Almost All Functions Require Exponential Energy

Neal Barcelo[1,⋆], Michael Nugent[1,⋆⋆], Kirk Pruhs[1,⋆⋆⋆], and Michele Scquizzato[2,†]

[1] Department of Computer Science, University of Pittsburgh, Pittsburgh, USA
[2] Department of Computer Science, University of Houston, Houston, USA

Abstract. One potential method to attain more energy-efficient circuits with the current technology is Near-Threshold Computing, which means using less energy per gate by designing the supply voltages to be closer to the threshold voltage of transistors. However, this energy savings comes at a cost of a greater probability of gate failure, which necessitates that the circuits must be more fault-tolerant, and thus contain more gates. Thus achieving energy savings with Near-Threshold Computing involves properly balancing the energy used per gate with the number of gates used. The main result of this paper is that almost all Boolean functions require circuits that use exponential energy, even if allowed circuits using heterogeneous supply voltages. This is not an immediate consequence of Shannon's classic result that almost all functions require exponential sized circuits of faultless gates because, as we show, the same circuit layout can compute many different functions, depending on the value of the supply voltages. The key step in the proof is to upper bound the number of different functions that one circuit layout can compute. We also show that the Boolean functions that require exponential energy are exactly the Boolean functions that require exponentially many faulty gates.

⋆ This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1247842. E-mail: ncb30@pitt.edu.

⋆⋆ E-mail: mpn1@pitt.edu.

⋆⋆⋆ Supported in part by NSF grants CCF-1115575, CNS-1253218, CCF-1421508, and an IBM Faculty Award. E-mail: kirk@cs.pitt.edu.

† Work done while at the University of Pittsburgh. E-mail: michele@cs.uh.edu.

1 Introduction

The threshold voltage of a transistor is the minimum supply voltage at which the transistor starts to conduct current. However, if the designed supply voltage was exactly the ideal threshold voltage, some transistors would likely fail to operate as designed due to manufacturing and environmental variations. In the traditional approach to circuit design the supply voltages for each transistor/gate are set sufficiently high so that with sufficiently high probability no transistor fails, and thus the designed circuits need not be fault-tolerant. One potential method to attain more energy-efficient circuits is *Near-Threshold Computing*, which simply means that the supply voltages are designed to be closer to the threshold voltage. As the power used by a transistor/gate is roughly proportional to the square of the supply voltage [4], Near-Threshold Computing can potentially significantly decrease the energy used per gate. However, this energy savings comes at a cost of a greater probability of functional failure, which necessitates that the circuits must be more fault-tolerant, and thus contain more gates. For an example of this tradeoff in an SRAM cell, see [7].

1.1 Our Contributions

As the total energy used by a circuit is roughly the sum over all gates of the energy used by that gate, achieving energy savings with Near-Threshold Computing involves properly balancing the energy used per gate with the number of gates used. In principle, for every function f there exists a circuit C computing f with probability of error at most δ that uses minimum energy. It is natural to ask questions about the minimum energy required for various functions. Pippenger showed that all Boolean functions with n inputs can be computed by circuit layouts with $O(2^n/n)$ noisy gates (i.e., gates that fail independently with some known, fixed probability) [12]. Using that construction, it immediately follows that all Boolean functions can be computed by some circuit C that uses $O(2^n/n)$ energy when δ is a fixed constant. Our main result, which we state somewhat informally below, is that this result is tight for almost all functions.

Theorem 1. *Almost all Boolean functions on n variables require circuits that use $\Omega(2^n/n)$ energy.*

The main component of the proof is to show that most functions require circuit layouts with exponentially many gates. Note that in this setting, this is not an immediate consequence of Shannon's classic result [15] that most functions require circuit layouts with exponentially many faultless gates. To understand this point better, let us consider the simple counting-based proof of the following somewhat informal statement of Shannon's classic result:

Theorem 2 (Shannon [15]). *Almost all Boolean functions on n inputs require circuits with faultless gates of size $\Omega(2^n)$ bits (and of size $\Omega(2^n/n)$ gates).*

Proof. We will associate circuit layouts with their binary representation in some standard form. Each string of k bits specifies at most one circuit layout. There are 2^k bit strings of length k. Thus using k or less bits, at most $\sum_{i=0}^{k} 2^i \leq 2^{k+1}$ different circuit layouts can be specified. But there are 2^{2^n} Boolean functions with n input bits, hence $k = 2^n - \ell$ bits are only sufficient to specify a $2^{2^n-\ell+1}/2^{2^n} = 1/2^{\ell-1}$ fraction of all the possible Boolean functions. The bound on the number of gates follows by noting that the number of bits per gate is logarithmic in the number of gates. □

