How Device Properties Influence Energy-Delay Metrics and the Energy-Efficiency of Parallel Computations

Phillip Stanley-Marbell
MIT

ABSTRACT

Semiconductor device engineers are hard-pressed to relate observed device-level properties of potential CMOS replacements to computation performance. We address this challenge by developing a model linking device properties to algorithm parallelism, total computational work, and degree of voltage and frequency scaling. We then use the model to provide insight into how device properties influence execution time, average power dissipation, and overall energy usage of parallel algorithms executing in the presence of hardware concurrency. The model facilitates studying tradeoffs: It lets researchers formulate joint energy-delay metrics that account for device properties.

We support our analysis with data from a dozen large digital circuit designs, and we validate the models we present using performance and power measurements of a parallel algorithm executing on a state-of-the-art low-power multicore processor.

CCS Concepts

•**Hardware** → **Chip-level power issues; Emerging technologies;** *Logic circuits;* •**Theory of computation** → **Parallel algorithms;** •**Computer systems organization** → **Parallel architectures;**

Keywords

Transistors, Parallelism, Energy-Efficiency, Power, Measurement

1. INTRODUCTION

There are several candidate device architectures to carry computing systems through deeply-scaled CMOS and beyond. The candidate devices range from band-to-band tunneling field-effect transistors (TFETs) [16] and nanoscale-electro-mechanical-system (NEMS) relay logic [6], to devices employing electron spin [13], and graphene [15]. These varied devices often have different characteristics from traditional bulk CMOS devices. For example, NEMS proposals have limited achievable clock speeds; they however have very low leakage, potentially permitting designs with large transistor counts that make up for their limited clock speeds by employing architectural parallelism.

In addition to the many alternative devices and alternative tokens for representing logic state (charge, spin, and so on), devices of a given type may be tuned for different regions of operation. Fundamental to the choice among operating regimes, devices, and architectures, are performance, power dissipation, potential for dense integration, and the opportunities for tradeoffs between these. The device characteristics which should be pursued by device engineers will therefore depend on the existence of a set of metrics which capture the constraints under which devices will be operated.

In this article, we derive a set of relations linking algorithm parallelism to device properties, and to tradeoffs between performance, power, and energy. The analysis we present is based on devices that represent logic values with voltages, and in which logic transfer between stages is via the charging of a capacitive load. For devices with other state tokens (e.g., electron spin), our analysis can still serve as a basis for extension. In this work, we present:

- **The derivation of relations between algorithm parallelism and device properties**, presented in Section 3.
- **Derivation and new insight into what metrics should be used for comparing joint energy-efficiency and performance, as a function of device characteristics**, and under what conditions these metrics are valid (Section 4).
- **Experimental measurement of power consumption and performance** of a parallel algorithm under voltage scaling on a state-of-the-art multi-core processor (Section 5).

2. RELATED RESEARCH

Theis and Solomon [21] outline two methods for reducing power dissipation in future devices: Reducing the energy lost during logic value transitions by lowering supply voltages, and using adiabatic logic. Given the challenges involved in designing efficient adiabatic logic circuits, the device research community has thus far focused on finding alternative logic devices that enable a significant lowering of supply voltage.

Dynamic supply voltage and frequency scaling (DVFS) in microprocessors as a means of reducing power dissipation, has been of interest for several decades [23]. This long-standing interest has been due to the quadratic dependence of dynamic power dissipation on supply voltage, for a given implementation circuit. Lowering supply voltages to reduce power dissipation however often leads to a loss in performance (although the overall energy usage is still usually reduced). This is because drain current (and hence gate delay) depends on supply voltage. For long-channel devices, this dependence, captured by the Shockley model, was linear in the region of transistor operation typically of interest. For short-channel devices, the improved delay model of Sakurai and Newton [14] generalized the Shockley model to account for velocity saturation.

HotPower '15, October 04-07, 2015, Monterey, CA, USA.
© 2015 ACM. ISBN 978-1-4503-3946-9/15/10. . . $15.00
DOI: 10.1145/2818613.2818744

Given the conflicting influences of supply voltage on performance of a fixed circuit (higher is better) and energy-efficiency (lower is usually better), it has been of interest to jointly consider both energy-efficiency and delay in quantifying system efficiency. For this, the energy-delay product [8] is often used. However, as noted by Pénzes and Martin [10], the energy-delay product is dependent on supply voltage, thus conclusions reached in comparing the energy-delay for two systems at one supply voltage might change when the systems operate at different supply voltages. They thus proposed the use of energy-delay^2 ($E \cdot T^2$), which they showed, empirically for a design in 0.6 μm CMOS, to be largely independent of supply voltage. We generalize this idea in Section 4 with the concept of *parameter-independent metrics*, with the voltage-independent metric of Pénzes and Martin being a special case, and we demonstrate how the parameter-independent metrics are functions of device technology parameters.

