

On the energy consequences of information for spacecraft systems

James Lyke, Jesse Mee, Arthur Edwards, and Andrew Pineda

Air Force Research Laboratory, 3550 Aberdeen Ave. SE, Kirtland AFB, New Mexico, USA

Erik DeBenedictis, Michael Frank

Sandia National Laboratories, Albuquerque, New Mexico, USA

Abstract— Conventional wisdom in the spacecraft domain is that on-orbit computation is expensive, and thus, information is traditionally funneled to the ground as directly as possible. The explosion of information due to larger sensors, the advancements of Moore’s law, and other considerations lead us to revisit this practice. In this article, we consider the trade-off between computation, storage, and transmission, viewed as an energy minimization problem..

Keywords—low-power microelectronics, energy minimization, Landauer limit, thermodynamic energy bound, adiabatic circuitry

I. INTRODUCTION

What is the most energy-efficient way to compute? The question can be considered from many levels of regard, from the shifting of states used to represent a single bit of information to the operation of data centers that drive global information networks for commercial, scientific, and military applications. It also impacts the many edge nodes, billions of platforms from disposable radio-frequency identification tags, embedded systems (spacecraft, cars, ships), to buildings.

A. Why Consider Spacecraft?

Space is an ideal vantage point around which many mission ideas involving information services can be formed, ranging from the earliest communications experiments to the modern creative concepts that involve the monetization of surveillance-based services, such as counting cars in the parking lots [1].

Over the decades, the size and frame rate of imaging sensors has increased, leading to much greater volumes of data to transmit, while at the same time, the Moore’s law trend in integrated circuits has led to dramatic reduction in the size, weight, and power of computing hardware that can be put in even the smallest spacecraft. We have ultimately three choices that can be applied in total or in combination to any data collected in the spacecraft: (1) process it on the spacecraft (which in effect reduces the amount to later be transmitted); (2) transmit it; or (3) store it. While tradition favors transmission (option 2), it may in fact be a more effective idea to selectively process and/or store information.

The engineering of spacecraft to implement these missions is challenging, due to the expense of launch, which puts pressure on the mass and dimensional envelope of spacecraft. These have led to the popular concept of very tiny spacecraft, commonly referred to as “CubeSats,” whose sizes are usually defined as integral multipliers of a 1U (10cm x 10cm x 10cm) envelope. At this design point, the pursuit of space has become more affordable (as of this writing over 700 CubeSats have been

launched [2]). At the other end of the spectrum, large spacecraft (e.g., > 10,000 kg) for communications, surveillance, and other applications have been fielded, albeit at a far greater expense.

In this paper, we seek to illuminate the problems of energy and information at different levels of regard :

Bit Level – A fundamental state representation of the most primitive quantity of information.

Circuit Level – Functions that transform one state representation into another one. Circuit-level usually refers to those types of functions that are in the NC computational complexity class [3], which are those “circuitize-able” functions capable of compact (polynomially-sized) circuit-based implementations

Algorithm Level – Algorithms are computations that can be mapped to Turing machines or optimized hardware.

Processing System Level. Considers the need to profile complex algorithms to better match worst-case execution time of code fragments to potentially widely different computing styles, such as CPU, GPU, FPGA, and ASIC.

Platform/Mission Level – Energy consumption at the mission level considers the interaction of computation with other functions in a product, platform, or mission.

II. THERMODYNAMIC LIMITS OF ENERGY IN COMPUTATION

The thermodynamic basis for the lower bounds of energy of information is referred to as the Landauer limit, which refers to the energy consumed by a single bit erasure, and the limit for a bit $k_B T \ln 2$, where k_B is Boltzmann’s constant and T is temperature [4]. Practically, Feynman suggested a more realistic engineering bound as being 80-100 $k_B T$ [5], which provides for some noise margin and useful (short) computing timespans. It appears that the practical energy limit in present-day CMOS is around 1,000-10,000 $k_B T$ [6], a bound that is today not improving dramatically even with additional gains in Moore’s law advancement. Breaking through this wall is, if not an obsession, very important to the industry