The reason that this proof does not work in a Near-Threshold Computing setting is because a circuit now not only consists of a layout, but also of a set of supply voltages. Thus in principle a circuit may compute different functions for different settings of the supply voltages. We start by showing that, perhaps somewhat surprisingly until one sees the trick, this can in fact actually happen. In Section 2 we show that when supply voltages must be homogeneous, that is every gate of the circuit is supplied with the same voltage, there are simple circuits with n inputs and $O(n)$ gates that compute $\Omega\big(\log n/\log(\frac{1}{\delta}\log n)\big)$ different functions with probability of error at most δ, and when heterogeneous supply voltages are allowed, there are circuits with n inputs and $O\big(n^2\big)$ gates that compute $\Omega(3^n)$ different functions. Here by heterogeneous voltages we simply mean that different gates could be supplied with different voltages.

In contrast, in Section 3 we show that, for each $\delta < 1/2$, every homogeneous circuit with n inputs and s faulty gates computes at most $s2^n + 1$ different functions, and every heterogeneous circuit with s faulty gates computes at most $(8e2^n)^s$ different functions. These upper bounds are then sufficient to prove our main result using the same counting-based technique as in the proof of Shannon's classic result. Since a homogeneous voltage setting is also a heterogeneous voltage setting, the result that almost all functions require heterogeneous circuits using exponential energy is strictly stronger than the corresponding result for homogeneous circuits. Nevertheless, we include the latter as it demonstrates how much simpler homogeneous supply voltages are, and as we are able to obtain a slightly stronger bound in terms of the required error probability δ.

These results leave open the possibility that some Boolean functions that do not require circuits with exponentially many gates still require exponential energy. For example, it could be the case that for some function the energy-optimal circuit has sub-exponentially many gates, with many of them requiring exponential energy. We show in Section 4 that this is not the case, i.e., the Boolean functions that require exponential energy are exactly the Boolean functions that require exponentially many faulty gates.

1.2 Related Work

The study of fault-tolerant circuits started with the seminal paper by von Neumann [16]. Subsequent work can be found in [5, 6, 12–14, 9, 8, 10]. The results of [16, 6, 12] show that any circuit layout of s faultless gates can be simulated by a circuit with $O(s \log s)$ noisy gates. As already mentioned, Pippenger [12] showed that all

Boolean functions can be computed by circuit layouts with $O(2^n/n)$ noisy gates. In fact, he proved this result in a stronger model in which the error probabilities of the gates could be adversarially set in the range $[0, \epsilon]$. In this model, the fact that almost all functions require $\Omega(2^n/n)$ noisy gates immediately follows from the classic result of Shannon that most functions require $\Omega(2^n/n)$ faultless gates and noting that the circuit must compute correctly if there are no gate failures. It is also known that functions with sensitivity m (roughly, the number of bits that affect the output on any input) require $\Omega(m \log m)$ noisy gates [5, 13, 9]. A more detailed history can be found in [9, 8].

The general idea of trading accuracy of a hardware circuit and computing architecture for energy savings dates back to at least [11]. An excellent survey on Near-Threshold Computing can be found in [7]. A theoretical study of Near-Threshold Computing was initiated in [3]. The four main results in [3] are: (1) to compute a function with sensitivity m requires a circuit that uses energy $\Omega(m \log m)$, (2) if a function can be computed by a circuit with s faultless gates, then it can be computed by a circuit with energy $O(s \log s)$ when δ is a fixed constant, (3) there are circuits where there is a feasible heterogeneous setting of the supply voltages which uses much less energy than any feasible homogeneous setting of the supply voltages, and (4) there are functions where there are nearly optimal energy circuits that have a homogeneous setting of the supply voltages when δ is a fixed constant. [2] considered the problem of setting the supply voltage of a given circuit in such a way that the circuit has a specified reliability with the objective of minimizing energy. [2] showed that obtaining a significantly better approximation ratio than the traditional approach, which sets the voltage sufficiently high so that with sufficiently high probability no gate fails, is NP-hard.

1.3 Formal Model

A *Boolean function* f is a function from $\{0,1\}^n$ to $\{0,1\}$. A *gate* is a function $g : \{0,1\}^{n_g} \to \{0,1\}$, where n_g is the number of inputs (i.e., the *fan-in*) of the gate. We assume that the maximum fan-in is at most a constant. A *Boolean circuit* C with n inputs is a directed acyclic graph in which every node is a gate. Among them there are n gates with fan-in zero, each of which outputs one of the n inputs of the circuit. The *size* of a circuit, denoted by s, is the number of gates it contains. For any $I \in \{0,1\}^n$, we denote by $C(I)$ the output of the Boolean function computed by Boolean circuit layout C.

