# Using Neuromorphic Computing Methods for General Computer Performance Growth*

Erik P. DeBenedictis, *Member, IEEE*

***Abstract*—Neuromorphic computing is a leading option for reestablishing growth of the IT sector. This growth is often credited to Moore's Law, but it is just a prediction that incremental progress will continue until the physical limits are reached. These limits are now just a decade away, turning the physical limits into an argument that progress must end.**

**This paper shows that a careful reading of Landauer's seminal paper on physical limits reveals the community has generalized an AND-gate example beyond what he intended. This paper applies the same analysis to a carefully designed neuromorphic synapse and finds a maximum energy efficiency 100× higher than what is often claimed to be the limit.**

**This paper shows how to update ideas on theoretical maximum energy efficiency to neuromorphic computers in three steps, creating a theoretical framework for neuromorphic computing that can leverage CMOS infrastructure and continue a growth path like Moore's Law for longer.**

## I. Introduction

The computer industry grew exponentially for decades, establishing exponential growth in computer capability as the baseline for industry health. Roadmaps project semiconductor chips only have about another decade of scaling at the same power per chip, meaning per-chip computing capability must flat line. If neuromorphic, quantum, and a few other less-known computing approaches were better understood, they could possibly enable continuation of traditional growth rates.

In this paper, we will explain conflicting interpretations of the energy efficiency limits of computation. Are they fundamental or simply artifacts of the way to do things? This distinction could be a relevant because nature used an independent development path for neural systems, so their pertinent characteristics could be different.

The power dissipation limits of current computers were studied by Landauer [1], who stated in the abstract of his paper that there would be a "minimum heat generation, per machine cycle, typically of the order of $kT$ for each irreversible function." Using an AND gate as an example, his paper computes a minimum dissipation of .82 $kT$[1] using a table duplicated within the blue outline in Fig. 1A (annotations outside the blue outline are due to the current author). "Reversible" computing had not been invented at the time of [1], so the qualifier "irreversible" seems to have been ignored. The information just presented is apparently the justification for a widespread view that there is a minimum dissipation of ~$kT$ per use (clock cycle) of a binary logic gate. This interpretation is excessively narrow and misleading.

The body of Landauer's paper [1] contains other reasoning that is more sophisticated. The table in Fig. 1A is explained in the body text as an AND gate's transformation of input combinations to output combinations[2]. Each row $i$ represents an input combination that occurs with probability $p_i$ during the system's operation. The paper states that the example's inputs are in "thermodynamic equilibrium," which is a way of saying that all input combinations occur with equal probability (i. e. .125 for each of the eight rows in Fig. 1A—however, the probability markings in orange were not in the original paper). When a function maps multiple input combinations to the same output values, the rows merge into a single output state and dissipate a minimum energy based on the various $p_i$'s. Rows not participating in any merging do not contribute to dissipation—however, the probabilities in purple in Fig. 1A were not in the original paper. It is also notable that Landauer's analysis is for stateless gates; however the analysis method can be readily extended to state machines or memories.

Neuromorphic computing is based on synapses rather than AND gates, and they have different behaviors. Recent breakthroughs on artificial neural networks suggest learning is more computationally intensive than non-learning operation (called performance) [2]. So let us concentrate on the energy efficiency of learning. While learning is essential, most experiences do not cause a given synapse to change state. For example, most readers of this paper will have learned the English alphabet as a child. By now, there is nothing more to learn by seeing the letter "L" for the millionth time. However, seeing the letter "Л" may invoke learning and cause synapse changes for readers unfamiliar with the letter equivalent to "L" in Russian (Cyrillic).

The discussion above leads to an open door. Synapses have three characteristics that were covered in the body of Landauer's paper but are not part of the currently popular interpretation: Synapses are essentially (1) storage devices where learning is a (2) critical consumer of energy, but they change state (3) only infrequently. Perhaps these characteristics can be exploited to create a theory of neuromorphic system energy efficiency that has lower energy minimum than the ~$kT$ "limit" that appears in the literature.

*Approved for unclassified unlimited release SAND2016-6400C. Research supported by Sandia National Laboratories, a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Erik P. DeBenedictis is with Sandia National Laboratories, Center for Computing Research, Albuquerque, NM 87185-1319 USA (505-284-4017; fax: 505-845-7442; e-mail: epdeben@sandia.gov).