Landauer’s limit on the energy dissipation of irreversible computational operations can be understood as a consequence of the Second Law of Thermodynamics, which states that total entropy never decreases; this implies the reversibility of fundamental physics. Since empirically the Second Law holds, we know that distinct physical microstates can never evolve to the same future state, since then the entropy of a probability distribution over those states would decrease. Thus, fundamental physical dynamics necessarily evolves past states to future states

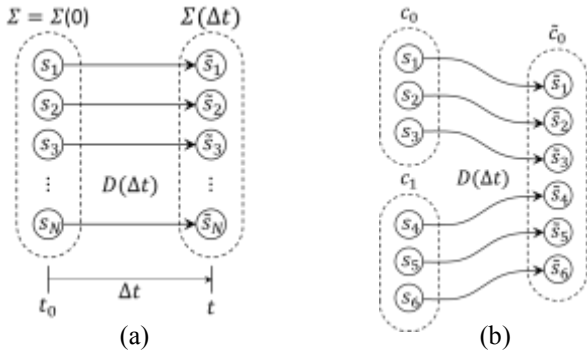


Fig. 1. (a) Illustration of a bijective functional relation $D(\Delta t)$ implementing the fundamental dynamical law that maps physical microstates at any given time t_0 to the states that they will evolve into at a future time $t = t_0 + \Delta t$. All viable modern theories of fundamental physics exhibit this bijectivity property. (b) Bit erasure and entropy increase. A computational state is a set of equivalent physical states. When two equiprobable computational states are merged, the entropy of the non-computational state necessarily increases by at least $k_B \ln 2$ (1 bit’s worth of entropy) due to bijectivity.

via an injective (not many-to-one) transformation. In standard expressions of fundamental dynamics, such as the unitary time-evolution operator of quantum mechanics, the transformation is also deterministic (not one-to-many); thus, it is a one-to-one (bijective) function, as illustrated in Fig. 1(a). Bijectivity directly implies the Landauer limit, which states that the minimum energy dissipation for an operation that loses (erases or overwrites) one bit’s worth of computational information is $k_B T \ln 2$. To understand why, see Fig. 1(b), which illustrates two computational states c_0, c_1 (sets of computationally equivalent physical microstates) merging to a single resultant state. If c_0 and c_1 are equally probable before the operation, the entropy of the computational state is $k_B \ln 2 = 1$ bit. After the operation, the entropy of the computational state is 0. Therefore, the entropy of the non-computational state (*i.e.*, the entropy of the physical state, conditioned on the computational state) has increased by the same amount, if the dynamics is known, so that the probability mass simply follows the arrows. If the dynamics is uncertain, then it is effectively a one-to-many stochastic transformation, and entropy can increase by a greater amount. Whatever amount ΔS of entropy is ejected from the computational state, it must eventually be emitted into a thermal environment at some temperature T ; this requires the dissipation of energy $\Delta E = T\Delta S$ to the form of heat in that environment. Thus, the minimum energy dissipation for bit erasure is $\Delta E_{\min} = k_B T \ln 2$. This constitutes a simple proof of Landauer’s limit, showing that it follows from basic thermodynamics.

III. THE ENERGY MINIMIZATION OF NON-TRIVIAL CIRCUITS

We consider here those computations facilitated by circuit-level manipulations, especially those functions that are usually represented by large combinational logic blocks, which as a rule of thumb can often be considered to be those practically limited by a circuit depth less than two dozen fanout of four (FO4) logic elements. Such circuits typically scale polylogarithmically in the size of the input. A cascade of such circuits, punctuated by registration circuits (storage elements, such as flip-flops or

register arrays) constitutes the dominant design pattern of clocked sequential circuitry. Such circuits (finite state machines) are themselves a building block for more complex designs, such as Turing machines.

How can we even estimate the minimum energy bound for circuits and algorithms? For circuits in a given traditional logic family, computing a given function of fixed-size inputs, an optimally energy-efficient solution can be found, in principle, by enumerating all circuits in the logic family in order of increasing complexity, and testing them on all possible inputs until the minimum-complexity circuit that computes the desired function is found. However, such an exhaustive search is generally not tractable, and the problem is made more difficult when we widen the design space to encompass arbitrary circuits that trade off hardware complexity against energy efficiency. It may even be that finding the globally most energy-efficient physical mechanism that can compute a given function is generally uncomputable, analogously to how, in the theory of Kolmogorov complexity, finding the simplest program that outputs a given bit-string is uncomputable [7]. We may recognize more efficient circuits when we discover them, but we will never know if our search is complete.