The joint treatment of device properties, algorithm properties, and the resulting performance and energy-efficiency that we present in this article provides new insight into prior efforts [2, 7, 9] to investigate the energy-efficiency of the use of parallelism.

3. ENERGY AND PARALLELISM MODELS

The power dissipation of a CMOS transistor can be decomposed into the primary components of dynamic, short-circuit, gate, and sub-threshold channel leakage. The analysis that follows will focus on the dynamic power dissipation and sub-threshold channel leakage; gate leakage has been addressed in recent years through the use of high-κ dielectrics, while short-circuit currents are typically small when signal rise and fall times are short.

3.1 Energy model

The energy for operation of a CMOS circuit at clock frequency f and supply voltage V, with effective circuit switching capacitance C_{eff}, for an execution duration T, is given by

$$E = C_{\text{eff}} \cdot V^2 \cdot f \cdot T + I_{\text{lkg}}(V, V_T, \theta) \cdot V \cdot T, \qquad (1)$$

where

$$I_{\text{lkg}}(V, V_T, \theta) = K_{\text{lkg},1} \cdot e^{\frac{K_{\text{lkg},2} \cdot q(V - V_T)}{k \cdot \theta}}.$$

V_{T}, is the threshold voltage, $K_{\text{lkg},1}$ and $K_{\text{lkg},2}$ subsume several device properties, k is Boltzmann's constant, q is the electron charge, and θ is the operating temperature in Kelvin.

3.2 Capturing the relation between V and f_{max}

Supply voltage also influences the gate drive current, which in turn determines the speed at which capacitive loads can be charged and discharged. The supply voltage therefore determines the *maximum clock frequency*, f_{max}:

$$f_{\text{max}} = \phi \frac{(V - V_{\text{min}})^\alpha}{V}. \qquad (2)$$

The constant ϕ subsumes several device and circuit parameters, and is treated as a monolithic constant in this work. For devices that operate purely in the super-threshold region, V_{min} equals V_{T}; for devices which span the sub- and super-threshold regions, V_{min} however loses its physical interpretation. The parameter α, which must be greater than or equal to unity[1], is treated in this work as a parameter with no direct physical interpretation.

Although the alpha-power-law voltage-versus-frequency dependence was originally derived by Sakurai and Newton [14] to account for short-channel effects (velocity saturation) in CMOS, it is observed to capture the behavior of a wide variety of circuits, even those with mixed super- and sub-threshold modes. To illustrate this, Figure 1 plots published voltage versus frequency "Shmoo" characterizations for 12 large programmable digital designs from the *IEEE International Solid-State Circuits Conference (ISSCC)*, along with the resulting multi-parameter fit to α.

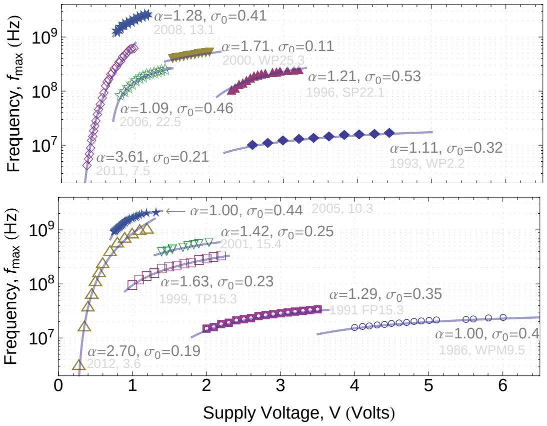

Figure 1: Empirical data from voltage-versus-frequency characterizations (points) and fits to Equation 2 (lines), for several large digital designs published in ISSCC (1986–2012).

The values of α providing the best fit, shown in Figure 1, range from 1.0 to 3.61. Both of the designs with large values of α ("2011, 7.5" [5] and "2012, 3.6" [12]) were explicitly designed to enable a wide supply operating voltage range. In what follows, we therefore treat α as a parameter that may be controlled even for a fixed technology node.