[1] Landauer's original paper had a math error and actually reported 1.18 $kT$. Landauer corrected this in a later paper.

[2] The circuit actually comprises a 2-input AND gate plus an extra wire that has no function, but the wire serves to create 8 input combinations.

| A. Landauer's paper figure 5 (AND gate plus wire) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | BEFORE CYCLE | | | | AFTER CYCLE | | | FINAL | | |
| Prob, | *p* | *q* | *r* | | *p* 1 | *q* 1 | *r* 1 | STATE | *S* i (k's) | *S* f (k's) |
| 0.125 | 1 | 1 | 1 | → | 1 | 1 | 1 | Alpha | 0.2599302 | 0.25993 |
| 0.125 | 1 | 1 | 0 | → | 0 | 0 | 1 | Beta | 0.2599302 | 0.25993 |
| 0.125 | 1 | 0 | 1 | → | 1 | 1 | 0 | Gamma | 0.2599302 | 0.367811 |
| 0.125 | 1 | 0 | 0 | → | 0 | 0 | 0 | Delta | 0.2599302 | 0.367811 |
| 0.125 | 0 | 1 | 1 | → | 1 | 1 | 0 | Gamma | 0.2599302 | 0 |
| 0.125 | 0 | 1 | 0 | → | 0 | 0 | 0 | Delta | 0.2599302 | 0 |
| 0.125 | 0 | 0 | 1 | → | 1 | 1 | 0 | Gamma | 0.2599302 | 0 |
| 0.125 | 0 | 0 | 0 | → | 0 | 0 | 0 | Delta | 0.2599302 | 0 |
| | | | | | | | | *S* f (k's) | 2.0794415 | 1.2554823 |
| B. Synapse Example | | | | | | | | *S* i-*S* f (k's) | | 0.8239592 |
| probability of a learning event: | | | | | | | | | **0.001** | |
| | cause | effect | field dir. | | cause | effect | field dir. | State | *S* i (k's) | *S* f (k's) |
| 0.0624375 | -1 | -1 | -1 | → | -1 | -1 | -1 | A | 0.173176 | 0 |
| 0.0624375 | -1 | 0 | -1 | → | -1 | 0 | -1 | B1 | 0.173176 | 0.173176 |
| 0.0624375 | -1 | 1 | -1 | → | -1 | 1 | -1 | C1 | 0.173176 | 0.173176 |
| 0.0624375 | 0 | -1 | -1 | → | 0 | -1 | -1 | D1 | 0.173176 | 0.173176 |
| 0.0624375 | 0 | 0 | -1 | → | 0 | 0 | -1 | E1 | 0.173176 | 0.173176 |
| 0.0624375 | 0 | 1 | -1 | → | 0 | 1 | -1 | F2 | 0.173176 | 0.173176 |
| 0.0624375 | 1 | -1 | -1 | → | 1 | -1 | -1 | G1 | 0.173176 | 0.173176 |
| 0.0624375 | 1 | 0 | -1 | → | 1 | 0 | -1 | H1 | 0.173176 | 0.173176 |
| 0.0005 | 1 | 1 | -1 | → | 1 | 1 | 1 | I | 0.0038005 | 0.1740608 |
| 0.0005 | -1 | -1 | 1 | → | -1 | -1 | -1 | A | 0.0038005 | 0.1740608 |
| 0.0624375 | -1 | 0 | 1 | → | -1 | 0 | 1 | B2 | 0.173176 | 0.173176 |
| 0.0624375 | -1 | 1 | 1 | → | -1 | 1 | 1 | C2 | 0.173176 | 0.173176 |
| 0.0624375 | 0 | -1 | 1 | → | 0 | -1 | 1 | D2 | 0.173176 | 0.173176 |
| 0.0624375 | 0 | 0 | 1 | → | 0 | 0 | 1 | E2 | 0.173176 | 0.173176 |
| 0.0624375 | 0 | 1 | 1 | → | 0 | 1 | 1 | F2 | 0.173176 | 0.173176 |
| 0.0624375 | 1 | -1 | 1 | → | 1 | -1 | 1 | G2 | 0.173176 | 0.173176 |
| 0.0624375 | 1 | 0 | 1 | → | 1 | 0 | 1 | H2 | 0.173176 | 0.173176 |
| 0.0624375 | 1 | 1 | 1 | → | 1 | 1 | 1 | I | 0.173176 | 0 |
| | | | | | | | | *S* f (k's) | 2.7784165 | 2.7725852 |
| | | | | | | | | **Si-Sf (k's)** | | **0.005831** |

Such a theory would not invalidate the statement in the abstract of Landauer's paper, since artificial synapses were not typical in the 1960s.