In this paper we consider circuits $(C, \bar{v})$ that consist of both a traditional circuit layout C as well as a vector of supply voltages $\bar{v}$, one for each gate of C. Every gate g is supplied with a voltage v_g. We say that the supply voltages are *homogeneous* when every gate of the circuit is supplied with the same voltage, and *heterogeneous* otherwise. A circuit is said to be homogeneous when its supply voltages are homogeneous, and heterogeneous otherwise. We say that a gate *fails* when it produces an incorrect output, that is, when given an input x it produces an output other than $g(x)$.

Each (*faulty*[3]) non-input gate g fails independently with probability $\epsilon(v_g)$, where $\epsilon : \mathbb{R}^+ \to (0, 1/2)$ is a decreasing function. The voltage supplied to a gate determines both its energy usage and its failure probability, thus we define $\epsilon_g := \epsilon(v_g)$ and drop all future formal reference to supply voltages. Finally we assume there is a failure-to-energy function $E(\epsilon)$ that maps the failure probability ϵ to the energy used by a gate. The only constraints we impose on $E(\epsilon)$ are that it is decreasing and $\lim_{x\to 0^+} E(1/2 - x) > 0$. In practice $E(\epsilon)$ is observed to be roughly $\Theta(\log^2(1/\epsilon))$ [7, 3]. The energy used by a circuit C is simply the aggregate energy used by the gates, $\sum_{g\in C} E(\epsilon_g)$ in our notation.

A gate that never fails is said to be *faultless*. Given a value $\delta \in (0, 1/2)$ (δ may not be constant), a circuit $(C, \bar{\epsilon})$ that computes a Boolean function f is said to be $(1 - \delta)$-*reliable* if for every input I, $C(I)$ equals $f(I)$ with probability at least $1 - \delta$. We say that C can compute ℓ different Boolean functions $(1 - \delta)$-reliably if there exist $\bar{\epsilon}_1, \bar{\epsilon}_2, \ldots, \bar{\epsilon}_\ell \in (0, 1/2)^{|C|}$ and different Boolean functions $f_1, f_2, \ldots, f_\ell$ such that $(C, \bar{\epsilon}_i)$ computes f_i $(1 - \delta)$-reliably, for each $i \in 1, 2, \ldots, \ell$.

2 A Lower Bound on the Number of Functions Computable by a Circuit

In this section we show that, in both the homogeneous case and the heterogeneous case, a single circuit can $(1 - \delta)$-reliably compute many different functions, by changing the supply voltage(s). Both of these lower bounds demonstrate that Shannon's counting argument will not be sufficient to show that almost all functions require exponential energy.

2.1 Homogeneous Supply Voltages

We start with the homogeneous case, giving an explicit construction of a circuit that computes approximately $\log n$ different functions. The key concept used throughout is that for a large enough perfect binary tree of AND gates (referred to as an AND tree) there is some ϵ such that, regardless of the input, the tree will output 0 with high probability. By combining such trees of different sizes into a single circuit we can essentially ignore different parts of the input depending on ϵ. The statement and proof are formalized below.

Theorem 3. *For any $\delta \in (0, 1/2)$ and $n \in \mathbb{N}$, there exists a homogeneous circuit C with n inputs and size $O(n)$ that computes $\Omega\left(\frac{\log n}{\log(\frac{1}{\delta}\log n)}\right)$ different Boolean functions $(1 - \delta)$-reliably.*

Proof (sketch). The circuit, which we indicate with C, consists of k perfect binary trees of AND gates, which we refer to as $\mathrm{AND}_1, \ldots, \mathrm{AND}_k$, and of a complete

[3] In previous work *faulty* and *noisy* are often used as synonyms, however, in order to provide additional clarity in regards to which model is currently being referred to, we use noisy when referring to gates in the fault-tolerant model, and faulty when referring to gates in the near-threshold model.

