCA), a circuit technology reaching molecular device densities and extremely high signal energy conservation. The results are generally applicable to all other emerging technologies based on the majority logic gate.**

I. Introduction

Computing power has been for a long time closely related to the energy-efficiency of the underlying technology, and heat dissipation is the limiting factor for modern integrated circuits, with the energy waste of the CMOS inherent to the operating principle of the voltage mode logic. While the technologies foreseen in the future are expected to improve the energy-efficiency by orders of magnitude, they will still be limited by the electrical power. However, the underlying mechanism heating the circuits might be fundamentally different. [1]

A significant and potentially also the major source of heat in the circuits constructed with the foreseen nanoscale components is the *irreversibility induced dissipation*, originating from the thermodynamics of decreasing system entropy during the computation. This effect has remained hidden behind the huge energy waste inherent to the transistor technologies. Quantum-dot cellular automata (QCA) is a promising computing paradigm with a path to transistor-less molecular implementations, based on bistable cellular automata used to construct both the logic and the interconnects. A major benefit of this approach is the reuse of signal energy throughout the circuit, resulting in very high power efficiency [2]. Combined with the ultra-high device density and switching frequency reaching the terahertz regime, this raises the irreversibility induced dissipation as a significant design factor [3].

Recent work on the logical irreversibility presented surprisingly tight operating frequency limits for computer arithmetic based on high density nanoscale components. For dense layout binary adders implemented with molecular QCA, the predicted operating frequency was limited to at most tens of gigahertz [4], while multiplier designs with significantly lower logic density were limited to around two hundred gigahertz in a very optimistic analysis [5]. The combination of nanometer wide devices and information loss clearly prevents the technology switching rate to be transformed into the desired clock frequencies in the hundreds of gigahertz regime, which would otherwise be reasonable for the pipelined logic designs. A comparison of standard adder structures proposed for QCA indicates that a simple pipelined ripple carry adder (RCA) would have the smallest loss of logical information [6].

This paper quantifies the information loss in standard hardware structures for the binary multiplication, which is one of the most common arithmetic operations. The heat per operation varies between the structures, which can now be evaluated from new perspective based on this pioneering work, to the best knowledge of the authors. We will start by defining the relationship between information loss and heat generation, followed by the development of bounds for the multiplication operation with considerable cost savings via exclusion of zero operands. Next, the worst case bit erasures in the majority logic QCA implementations are compared, based on the gate counts and runtime clock cycles, indicating that direct accumulation of the summands with an array multiplier would outperform the existing serialized approaches.

II. Irreversibility and Heat Generation

The continued evolution of modern CMOS processing chips face many challenges related to scaling device sizes, arguably the most fundamental being the power dissipation problem. Present microprocessors with dissipation of 50–100 W/cm^2 press the limits of air cooling, while the ITRS 2009 Roadmap projects fully scaled CMOS at a device density of 10^{10} cm^{-2}, a switching speed of 12 THz, and a switching energy of 3 aJ [1]. If all of these devices were switching at full rate, the chip would generate heat at a rate of 360 kW/cm^2. In the short term this has motivated sophisticated power management and heat sinking, but also forced examination of more fundamental issues. It has become clear that logic transistor operation is inherently wasteful in terms of heat production. Overcoming this will require a new basic element, the ”next switch.” The QCA paradigm is a promising candidate in this search, though *any* nano-scale transistor replacement technology must satisfy the essential requirement of exhibiting dramatically lower power dissipation during switching. But how good could it be? Is heat generation somehow fundamentally necessary for computation? In particular, is there a minimum amount of heat that must be dissipated to compute a bit of information? Why are heat and information in any way connected?

978-1-4244-9474-3/11/$26.00

For an isolated system, the microscopic laws of physics are time reversible (excluding some exotic processes involving the electroweak force). This *physical reversibility* means that 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. For a computation to be *logically reversible* means that given the outputs of the computation, one could deduce what the inputs must have been. Implementing a 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.

The logical reversibility of a computation and the physical reversibility of a system which implements it are 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 inputs from the outputs. A consequence is that 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, many, untrackable degrees of freedom in the environment, and is now unrecoverably 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, then we could deduce from the much larger final state the input state. 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, lack of information, regarding the detailed motion of large systems.

