Energy Efficiency Limits of Logic and Memory

Sapan Agarwal[1], Jeanine Cook[2*], Erik DeBenedictis[2], Michael P. Frank[2]
[1]Microsystems Science and Technology
[2]Center for Computing Research
Sandia National Laboratories
Albuquerque, NM, USA
[*]corresponding author jeacook@sandia.gov

Gert Cauwenberghs[3]; Sriseshan Srikanth[4], Bobin Deng[4], Eric R. Hein[4], Paul G. Rabbat[4], Thomas M. Conte[4];
[3]Department of Bioengineering, Jacobs School of Engineering, and Institute for Neural Computation, University of California San Diego, La Jolla, CA,
[4] Schools of CS and ECE
Georgia Institute of Technology, Atlanta GA

***Abstract*—We address practical limits of energy efficiency scaling for logic and memory. Scaling of logic will end with unreliable operation, making computers probabilistic as a side effect. The errors can be corrected or tolerated, but overhead will increase with further scaling. We address the tradeoff between scaling and error correction that yields minimum energy per operation, finding new error correction methods with energy consumption limits about 2× below current approaches. The maximum energy efficiency for memory depends on several other factors. Adiabatic and reversible methods applied to logic have promise, but overheads have precluded practical use. However, the regular array structure of memory arrays tends to reduce overhead and makes adiabatic memory a viable option. This paper reports an adiabatic memory that has been tested at about 85× improvement over standard designs for energy efficiency. Combining these approaches could set energy efficiency expectations for processor-in-memory computing systems.**

Keywords—Moore's Law, Shannon, Landauer, limits of computing, adiabatic, reversible, reversible logic, millivolt switch

I. INTRODUCTION

We address an apparently novel tradeoff between two well-known issues. Semiconductor scaling is widely known to reduce energy consumption today, but it will eventually lead to a rise in errors due to insufficient energy to distinguish between 0s and 1s. Algorithm-Based Fault Tolerance (ABFT) and error correction are well-known methods that allow logic and memory to function in the presence errors, albeit with progressively more overhead as the error rate rises. We discuss how continued scaling will initially reduce energy consumption, but scaling beyond an optimal point will cause energy to increase again due to the overhead of handling the errors. This paper finds two error correction methods that reduce minimum energy consumption for logic, yet finds a different approach that is more suitable to memory.

When Moore's Law was formulated in the 1960s [1], it projected practical, manufacturable semiconductor technology from that point in time into the future. In the same decade, theorists identified energy efficiency limits for computer technology [2] that were unimaginably far ahead of the technology of the day. The big gap led to 50 years of exponential growth. Now in the 2010s, we find the energy efficiency of manufactured devices being in the range of 10× to 10,000× above the theoretical limits.

Device size is approaching theoretical limits as well. However, the expected change from 2D to 3D manufacturing will allow module-level density to rise and further exacerbate energy efficiency challenges.

Each 2× scaling generation enables new products for a few years in the huge global information technologies (IT) sector. Given the stakes, it would be useful to find the endpoint of scaling more accurately than just 10×-10,000× beyond where we are now.

The best known work on minimum dissipation is due to Landauer [2], who stated that the minimum energy is typically on the order of kT for each irreversible logic operation. The expression $kT \approx 4\times10^{-21}$ joules at room temperature comprises Boltzmann's constant k times the absolute temperature T. Landauer's minimum is a very solid lower bound yet often misinterpreted; another paper at this conference analyzes this issue [3].

Today's Complementary Metal-Oxide Semiconductor (CMOS) can never reach Landauer's minimum, but it is important to know how close it can come. CMOS is a term that denotes both a complementary pull-up/pull-down logic circuit and the MOSFET transistor. The CMOS circuit design is very simple, but the circuit's simplicity forces the charging and discharging of capacitors defining signal values directly from DC power supplies. The simple charging circuit limits energy efficiency to what is called the Landauer-Shannon limit [4] that is about 50× higher than Landauer's minimum.

However, the transistors have issues as well. Roadmaps for transistors project additional reduction in power supply voltage for a device class called millivolt switches. This phrase refers to devices that could operate in CMOS-like circuits below the limits of MOSFET devices. MOSFETs are limited to $\ln(10)\ kT/q \approx 60$ mV/decade sub threshold slope, which prevents practical operation below about 0.5 V. While a specific MOSFET replacement with steeper slope has not been selected for full-scale development, Tunnel FETs and

Approved for unclassified unlimited release SAND2016-7245 C. 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

978-1-5090-1370-8/16/$31.00 ©2016 IEEE

Piezotronic FETs are candidates. If scaling continues after these devices go into production, technology should be able to reach the Landauer-Shannon limit (and we argue it can go further).

The theoretical minimum energy for CMOS Boolean logic has been studied by considering the wires between logic gates as communications links and applying Shannon's communications theory, but we will argue this does not find the true minimum. Just as a cell phone picks up more and more static as it moves further from the transmitter tower, the theoretical minimum energy of a Boolean Logic gate is not a single number but a function of the acceptable error probability. The earliest analysis the authors can find [4] yields the familiar expression $P_e = \exp(-E_{signal} / kT)$ [1]. Modern textbooks [5, p. 595] sometimes use the asymptotically equivalent complementary error function (erfc). The erfc version appears in [6] as applicable to minimum energy for CMOS, however Theis and Solomon follow their analysis with the statement "[a]s thermal voltage fluctuations become significant, we must incorporate redundancy and error correction in the logic to keep the error rate in bounds" [6].