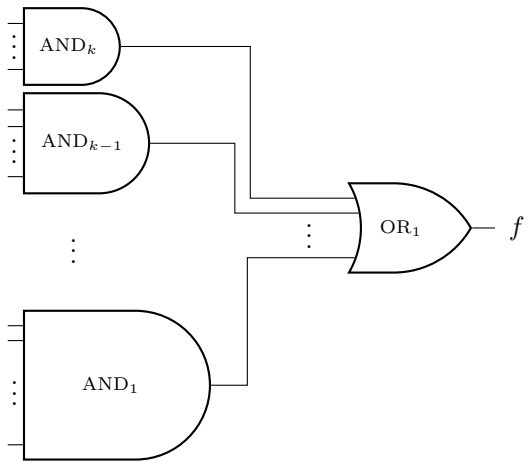

Fig. 1: The circuit used in the proof of Theorem 3.

binary tree of OR gates, denoted OR_1. The size of AND_i, which will be determined later but decreases exponentially as i increases, is denoted by s_i, and the size of OR_1 is $k-1$. Each AND tree receives its own set of input bits. The outputs of these k trees are fed into the tree of OR gates, and the output of the latter tree is the output of the circuit. Thus, when $\epsilon = 0$, the circuit C computes $\mathrm{OR}(\mathrm{AND}_1, \ldots, \mathrm{AND}_k)$ (see Figure 1).

The high level approach is to show that as ϵ grows larger, the larger AND trees switch from computing the AND function to computing the 0 function. In other words, the result is completely determined by the remaining functional AND trees. By choosing the sizes s_i to be sufficiently different, we can show that each AND_i will switch to computing the 0 function at a different ϵ, and further, when this switch occurs all of the smaller trees will still be functioning correctly with high probability. The details are left to the full version. □

2.2 Heterogeneous Supply Voltages

We now show that with heterogeneous voltage settings, we can construct a circuit that computes exponentially many functions $(1-\delta)$-reliably. We leverage the power of heterogeneity to ensure that certain parts of the circuit compute correctly with high probability, while other parts can fail with high probability. In particular, we build a circuit for a conjunctive normal form (CNF) Boolean formula where the literals of the formula can be determined dynamically by forcing certain gates to fail while preserving the correctness of the CNF calculation. This allows a single circuit to compute all possible functions representable by CNF formulas with n variables and a fixed number of fixed-sized clauses.

Theorem 4. *For any constant $\delta \in (0, 1/2)$ and $n \in \mathbb{N}$, there exists a heterogeneous circuit C of size $O(n^2)$ that computes $\Omega(3^n)$ different Boolean functions $(1-\delta)$-reliably.*

Proof (sketch). We give a circuit that computes at least 3^n different functions. We delay the discussion of voltages and correctness until we have completely described the circuit. Consider a 3CNF formula Φ with n variables and m clauses, i.e., $\Phi(x)$ is 1 if x satisfies all the clauses and 0 otherwise. To build a circuit that computes Φ, for each clause $(\ell_1 \vee \ell_2 \vee \ell_3)$ we have a single OR gate the inputs of which are variables ℓ_1, ℓ_2, ℓ_3 (note these need not be different and we are ignoring negations here). The output of each such OR gate is fed into an AND tree which outputs the conjunction of all such clauses. This circuit computes f_Φ, the Boolean function computed by 3CNF formula Φ.

We now give the construction of the circuit C. Consider a generic 3CNF formula $\Phi = (\ell_1 \vee \ell_2 \vee \ell_3) \wedge \cdots \wedge (\ell_{3m-2} \vee \ell_{3m-1} \vee \ell_{3m})$, and the corresponding series of OR and AND gates as described above, however with input wires coming into each ℓ_i removed. We will use a selection circuit to dynamically connect each ℓ_i to some x_j, depending on the supply voltages.

We define the selection circuit for ℓ_i, S_i as follows. This circuit takes as input $\log 2n$ bits as selectors as well as the $2n$ bits $(x_1, \neg x_1, \ldots, x_n, \neg x_n)$. The output of S_i is the bit corresponding to the location determined by the first $\log 2n$ bits. Note that Pippenger provides such a circuit of size $O(n)$ in [12]. Hence for all possible Φ, by appropriately setting the $\log 2n$ bits of each selection circuit, this circuit computes the function f_Φ.

The last piece necessary to define C is describing how the $\log 2n$ input bit b_k of each selection circuit are set. For each such b_k, we have a tree of AND gates with $\Theta\left(\log \frac{m \log n}{\delta}\right)$ inputs, the output of which is fed into b_k. The input to these AND gates are constant 0's that go through a single NOT gate, which have failure probability close to 0 if we want b_k to be 1, and close to $1/2$ if we want b_k to be 0. We leave to the full version the details of showing that for any fixed Φ there are voltage settings such that, for all x, with probability at least $1-\delta$, $C(x) = f_\Phi(x)$.

