

Performance and Variation Robustness of Near-Threshold Differential Cascode Voltage Switch Logic

Andrew G. Virga*, Ross Seltzer Richman†, Timothy N. Miller†, and Aaron Carpenter*

*Department of Electrical & Computer Engineering

†Department of Computer Science

Binghamton University

{avirga1, rseltze1, millerti, carpente}@binghamton.edu

ABSTRACT

Near-threshold voltage (NTV) computing is a popular approach to substantially improve energy efficiency in modern microelectronics devices. Two challenges have hindered the integration of NTV into the mainstream: (1) reduced performance and (2) greater vulnerability to the effects of process variation, particularly as transistor dimensions decrease. Alternative logic families (besides static CMOS) provide new opportunities that are not well-explored in low voltage environments. This paper explores the use of Differential Cascode Voltage Switch Logic (DCVSL) as a replacement to CMOS for near-threshold voltage circuits, demonstrating faster overall speeds when voltages are below 0.5V. DCVSL does generally require more energy than CMOS, but judicious use in critical components can confer a substantial performance advantage, offsetting some of the performance lost in near-threshold operation.

Additionally, these circuits are evaluated for variation-robustness. DCVSL circuits demonstrate greater vulnerability in high variation (15-20%) environments. We identify the main point of vulnerability with the DCVSL circuits as the PMOS transistors used in the differential pull-up network. Careful sizing can mitigate some of this vulnerability but sacrifices some of the gained performance benefits. Overall, we present DCVSL as an alternative logic family for near-threshold computation, facilitating design of circuits that are faster than CMOS and variation-robust.

1. INTRODUCTION

Circuit designers operate in an environment where power and energy are first-class considerations. High-end CPUs and GPUs operate at the top end of cost-effective heat dissipation, and energy consumption of microelectronics is a dominating factor in mobile battery life. Two main approaches to reducing power and energy are technology scaling and lower voltage. In the past, scaling transistors down to the next smaller technology node resulted in substantial improvements in performance and power, but in recent years, manufacturing has suffered diminishing returns. This is in part because supply voltage is no longer scaling with geometry, due to both performance requirements and

greater uncertainty in transistor switching characteristics. Indeed, power density now increases with technology scaling, placing severe restrictions on the practical use of any theoretical benefits of scaling.

To compensate for the increasing power density of shrinking transistors, some designs use low operating voltages. This technique typically reduces power consumptions by orders of magnitude but suffers significantly in terms of performance and delay variation. Alternative logic families may offer faster circuit speed at low voltage, while maintaining a lower energy and power consumption than high voltage CMOS.

Meanwhile, in the deep submicron fabrication realm, transistors are increasingly vulnerable to process, temperature, and voltage variations [11, 13]. These variations shift device parameters, such as threshold and supply voltages. In order to safeguard circuits from these variations, designers add margins to ensure that variation does not cause errors. These margins impact both performance and power. Lowering the supply voltage simultaneously shrinks the available margins and makes each type of variation's impact more severe.

In this paper, we compare static CMOS and Differential Cascode Voltage Switch Logic (DCVSL), specifically for 1-bit full adders, across voltage variations and supply voltage levels. DCVSL, when sized similarly to CMOS, suffers from longer delays at nominal voltage. As we lower the voltage of both CMOS and DCVSL, both circuits slow down significantly but also use proportionately less power. By considering both variation and voltage, we find optimal points for both DCVSL and CMOS and demonstrate that DCVSL is a practical alternative to CMOS, particularly for low operating voltages and minimal variation. Specifically, we will show that DCVSL is a faster choice than CMOS at near-threshold voltages. We also show that DCVSL is more vulnerable to variation and identify that this is largely due to the PMOS transistors that act as the pull-up network to the differential logic paths.