With those points in mind, we consider the energy consequences of at least one well-known algorithm, the SHA-256 hashing function, which is at the heart of the Bitcoin economy.

Bitcoin is a cryptocurrency that has gained immense popularity as an electronic payment system since its inception in 2008 [8]. The novelty of the Bitcoin transaction model is that it does not depend on an intermediary to validate transactions, but rather, the transactions are validated by a distributed network of machines known as “miners.” The miners compute cryptographic hash functions which transform data into a string with fixed length. Hash functions are essentially impossible to reverse (*i.e.* you cannot predict the input that will lead to a specific hash value). The data includes a hash value depending on information about all previous Bitcoin transactions, a new transaction block and a nonce, or random, value that is adjusted until the Bitcoin difficulty criteria is satisfied—the hash must start with a certain number of leading zeros. When a miner is successful in meeting the difficulty requirement, the validated transactions are added to the blockchain (public transaction ledger) and the successful miner is rewarded with payment. The goal for any miner is to make more money from transaction rewards than they spend for the energy and hardware to perform the computations.

The insatiable drive to compute the SHA-256 hashing algorithm with greater efficiency (lower power and higher performance) led to the rapid evolution from general purpose processing on a CPU to the development of highly optimized Application Specific Integrated Circuits (ASICs). To the best of our knowledge, there is no better case of a single algorithm running across multiple hardware architectures and technology generations than that of the SHA-256 hashing algorithm employed to validate Bitcoin transactions.

We performed an extensive literature review of published efficiency data for Bitcoin mining. Fig. 2 illustrates the energy-per-operation (J/GHash) of SHA-256 hashing algorithm imple-

mentations on four different computing architectures: the General Purpose CPUs, Graphics Processing Units (GPUs), FPGAs, and ASICs. It is a comprehensive survey, compiled from a variety of sources [9-20]. The efficiency data is plotted across multiple processor technology generations from 130 to 16 nm, representing approximately 15 years of Moore’s law scaling. A staggering 10^7 improvement in efficiency was realized in less than a decade by moving from a software implementation of SHA-256 executed in a CPU to the SHA-256 algorithm implemented in hardware on an optimized custom ASIC!

There is a predictive quality to the data presented in Fig. 2; specifically, the plot establishes a baseline for the relative gain in efficiency that one could reasonably expect from transitioning to a smaller technology node or computing architecture. For example; let’s assume there is an application running in a 55 nm GPU, and the user of this application requires a $100\times$ improvement in power efficiency. The data would suggest that even the most advanced GPU technology would only provide another $10\times$ net improvement in efficiency. To realize a $100\times$ improvement, a hardware design in a 40 nm FPGA or a custom ASIC would be required. This type of predictive analysis provides valuable insight to system engineers who need to make decisions about what technologies will meet their system requirements.

1) Estimating Landauer limit for the bitcoin hash algorithm

One of the difficulties in assessing energy minimization for computations based on circuits is a methodology that is traceable to the thermodynamic bounds implied by the Landauer limit for irreversible bit operations. One methodology would have considered it as a crude approach referred to as “Landauer substitution.” The method is a simplistic brute force approach in which every primitive bit level operation is assumed to be replaced with an idealized operation at the Landauer bound. If a digital logic circuit for example can be rationalized as having 50 bit-level operations, the bound based on Landauer substitution would be $100 k_B T$. The method is crude, because we cannot be certain of the actual equivalence of every logic operation as having the same Landauer bound, and there may be practical considerations that preclude the possibility of achieving this lower bound. Nevertheless, this approach gives us some minimal implement for which to examine energy reduction targets. Minimal forms for many classes of circuits have been studied extensively through logic synthesis and computational complexity theory. In these cases, Landauer substitution can provide us an estimated lower bound.