Consider the case where $m = n$. We now compute the size of C. The size of the 3CNF circuit is at most $3n$. For each of the $3n$ literals, there is a circuit of size $O(n)$ that uses $\log 2n$ bits to map an input or its negation to that literal. Each of the $O(n \log n)$ bits is created by a tree of size $O(\log(n \log(n)/\delta))$. Thus C has size $O\big(n^2 + n \log(n) \log(1/\delta)\big)$.

The last step is to show that there are $\Omega(3^n)$ unique functions $f_\Phi(x)$ with m clauses. Consider some subset $S = \{s_1, \ldots, s_{|S|}\} \subseteq [n]$ and some assignment $x = (x_{s_1}, x_{s_2}, \ldots, x_{s_{|S|}})$ for the variables x_i such that $i \in S$. Then, for each such x_i, if $x_i = 1$ create the clause $(x_i \vee x_i \vee x_i)$ and if $x_i = 0$ create the clause $(\neg x_i \vee \neg x_i \vee \neg x_i)$. Create $n - |S|$ additional clauses that are a duplicate of one of these clauses. Note that the resulting formula Φ returns 1 exactly when the input bits S are set to x, regardless of the value of the rest of the input bits, and 0 otherwise. Thus for each unique assignment of x and each unique S we obtain a new function. Since there

are $\binom{n}{|S|}$ ways to choose S and $2^{|S|}$ possible assignments for x, by the binomial theorem we have that the sum over $0 \leq |S| \leq n$ of $\binom{n}{|S|}2^{|S|}$ is 3^n. □

3 Almost all Functions Require Exponential Energy

In this section we show that, despite the ability of a single circuit to compute multiple functions, an upper bound on the number of such functions and an adaptation of Shannon's argument allows us to show that almost all functions require exponential energy, both in the homogeneous and heterogeneous case. In some sense, this is evidence that the advantages heterogeneity provides are somewhat limited, as even though some heterogeneous circuits can compute many more functions than any homogeneous circuit of the same size, this advantage is not sufficient to reduce the minimal circuit size by more than a constant for almost all functions.

3.1 Adaptation of Shannon's Argument

Inspired by Shannon's counting argument that almost all Boolean functions require exponentially-sized circuits, we show first that, in circuit models where circuits can compute multiple functions, as long as the number of functions a single circuit can compute is not too many, almost all functions still require exponentially-sized circuits. We will combine this with upper bounds on the number of functions homogeneous and heterogeneous circuits can compute to obtain our main results. Note that the following lemma assumes gates have fan-in at most two, and thus all of our results assume gate fan-in is at most two; It is straightforward to generalize this lemma and our results to any setting where the fan-in of the gates is a constant.

Lemma 1. *Suppose a circuit of size s can compute at most $f(s)$ Boolean functions in some circuit model where gates have fan-in at most two. If there exists some constant $c > 0$ such that $s^{4s} f(s) = o\big(2^{2^n}\big)$ for $s = 2^n/cn$, then almost all Boolean functions require $\Omega(2^n/n)$ gates in that model.*

Proof. Consider the set of circuits with at most s gates. A standard counting argument shows that any circuit in this set can be represented with $4s \log s$ bits, and therefore there are at most s^{4s} circuits with size at most s. Thus, if for some $c > 0$ and $s = 2^n/cn$ it holds that $s^{4s} f(s) = o\big(2^{2^n}\big)$, then almost all Boolean functions require circuits of size at least $2^n/cn = \Omega(2^n/n)$. □

3.2 Homogeneous Supply Voltages

In this subsection we show that almost all functions require exponential-energy homogeneous circuits. In some sense, this result is a corollary of the later result that almost all functions require exponential-energy heterogeneous circuits; However, we include this result as it illustrates how homogeneous circuits are simpler than heterogeneous circuits, and we are able to obtain a slightly stronger lower

bound on the energy used by almost all functions. Our proof aims to bound the number of functions a circuit of size s can compute, which is necessary, since, as we showed in the previous section, a single circuit can compute many functions.