2. RELATED WORK

Following the introduction of DCVSL [9], a number of papers have used or evaluated DCVSL and other non-static CMOS logic, particularly at sub-nominal operation voltages. These papers are aimed primarily

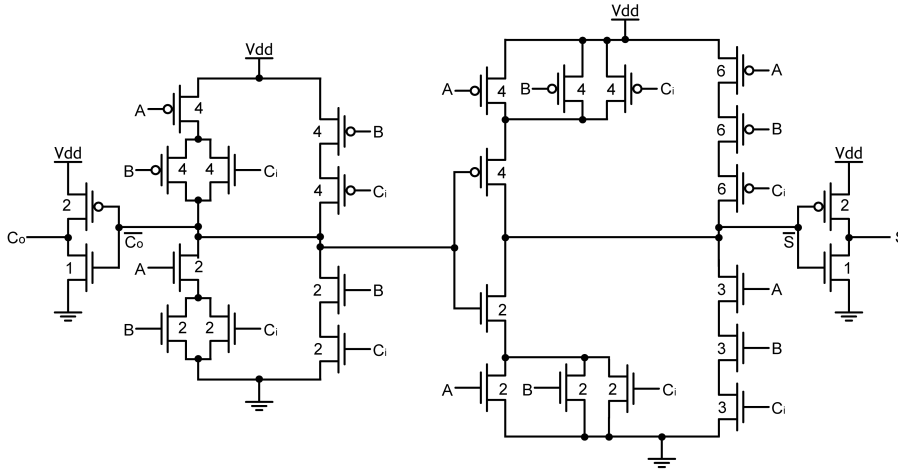


Figure 1: 1-bit Static CMOS full adder with relative gate widths.

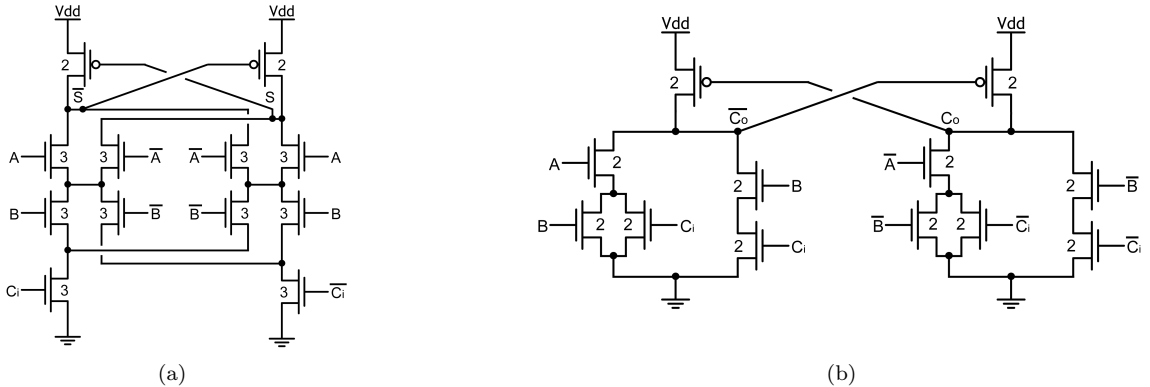


Figure 2: 1-bit DCVSL (a) sum logic and (b) carry logic with relative gate widths.

at addressing energy-delay-product and/or performance [3, 10, 12, 15, 17–19, 23, 24, 28, 29]. Unfortunately, because they do not account for variation, energy numbers reported are not 100% accurate, as variation can have significant impact on the delay, power, and energy of the circuits. A few significant works compare DCVSL and Domino logic, accounting for variation, but do not consider voltage scaling [14, 25]. By simultaneously exploring voltage scaling, parameter variation, and logic family, we hope to identify more efficient circuit design methodologies that have previously been overlooked.

A few papers have explored the use of DCVSL in level-shifters, accounting for process variation, and represent a step in the right direction [16, 26]. Our work is an effort to bridge that information gap and introduce DCVSL as a solution for higher-performance, low-voltage digital circuits for CPUs.

There are a number of variation-robustness techniques that warrant mention, ranging from circuit-level techniques for error avoidance and correction to system-level redundancy [4–8, 20–22]. The work presented here is not necessarily an alternative to these techniques but can be used as an enhancement. For brevity, details of other variation-robustness techniques are left to the reader.