There is about a 50× "end zone" to energy scaling that this paper begins to address with error correction. CMOS circuits made of extremely good millivolt switches should be able to reach the Landauer-Shannon limit of $P_e = \exp(-E_{signal} / kT)$, but clever circuits will be needed to get closer to Landauer's minimum of kT. The historical literature contains (unimplemented) examples of designs [4] [7], supporting the idea that such approaches are possible and we find some here.

There are methods of correcting gate errors without concern to energy consumption. For example, Triple Modular Redundancy (TMR) [8] votes the results of three redundant calculations and uses the winner as the answer. This is effective, but more than triples the gate count and hence energy. Another class of techniques is called Algorithm-Based Fault Tolerance (ABFT) [9], which typically performs a full calculation and an estimate. For example, the estimate might be just the least significant bit of the full computation. The calculation is repeated if the two do not match. ABFT may have low overhead, but the versions in the literature are specific to just one algorithm rather than applying generally.

To the knowledge of the authors, this is the first paper to connect the exponential error probability $P_e = \exp(-E_{signal} / kT)$ with error correction overhead. For example, cutting signal energy in half ($E_{signal}' = E_{signal}/2$) will cut gate energy in half but change $P_e' = \sqrt{P_e}$. If error correction can square the error probability for less than a factor of two in overhead, the energy efficiency can rise above the Landauer-Shannon limit.

Reversible computing [10] is a second approach to beating energy efficiency limits, yet it faced practical limitations in the past. Reversible computing can use existing MOSFET transistors, yet uses different circuits that recycle energy. Readers are referred to [11] for extensive additional details, but the principle for energy efficiency scaling is to recycle a rising fraction of the energy as the technology improves. One generation might recycle 99% of operating energy, drawing only the remaining 1% from the power supply. A few generations later, 99.9% might be recycled with 0.1% drawn from the power supply. And so forth.

There have been challenges in applying reversible computing to logic, but there is progress in the area including a paper at this conference by Snider et. al. [12]. The circuitry needed to recycle energy is more complicated than conventional Boolean logic, using more transistors in some approaches and using large numbers of clock signals in others. The cost to power a processor over its lifetime has recently started to exceed the purchase cost, making more complexity a good investment if it can lower energy consumption.

Reversible computing principles can be applied to memory as well, which is being reported in this paper. Complexity has different meanings for memory and logic. Memories are divided into addressing logic and a memory storage array. While smaller storage cells are preferred, the user wants the largest possible array for a given storage cell size. This is because the array holds the data that is valuable to the user. As long as the addressing logic is small compared to the memory array, the user will not care about its complexity. Logic associated with memory addressing is a large consumer of energy in current computer systems, so making the addressing more energy efficient is a priority as long as it does not increase complexity very much.

In section IV, we report on a memory using adiabatic principles that has been measured as reducing energy by 85× (i. e. recovers a fraction 84/85 of the delivered energy drawn from the power supply) owing to resonant energy exchange.

II. Redundant Residue Number Systems

This section shows that an RRNS-based processor can exceed the energy efficiency predicted by the Landauer-Shannon limit. A companion paper in these same proceedings [13] describes the Computationally-Redundant Energy-Efficient Processing for Y'all (CREEPY) architecture. CREEPY uses a n=4 sub cores to represent ~32-bit numbers using a Residue Number System (RNS), with one residue per sub core. CREEPY also has r=2 additional redundant sub cores that extend the RNS into a Redundant RNS (RRNS) and allow detection and correction of a single logic error. The RRNS was developed in [14], but that paper did not consider energy efficiency.

CREEPY (or any RRNS processor) can be used as a baseline for comparisons by ignoring both the energy consumption of the redundant sub cores and their ability to correct errors, a method developed in [15] for general circuits. The energy efficiency of the baseline can be improved by reducing E_{signal} and increasing the energy efficiency of the underlying gates up to the Landauer-Shannon limit in [4] and [5, p. 595], as described above. Any single error that occurs with probability $P_e = \exp(-E_{signal} / kT)$ per gate operation would cause a system failure. However, [6] suggests that incorporation of redundancy and error correction might be helpful to further increase energy efficiency.

[1] The Landauer-Shannon limit is usually written as $E_{signal} = kT \ln(1/P_e)$

CREEPY can then model redundancy and error correction by including the energy consumption of the redundant sub cores and assuming they will correct single errors. In this case, a system failure occurs only when two or more errors occur within a time window. The analysis below shows that redundancy and error correction can help. If E_{signal} is lowered the precise amount that keeps overall system energy unchanged given the additional gates, the probability of a system failure declines—at least in some useful operating ranges. If the baseline was operating at the Landauer-Shannon limit, the RRNS version would operate below the limit.

Consider the baseline system where each of n residues is implemented by G gates, so the baseline comprises $N = Gn$ gates. While the baseline does not check or correct errors, we will derive the error probability on batches of f_n sequential arithmetic operations, using notation consistent with [13]. Since all errors will be undetected, the probability of an undetected error per batch as a function of E_{signal} is

$$P_u(E_{signal}) = Nf_n \exp(-E_{signal} / kT). \quad (1)$$

Now consider the additional r redundant residues for a total of $t = n+r$ residues, and an additional $R = Gr$ gates. This RRNS circuit will be distinguished by primes (') and operated with a signal energy E_{signal}'. The probability of an undetectable double error in a batch will be

$$P_u'(E_{signal}') = \tfrac{1}{2}\, t(t-1)\, (G f_n)^2 \exp(-2E_{signal}' / kT), \quad (2)$$

which are the $\tfrac{1}{2}\, t(t-1)$ combinations of two residues being in error multiplied by the square of the probability of each residue being in error over the time of an entire batch. We are not detecting or correcting errors at this point; errors just become inconsistent encodings of the RRNS residues.