Landauer argued that the loss of information from the physical system results in a fundamental lower bound on how much heat is generated by a computation [7]. This result follows quickly from a thermodynamic (Boltzmann) entropy argument. One bit can be in 2 states, so the associated entropy is $S = k_{\mathrm{B}} \ln(2)$. Erasing an unknown bit, say changing either 0 or 1 to a NULL state, means there is a transfer of this entropy to the environment with associate free energy $\Delta E = TS = k_{\mathrm{B}} T \ln(2)$. Thus a physical implementation of any logical operation that loses 1 bit of information must necessarily dissipate at least ΔE of heat. Note that at room temperature this is about 0.004 aJ, nearly three orders of magnitude lower than end-of-the-roadmap CMOS transistors.

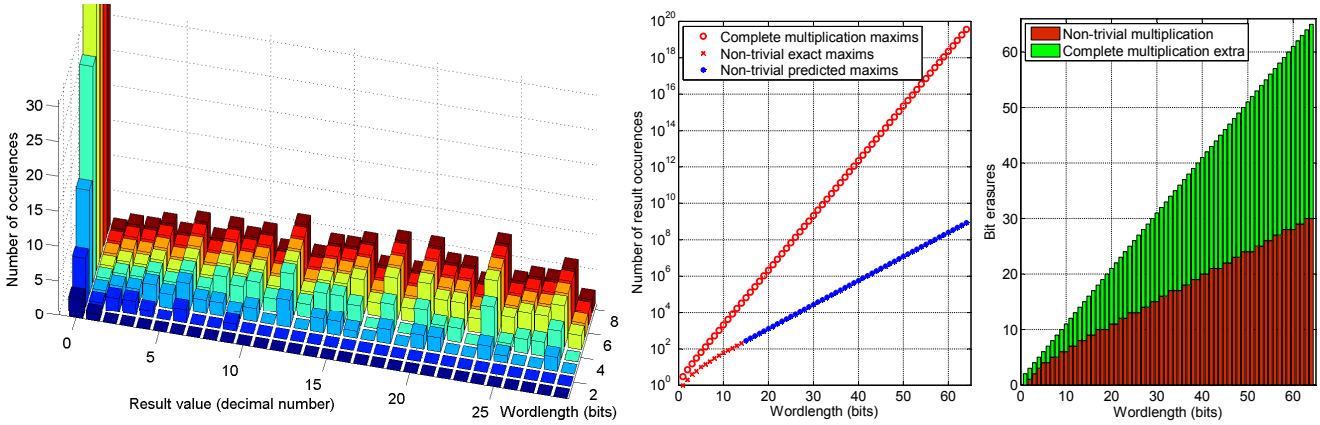
The fundamental lower limit is even lower. Bennett subsequently showed how any logically irreversible computation could be embedded within a logically reversible computation [8]. Therefore even logically irreversible operations, by embedding them in a larger computation, can be implemented with physically reversible processes. As a practical matter, this has often a prohibitive cost in the layout complexity, because all intermediate results have to be preserved. Nevertheless, the fundamental lower limit for heat generation is 0.

The important design considerations are therefore practical issues. Present CMOS effectively performs an erasure every time a transistor switches states—generating hugely unnecessary levels of heat. To make it to the nanoscale, we must do much better than that. Now switching the state of virtually anything in the physical world dissipates some small amount energy as heat because it's impossible to completely isolate a system. There will always be some small residual energy transfer with the environment through mechanisms like phonon, plasmon, or molecular vibronic coupling. These are the quantum mechanical versions of friction. Friction-type dissipation can always be made smaller, either by more clever design or by simply moving more slowly; a characteristic of friction is that it is proportional to the velocity. Erasure events, by contrast, aren't amenable to such techniques. They *each* have a fundamental heat dissipative cost, and these costs can accumulate catastrophically at nanoscale densities (1 nm^2 footprints corresponds to 10^{14} cm^{-2} densities). Even for molecular nanoelectronics, it is not worth the overhead to do full Bennett-style embedding in order to completely eliminate erasure costs, though partial implementation can be very helpful [9]. Designing circuits to be "erasure-aware" means using computational elements, like QCA, that don't unnecessarily erase information, and managing carefully where and how often bit erasure occurs. Making such practical trade-off decisions, informed by the fundamental thermodynamic issues, is necessary for achieving nanoscale levels of integration without vaporizing our cleverly-designed ultrasmall devices.