The Bitcoin Hash algorithm is the so-called double SHA-256 hashing algorithm in which the output of the SHA-256 algorithm (NIST) is passed through the SHA-256 algorithm again. The core 32-bit algorithm underlying SHA-256 operates on 512-bit (64 byte) blocks of data using a fixed number of iterations of a set of bit-level 32-bit operations including ANDs, NOTs, XORs, bit shifts/rotations (circular bit shifts), and load/store operations. Hence, it is relatively straightforward to perform a back of the envelope Landauer analysis by counting up the numbers of each type of 32-bit operation and assigning to each of these a Landauer energy upper bound in multiples of $k_B T \ln 2$. For our purposes, we take the coefficient of $k_B T \ln 2$ to be the product of the number of bits and a factor related to the number of

transistors used in implementing the operation. We estimate approximately $440,803 k_B T \ln 2/\text{hash}$ ($=1.45\times 10^{-15}$ J/hash= 1.45 fJ/hash) for the Landauer-based energy assumption: We indicate this is a lower bound, but likely not *the* lower bound.

IV. COMPUTING STYLES, ALGORITHMS, AND THE END OF ARCHITECTURES (WE STILL AREN’T BEATING LANDAUER)

We briefly consider other factors that affect energy optimization of large-scale computation.

A. The Moore’s Law Progression of Power Reduction

Since the inception of the integrated circuit, Moore’s law has resulted in gains in (per transistor) cost, density, performance (propagation delay), and energy consumption. The limits cannot be taken for granted. Energy scaling has been compromised by the more complex material and geometry limitations in sub-16 nm device configurations. As such, we can no longer rely on Moore’s law alone for future power reduction.

B. Getting Rid of Clocks (Asynchronous Methods)

The vast majority of digital circuitry is clock-mode or syn-

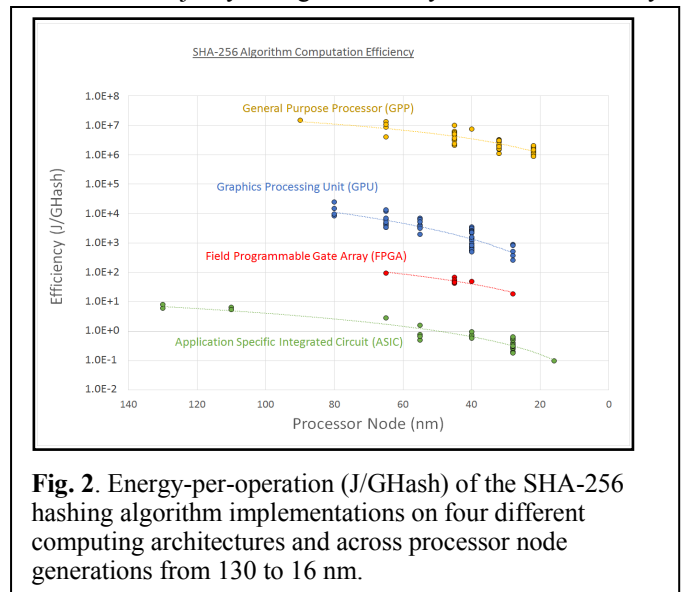


Fig. 2. Energy-per-operation (J/GHash) of the SHA-256 hashing algorithm implementations on four different computing architectures and across processor node generations from 130 to 16 nm.

chronous. In clock-mode designs, the clock is always supplied, whether the inputs to a logic stage change are not. Most digital designs can have large sections of logic that are relatively idle compared to the most active portions. In those cases, synchronous design can be energetically wasteful.

There are several broad strategies for minimizing these clocks or getting rid of them all together. Asynchronous digital design approaches eliminate clocks, predominantly by using handshaking signals to replace the registration approaches that involve clocked circuits (e.g., flip-flops). Instead of constant clocking of all circuit elements, computations are self-timed, based for the most part only on activity. As such, circuitry with large inactive regions become more efficient [21]. Another design pattern is the so-called “globally asynchronous, locally synchronous” (GALS) approach [22].

C. Dynamic Voltage Scaling, Activity gating

Dynamic voltage scaling (DVS) exploits the quadratic benefit in reducing voltage ($P \propto CV^2f$, where C is capacitance, V is voltage, and f is clocking frequency). Reduced voltage affects slew rate and noise margins, hence it is usually accompanied by reduced clock frequency. DVS has been standard practice for many years. Unfortunately, in modern CMOS processes, static power (which is negligible in ideal CMOS) has become a more dominant fraction of the total power budget in complex designs due to increased leakage current. As such, more sophisticated designs employ approaches like activity gating, which suspend clocking altogether, effectively eliminating the dynamic power contribution.