## II. The Minimum Dissipation of a Synapse

We will repeat the procedure in [1] but replacing the AND gate with an artificial synapse of sorts. A synapse changes state during learning when a "cause" is different from the desired "effect." There are different learning algorithms, but an old-style magnetic core memory has the necessary behavior. In such an implementation, the row conductors would be the cause, the column conductors would be the effect, and the magnetic field direction in the cores would be the learned behavior. Cause and effect would each be turned into a current of magnitude 0 or ±1, with the core flipping at a current of ±1.5 through its center in the same units. If cause and effect are the same direction, a current of ±2 would go through the center and cause a flip—but only if the core was not already in the correct state. When cause and effect are different (including either or both being 0), the maximum total current through the core will be too small to cause a flip. A core's flipping is an energetic event and causes energy dissipation, but a fluctuating current through a core dissipates very little energy otherwise.

Figure 1B is in the same form as the AND gate in [1]. However, the table is actually derived from an Excel spreadsheet where the green parameter labeled "probability of a learning event" triggers a sequence of computations. The parameter in green is the probability of a state change in the synapse being analyzed during one "machine cycle." This parameter is part of the problem definition and relates (per previous discussion) to the probability of an unexpected event in the input data, such as a "JI." The spreadsheet distributes this probability (.001) equally between the two states in the center of the chart (.0005). The remaining probability (.999) is equally distributed across the other states. To model the behavior of the core, only states A and I merge.

In accordance with Landauer's method, the spreadsheet computes initial entropy ($S_i$), final entropy ($S_f$), and the minimum dissipation as their difference. The computed value (.005831) is not fundamental; almost any dissipation is possible by changing the value in green.

## III. Three Changes Needed

The previous section analyzed a system that yielded a minimum energy below the widely believed ~$kT$ limit, but is there a way to synthesize systems with this property on demand? The basic process and its implications are described below, with more detail in [3].

The actual process for creating the core-based synapse model involved the author reverse-engineering Landauer's method of computing minimum energy and figuring out how to synthesize systems that would have an unusually low minimum energy. The example in Fig. 1B also involved searching for a computational primitive useful in the learning phase of neuromorphic computing.

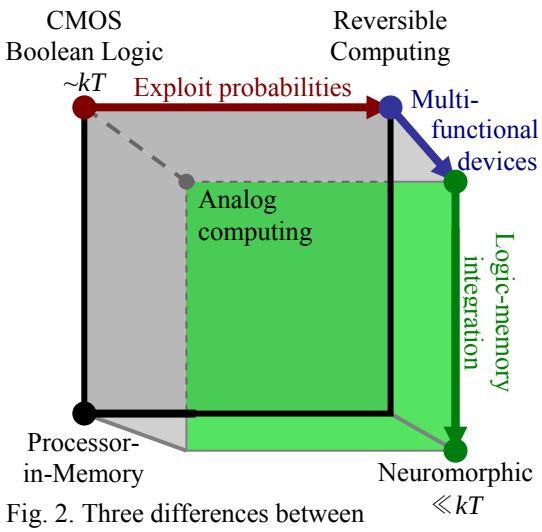


Fig. 2. Three differences between CMOS and neuromorphic energy efficiency

It appears three changes in computer design will be required to systematically create systems with the energy efficiency of Fig. 1B. These are illustrated as a cube in Fig. 2. The cube is not intended to represent a geometric structure, but rather a series of ideas related in a way that has the same topology as vertices and edges on a cube. Starting with CMOS and Boolean logic on the upper left, each dimension of the cube represents a design change. Traversing all three dimensions in any order gets to the lower right corner that produced the low minimum energy in Fig. 1B. Interestingly, traversing just one or two dimensions sometimes leads to a known computing approach (e. g. reversible computing, processor-in-memory, and analog computing).