3. DETAIL ON DCVSL AND STATIC CMOS CIRCUITS

Figures 1 and 2 show schematics for static CMOS and DCVSL full adders, respectively. Static CMOS is widely accepted as the default logic type for modern digital circuits. DCVSL, essentially a variant of CMOS, does not contain as many large and slow PMOS transistors; instead logic computations are done in NMOS with only a pair of pull-up PMOS transistors per macro block. DCVSL utilizes complementary logic in two differential cross-coupled paths with any number of NMOS transistors for pull-down, while pull-up is provided by only two PMOS transistors. This does raise potential issues with variation as the DCVSL PMOS transistors are a single point of failure. We address this further in Section 5.

4. EXPERIMENTAL METHODOLOGY

Circuit delay and power are calculated based on a population of 1000 circuits with random threshold voltage variation, and six rise and fall transitions are simulated for each of CMOS and DCVSL. Power is calculated from voltage and the total current that flows from the voltage source, V_{dd} . Delay is measured from the

	1.0V	0.8V	0.6V	0.55V	0.5V	0.45V	0.4V	0.3V
0%	1.481	1.925	1.614	1.271	0.933	0.685	0.546	0.444
3%	1.634	2.447	2.577	2.014	1.373	0.905	0.643	0.498
7%	1.940	4.024	8.428	6.625	4.000	2.302	1.260	0.602
10%	2.307	7.544	26.275	23.771	15.518	7.433	3.458	0.902
13%	2.771	14.224	66.182	62.694	42.786	21.761	9.234	1.233
17%	4.333	31.956	169.672	93.213	67.086	25.083	9.647	1.079
20%	6.266	57.105	148.872	120.648	59.415	22.411	7.946	0.997

Table 1: Delay table: Given variation (left-most column) and a supply voltage (top row), the table shows the ratio for worst-case DCVSL circuit delay divided by the worst-case CMOS circuit delay.

	1.0V	0.8V	0.6V	0.55V	0.5V	0.45V	0.4V	0.3V
0%	1.444	1.562	1.980	1.878	1.675	1.535	1.372	1.162
3%	1.444	1.569	2.006	1.905	1.698	1.551	1.379	1.165
7%	1.453	1.610	2.152	2.068	1.841	1.647	1.441	1.193
10%	1.462	1.654	2.428	2.396	2.147	1.861	1.566	1.231
13%	1.471	1.734	2.969	3.136	2.853	2.358	1.841	1.301
17%	1.491	1.879	4.493	5.103	4.701	3.527	2.390	1.383
20%	1.511	2.058	6.328	7.463	6.534	4.457	2.788	1.429

Table 2: Energy table: Given variation (left-most column) and a supply voltage (top row), the table shows the ratio for average DCVSL circuit energy divided by the average CMOS circuit energy.

time the inputs reach $V_{dd}/2$ until the time the last output has transitioned to $V_{dd}/2$. For each supply voltage (300mV to 1V) and degree of process variation (0% to 20%), we simulate the entire population, and we record maximum delay and average energy for the population. Data was collected from a total of 672,000 SPICE simulations, using the 45nm high performance predictive transistor model from ASU [1].

To calculate process variation, 1000 sets of Gaussian-distributed random numbers $R_{i,j}$ are computed where $\sigma=1$ and $\mu=0$. Mean V_{th} is 180mV, and the degree of process variation v is specified in terms of σ/μ . σV_{th} increases in proportion to the square root of the reduction in channel area [2]. For transistor i, j with variation v at nominal width W_R and scaled width W , threshold voltage is computed as in Equation 1. Threshold voltage variation is applied via SPICE instance parameters.

$$V_{th}(i, j) = \mu + \mu v R_{i,j} \sqrt{W_R/W} \quad (1)$$

Transistor sizes for both CMOS and DCVSL were chosen based on a simple 2:1 (PMOS to NMOS) width ratio and were scaled when needed in order to balance the current flowing through both the pull-up and pull-down networks. The sizes chosen are similar to previously published static CMOS circuits [10, 18, 27]. For completeness, we also used smaller and larger DCVSL PMOS, but found that the sizing of 2:1 gives the optimal results at near-threshold voltages. The overall area of the DCVSL and CMOS circuits are comparable, with CMOS taking larger transistor area, but smaller routing than the DCVSL alternative.