Assume the single error detection and correction is performed once for each batch of f_n operations by a circuit comprising C gates. We will not model the gates explicitly, so C will be an equivalent value.

The two circuits will consume the same total energy if we set

$$E_{signal}' = N / (N+R+C/f_n)\, E_{signal}. \quad (3)$$

The ratio of the two error probabilities above form a figure of merit $\mathcal{M}$ if the probabilities are computed at constant total energy, which can be the result of adjusting the signal energies of the two circuits to match as in (3)

$$\mathcal{M} = \frac{P_u(E_{signal})}{P_u'(E_{signal}')}. \quad (4)$$

The figure of merit can be expressed either as a function of E_{signal}' or E_{signal}. We choose E_{signal}. If we designate $\beta = \tfrac{1}{2}\, t(t-1)/n$ as an RRNS property and $G^* = N + R + C/f_n$, (4) becomes

$$\mathcal{M} = \frac{Nf_n \exp(-E_{signal} / kT)}{\beta\, n (G f_n)^2 \exp(-2N/G^*\, E_{signal} / kT)}. \quad (5)$$

Now (5) simplifies to

$$\mathcal{M} = 1/(\beta\ Gf_n) \exp((2N-G^*)/G^*\, E_{signal} / kT). \quad (6)$$

The error detection yields benefit when $\mathcal{M}$ is greater than 1, or beyond the break even point. Shifting to the inequality and taking the logarithm of both sides yields

$$0 < \ln(1/(\beta\ Gf_n)) + (2N-G^*)/G^*\, E_{signal} / kT, \quad (7)$$

which can be solved for signal energy in units of kT as

$$E_{signal} / kT > \ln(\beta\ Gf_n)\, G^*/(2N-G^*), \text{ or} \quad (8)$$

$$E_{signal}' / kT > \ln(\beta\ Gf_n)\, N/(2N-G^*). \quad (9)$$

Let us explore the range of situations where RRNS is helpful in raising energy efficiency. Assume the number of logic gates in a residue calculation is $G = 2000$, a number used in [13]. The example number system from [14] used in [13] uses n=4, r=2, and therefore $\beta = 3.75$, $N = Gn = 8{,}000$, and $R = Gr = 4{,}000$. From an inspection of diagrams in [14], let us assume detection and correction is equivalent to 3 arithmetic operations or 12 residue operations. This implies $C = 12G = 24{,}000$.

The spreadsheet in Table I shows RRNS can raise energy efficiency as long as the signal energies are above the break even point. As long as f_n is more than 10 or so, the break even signal energies are below anything useful in a design, so the break even point is not an obstacle. Even though Table I establishes the boundary where energy efficiency rises, the specific numbers in Table 1 are the point where RRNS makes precisely no difference.

Based on the $f_n = 100$ reliability analysis in the companion paper [13], we conclude RRNS gives benefit in at least one

Parameters				Break even	
f_n	N	R	C	e_{signal} / kT	e_{signal}' / kT
7	8,000	4,000	24,000	293.45	152.16
8	8,000	4,000	24,000	165.03	88.02
9	8,000	4,000	24,000	122.32	66.72
10	8,000	4,000	24,000	101.03	56.13
11	8,000	4,000	24,000	88.30	49.81
12	8,000	4,000	24,000	79.85	45.63
25	8,000	4,000	24,000	51.76	31.95
50	8,000	4,000	24,000	45.50	29.17
100	8,000	4,000	24,000	44.04	28.78
G	n	r	const	$\exists$	
2000	4	2	12	3.75	

TABLE I. RRNS BREAK EVEN

example situation. The scenario above as analyzed in [13] shows a sharp increase in reliability between $E_{signal}' = 42$ and 43 kT. Table I shows the break even point for $f_n = 100$ to be $E_{signal}' = 28.78\ kT$, which is lower.

A short discussion may be in order on how to extend the approach. The Shannon-Landauer "limit" has merit, yet we found an exploitable property. Scaling causes a linear reduction in energy consumption but an exponential rise in raw error rate. We searched for a low-overhead error-correction approach and checked to see if there could be a net savings in energy before the exponential dominated. The steepness of the exponential was crucial to how far we could push, e. g. we should have been able to push further with a gentler polynomial but a step function would have been impenetrable. Simply reducing the supply voltage to a semiconductor circuit causes it to fail uncontrollably, which is like a step function. Thus the onset of thermal noise is a special case not expected to be seen before the end of scaling. The authors have no idea how far this could go; it may be the equivalent of one semiconductor generation.

III. Temporal Error Correction

The authors propose a second form of error correction using samples collected over time, which also reduces minimum energy below the Landauer-Shannon limit. Typically, the wire from one gate's output to another's input is modeled as a communications channel. This is shown in Fig. 1A where a wire connects a driving gate through a hypothetical switch to the combined capacitance C of the wire and the next gate's input [6]. The capacitance is charged and then the switch opened. The disconnection leaves a reset noise of voltage kT/C on the input of the gate being analyzed, meeting the Landauer-Shannon limit of $P_e = \exp(-E_{signal} / kT)$ on the gate's output.