One design change would be to exploit probabilities in data and in the design of logic. All gates in today's CMOS have the same schematic diagram, notably complementary pull-up and pull-down trees. If it is known that one bit possibility occurs more often than another [1], the design of a gate can be customized to the circumstance and yield a lower dissipation limit. This idea is actually in Landauer's paper [1], yet is a topic of continued research area [3] [4].

This idea was exploited in the design of the synapse used in Fig. 1B by arranging for the most common data inputs to result in no dissipation. Specifically, a synapse mostly verifies that it has learned what it needs to know as opposed to actually changing state. Fig. 1B maps these most common inputs into the rows of the chart that do not merge and hence do not have dissipation.

A second design change is to try and discover physical devices that execute higher-level functions in one step. Boolean logic is universal and hence can compute anything, but it is almost never true that a group of maximally energy-efficient Boolean logic gates will compute a function with maximum energy efficiency. This is because each logic gate cleans up signal noise through overdriven amplification and clipping. This restorative process essentially forces the minimum dissipation analysis of Fig 1 to be executed once per gate and the dissipations added. Since the magnetic core example in figure 1B performs a function equivalent to four NAND gates, a Boolean logic implementation should have a minimum of ~$kT$ dissipation.

The open door is to engage in physical science research to find devices with new functions. This might allow creating an inventory of devices that could be combined in many ways. Alternatively, some important functions could be identified and then physical science research would seek the device.

Another idea is to tightly integrate logic and memory. The ideas above may not be helpful to designers intending to create the processor portion of a von Neumann-style computer. Today's logic design style assumes memory is concentrated in a specific subsystem. The user gets happier as the number of devices in the memory subsystem increases, because this means more memory and the computer becomes more useful. However, the designer is supposed to minimize the number of devices in the processor portion because these devices tend to be larger, more expensive, and consume more energy. Logic with highly skewed probabilities of 0s and 1s are wasteful and the designer would be encouraged to redesign irrespective of any discussion in this paper.

However, synapses in biological neural networks both contain state (i. e. they are memory) and perform some logic. As Fig 1B shows, integrating these functions allows much lower minimum energy. If the data was stored elsewhere, it would have to be moved with the consequent increase in energy just for the movement.

## IV. Conclusions

Moore's Law was never a physical law, but could be considered a statement of optimism whose believability depended on its not violating physical law. It is difficult to imagine the optimism continuing with physical limits looming just a decade ahead. Alternative computing paradigms should not be able to beat the physical limits either, if they are real limits. We show in this paper that the popular interpretation of computational energy-efficiency limits became tied to the way we do things, such as using Boolean logic and the von Neumann architecture.

If we reapply theory on the limits of computation to society's evolving expectations of computers instead of historical artifacts, we find the limits to be further out. The desire for more arithmetic prowess is giving way in some user communities to interest in machines that learn—such as learning to drive cars or learning our preferences on what consumer products to buy from online retailers.

This gives theoretical support for continuing R&D in neuromorphic computing. This paper specifically shows how integrated logic and memory can improve energy efficiency by reducing energy-consuming communications, creating a role for new physical devices with functional diversity beyond just switches and transistors, and shows how these systems could be well matched to artificial neural networks or neuromorphic computing.

## Acknowledgment

This work has been significantly extended already due to collaboration with Michael P. Frank, Natesh Ganesh, and Neal G. Anderson. These extensions have been put into a paper [3], due for publication in October.

## REFERENCES


[1] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM journal of research and development* 5.3 (1961): 183-191.

[2] D. Silver, et al., "Mastering the game of Go with deep neural networks and tree search," Nature 529.7587 (2016): 484-489.

[3] E. DeBenedictis, M.P. Frank, N. Ganesh, N.G. Anderson, "A Path Toward Ultra-Low-Energy Computing," accepted to International Conference on Rebooting Computing, October 2016.

[4] P. Zulkowski and M. DeWeese, "Optimal finite-time erasure of a classical bit," Physical Review E 89.5 (2014): 052140.