The inputs, A, B, and Carry_in, were switched individually to provide both positive and negative edges and observe both rise and fall times. Average rise and fall times can be misleading, because they de-emphasize extreme cases that may impact yield. Therefore we present only worst-case propagation delay values here.

5. EXPERIMENTAL RESULTS

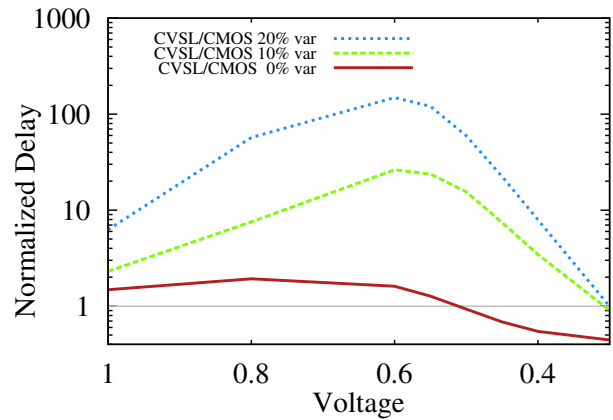
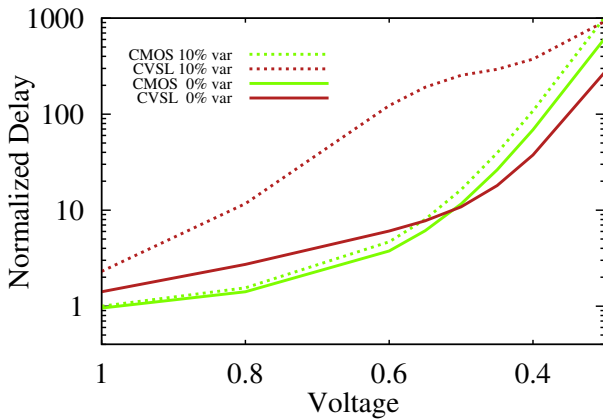
5.1 Delay & Energy Effects of Supply Voltage with No Variation

The CMOS circuits are significantly faster than the DCVSL counterparts at high voltages, as shown in Figures 3 and Table 1, by as much as 2x with the 0.8V voltage supply. Energy for CMOS is consistently lower, ranging from 50% to 86% of the DCVSL adder’s energy (see Table 2 and Figure 5), most likely due to the greater short-circuit energy of differential logic.

As we approach the threshold voltage, the DCVSL adder becomes as fast as or faster than CMOS. At 0.5V supply voltage, DCVSL is 7% faster compared to CMOS at the same voltage, and as the voltage decreases further, DCVSL improves to more than twice as fast as CMOS. This trend has been noted previously, but not promoted as the central benefit of DCVSL [27]. Selective use of DCVSL circuits for critical timing paths can reduce the effects of using near-threshold voltages, and thus, the additional energy can be tolerated. Unfortunately, the performance advantage is true only for low variation; at 0.3V, DCVSL loses its performance advantage beyond 10% variation while still requiring more energy.