3.3 Decoupling V and f_{max}

The frequency f at which a circuit operates (in Equation 1) can be chosen at will under the constraint $0 < f \leq f_{\text{max}}$; doing so while leaving supply voltage fixed however results only in a reduction in average power, but no gain in energy-efficiency. The following analyses are therefore restricted to the mode of operation where the supply voltage is always the lowest for a given target operating frequency.

We use V_{min} and V_{max} to denote the minimum and maximum supply voltages at which a system operates (technology parameters), and we use σ to denote the degree of voltage scaling (a system-configuration-dependent parameter). Let $\sigma = 1$ indicate no voltage/frequency scaling, and let $\sigma = \sigma_0$ denote a maximally-scaled voltage, i.e., $\sigma_0 < \sigma \leq 1$, and

$$\sigma_0 = \frac{V_{\text{min}}}{V_{\text{max}}}. \qquad (3)$$

Expressed in terms of σ and σ_0, Equation 2 becomes

$$f_{\text{max}} = \phi V_{\text{max}}^{\alpha - 1} \frac{(\sigma - \sigma_0)^\alpha}{\sigma}. \qquad (4)$$

The supply operating point (σ) employed in a system will depend on the desired tradeoff between performance and energy-efficiency, and, importantly, on the possibility to make up for lower clock frequencies through the use of architectural parallelism.

3.4 Algorithm parallelism model

The dynamic execution of an algorithm can be represented with a data dependence graph, a directed acyclic graph (DAG) in which nodes are units of work and edges represent dependencies. These units may be instructions, basic blocks, or coarser. In the DAG model for dynamic parallelism [3], on which the following analysis is based, the units are sections of the dynamic instruction stream

[1] $\alpha < 1$ would permit *decreasing* power dissipation with *increasing* performance.

between points of creation or merging of parallel threads.

The number of nodes in the execution DAG constitutes the total amount of *work*, W_1, that must be completed. In a serial execution, this corresponds to the computation performed by a single processor. The length of the longest dependence chain of work units, or the *span*, is denoted by W_∞, and the average amount of parallelism, in units of work, over the course of execution, is W_1/W_∞.

In an execution employing p processors, the available parallelism must be at least p in order to achieve linear speedup. We restrict our analysis in this work to computations which occur in the region where there is sufficient algorithm parallelism for the chosen number of processors. Under these conditions, and assuming perfect load balancing, the work per processor, W_p, is

$$W_p = \frac{W_1}{p}. \tag{5}$$

For the remainder of the analysis, we assume that communication overheads are minimal, to simplify the derivation of relations for the interaction between algorithm parallelism and device properties. For applications with significant amounts of communication, we have recently derived analogous expressions for performance and power [19]. As we will demonstrate in Section 5, there are important real-world problems for which these assumptions of algorithm parallelism and communication overheads hold. The model presented can be extended further to capture properties such as limited parallelism and the memory hierarchy, by building on existing research into analytic models for computing system performance [4]; this is one area of future work.

Our succinct model of parallelism in the dynamic execution of algorithms can now be combined with the device-specific power and timing relations of Section 3.2.

3.5 How device properties affect performance, power, and energy-efficiency of parallelism

Given the definitions for maximum clock frequency and energy in Equations 1 and 2, and per-processor parallel workload in Equation 5, we reformulate the execution time, T, for a parallel computation, as

$$T = \frac{W_1}{p \cdot f_{\max}} = \frac{W_1 \cdot V_{\max}^{1-\alpha}}{p\phi} \frac{\sigma}{(\sigma - \sigma_0)^\alpha}. \tag{6}$$

Substituting Equation 6 into Equation 1 yields the expression for the energy usage of the parallel algorithm execution as

$$E = \frac{W_1}{p}\sigma^2 V_{\max}^2 \left(C_{\text{eff}} + \frac{I_{\text{lkg}}(V, V_T, \theta)(\sigma V_{\max} - \sigma_0 V_{\max})^{-\alpha}}{\phi} \right). \tag{7}$$

The average power over the course of the execution is thus also

$$P = \sigma V_{\max} \left(C_{\text{eff}} \phi (\sigma V_{\max} - \sigma_0 V_{\max})^\alpha + I_{\text{lkg}}(V, V_T, \theta) \right). \tag{8}$$