III. Reversibility of Binary Multiplication

The binary multiplication operation as such is not reversible, since it performs compression between the input and output state spaces by a non-bijective mapping. The multiplication of two n-bit integer operands produces at most a $2n$-bit result, but despite the apparent equivalence in the number of bits, significant logical information is lost. The reason is the unbalanced state compression inherent to the highly complicated result value spectrum shown in Fig. 1(a) for the unsigned operation. We treat separately the complete spectrum multiplication and non-trivial multiplication without zero operands.

Complete multiplication. The result value zero compacts the maximum number of operand pairs in all wordlengths. Illustrated by the top curve in Fig. 1(b), the number of zero results c_0 is dependent on the wordlength and follows exactly the power of two's law defined in Table I, found analytically by considering the number of trivial operand combinations leading to the single result. The logical reversal of this worst case compression requires $b_0 = n+1$ extra bits, which identify the specific operand value pair uniquely. The extra bits required to logically reverse the multiplication as one indivisible



(a) (b) (c)

Fig. 1. Unsigned integer multiplication with n-bit operands: a) Beginning of the result value spectrum, b) maximum number of result value occurences in the complete and non-trivial operations, and c) logical bit erasures per operation, with the total bar height representing the complete multiplication.

logical operation can be interpreted as the minimum amount of information lost in any irreversible multiplier structure at best. This loss determines the minimum achievable energy cost per operation, and provides a common reference point for comparing the various hardware implementations.

Non-trivial multiplication. Multiplication with zero value operands can be detected and bypassed past the computation hardware, and therefore, it is reasonable to consider a logical operation, which does not include the zero operands and the corresponding zero result. In this operation, the maximum number of operand pairs is compacted behind the second highest bars in the result value spectrum in Fig. 1(a), and several different results reach this level on each wordlength. The number of the operand pairs reduced to each second most occuring result value is illustrated by the bottom curve in Fig. 1(b) and appears to follow an exponential law, but we have not determined an exact analytical expression for this. Based on small wordlength brute force data, a parametric regression model fitted using nonlinear least squares criteria in Matlab for the number of operand pairs c_{nt} leading to the specific result is defined in Table I. This model has a trend of diminishing relative error with increasing wordlength, making it suitable for extrapolation. For the logical reversal of the state compression, the number of extra bits b_{nt} required to identify the operands is *sub-linear* in respect to the wordlength, following the logarithmic form defined in Table I.

The number of bit erasures in both types of indivisible logical operations are shown in Fig. 1(c). The total height of each bar represents the exact information loss in complete multiplication, while the bottom portion of the bar represents the predicted loss in the non-trivial multiplication. The top portion corresponds to the lost bits related only to the trivial zero results, and asymptotically with the wordlength, this part settles to about 54% of the bar height. Although this partitioning is based on a prediction model, it seems likely that the non-trivial multiplication would typically save more than half of the bit erasures, compared to the complete operation.

TABLE I
MODELS FOR THE MAX NUMBER OF RESULTS AND BIT ERASURES.

	Complete multiplication	**Non-trivial multiplication**
Results:	$c_0 = 2^{n+1} - 1$	$c_{nt} = p_1^{n+p_2} + p_3$
Erasures:	$b_0 = n + 1$	$b_{nt} = p_4 * log(n + p_5) + p_6$
$p_1 = 1.3560$, $p_2 = 3.5405$, $p_3 = -3.5579$		
$p_4 = 122.9461$, $p_5 = 240.9868$, $p_6 = -673.3972$		

IV. BIT ERASURES IN SUMMAND ACCUMULATION

The paper-and-pencil multiplication algorithm of unsigned binary operands $A = (a_{n-1}, \ldots, a_1, a_0)$ and $B = (b_{n-1}, \ldots, b_1, b_0)$, producing the result $M = (m_{2n-1}, \ldots, m_1, m_0)$, where a_0, b_0, and m_0 are the least significant bits, is defined as follows:

$$
\begin{array}{cccccccc}
 & & & & a_{n-1} & \cdots & a_1 & a_0 \\
\times & & & & b_{n-1} & \cdots & b_1 & b_0 \\
\hline
 & & & & a_{n-1}b_0 & \cdots & a_1b_0 & a_0b_0 \\
 & & & a_{n-1}b_1 & \cdots & & a_1b_1 & a_0b_1 \quad 0 \\
 & & & & \vdots & & & \\
 & & & & & & 0 & 0 \\
+ & a_{n-1}b_{n-1} & \cdots & a_1b_{n-1} & a_0b_{n-1} & 0 & 0 & 0 \\
\hline
 & m_{2n-1} & \cdots & \cdots & \cdots & \cdots & m_1 & m_0
\end{array}
$$