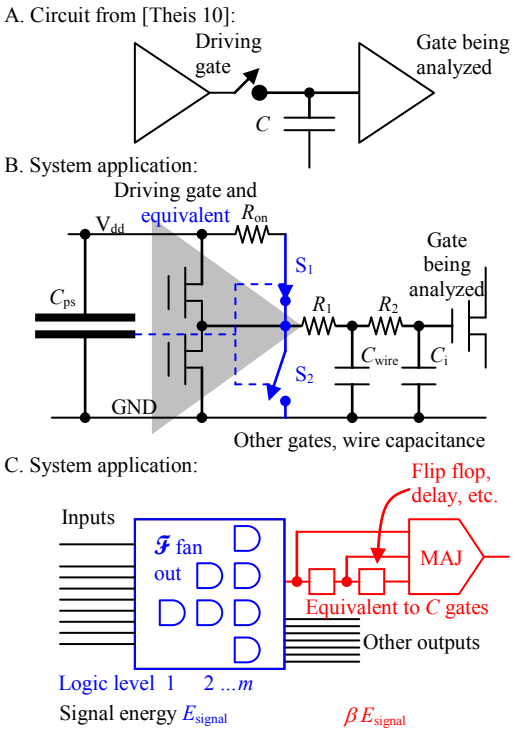

Fig. 1. Temporal error correction

Now imagine that the switch in Fig. 1A is thrown back and forth several times. Since the capacitance would charge to its proper value on the first cycle, later cycles would not consume additional energy. However, the analysis in [6] would be independently valid on each cycle, yielding the same data value but a different sample of the random reset noise. Applying error correction to multiple samples could reduce the error probability—or keep the error probability unchanged while reducing the gate energy. We show an error correction method below that allows more energy to be saved in the logic than is consumed by error correction. The switch is extraneous; it can be left closed all the time or replaced by a wire.

This does not violate the communications theory analysis as the wire between stages will also conduct noise from the receiver's gate backwards though the wire and ultimately to the output gate's power supply. The equivalent circuit model is shown in Fig. 1B in enough detail to convey the basic idea. If R_{on}, R_1, and R_2 are large, it should be clear that the input of the gate being analyzed will have noise voltage kT/C_i. If R_{on}, R_1, and R_2 are small or shorts, they connect the three capacitors C_{ps}, C_{wire}, and C_i in parallel and the noise voltage drops to $kT / (C_{ps}+C_{wire}+C_i)$. The R's will never be 0 or ∞ ohms in practice, but this effect seems not to have been considered before.

This type of error correction will need to be applied to multiple levels of logic to amortize the overhead of the error correction circuit. Fig. 1C shows this method being applied to a field of gates in blue organized as a rectangle with m logic levels and n rows, each row comprising an input gate, $m-2$ intermediate gates, and an output. We assume the signal from any given input will affect more than one output, fanning out by a factor $\mathcal{F}$ as it goes from one logic level to the next. Similarly, an output will be controlled by multiple inputs, with a fan-in factor of $\mathcal{F}$ at each level. Each of the n outputs will need to have its own error correction circuit and so we will consider the energy of one row at a time. Data applied to the inputs is transformed to output data in m time steps equal to the propagation delay of a gate, with no error correction between logic levels.

Fig. 1C represents both the baseline and error corrected cases. The baseline circuit comprises only the blue graphics, which are operated at signal energy E_{signal}, using the same terminology as section II. The error corrected case includes the blue graphics operating at a reduced signal energy E_{signal}' followed by the error correction circuitry in red operating at the original signal energy E_{signal}. We define $E_{signal} = \beta E_{signal}'$.

The example in Fig. 1C uses majority voting over three samples for error correction, but the analysis below uses majority voting of α samples, $\alpha \geq 3$ and odd. The inputs must remain stable for least α steps so the outputs can be sampled at the end of steps m, $m+1 \ldots m+\alpha-1$. The propagation delay time is approximately the same as the minimum time needed to get independent error samples. The samples could also be taken by using a single device with a low-pass filter (large capacitance) that averages the signal over multiple propagation delay times.

A. Error rate P_u of a row of the baseline circuit

For both error rate and energy calculations, let thermal noise cause a gate to generate an erroneous signal with probability p. The signal will then propagate to the right, spreading to additional rows by a factor of $\mathcal{F}$ each time it moves from one layer to the next. This means the uncorrected error rate from the rightmost blue gate will be the probability of error in a given gate times number of parent gates, $\mathcal{N}$, that fan in to an output:

$$p_{\text{raw}}(p) = p\,\mathcal{N} \tag{11}$$

where:

$$\mathcal{N} = (\mathcal{F}^{m-1} + \ldots\ \mathcal{F}^2 + \mathcal{F} + 1) = (\mathcal{F}^m - 1)/(\mathcal{F} - 1). \tag{12}$$

For the baseline error calculation with no error correction, $p = \exp(-E_{\text{signal}} / kT)$ and

$$P_{\text{u}}(E_{\text{signal}}) = \mathcal{N} \exp(-E_{\text{signal}} / kT) \tag{13}$$

B. Error rate of a row of the error-corrected circuit (P_u')

For the calculation with error correction, an undetected error results when a majority of the α samples of the blue logic circuitry are erroneous. However, an undetected error could also result from an error in the red error correction circuit itself. We will call this probability q. The probability of an undetected error is then:

$$\begin{aligned} P_{\text{u}}'\left(E_{\text{signal}}'\right) &= \sum_{x=\frac{\alpha+1}{2}}^{\alpha} \mathbf{C}(\alpha, x) \times \left(e^{-\frac{E_{\text{signal}}'}{kT}}\mathcal{N}\right)^{x} \times \left(1 - e^{-\frac{E_{\text{signal}}'}{kT}}\mathcal{N}\right)^{1-x} + q \\ &\approx \mathbf{C}\left(\alpha, \frac{\alpha+1}{2}\right) \times \left(e^{-\frac{E_{\text{signal}}'}{kT}}\mathcal{N}\right)^{\frac{\alpha+1}{2}} + q \end{aligned} \tag{14}$$

where $\mathbf{C}(\alpha, x)$ is the number of ways to choose x of the α samples.