Equations 6, 7, and 8 encapsulate the relation between algorithm properties (W_1), hardware concurrency (p), implementation (C_{eff}), device properties ($V_{\max}$, α, ϕ, and σ_0), and system operating point (σ). To understand how a candidate transistor or circuit-level technique will influence the energy-efficiency of algorithms in the context of the model presented (e.g., for spin logic [1], nanowire TFET [17], or carbon nanotube FETs [22]), we must therefore be able to characterize $V_{\max}$, α, ϕ, and σ_0. One method to do this is to fit the model of Equation 4 to characterization data for the maximum operating frequency of the candidate device and circuit technology as a function of supply voltage scaling level. Examples of such characterizations were presented in Figure 1, and we present another more detailed study in Section 5.

Both execution time and energy usage can be reduced by more efficient algorithms (smaller W_1) or increased parallelism (larger p). The appropriate transistor and circuit characteristics and resulting values of α, ϕ, and $V_{\max}$ however depend on the desired performance versus energy tradeoff; we explore this further in Section 4.

Even though the relations presented thus far are structured based on transistor-level equations, they also accurately capture the aggregate behavior of the millions of transistors making up an integrated circuit such as a microprocessor. We show this in Section 5 by fitting data from empirical measurements for a multi-core processor to the models.

Because candidate CMOS replacements such as TFETs and NEMS have very low leakage compared to CMOS, we will focus our analysis in the following section on the dynamic component of energy, for simplicity.

4. PARAMETER-INDEPENDENT METRICS

When a single metric is of interest (e.g., only timing performance or average power dissipation), it is possible to use Equations 6 through 8 to determine which combinations of algorithms and system parameters satisfy a given time, energy, or power constraint.

In practice however, multiple metrics are often of interest. The traditional approach is to use a product of the metrics of interest, such as the energy-delay ($E \cdot T$) product proposed by Horowitz et al. [8]. Pénzes and Martin [10] previously argued that the $E \cdot T$ metric is voltage-dependent, arguing instead for $E \cdot T^2$. We generalize this concept further to the idea of *parameter-independent metrics*. Using Equations 6 and 7, the appropriate form of these parameter-invariant metrics can be formulated as functions of device technology parameters.

To minimize *both* energy and delay independent of a given parameter (e.g., supply voltage, V), an appropriate parameter-independent metric is of the form $E^a \cdot T^b$. The constants a and b are picked to be nonzero and such that all terms of the parameter from which independence is desired cancel in the product $E^a \cdot T^b$.

4.1 $V_{\max}$-independent metric

The maximum supply voltage, $V_{\max}$, at which a design operates, may be constrained due to power supply design, supply noise, supply current, or circuit reliability concerns. It may therefore be of interest to be able to compare algorithms paired with hardware designs, independent of specific values of $V_{\max}$.

From Equation 7, the dynamic component of energy is a function of $V_{\max}^2$, while delay (Equation 6) is a function of $V_{\max}^{1-\alpha}$. Thus, the $V_{\max}$-independent energy-delay product is achieved when

$$-2 \cdot a = (1 - \alpha) \cdot b,$$

with both a and b nonzero. One valid solution is achieved with $a = 1$ and $b = \frac{2}{\alpha - 1}$. For $\alpha = 2$ (Shockley model), the $V_{\max}$-independent metric is therefore $E \cdot T^2$. As α approaches unity, jointly minimizing energy and performance, independent of $V_{\max}$, requires placing more effort on minimizing delay.

4.2 W_1-independent metric

A W_1-independent metric is desired when comparing the steady-state behavior of an algorithm and platform combination, regardless of the total amount of computational work (W_1). An example of such a scenario is when studying the steady-state behavior of a streaming application.

From Equations 6 and 7, both energy and delay are functions of W_1 raised to the same exponent (unity). The W_1-independent energy-delay product is therefore achieved when $a = -b$, yielding

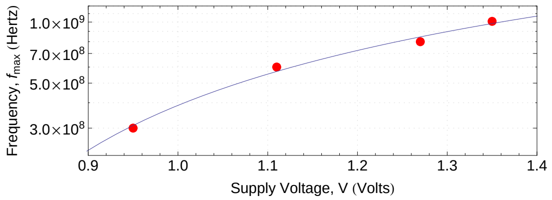

Figure 2: Voltage versus frequency dependence of the OMAP4430 (points), fit to the model of Equation 2 (line).