Lemma 2. *For any circuit C on n inputs with s gates, and any $\delta > 0$, let $\mathcal{F}$ be the set of all Boolean functions f for which there exists some $\epsilon \in (0, 1/2)$ such that (C, ϵ) is $(1-\delta)$-reliable for f. Then, $|\mathcal{F}| \leq s2^n + 1$.*

Proof. Fix some circuit C and input I, and let $C_I(\epsilon)$ be the probability that (C, ϵ) outputs a 1 on input I. Note that by definition for (C, ϵ) to compute some function f we must have that for all inputs I, either $C_I(\epsilon) \geq 1 - \delta$ or $C_I(\epsilon) \leq \delta$. Fix some input I and consider how the output of C changes as we vary ϵ. Note that the above observation implies that C will only switch the function it is computing due to input I if $C_I(\epsilon) = 1-\delta$ and $C_I(\epsilon)$ is decreasing or $C_I(\epsilon) = \delta$ and $C_I(\epsilon)$ is increasing. However note that $C_I(\epsilon)$ is a polynomial in ϵ of degree s,[4] and therefore there are at most s such points since between any two of them the function must change at least once from increasing to decreasing or vice versa. This means that each input I can cause C to switch the function it is computing at most s times. Since there are 2^n distinct inputs, this means that C can switch functions at most $s2^n$ times, and therefore it is able to compute at most $s2^n + 1$ different functions. □

Since $E(\epsilon) = \Omega(1)$ for $\epsilon > 1/2$, we need only show that almost all functions require exponentially many gates in this model to show that almost all functions require exponential energy. However, the following lemma will allow us to strengthen our theorem statement.

Lemma 3. *Let C be a homogeneous circuit that is $(1-\delta)$-reliable. Then, $\epsilon \leq \delta$.*

Proof. Let f be the function C is trying to compute, and fix some input I. It suffices to show that the output gate, g_o, must fail with probability less than δ. Let p be the probability that g_o receives an input I' such that $g_o(I') = f(I)$. Then, note that $\mathbf{Pr}[g_o(I') = f(I)] = p(1-\epsilon) + (1-p)\epsilon \leq 1 - \epsilon$. Since by hypothesis $\mathbf{Pr}[C(I) = f(I)] \geq 1 - \delta$, it follows that $\epsilon \leq \delta$. □

We can now prove the desired theorem.

Theorem 5. *For any $\delta \in (0, 1/2)$, almost all Boolean functions on n variables require homogeneous circuits using $\Omega(E(\delta)2^n/n)$ energy.*

Proof. From Lemma 2 we know that each circuit of size s computes at most $s2^n+1$ different functions. We now show that for $s = 2^n/4n$, the quantity $s^{4s}(s2^n + 1)$ is asymptotically smaller than 2^{2^n}, the number of functions on n inputs. Plugging in and simplifying we have

$$\left(\frac{2^n}{4n}\right)^{4\frac{2^n}{4n}} \left(\frac{2^n}{4n}2^n + 1\right) \leq \frac{2^{2^n}}{n^{\frac{2^n}{n}}}2^{2n} = o(2^{2^n}).$$

[4] If we fix which gates fail, then the output of C on I is fixed to either 1 or 0. A fixed set of q gates fail with probability $\epsilon^q(1-\epsilon)^{s-q}$, a polynomial of degree s in ϵ. $C_I(\epsilon)$ can be viewed as the sum over the sets of gates that, when failing, cause C to output 1 on I, of the probability of that set failing.

Hence, Lemma 1 implies that almost all homogeneous circuits require $\Omega(2^n/n)$ gates. By Lemma 3, we have $\epsilon \leq \delta$, so each gate uses at least $E(\delta)$ energy. □

3.3 Heterogeneous Supply Voltages

In this section we show that almost all functions require exponential energy, even when allowed circuits with heterogeneous voltages. The approach is similar to the one for the homogeneous case, however the bound on the number of functions a heterogeneous circuit can compute requires a technical result from real algebraic geometry, which was proved by Alon [1].

Lemma 4. *For any circuit C on n inputs with s gates, and any $\delta > 0$, let $\mathcal{F}$ be the set of all Boolean functions f for which there exists some $\bar{\epsilon} \in (0, 1/2)^{|C|}$ such that $(C, \bar{\epsilon})$ is $(1-\delta)$-reliable for f. Then, $|\mathcal{F}| \leq (8e2^n)^s$.*

Proof. Let $\mathcal{P} \subset \mathbb{R}[X_1, \ldots, X_k]$ be a finite set of p polynomials with degree at most d. A *sign condition* on $\mathcal{P}$ is an element of $\{0, 1, -1\}^p$. The *realization* of the sign condition σ in $\mathbb{R}^k$ is the semi-algebraic set

$$\mathcal{R}(\sigma) = \left\{ x \in \mathbb{R}^k : \bigwedge_{P \in \mathcal{P}} \operatorname{sign}(P(x)) = \sigma(P) \right\}.$$

Let $N(p, d, k)$ be the number of realizable sign conditions, i.e., the cardinality of the set $\{\sigma : \mathcal{R}(\sigma) \neq \emptyset\}$. The following theorem is due to Alon.