The error correction circuit is shown in red in Fig. 1, comprises of two latches and a majority gate. Let us model the circuit as C gates per sample α, where the energy per gate is βE_{signal}'. The probability of an error occurring in the correction circuit itself is:

$$q = C\alpha \times e^{\frac{-\beta E_{\text{signal}}'}{kT}} \tag{15}$$

So the undetected error rate is

$$P_{\text{u}}'\left(E_{\text{signal}}'\right) \approx \mathbf{C}\left(\alpha, \frac{\alpha+1}{2}\right) \times \left(e^{-\frac{E_{\text{signal}}'}{kT}}\mathcal{N}\right)^{\frac{\alpha+1}{2}} + C\alpha \times e^{\frac{-\beta E_{\text{signal}}'}{kT}} \tag{16}$$

We can find the error corrected signal energy, E_{signal}', required to get the same error rate as the baseline circuit by setting (13) equal to (16):

$$E_{\text{signal}}' \approx \frac{2}{\alpha+1} E_{\text{signal}} + kT\,\frac{\alpha-1}{\alpha+1}\ln(\mathcal{N}) + kT\,\frac{2}{\alpha+1}\ln\left[\mathbf{C}\left(\alpha, \frac{\alpha+1}{2}\right)\right] \tag{17}$$

Here we assumed the number of error correction gates is much less than the number of gates in the logic, $C\alpha \ll \mathcal{N}$, and simplifed the result.

C. Energy E' of a row of the error corrected circuit

To calculate the total energy used in a row of error-corrected logic, we need to add the energy consumption of the baseline circuit (blue only in Fig. 1) to compute the correct result, E_0', the energy of the error correction circuit, E_C', and the energy due to errors, E_x'.

The energy of the logic is given by the signal energy E_{signal}' times the number of gates in a row:

$$E_0' = m\,E_{\text{signal}}'. \tag{18}$$

The energy for the error correction circuit is the number of error correction gates times βE_{signal}':

$$E_C' = C \times \alpha \times \beta \times E_{\text{signal}}' \tag{19}$$

Next, consider the energy drawn from the power supply by an ideal CMOS circuit due to thermal errors, E_x' in a single row. Let us assume a thermal error occurs with probability p. Errors in gates on the left side of the network will each propagate to $\mathcal{F}$ rows as they go from layer to layer until they reach level m. An error takes twice the signal energy (an error signal and then a return signal) from the power supply at each level. The number of errors per logic level in one row of the blue gates in Fig. 1C will be:

$$p,\ p(\mathcal{F}+1),\ p(\mathcal{F}^2+\mathcal{F}+1),\ \ldots\ p(\mathcal{F}^{m-1}+\mathcal{F}^{m-2}\ldots 1). \tag{20}$$

The series above can be summed and simplified, yielding an expression for the number of errors in the gates of Fig. 1C per row.

$$N_{\text{err}} = p\,\mathcal{E} = p\,[(\mathcal{F}^m - 1)/(\mathcal{F} - 1)^2 - (m+1)/(\mathcal{F} - 1)], \tag{21}$$

which defines $\mathcal{E}$ as a constant related only to logical circuit structure. The energy due to errors is given by:

$$E_x' = 2\,\alpha\,p\,\mathcal{E}\,E_{signal}' \tag{22}$$

where 2α originates from 2 signal transitions per error times α samples. Thus the total switching energy is:

$$E' = E_0' + E_C' + E_x' = (m + C\times\alpha\times\beta + 2\,\alpha\,p\,\mathcal{E})\times E_{signal}' \quad (23)$$

D. Energy of a row of the baseline circuit (E)

The energy consumption of the baseline circuit (blue only in Fig. 1) will be the signal energy to compute the correct result, E_0, assuming no errors plus the energy, E_x, due to errors. Clearly,

$$E_0 = m\,E_{signal} \quad (24)$$

and in the baseline case, $p = \exp(-E_{signal} / kT)$, so

$$E_x = E_{signal}\,exp(-E_{signal} / kT)\,\mathcal{E} \quad (25)$$

with the sum of equations (24) and (25) being the energy of the baseline, uncorrected, circuit.

Fig. 2 plots energies for circuit representative of a 16×16 bit multiplier with three samples for error correction. The circuit is modeled by m=48 layers of logic with $\mathcal{F} = 32^{1/m}$ such that an error propagates to at most 32 outputs (which is all the outputs that exist on the multiplier). The error correction circuit is modeled by C=α gates. The horizontal axis is the baseline signal energy; an engineer would assess end-user requirements and pick a signal energy for an uncorrected circuit on the basis of $p_{error} = \exp(-E_{signal} / kT)$. For example, this signal energy might be 60-100 kT for a supercomputer but only 40 kT for a consumer device.