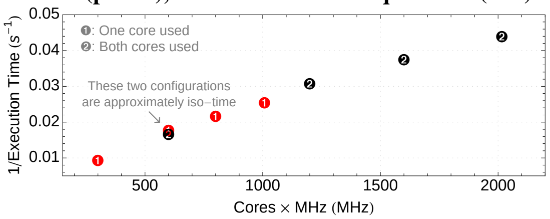

Figure 3: Measured performance scaling.

the product $E \cdot T^{-1}$. This is intuitively pleasing, as it corresponds to the average power dissipation.

4.3 Nonexistent independences

The dynamic component of energy for a parallel computation is a function of σ^2, while the delay is a function of $\sigma/(\sigma - \sigma_0)^{\alpha}$. For $\sigma \gg \sigma_0$ (i.e., when far above the minimum supply setting in a highly voltage-scalable system), delay becomes independent of the degree of supply voltage scaling, σ. This makes a truly σ-independent $E \cdot T$ metric unattainable: i.e., jointly minimizing energy and delay cannot be made independent of the degree of voltage scaling, σ.

A metric that is jointly independent of both V_{max} and W_1 has values of the exponents a and b of E and T respectively that satisfy the system of simultaneous equations

$$-a = b,$$
$$-2 \cdot a = (1 - \alpha)b,$$

which has no valid solutions given the constraint that $\alpha \geq 1$. Thus, one cannot jointly minimize energy and delay independent of both the total amount of computational work W_1 (algorithm dependent) and the maximum supply voltage V_{max} (technology dependent).

4.4 How device properties influence energy-delay metrics

Sections 4.1 and 4.2 provided formulations of the joint energy-delay metrics to be used under two different system usage models, yielding three main insights.

First, the metric of interest depends on the system's evaluation and usage criteria. For example, for a streaming workload, a W_1-independent metric is likely desirable, and the metric of interest is therefore $E \cdot T^{-1}$. Second, the V_{max}-independent metric, $E \cdot T^{\frac{2}{\alpha - 1}}$, is a function of α. Its precise form therefore depends on the voltage versus frequency characteristics of computing architectures implemented in a given device and circuit technology. ($E \cdot T^{-1}$, by contrast, is always the W_1-independent metric, independent of device properties.) Finally, jointly minimizing dynamic energy and delay cannot be made independent of the degree of voltage scaling.

5. EMPIRICAL MEASUREMENTS

The preceding sections outlined a model for capturing the interaction between device properties (α, σ_0, ϕ, V_{min}, and V_{max}), algorithm properties (W_1), implementation/architecture (C_{eff}), and the system operating point (σ). Although empirical values of device-level parameters were provided to support the argument, one question remains: Do the performance, energy, and power models of Equations 6, 7, and 8 truly reflect the behavior of complete integrated circuits executing parallel algorithms? To address this question, we carried out performance and power measurements of a parallel algorithm executing on a multi-core platform.

For the evaluation, we employed a cache-oblivious parallel matrix-matrix multiplication (MMM) kernel. Parallel matrix-matrix multiplication was chosen as a benchmark as matrix-matrix multiplication is a crucial subroutine in many compute-intensive scientific, machine learning, and commercial data analytics workloads. The kernel, which was written in the Cilk dynamic multithreading language [11], was run over the Cilk 5.4.6 runtime, which we ported to the ARM architecture to facilitate the experiments. For input data, 4 M-entry product matrices were employed, populated with uniformly distributed random data in the range of 0.0 to 1.0, to maximize switching activity in the processor datapath.

5.1 Measuring α, ϕ, σ_0, V_{min}, and V_{max}

For empirical measurements, we used an OMAP4430 dual-core ARM Cortex-A9 system-on-chip (SoC) from Texas Instruments, implemented in 45 nm CMOS. To facilitate our measurements, we modified a hardware evaluation board containing the target SoC to isolate the power supply rails for just the Cortex-A9 subsystem (`VCORE1` on the OMAP4430). This enabled us to measure power dissipated in just the processor cores, separate from power dissipated by other on-chip and board-level peripherals. For measurements, we used a Fluke 289 Logging Multimeter, sampling the voltage across a 40 mΩ resistor at 1 Hz, with 1 μV resolution.