While these and other methods have become standard practice in modern design, activity reduction in a larger sense contributes to the problem referred to as “dark silicon” [23]. Dark silicon refers to the consequence of nontrivial fractions of integrated circuits being inactive due to thermal management considerations. While inactivity seems like a good idea for energy reduction, inefficient silicon is not considered the best way to achieve it. In cases where inactivity is a byproduct of how silicon works in a given application, approaches such as DVS and activity gating are effective tools. When performance is thermally limited, however, dark silicon is an undesired consequence of a particular strategy to cope with the problem.

D. Reduced Precision

Representing precise numbers in calculations requires longer numeric formats and these drive circuit size. Floating point representations drive size, complexity, and energy. One way to reduce energy is to use reduced / variable precision approaches. The implementations are particularly popular in FPGAs [24], since it is possible to fluidly shape the precision throughout an algorithmic processing chain, choosing to implement low precision in some stages and higher precision in other parts.

E. Probabilistic/Inexact

A variety of probabilistic computation and inexact computing [25] concepts have been proposed, many of these embracing the notion of achieving “good enough” results, based on approaches where substantial improvements in energy efficiency can be gained for tolerable losses in accuracy or precision. The terms “probabilistic” or “inexact” can be assessed in many different contexts.

F. Reconfigurable Computing

Much has been said about the benefits of reconfigurable computing [26] [27], and we shall not undertake a full discussion here. Conceptually, the approach involves the structuring of configurable digital “fabrics” (in which logic, memory, and interconnect can be arbitrarily shaped to optimally solve a computing problem. It is well established that in some use cases, FPGA-based computers can significantly outperform traditional ones. As illustrated in our Bitcoin discussion, it appears that the narrowly focused problem is best solved with custom ASICs. FPGAs, being configurable fabrics, carry with them the overhead necessary to support this flexibility. ASICs, for the most part, can be brittle, being unable to accommodate variations in a given problem. Hence, in cases where the

overhead in size, power, and performance can be compensated by flexibility, FPGAs can give better results.

G. Analog and Neuromorphic Approaches

There has been a long-standing debate as to whether computations are more efficient in analog or digital form. Some argue the world is analog, and even digital logic is simply a restriction placed on an analog circuit. As such, should we not expect a better result when we can render problems in a form amenable to direct computation by manipulating continuously variable signals within like building blocks?

Neuromorphic computing is a broad category, covering any computation that takes advantage, either algorithmically or physically, of potential advantages of using paradigms derived from biological neural systems. The term *neuromorphic* was originally suggested by Mead [28] to exploit device physics to mimic neural behavior, noting even the simplest biological neural system far outstrips the energy efficiency of digital implementations. In recent years there has been an explosion of activity using so-called deep neural networks, with many hidden layers, paradoxically using digital methods. This activity can be categorized as machine learning, where a network is trained offline, prior to live use. More recently, a thrust to develop machine intelligence—behavior that would be closely related to activity in the mammalian neo-cortex—has emerged [29]. Hallmarks of cortical processing are online, continuous learning, a hierarchical memory architecture, and a remarkably high level of feedback between input sensors and the neural substrate. As both machine learning and machine intelligence are currently of great interest, we adopt the terms “neuromorphic computing” and “cortical computing” to describe each. The distinction is important because their computation architectures and energy consumption to be radically different. They can be CMOS-based or employ newer technologies, such as memristors or coupled oscillators [30].