5.2 Delay & Energy Effects of Process Variation

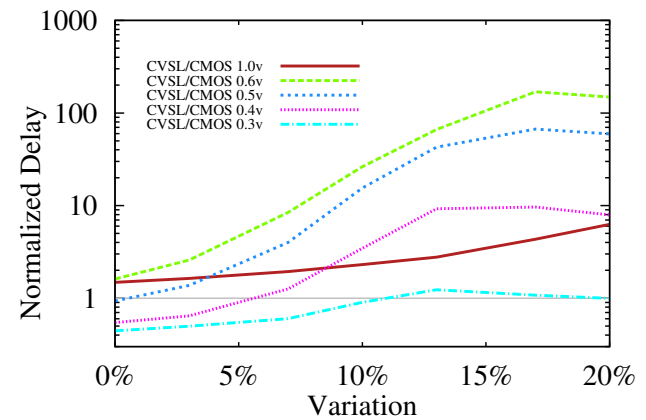
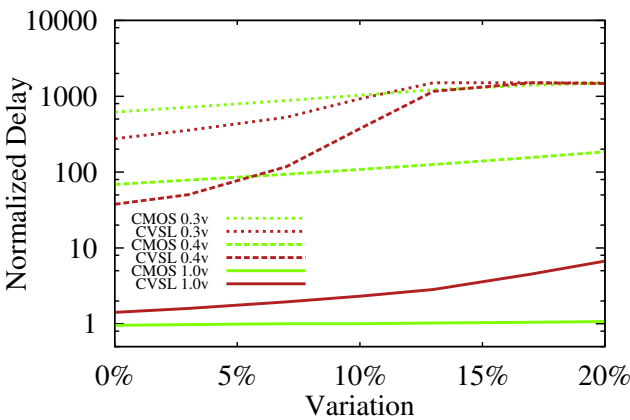
Transistors with modern and future fabrication techniques must account for a significant amount of variation. At the 22nm node, we expect approximately 10% variation, and the 11nm node will have up to 20% variation [11, 13]. Unfortunately, the baseline DCVSL full adder performs poorly in the presence of significant variation, as Figure 4 illustrates. At 10% and 20% variation and 1.0V voltage supply, the DCVSL is 43% and 16%



(a) Absolute: Normalized to CMOS at 1V and no variation

(b) Relative: DCVSL delay divided by CMOS delay

Figure 3: Delay: Comparison of circuit delay for DCVSL and CMOS, as a function of voltage, for various degrees of process variation. Delay reported is the maximum across the population of simulated adder circuits.



(a) Absolute: Normalized to CMOS at 1V and no variation

(b) Relative: DCVSL delay divided by CMOS delay

Figure 4: Delay: Comparison of circuit delay for DCVSL and CMOS, as a function of variation, for various voltages. Delay reported is the maximum across the population of simulated adder circuits.

(respectively) as fast as CMOS, which would be unacceptable for most circuit implementation. Meanwhile, the CMOS circuit is relatively robust when variation is introduced. Figure 4 shows some CMOS circuit timing fluctuation, but it is minimal compared to DCVSL.

For PMOS sizing that confers a performance advantage, DCVSL never has an energy advantage, as seen in Figure 6 and Table 2. With smaller PMOS transistors (Section 5.3), DCVSL gains an energy advantage but loses much its performance advantage at near-threshold. At high voltages, the smaller PMOS does have better performance and better variation robustness.

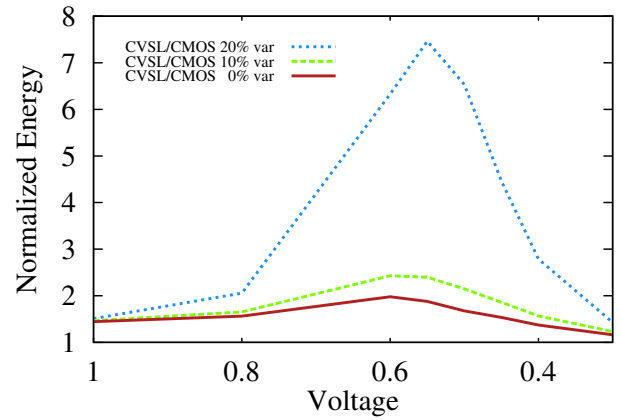
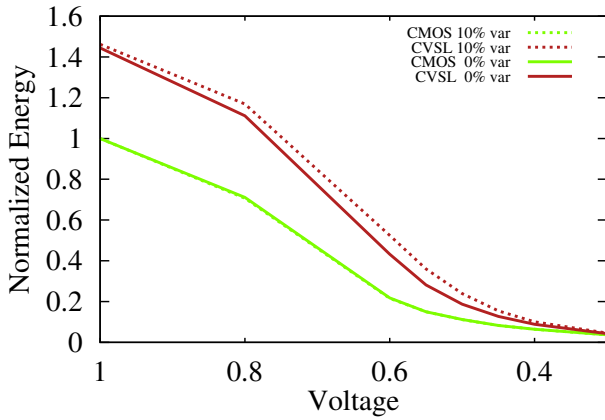
5.3 Improving Variation-Robustness of DCVSL Circuits