The SoC's Cortex-A9 cores support execution at clock frequencies of 300 MHz, 600 MHz, 800 MHz, and 1008 MHz. The SoC contains a hardware subsystem ("SmartReflex") which cooperates with an external voltage regulator to set supply voltages, based on the requested clock frequency. On the test hardware platform, we measured core supply voltages at these aforementioned frequencies, of 0.95 V, 1.11 V, 1.27 V, and 1.35 V.

Figure 2 plots the measured supply voltage at the processor core (`VCORE1` on the OMAP4430) across operating frequencies. Fitting the measurements to Equation 2 yields values of the device technology parameters $\phi = 2.6 \times 10^9$, $\alpha = 1.69$, $V_{\text{min}} = 0.67$, $V_{\text{max}} = 1.35$, and $\sigma_0 = 0.49$.

5.2 Model, measurements, and observations

Figures 3, 4, and 5 show measurements of performance, total energy, and average power of the MMM application, for single- and dual-core configurations (i.e., $p = 1$ and $p = 2$). In the figures, the points represent measurements, and the dashed lines are the trends predicted by Equations 6, 7, and 8 for the model constants estimated in Section 5.1.

From Figure 3, the performance of the MMM application scales with increases in clock frequency, as well as with increases in the number of parallel threads. Even though the clock frequency doubles from 300 MHz to 600 MHz between the slowest two configurations in both the single- and dual-core workloads, this doubling of performance is accompanied by a smaller than two-fold increase in power dissipation (Figure 5). The energy for task completion is therefore reduced in going from 300 MHz to 600 MHz. As the clock frequency is further increased from 600 MHz to 1008 MHz, which requires operation at yet higher supply voltages, the rate of increase in power dissipation with clock frequency is observably higher for both the single and dual-core workloads (Figure 5). This observed behavior is determined by the device technology parame-

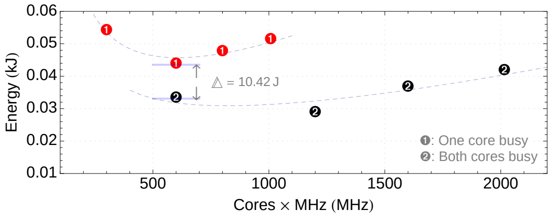

Figure 4: Measured core-only energy (points) and fit to the model of Equation 7 (dashed lines). The compared configurations have approximately equal runtimes.

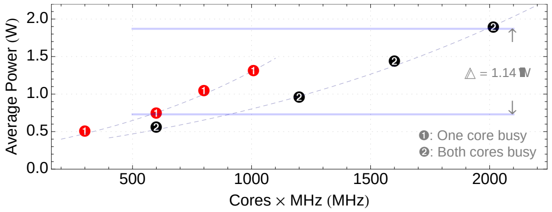

Figure 5: Measured core-only power (points) and fit to the model of Equation 8 (dashed lines). The compared points are approximately iso-energy in Figure 4.

ters in the models of Equations 6, 7, and 8. In particular, the shape of Figure 2 and the value of α determine the rate of increase of clock frequency with increases in supply voltage (and hence increases in power dissipation).

For low values of α, the optimum operating frequency ($f_{\max}$) does not increase significantly with increasing supply voltages. For a device technology that leads to an inherently low α therefore, it may be better to operate at lower voltages and low clock frequencies, and to make up for the lost performance with parallelism. However, as both the model and measurements show, the operating point with the lowest supply voltage and frequency might not be the most energy-efficient, due in part to the effects of leakage.

The models of Equations 7 and 8 enable a number of additional insights. For the parameter values extracted in Section 5.1, the model of Equation 7 predicts lower average power and lower total energy usage across all degrees of voltage scaling σ, if $p = 2$; this is corroborated by the measurements in Figure 4. Similarly, due to leakage, the model predicts a minimum in energy for both the single-core and dual-core cases, near 600 MHz, which is again validated by the measurements.

Even though the performance of the 300 MHz dual-core configuration is only 2% lower than for a single core at 600 MHz (Figure 3) its energy usage is 24% lower (Figure 4). This measured improvement is within 12 percentage points of the improvement predicted by Equations 6 and 7.

6. CONCLUSIONS AND INSIGHTS

This article presented a set of relations between the properties of parallel algorithms, properties of the device technologies of the architectures on which they execute, and the resultant performance, power, and energy-efficiency. Using performance and power measurements on a dual-core ARM Cortex-A9, we demonstrated that the derived relations capture the behavior of real systems in today's semiconductor technologies.