The top magenta (E) and dark blue (E') curves represent system energy without and with error correction respectively. The magenta curve for E is represents the exponential Landauer-Shannon limit, although accounting for error propagation. Error correction allows signal energy to be reduced at the expense of overhead for the error correction logic. The dark blue curve (E') shows total system energy at constant error rate. The blue curve is lower on the right of the graph, asymptotically approaching about 2:1 reduction in energy for the 7-input gate case. The error correction stops working below 15 kT or so, meaning the error-corrected circuit consumes more energy for the same output error rate.

The limiting factor at low signal energy for error correction is illustrated by the yellow (E_x'+E_c') curve representing energy consumed by the errors and error correction circuitry. The blue-green (E_x) curve represents the energy consumed by the errors themselves in the uncorrected case. The number of errors rises exponentially as signal energy drops. Furthermore, errors propagate through the circuit with fanout such that each error produces a large number of downstream errors in the circuit. Both the yellow and green curves include an exponential rise when moving to the left.

We conclude that there is an opportunity to exceed the purported Landauer-Shannon limit. However, the upside potential depends on many variables. Even ideal millivolt switches that have no leakage will consume power due to the energy of creating and propagating thermal errors. This effect becomes dominant for logic nets operating below 20 kT in this example but varies based on fanout and circuit depth.

IV. Adiabatic Charge-Recycling Memory

Aside from energy efficiency improvements resulting from monolithic integration of memory and logic, adiabatic charge-recycling has been explored to further increase energy efficiency for matrix-vector multiplication in massively parallel connectionist neural computation [16]. It saves substantial energy by conserving charge through capacitive coupling, rather than destructive charge transfer.

Here, we investigate the energy efficiency of the same adiabatic charge-recycling circuit when used in a memory. Since the proposed memory uses the same adiabatic circuit that was fabricated, tested, and reported as a neural network array processor [16], will report on its performance in the context of

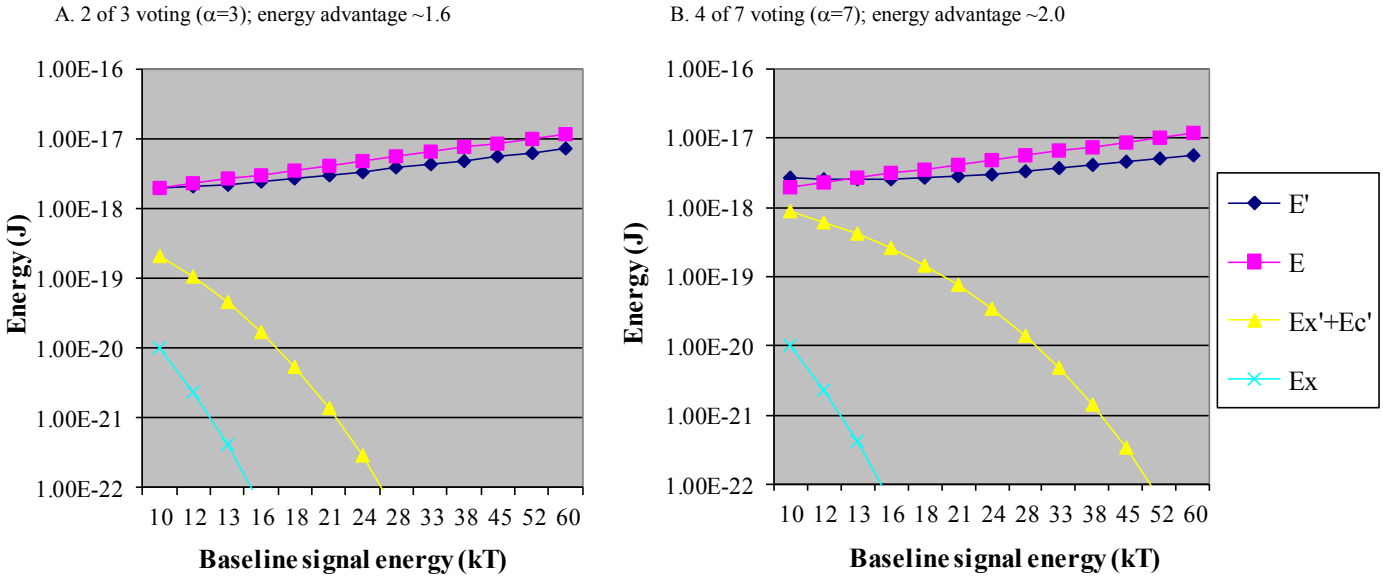

Fig. 2. Modeling of temporal error correction, α=3 and 7

a memory by reinterpreting past experimental results.

Memories dissipate energy in the addressing logic, capacitance between row and column conductors and ground, and in storage devices. The approach illustrated in Fig. 3 reduces all these energy losses by recovering energy recirculating between electrostatic and inductive forms in an LC tank circuit at its resonant oscillation frequency.

Fig. 3A illustrates the principle using a small memory array with columns in green, rows in orange, and charge-injection device (CID) storage cells in grey rectangles. Each CID cell is abstracted as an open circuit for a 0 and full-charge capacitive coupling for a 1. The current flow path for a read cycle is shown in blue. Energy stored in the inductor is connected to one row at a time through the addressing switch. The remainder of the circuit is an arrangement of one or four capacitors depending on the state of the storage cell. The blue-indicated elements form a tank circuit, which will oscillate at its resonant frequency. The principle of adiabatic resonant energy recovery is as follows: As long as the timing of the opening and closing of the addressing switch is properly synchronized with the oscillation, the resonant energy in the tank is conserved as charge is circulated, except for minor energy losses due to parasitic resistances.