Theorem 6 (Proposition 5.5 in [1]). *If $2p > k$, then $N(p, d, k) \leq \left(\frac{8edp}{k}\right)^k$.*

Let $I \in \{0,1\}^n$ be some input to C, and let $P_I(\epsilon_1, \ldots, \epsilon_s)$ be the probability that C outputs 1 on I, when gate i fails with probability ϵ_i. Observe that $P_I \in \mathbb{R}[\epsilon_1, \ldots, \epsilon_s]$ and that P_I has degree at most s, since we can compute P_I by summing over all possible subsets of gates that could fail and cause C to output a 1, of the probability that exactly those gates fail and no others (which is a polynomial is $\epsilon_1, \ldots, \epsilon_s$, where each ϵ_i has exponent 1).

Let $\mathcal{P} = \{P_I - (1-\delta) | I \in \{0,1\}^n\}$. Clearly, the cardinality of $\mathcal{P}$ is at most 2^n. Observe that every different function f that C calculates must correspond to a unique realizable sign condition of $\mathcal{P}$, in the sense that there is some setting of $\bar{\epsilon} = (\epsilon_1, \ldots, \epsilon_s)$ such that

1. $P(\bar{\epsilon}) - (1-\delta) > 0$ on inputs I such that $f(I) = 1$, and
2. $P(\bar{\epsilon}) - (1-\delta) < 0$ on inputs I such that $f(I) = 0$ (in fact, we need $P(\bar{\epsilon}) - \delta < 0$, an even stronger condition).

By Theorem 6, if the size of $\mathcal{P}$ is at least $n/2$, the number of realizable sign conditions of $\mathcal{P}$ is at most $(8e2^n)^s$. Otherwise, if the size of $\mathcal{P}$ is at most $n/2$, the total number of sign conditions is at most $3^{n/2} = o((8e2^n)^s)$. Thus, we have obtained an upper bound on the number of different functions C can compute. □

We can now prove our main theorem.

Theorem 7. *For any $\delta \in (0, 1/2)$, almost all Boolean functions on n variables require heterogeneous circuits using $\Omega(2^n/n)$ energy.*

Proof. From Lemma 4 we know that each circuit of size s computes at most $(8e2^n)^s$ different functions. We now show that for $s = 2^n/8n$, the quantity $s^{4s}(8e2^n)^s$ is asymptotically smaller than 2^{2^n}, the number of functions on n inputs. Plugging in and simplifying we have

$$\left(\frac{2^n}{8n}\right)^{4\frac{2^n}{8n}} (8e2^n)^{\frac{2^n}{8n}} \leq 2^{\frac{2^n}{2}} 2^{\frac{2^n(3+2-12-4\log n)}{8n}} 2^{\frac{2^n}{8}} \leq 2^{\frac{5\cdot 2^n}{8}} = o(2^{2^n}).$$

Hence, Lemma 1 implies that almost all heterogeneous circuits require $\Omega(2^n/n)$ gates. The theorem follows since $E(1/2) = \Omega(1)$ and E is decreasing in the interval $(0, 1/2)$. □

4 Relating Energy and the Number of Faulty Gates

In this section, we show that the Boolean functions that require exponential energy are exactly the Boolean functions that require exponentially many faulty gates. Before formalizing this notion we introduce some additional notation. For any Boolean function f on n variables and any reliability parameter δ, let $NG(f, \delta)$ denote the minimum size of any (heterogeneous) circuit that $(1-\delta)$-reliably computes f, and $\widetilde{NG}(f, \delta)$ denote the minimum size of any homogeneous circuit that $(1-\delta)$-reliably computes f. Similarly define $\mathcal{E}(f, \delta)$ to be the minimum energy used by any (heterogeneous) circuit that $(1-\delta)$-reliably computes f, and $\widetilde{\mathcal{E}}(f, \delta)$ the minimum energy used by any homogeneous circuit that $(1-\delta)$-reliably computes f. We are now ready to state the main result of this section.