When energy and delay are required to be jointly optimized, the *parameter-invariant energy-delay metrics* introduced in Section 4 specify the precise form of the appropriate joint energy-delay metrics, as a function of device properties. The relations presented linking algorithm and device properties, together with the parameter-invariant energy-delay metrics, provide an analytic basis for exploring the role of algorithm parallelism in the search for an energy-efficient CMOS successor.

7. ACKNOWLEDGEMENTS

The work was influenced by discussions [20] with Volker Strumpen, who provided the idea of the formulation of the interrelation of V and $f_{\max}$ in terms of σ and σ_0, and proposed the expression of work and span in terms of instructions (W_1, W_∞) as opposed to time (T_1, T_∞). Christoph Hagleitner and various anonymous reviewers of earlier drafts [18] of this paper provided valuable suggestions on improving the exposition.

8. REFERENCES

[1] B. Behin-Aein et al. Switching energy-delay of all spin logic devices. *Applied Physics Letters*, 98:123510, 2011.

[2] B. D. Bingham and M. R. Greenstreet. Modeling energy-time trade-offs in vlsi computation. *IEEE TCOMP*, 61:530–547, 2012.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 2009.

[4] D. Culler et al. LogP: Towards a realistic model of parallel computation. PPoPP '93, pages 1–12. ACM Press, May 1993.

[5] G. Gammie et al. A 28nm 0.6v low-power dsp for mobile applications. In *IEEE ISSCC*, pages 132–134, Feb 2011.

[6] J. Jeon et al. Perfectly complementary relay design for digital logic applications. *IEEE Electron Device Letters*, 31(4):371–373, 2010.

[7] V. A. Korthikanti and G. Agha. Analysis of parallel algorithms for energy conservation in scalable multicore architectures. ICPP '09, pages 212–219. IEEE Computer Society, 2009.

[8] M. Horowitz et al. Low-power digital design. In *IEEE Symposium on Low Power Electronics*, pages 8–11, Oct 1994.

[9] N. Pinckney et al. Assessing the performance limits of parallelized near-threshold computing. DAC '12, pages 1147–1152. ACM, 2012.

[10] P. I. Pénzes and A. J. Martin. Energy-delay efficiency of VLSI computations. In *ACM GLSVLSI*, pages 104–111, Apr. 18–20 2002.

[11] R. Blumofe et al. Cilk: an efficient multithreaded runtime system. PPOPP '95, pages 207–216. ACM, 1995.

[12] S. Jain et al. A 280mv-to-1.2v wide-operating-range ia-32 processor in 32nm cmos. In *IEEE ISSCC*, pages 66–68, Feb 2012.

[13] S. Wolf et al. Spintronics: A spin-based electronics vision for the future. *Science*, 294(5546):1488, 2001.

[14] T. Sakurai and A. Newton. Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. *IEEE Journal of Solid-State Circuits*, 25(2):584–594, 1990.

[15] F. Schwierz. Graphene transistors. *Nature nanotechnology*, 5(7):487–496, 2010.

[16] A. Seabaugh and Q. Zhang. Low-voltage tunnel transistors for beyond cmos logic. *Proc. IEEE*, 98(12):2095–2110, 2010.

[17] P. Solomon, D. Frank, and S. Koswatta. Compact model and performance estimation for tunneling nanowire fet. In *69th Annual Device Research Conference (DRC)*, pages 197–198. IEEE, 2011.

[18] P. Stanley-Marbell. The influence of transistor properties on performance metrics and the energy-efficiency of parallel computations. IBM Research Report RZ 3829, 2012.

[19] P. Stanley-Marbell. L24: Parallelism, performance, energy efficiency, and cost trade-offs in future sensor platforms. *ACM Trans. Embed. Comput. Syst.*, 13(1):7:1–7:27, Sept. 2013.

[20] V. Strumpen. Energy efficiency of parallel computations under voltage scaling. In *Austrian Workshop on Microelectronics*, 2013.

[21] 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. *Proc. IEEE*, 98(12):2005 –2014, Dec 2010.

[22] L. Wei, S. Oh, and H. Wong. Performance benchmarks for Si, III–V, TFET, and carbon nanotube FET — re-thinking the technology assessment methodology for complementary logic applications. In *IEEE IEDM*, pages 16–2, 2010.

[23] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for reduced cpu energy. OSDI '94. USENIX Association, 1994.