Output data is sensed from the columns by a charge-sensitive amplifier. The amplifier detects a voltage change when the CID is in the fully charged capacitive form, but there is no voltage in the open circuit form.

The CID cell is realized by two charge-coupled MOSFETs in series (Fig. 3B, upper). The two MOSFET gates are connected to the row and column conductors. Since the two MOSFETs are isolated from their surroundings except for purely capacitive output coupling, charge is mostly conserved. Charge in the yellow isolation area in Fig. 3B can leak out through the gate oxide or reverse biased diodes that may exist in the substrate area, but this leakage is relatively slow and can be addressed with DRAM-like refresh (at Hz to kHz rates).

The CID-equivalent circuit (Fig. 3B, lower) changes from an open circuit to a nonlinear capacitor based on charge in the isolation region. With no stored charge (Q=0) both transistors are in a non-conductive state, and the equivalent circuit is an open circuit. On the other hand, a fully charged cell (Q=Q_{max}) effectively couples all its charge between the gates.

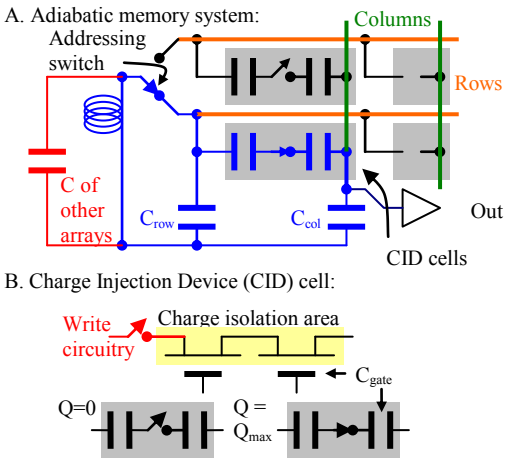

Fig. 3. Principle and simplified diagram of adiabatic charge-recovery

A memory chip would have single inductor for many banks with one row driven in each, in which case the tank's capacitor comprises total capacitance of the fully charged CID cells in all the selected rows at a given instant. The total capacitance and therefore the resonant frequency depends on the stored data. It is critical that the LC tank is maintained at its resonance peak for most efficient energy recovery.

The adiabatic circuit described above was fabricated and measured [16] as a 256×256 array multiplier, which would have similar circuit characteristics to a memory chip with 256 banks. Fig. 4 shows the impact of the energy recycling. The upper red lines are without energy recycling, which makes them comparable to production memories. With recycling turned on and tuned, energy consumption shown by the blue curves dropped by up to 85×, peaking when half the selected CID cells were storing 0 and the other half storing 1.

The adiabatic resonant energy recovery principle applies specifically to charge-based memory storage with capacitive readout, as described above. It readily extends to other memory types where charge can be inherently conserved—however, emerging resistive memory technologies such as memristor (ReRAM), PCRAM, and MRAM crossbars are inherently lossy and do not permit adiabatic energy recovery.

V. Discussion

Both error correction methods above rely on cooperation across multiple technology levels. Moore's Law presupposed improvement only at the individual device level, assuming it would be a "rising tide that lifts all ships" without redesign of the ships. However, the error correction methods above recognize that the desirable reduction in size and energy of devices results in an undesirable increase in error rate. These errors cannot be corrected at the device level with increasing energy levels, but the aggregate result of these errors can be corrected at higher levels of a computer's technology stack.

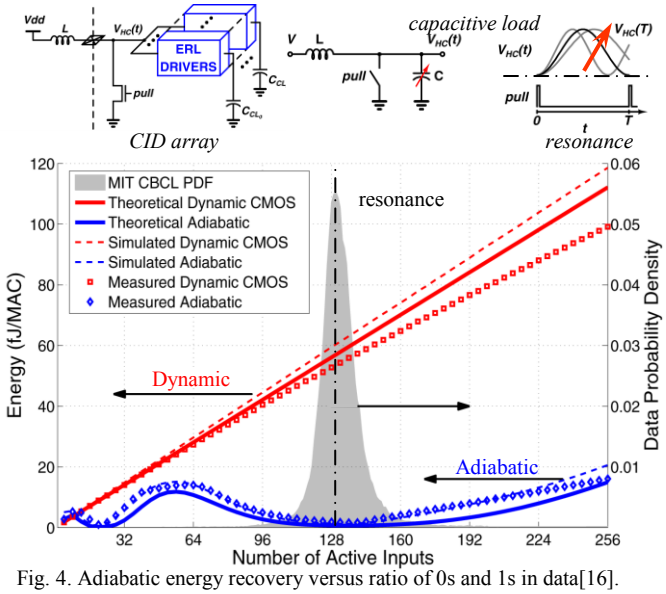

Fig. 4. Adiabatic energy recovery versus ratio of 0s and 1s in data[16].

The RRNS example uses mathematics to allow occasional errors to be corrected. Temporal error correction uses more than one gate's worth of energy to fix an error, but it can be applied to many gates at a time—leading to net advantage.