V. BEATING LANDAUER WITH REVERSIBLE APPROACHES

The derivation of Landauer’s limit (in its usual $k_B T \ln 2$ form) depends upon the assumption that the computational operation that is performed merges two equally-likely computational states. We can say that such an operation is *logically irreversible*, since the identity of the initial computational state cannot be recovered if only the final computational state is known. In contrast, we can consider computational operations that are *logically reversible*, in which the transformation of the computational states (or at least, the subset of states with non-zero probability) is one-to-one. In such a case, computational entropy is not ejected into the non-computational state, and so Landauer’s limit does not apply. This leads to the concept of *reversible computing*, that is, computation using logically reversible operations that avoid information erasure, as the only possible way to circumvent Landauer’s limit. In reversible computing, energy efficiency is limited only by the degree to which computational operations manage to conserve non-computational entropy through avoidance of parasitic energy loss mechanisms; *i.e.*, there is no known *fundamental* thermodynamic limit to the degree of energy reduction.

A. Reversible Computation

Typical approaches to reversible computing involve the adiabatic (asymptotically lossless) transformation of the state of the computational mechanism between distinct logical states under external control. For example, in an adiabatic version of ordinary voltage-coded CMOS logic circuits, conventional gates driven by constant power/ground rails would be replaced by new gate designs driven by externally-supplied power/clock signals that adiabatically transition between logic levels, causing the logic gate's output to transform between (say) some initial, default level, and the computed result of the gate. As such transformations are carried out more gradually, the energy dissipated per logic transition decreases. To ensure adiabaticity, one must never turn on a transistor with a voltage across it, or turn off a transistor that is carrying a current. In a CMOS circuit satisfying these conditions, the state transformation constitutes a conditionally-reversible operation [31] that is capable in principle of circumventing the Landauer limit, if sufficiently low-leakage transistors are utilized. An engineering challenge for adiabatic circuits that requires additional development effort is to design sufficiently high-quality resonant signal generators that can deliver and subsequently recover almost all of the energy in the clock/power signals that drive the adiabatic transitions. These variant CMOS circuits can approach and in principle even transcend the Landauer limit.

B. Extending Adiabatic Concepts to Asynchronous

One problem with existing adiabatic approaches to reversible computing is that they are thoroughly synchronous, that is, with the timing of every transition of every logic gate explicitly timed and controlled by ubiquitous clock signals. Distributing these clocks throughout the logic adds a fairly large overhead to the hardware complexity of implementing adiabatic reversible computations. An alternative approach that is just beginning to be explored is the possibility of asynchronous reversible computation [32], in which ballistically-propagating signal pulses pass, one at a time, through reversible devices, possibly altering their state. This approach can reduce the clocking overhead of reversible solutions dramatically, although occasional irreversible re-synchronization of signals is still needed. However, the technology readiness level of the asynchronous reversible approach is currently still very low.

VI. INFORMATION-CENTRIC ENERGY ASSESSMENTS OF SPACECRAFT MISSION

We have considered a number of facets on minimizing the energy of computation. But what do they mean for platforms, such as spacecraft? Or, conversely, how can we achieve the most computation possible for a given Joule of "investment?"

As an example, an imaging spacecraft may collect data from the earth. They almost always require contact with ground stations to receive commands and to export telemetry (it could be processed or unprocessed results of collections, combined with health and status data). In some cases, spacecraft can interact with users. Other missions, such communications and space situational awareness, offer slight variations in this theme. The primary differences are in the use of sensors (space situation awareness sensors examined the space environment instead of collecting data from Earth) and movement of information (communications missions predominantly are concerned with

relaying data between terrestrial users and possibly other spacecraft, and may not have any primary sensors).

A. A Day in the Life of the Spacecraft

The spacecraft is a particularly constrained platform. It must live on the photons it can harvest from the sun, and the mass of a spacecraft is limited by the economics of spacelift. A crude rule of thumb suggests that a typical spacecraft operates on ~ 1 - 2 W/kg, meaning that many small spacecraft (arbitrarily considered to be platforms < 1000 kg) deal with budgets comparable to a portable hair dryer.

Platform energy is about supply and demand. The source for almost any spacecraft is the sun. A small fraction of the available solar energy is intercepted by solar panels and converted into a form accessible to the spacecraft. The spacecraft itself is divided into the "payload" (which performs the mission) and a vehicular portion referred to as the "bus" (the conveyance that delivers the payload into operation and handles its "care and feeding"). Every Joule of energy harvested is not only invested in payload, but also in the bus. Within the payload, we similarly experience a variety of overhead factors, such as the energy cost to point mirrors, cool focal plane arrays, and these further erode the Joule balance available to contribute to information processing.