The basic structures compute each summand a_ib_j once with an AND-gate, which is based on a reduced three-input majority gate on QCA and in the worst case erases two bits of information. This is accompanied with the accumulation of the summands with a full or serial adder, containing three majority gates each erasing two bits. Thus, as the number of summands is n^2 and for each of them eight bits are erased, the cost of worst case bit erasures per finished operation is $8n^2$. The hardware specific erasures are illustrated in Fig. 2 and the estimation models presented in Table II, based on the logic gate and running cycle analysis described in the following.

Array multiplier. The full array structure [5] has only the basic cost in information loss, since the paper-and-pencil algorithm is mapped directly to hardware with normally 100% utilization rate of the pipeline. This guarantees that the logic gates neither have idle cycles nor compute with dummy operands, which would lead to unnecessary logic activity.

The summand generation and accumulation erasures are present also in the serial-parallel multipliers [10]–[12], which multiplex the rows and partial products of the algorithm onto shared hardware, with the penalty of additional serialization costs estimated in the following. However, the basic cost is not directly included in the radix-4 recoded multiplier [13], which operates using a more complex algorithm.

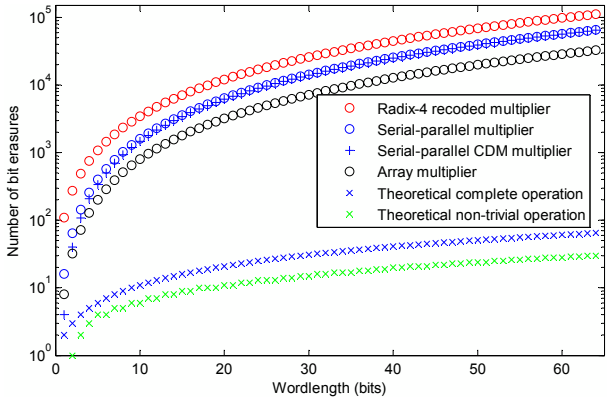


Fig. 2. Bit erasures in the proposed multiplier implementations.

V. Bit Erasures of Serialization

Serialization of the multiplication operation makes a trade-off between the computing performance and the circuit area, but there appears to be also a cost in information loss, due to underutilization of the pipeline or control of the algorithm.

Serial-parallel multipliers. The approach of collapsing one dimension of the algorithm into bit-serial operation leads to the necessity to feed dummy zero operands to inputs, for many cycles in the middle of computation, in order to avoid the corruption of pipeline data. This generates unwanted activity cycles for the logic gates, and can be seen as underutilization of the pipeline. The basic serial-parallel designs [11], [12] use a length n chain of AND-gates and adder slices, each pair losing eight bits per cycle. With the running time of $2n$ cycles per operation, the resulting total worst case erasures is $16n^2$. An improved carry delay multiplier (CDM) design in [10], [11] removes one of the slice adders and also $12n$ bit erasures.

Often both operands are available only in parallel format, and this requires the serial-parallel multipliers to utilize a data format converter. A simple design is included in [5], based on a chain of OR-gates. The $n-1$ gates each discard two bits of logical information, and during the runtime of $2n$ cycles, the additional loss is $4n^2 - 4n$ erasures.

Digit-serial multiplier. Recoded radix-4 modified Booth multiplier [13] represents a mid-point in the degree of parallelism of the proposed QCA implementations. While the partially digit-serial approach brings a performance gain compared to the bit-serial approach, the design is made unattractive by the huge complexity and area cost. The runtime per operation is limited to n cycles, but the number of gates active on each cycle is over $13n^2$. This leads to the worst case total information loss of $26n^2 + 86n - 2$ bits per multiplication.

Most of the radix-4 multiplier bit erasures occure in the multiplexer chain utilized to select the correct multiple of the multiplicand for summation. The muxes are responsible for about $20n^2$ erasures, which indicates that this would be a natural starting point for optimizations, since the simple logic could possibly be laid out in a way that helps to retain information, avoiding the worst case *per gate* loss.