We can construct error correctoin schemes that beat the Landauer-Shannon limit, but we do not know the limits of the approach. Some applications require extremely reliable computing at the user level, such as Exascale supercomputers that may not make a single mistake over a multi-year lifetime. The simple analyses in this paper suggest 2× advantage for supercomputers. On the other hand, users watching video on a smartphone are likely to tolerate a bad pixel once in a while so the reliability requirements are less. The energy reduction due to error correction will be less in this case.

The RRNS example in section II was based on number system devised in 1965 for significantly different purpose. A separate paper in this conference details work in devising number systems and architectures that may perform better [13]. However, we do not know the potential of the approach.

The temporal error correction in section III was susceptible to direct analysis. It had two energy levels: E_{signal}' for the logic and E_{signal} for the correction circuitry. In the opinion of the authors, temporal error correction to reduce energy is unlikely to become a specific circuit or device. It seems more likely that a computerized design tool might be able to optimize a logic layout for low energy by applying algorithms to each gate.

Adiabatic memory shows significant promise in an unusual area. The memory in today's computers does not produce much heat, but there is a lot of interest in computer applications that use a lot of data. In conjunction with emerging 3D manufacture of logic or memory, it is possible that today's low power single-layer memories will evolve to hundred-layer modules that dissipate a hundred times as much power. After a 100× rise, memory would no longer be low power. This would create a demand for energy-efficient memories such as described.

VI. Conclusions

By simple arithmetic, an Exascale supercomputer needs an uncorrected gate-level reliability equivalent to an E_{signal} of around 60 kT to avoid silent errors over its lifetime. In this range, the error correction described in this paper could reduce overall energy by around 2×, for a effective energy of 30 kT. Using [17] as reference, logic is heading towards an energy of, say, 10,000 kT per operation. Subject to evolution of transistors into millivolt switches (which is not assured), the remaining improvement would be 10,000 / 30 ≈ 300×.

Reversible logic seems to be emerging as a practical option for continued scaling, including both reversible processors [12] and the discussion of reversible memory in this article. It is likely that the energy efficiency of memories will improve as Moore's Law progresses, yet the CV^2 energy in the row/column lines will limit energy efficiency. The adiabatic approach discussed in this paper demonstrated an 85× boost in a laboratory demonstration that could be further improved. The energy reduction from adiabatic operation would combine multiplicatively with the improvement due to Moore's Law.

More study of the limits of current technology would seem indicated. The semiconductor industry spends billions of dollars on new fab lines to get each additional 2× energy efficiency. Theory work on error correction appears from this paper seem capable of getting the same result at reduced cost.

Continued research on millivolt switches is indicated. This paper shows the devices would make even more of a contribution than currently expected if accompanied by error correction.

Memory is moving to 3D now, which has obvious benefits for both energy efficiency and applications that may need to use a lot of memory. However, device physics research would be needed for memory cells that avoid dissipating large amounts of power during reads and write—such as the open circuit/capacitor CID cell discussed in the paper.

References

[1] G. Moore, "Cramming more components onto integrated circuits, Electronics, volume 38, number 8, April 19, 1965, pp. 114 ff."

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

[3] E. DeBenedictis, M. Frank, N. Ganesh, N. G. Anderson, "A path toward ultra-low-energy computing," International Conference on Rebooting Computing, 2016.

[4] M. Neyman, "The negentropy principle in information-processing systems," Telecommunications and Radio Engineering 21 (1966): 68.

[5] J. Bellamy, Digital telephony (Wiley Series in Telecommunications and Signal Processing), Wiley-Interscience, 2000.

[6] T. Theis and P. Solomon, 'In quest of the "next switch": prospects for greatly reduced power dissipation in a successor to the silicon field-effect transistor,' Proceedings of the IEEE 98.12 (2010): 2005-2014.

[7] R. Keyes and R. Landauer, "Minimal energy dissipation in logic," IBM Journal of Research and Development 14.2 (1970): 152.

[8] Robert Lyons and Wouter Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," IBM Journal of Research and Development 6.2 (1962): 200-209.

[9] Kuang-Hua Huang and Jacob A. Abraham, "Algorithm-based fault tolerance for matrix operations," IEEE transactions on computers 100.6 (1984): 518-528.

[10] C. Bennett, "Logical reversibility of computation," *IBM Journal of Research and Development* 17.6 (1973): 525-532.

[11] M.P. Frank, "Reversibility for efficient computing," Ph. D. thesis, Massachusetts Institute of Technology, 1999.

[12] César O. Campos-Aguillón, et. al., "A Mini-MIPS microprocessor for adiabatic computing," International Conference on Rebooting Computing, 2016.

[13] B. Deng, et. al., "Computationally-Redundant Energy-Efficient Processing for Y'all (CREEPY)," International Conference on Rebooting Computing, 2016.

[14] R. Watson and C. Hastings, "Self-checked computation using residue arithmetic," Proceedings of the IEEE 54.12 (1966): 1920-1931.

[15] E. DeBenedictis and H. Zima, "Millivolt switches will support better energy-reliability tradeoffs," Energy Efficient Electronic Systems (E3S), 2015 Fourth Berkeley Symposium on. IEEE, 2015.

[16] R. Karakiewicz, R. Genov, and G. Cauwenberghs, "1.1 TMACS/mW fine-grained stochastic resonant charge-recycling array processor," Sensors Journal, IEEE 12.4 (2012): 785-792.

[17] D. Frank, "Reversible adiabatic classical computation—an overview," 2nd IEEE Rebooting Computing Summit, 2014, slide 23; http://rebootingcomputing.ieee.org/images/files/pdf/7-rcs2-adiabatic-reversible-5-14-14.pdf.