B. "Custody Chain" of Energy

In the custody chain approach, we start with the source of energy harvested, usually with photovoltaic arrays, and then we follow its course through the spacecraft after conversion and regulation. Even the collection process can have a complex set of "care and feeding" requirements, such as control systems and actuators to maintain effective sun pointing. Distribution occurs often from a single "golden node," subject to further losses through a hierarchy of power regulators, bus wiring, and the interconnections of packaging structures that eventually result in the delivery of power to discrete and integrated circuit components. At the end of this custody chain, we finally can invest a *fraction* of the originally possible energy (as determined by solar flux calculations) into the electrical subsystems for use by the spacecraft bus and payload.

C. Energy-Based Accounting

The energy custody chain, when combined with "day in the life analyses," may be a useful concept for assessing impacts in spacecraft engineering and mission design, to include technology investments. One concept for comparing spacecraft design or technology options involves defining benchmarks for typical mission templates, such as a family of orbital models. These benchmarks can be evaluated to some definition of the normal or nominal baseline. This "norm" can then be compared against variations. We can examine if the substitution of a new technology or alternative component yields a net change in Joules remaining at the end of the benchmark evaluation.

D. The Energy Footprint of a Spacecraft Mission

The methodology suggested by the previous analysis need not be restricted to a platform, but could be extended to encompass an entire space mission. An overall space mission may in fact involve constellations of multiple spacecraft, interacting with many terrestrial nodes. We see why it may make sense to consider an extended energy footprint when we observe, for

example, a smartphone can have a larger energy footprint than the phone itself. The use of the smartphone touches cell towers, data servers, and has a variety of indirect factors that influence consumption of energy, resulting in an energy impact similar to a small refrigerator [33]. We may find other unexpected consequences the extended energy footprint of spacecraft. We may find, counterintuitively, that the more information we move to the ground, the larger the energy footprint. A deeper assessment may be needed to examine the ultimate purposes of the information and how spacecraft can be most optimally used to provide information on demand as opposed to “all the time.”

VII. CONCLUSIONS

While the relationship of energy consumption and information is an obvious limiting factor in many consumer devices such as laptop computers and smartphones, it is also becoming an increasingly important consideration facing a wider variety of platforms, such as spacecraft. One of the long-standing windfalls of Moore’s law—in addition to increased number of transistors per unit area and cost reduction per transistor—has been a dependable and dramatic reduction in the amount of energy per computing operation. At the time of this writing, many of these benefits have come into question, but the demand for information “services” has not slowed. If anything, there is an expectation of a continued progression.

In this paper, we seek to (1) increase appreciation of the increasingly pivotal relationship between energy and information in embedded systems; (2) identify the need to better understand lower energy balance for nontrivial functions and algorithms; and (3) understand how the relationship of energy and information might lead to a better understanding of product/mission lifecycles and can be used to guide better technology investments or mission designs. Physics informs us that there is no free lunch, and the reconciliation is important.

ACKNOWLEDGMENTS

We gratefully acknowledge support of the Air Force Office of Scientific Research and the Department of Energy.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

- [1] Kumagai, J. "5 Earth-imaging start-ups coming to a sky near you [News]," in *IEEE Spectrum*, vol. 51, no. 4, pp. 20-21, April 2014.
- [2] Swartwout, M. "Cubesat Database," as of March 2017 (<https://sites.google.com/a/slu.edu/swartwout/home/cubesat-database>).
- [3] Papadimitriou, Christos (1993). "Section 15.3: The class NC." *Computational Complexity* (1st ed.). Addison Wesley. pp. 375–381.
- [4] Landauer, R. Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.* 5, 183–191 (1961).
- [5] Richard P. Feynman and Anthony Hey, "Feynman Lectures on Computation", Westview Press (2000).
- [6] Berut, A. *et al.*, "Experimental verification of Landauer’s principle linking information and thermodynamics." *Nature* 483.7388 (2012).
- [7] Ming Li and P.M.B. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications (Texts in Computer Science)*, Springer; 3rd ed. 2008.