The source of the DCVSL circuit's variation-intolerance can be isolated to the two PMOS transistors in the pull-up networks for the differential logic macro blocks. As a single point of failure, high threshold voltage variation creates drastic variation in the maximum rise-time delay. Analysis shows that the rise times, even at zero variation, are between 2 to 5 times larger than the fall times, depending on voltage. As the size of the PMOS

increases, the rise and fall times converge, but at significant cost to the maximum delay. DCVSL delay is often presented as an average, over-representing its speed. Here we use worst-case delay.

As we introduce variation, these PMOS transistors' worst-case rise times increase substantially. This is experimentally confirmed by a variation sweep in which only the PMOS transistor was subject to variation, while the pull-down networks of NMOS transistors had zero variation. This revealed that the PMOS alone was responsible for 50% to 98% of the total slowdown from increasing variation, depending on voltage and threshold voltage variation. We also performed similar analysis varying only the NMOS pull-down network, and found similar trends. While the NMOS variation contributes to the overall slowdown, it is only a minor factor compared to the PMOS network. With careful design of the PMOS alone or with a redesign of the pull-up network, this vulnerability can be mitigated.

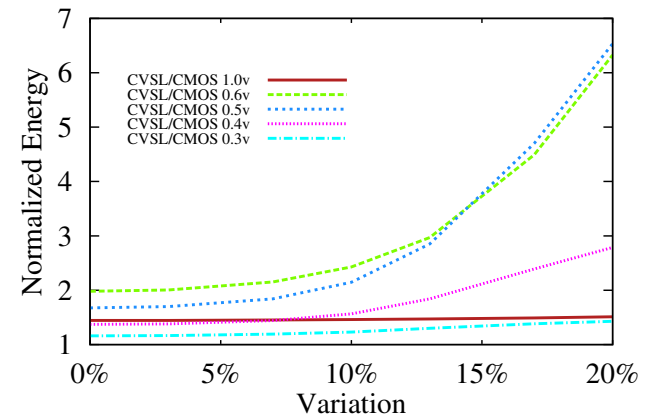
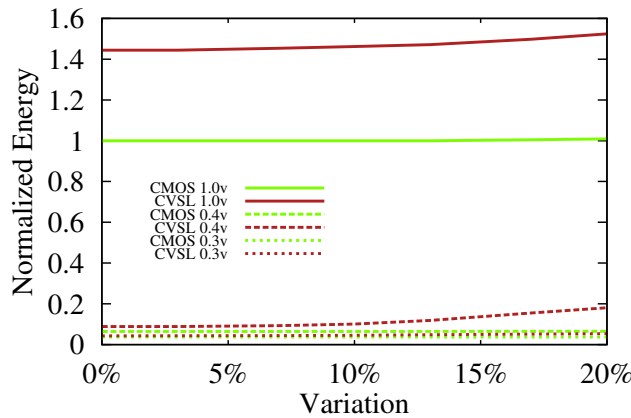
To test this hypothesis, we simulated different PMOS sizes. The recommended PMOS to NMOS width ratio (see Section 4) is 2:1, and we also tested 1:1 and 3:1. For near-threshold voltages, the original 2:1 sizes are



(a) Absolute: Normalized to CMOS at 1V and no variation

(b) Relative: DCVSL energy divided by CMOS energy

Figure 5: Energy: Comparison of circuit energy for DCVSL and CMOS, as a function of voltage, for various degrees of process variation. Energy is averaged across the population of simulated adder circuits.



(a) Absolute: Normalized to CMOS at 1V and no variation

(b) Relative: DCVSL energy divided by CMOS energy

Figure 6: Energy: Comparison of circuit energy for DCVSL and CMOS, as a function of variation, for various voltages. Energy is averaged across the population of simulated adder circuits.