TABLE II
Models for the worst case multiplier bit erasures.

Array multiplier [5]	$8n^2$
Serial-parallel [11], [12]	$16n^2$
Serial-parallel carry delay [10], [11]	$16n^2 - 12n$
Radix-4 recoded [13]	$26n^2 + 86n - 2$
**Optional parallel-serial converters [5]*	$4n^2 - 4n$

VI. Conclusion

The theory of complete binary multiplication requires a linear number of bit erasures, and without the zero operands, non-trivial multiplication has information loss determined by a sub-linear, logarithmic function of the operand wordlength. In contrast, all of the studied structures for the hardware implementation have the number of worst case bit erasures following a *square-law*, dependent on the wordlength. Based on this observation, the multipliers should yield to significant optimization for information loss. Of the QCA designs, the array multiplier performs best in this regard.

This work presented the worst case bounds for the implementations, but properly designed physical signal routing and layout would lead to higher level of retained information. Future work concentrates on gaining insight into how this could be achieved, developing deterministic models for the cost of improving both the logical and physical reversibility. Accurate models for the power density of the irreversible QCA layouts open a way to optimize the designs by taking into account the bit erasures, timing, and circuit area together.

Acknowledgment

This work was supported by the Academy of Finland under research grant 132869.

References

[1] International Technology Roadmap for Semiconductors. (2009) ITRS report. [Online]. Available: http://www.itrs.net/Links/2009ITRS/Home2009.htm

[2] C. Lent and P. Tougaw, "A device architecture for computing with quantum dots," *Proc. IEEE*, vol. 85, no. 4, pp. 541–557, Apr. 1997.

[3] J. Timler and C. Lent, "Maxwell's demon and quantum-dot cellular automata," *J. Appl. Phys.*, vol. 94, pp. 1050–1060, July 2003.

[4] I. Hänninen and J. Takala, "Binary adders on quantum-dot cellular automata," *J. Sig. Proc. Syst.*, vol. 58, no. 1, pp. 87–103, Jan. 2010.

[5] ——, "Binary multipliers on quantum-dot cellular automata," *Facta Universitatis*, vol. 20, no. 3, pp. 541–560, Dec. 2007. [Online]. Available: http://factaee.elfak.ni.ac.rs/fu2k73/15hanninen.html

[6] ——, "Irreversible bit erasures in binary adders," in *Proc. IEEE Conf. Nanotechnology*, Seoul, Republic of Korea, Aug. 17–20, 2010.

[7] R. W. Keyes and R. Landauer, "Minimal energy dissipation in logic," *IBM J. Res. Dev.*, vol. 14, no. 2, pp. 152–157, Mar. 1970.

[8] C. Bennett, "Logical reversibility of computation," *IBM J. Res. Dev.*, vol. 17, pp. 525–532, Nov. 1973.

[9] C. Lent, M. Liu, and Y. Lu, "Bennett clocking of quantum-dot cellular automata and the limits to binary logic scaling," *Nanotechnol.*, vol. 17, no. 16, pp. 4240–4251, Aug. 2006.

[10] H. Cho and E. Swartzlander, "Adder and multiplier design in quantum-dot cellular automata," *IEEE Trans. Comput.*, vol. 58, no. 6, pp. 721 –727, jun. 2009.

[11] ——, "Serial parallel multiplier design in quantum-dot cellular automata," in *Proc. IEEE Symp. Computer Arithmetic*, Montepellier, France, June 25–27, 2007, pp. 7–15.

[12] K. Walus, G. Jullien, and V. Dimitrow, "Computer arithmetic structures for quantum cellular automata," in *Rec. Asilomar Conf. Sign., Syst. Comp.*, Pacific Grove, CA, USA, Nov. 9–12, 2003, pp. 1435–1439.

[13] I. Hänninen and J. Takala, "Radix-4 recoded multiplier on quantum-dot cellular automata," in *Embedded Computer Systems: Architectures, Modeling, and Simulation*, ser. Lecture Notes in Computer Science, K. Bertels, N. Dimopoulos, C. Silvano, and S. Wong, Eds. Berlin/Heidelberg, Germany: Springer, 2009, vol. 5657, pp. 118–127.