- [8] Bitcoin Wiki, "Mining Hardware Comparison" (https://en.bitcoin.it/wiki/Mining_hardware_comparison)
- [9] BitFury Group, "16nm ASIC Chip", as of April 2017 (<http://bitfury.com/products#16nm-asic>)
- [10] Spondoolistech "Products", as of April 2017 (<http://spondoolistech.co/collections/products.html>)
- [11] Canaan, "AvaloMiner 741" as of April 2017 (<https://canaan.io/product/avalominer-741>)
- [12] Bitmain, "AntMiner S9 Worlds first consumer grade 16nm ASIC", as of April 2017 (<https://www.bitmaintech.com>)
- [13] Xilinx, "Spartan 6 FPGA Family", as of April 2017 (<https://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html>)
- [14] Xilinx, "Artix 7 FPGA Family", as of April 2017 (<https://www.xilinx.com/products/silicon-devices/fpga/artix-7.html>)
- [15] Altera, "Stratix III", as of May 2016 (<https://www.altera.com/products/fpga/stratix-series>)
- [16] Bitcoin Wiki, "Graphics Cards", as of April 2017 (https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison#Graphics_cards)
- [17] TechPowerUP, "GPU Database", as of April 2017 (<https://www.techpowerup.com/gpubd>)
- [18] Bitcoin Wiki, "Non-specialized hardware comparison", as of April 2017 (https://en.bitcoin.it/wiki/Non-specialized_hardware_comparison)
- [19] Intel, "Product Specifications", as of April 2017 (<http://ark.intel.com/products>)
- [20] AMD, "Processors for Desktops, Laptops and Servers", as of April 2017 (<http://www.amd.com/en-us/products/processors>).
- [21] Alain Martin and Mika Nystrom. "Asynchronous techniques for system-on-chip design." *Proceedings of the IEEE* 94.6 (2006): 1089-1120.
- [22] A. Rajakumari, N. S. Murthy Sharma, K. L. Kishore and V. K. Petta, "A power gating GALS interface implementation," 2013 IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Visakhapatnam, 2013, pp. 34-39.
- [23] M. Shafique and S. Garg, "Computing in the Dark Silicon Era: Current Trends and Research Challenges," in *IEEE Design & Test*, vol. 34, no. 2, pp. 8-23, April 2017.
- [24] G Alonzo Vera, Marios Pattichis, James Lyke, "A dynamic dual fixed-point arithmetic architecture for FPGAs", *International Journal of Reconfigurable Computing*, Volume 2011 (2011), Article ID 518602, 2011/3/29.
- [25] Palem/Linganeni, Krishna Palem and Avinash Lingamneni. 2013. Ten Years of Building Broken Chips: The Physics and Engineering of Inexact Computing. *ACM Trans. Embed. Comput. Syst.* 12, 2s, Article 87 (May 2013).
- [26] J. C. Lyke, C. G. Christodoulou, G. A. Vera and A. H. Edwards, "An Introduction to Reconfigurable Systems," in *Proceedings of the IEEE*, vol. 103, no. 3, pp. 291-317, March 2015.
- [27] A. DeHon, "Fundamental Underpinnings of Reconfigurable Computing Architectures," in *Proceedings of the IEEE*, vol. 103, no. 3, pp. 355-378, March 2015.
- [28] C. Mead, "Neuromorphic Electronic Systems," *Proceedings of the IEEE*, vol. 78, Oct. 1990.
- [29] While the definition of intelligence in general may be controversial, we adopt herein Hawkins conjecture that cortical intelligence is prediction—that we predict what we will see next, and when this differs from the construction from sensory input, we alter our model.
- [30] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature Nanotechnology*, vol. 8, pp. 13-24, 2013.
- [31] Frank, M. P. "Foundations of Generalized Reversible Computing," to appear, *Reversible Computation (RC 2017)*, Lecture Notes in Computer Science, vol. 10301. Springer, 2017, pp. tbd
- [32] Frank, M.P., "[Generalized Reversible Computing, Truly Adiabatic Circuits, and Asynchronous Ballistic Logic](#)," *Presentation*, Energy Consequences of Information Workshop, February 2017.
- [33] Smil, Vaclav, "Your Phone Costs Energy—Even Before You Turn It On", *IEEE Spectrum*, 26 April 2016.