Lemma 5. *For all Boolean functions f, and for all $\delta < 1/2$,*

$$E(1/2)NG(f, \delta) \leq \mathcal{E}(f, \delta) \leq \widetilde{\mathcal{E}}(f, \delta) \leq E\left(\frac{\delta}{\widetilde{NG}(f, \delta)}\right) \widetilde{NG}(f, \delta).$$

Proof. First observe that $\mathcal{E}(f, \delta) \leq \widetilde{\mathcal{E}}(f, \delta)$. We now prove the leftmost inequality. Let $(C, \bar{\epsilon})$ be the circuit achieving $\mathcal{E}(f, \delta)$ and note that by definition $\mathcal{E}(f, \delta) = \sum_{g \in C} E(\epsilon_g)$. Since E is decreasing, it follows that $E(\epsilon_g) \geq E(1/2)$ for all $g \in C$. Additionally, by definition, $|C| \geq NG(f, \delta)$, and the result follows.

To show the rightmost inequality, fix some Boolean function f, and some δ. Let C be a circuit of size $s = \widetilde{NG}(f, \delta)$, and ϵ the failure probability, such that (C, ϵ) is $(1-\delta)$-reliable on f. If $\epsilon \geq \delta/s$, we are done, since E is decreasing. Note that for a circuit of size s, if gates fail with probability at most δ/s, then by the union bound, the probability that any gate fails is at most δ. Thus, if $\epsilon < \delta/s$, the probability that any gate fails is at most δ. However, this implies that $(C, \delta/s)$ is $(1-\delta)$-reliable on f as well, and thus can use energy $E\left(\frac{\delta}{\widetilde{NG}(f,\delta)}\right) \widetilde{NG}(f, \delta)$. □

If $E(1/2)$ is $\Omega(1)$, and $E(\delta/\widetilde{NG}(f,\delta))$ is bounded above by a polynomial in $\widetilde{NG}(f,\delta)$ and $1/\delta$ (recall that in current CMOS technologies $E(\epsilon) = \Theta\big(\log^2(1/\epsilon)\big)$), this implies that any function that requires exponential energy requires exponential circuit size and vice versa.

References

1. N. Alon. Tools from higher algebra. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, volume 2, pages 1749–1783. MIT Press, 1995.
2. A. Antoniadis, N. Barcelo, M. Nugent, K. Pruhs, and M. Scquizzato. Complexity-theoretic obstacles to achieving energy savings with Near-Threshold Computing. In *Proceedings of the 5th International Green Computing Conference (IGCC)*, pages 1–8, 2014.
3. A. Antoniadis, N. Barcelo, M. Nugent, K. Pruhs, and M. Scquizzato. Energy-efficient circuit design. In *Proceedings of the 5th conference on Innovations in Theoretical Computer Science (ITCS)*, pages 303–312, 2014.
4. J. Butts and G. Sohi. A static power model for architects. In *Proceedings of the 33rd annual ACM/IEEE International Symposium on Microarchitecture (MICRO)*, pages 191–201, 2000.
5. R. L. Dobrushin and S. I. Ortyukov. Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements. *Problems of Information Transmission*, 13:59–65, 1977.
6. R. L. Dobrushin and S. I. Ortyukov. Upper bound for the redundancy of self-correcting arrangements of unreliable functional elements. *Problems of Information Transmission*, 13:203–218, 1977.
7. R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. N. Mudge. Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits. *Proceedings of the IEEE*, 98(2):253–266, 2010.
8. P. Gács. *Algorithms in Informatics*, volume 2, chapter Reliable Computation. ELTE Eötvös Kiadó, Budapest, 2005.
9. P. Gács and A. Gál. Lower bounds for the complexity of reliable Boolean circuits with noisy gates. *IEEE Transactions on Information Theory*, 40(2):579–583, 1994.
10. D. J. Kleitman, F. T. Leighton, and Y. Ma. On the design of reliable Boolean circuits that contain partially unreliable gates. *Journal of Computer and System Sciences*, 55(3):385–401, 1997.
11. K. V. Palem. Energy aware computing through probabilistic switching: A study of limits. *IEEE Trans. Computers*, 54(9):1123–1137, 2005.
12. N. Pippenger. On networks of noisy gates. In *Proceedings of the 26th Symposium on Foundations of Computer Science (FOCS)*, pages 30–38, 1985.
13. N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis. On a lower bound for the redundancy of reliable networks with noisy gates. *IEEE Transactions on Information Theory*, 37(3):639–643, 1991.
14. R. Reischuk and B. Schmeltz. Reliable computation with noisy circuits and decision trees–A general $n \log n$ lower bound. In *Proceedings of the 32nd Symposium on Foundations of Computer Science (FOCS)*, pages 602–611, 1991.
15. C. E. Shannon. The synthesis of two-terminal switching circuits. *Bell Systems Technical Journal*, 28:59–98, 1949.
16. J. von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 329–378. Princeton University Press, 1956.