15-21% (0.4V) and 44-76% (0.3V) faster than the 1:1 ratio at low variations (0-3%). Thus, if both variation and voltage are low, the 2:1 ratio is optimal. (For the 3:1 ratio, performance was universally worse.)

However, in high-variation environments or with higher supply voltages, the 1:1 ratio circuits can be up to 30 times faster and more than 12 times energy efficient (0.8V and 20% variation), compared to the 2:1 DCVSL. Compared to CMOS, the DCVSL 1:1 ratio circuit is up to 27% faster than CMOS at 0.55V, albeit less tolerant of variation. DCVSL still maintains a performance advantage at low voltages, although not as favorable as the 2:1 ratio. Overall, the 1:1 ratio provides a more robust circuit across all voltages and variations, but does not provide the same level of performance gain at low voltages as the 2:1 circuit does. This allows the designer to not only choose the logic family for its particular benefits, but also size the individual circuits to tailor them to particular design goals: the baseline 2:1 provides the best low-voltage performance while the 1:1 DCVSL ratio gives better variation robustness and high-voltage performance.

6. ON-GOING & FUTURE WORK

DCVSL is only one alternative logic family. Studying similar effects of variation and supply voltage scaling in other families, such as pass-gate or domino logic, would present circuit designers with a larger array of alternatives, beyond static CMOS. Past work has evaluated variation *or* voltage-scaling, but it is imperative to do both in order to find optimal approaches for future near-threshold computing. Each logic family will have different points of failure, and thus, each needs to be analyzed and optimized individually, similarly to the DCVSL's pull-up network sizing presented here.

Moreover, DCVSL is not fully explored. We will continue improving sizing and robustness. This work is a first step towards evaluating non-static CMOS logic.

Additionally, more complicated digital circuits will be designed and evaluated with a similar process, creating a library of each logic family for use in EDA tools. While a simple 1-bit adder provides a good foundation for comparison, more complex circuits are necessary in order to make system-wide conclusions.

It is also necessary in future research in this field to characterize variation for yield assessment. Because of

the probabilistic nature of variation, it is possible to discard outliers, improving the overall performance of a given circuit, while sacrificing yield. This analysis was not shown here, but is in progress.

7. CONCLUSIONS

Near-threshold computing is an effective method for designing low-energy devices. However, as the voltage is lowered, performance is lost because of increasing delays. To address this, we have presented the use of DCVSL as an alternative to static CMOS at near-threshold voltages. DCVSL requires more energy than CMOS at the same voltage, but targeted use in performance-critical components can partially mitigate the substantial performance reduction inherent in near-threshold designs. DCVSL, however, has poor performance in the presence of significant threshold voltage variation. To address this, we identify the PMOS transistors used for the DCVSL pull-up network as a single point-of-failure. In low-variation environments, DCVSL is an attractive alternative to CMOS at low-voltages. Further, with design effort, DCVSL will also be a promising solution for near-threshold circuits in future, high-variation systems.

8. REFERENCES

- [1] Predictive Technology Modeling. <http://ptm.asu.edu/>.
- [2] A. Asenov. Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm MOSFETs: A 3-D “atomistic” simulation study. *Electron Devices, IEEE Transactions on*, 45(12):2505–2513, 1998.
- [3] P. Corsonello, S. Perri, and G. Cocorullo. Performance comparison between static and dynamic CMOS logic implementations of a pipelined square-rooting circuit. *IEE Proceedings - Circuits, Devices and Systems*, 147(6):347–355, Dec. 2000.
- [4] R. Dreslinski, G. Chen, T. Mudge, D. Blaauw, D. Sylvester, and K. Flautner. Reconfigurable energy efficient near-threshold cache architectures. In *Proc. Int’l Symp. on Microarchitecture*, pages 459–470, December 2008.
- [5] R. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge. Near-threshold computing: Reclaiming Moore’s Law through energy efficient integrated circuits. *Proc. of the IEEE*, 2(98):253–266, 2010.
- [6] R. Dreslinski, B. Zhai, T. Mudge, D. Blaauw, and D. Sylvester. An energy efficient parallel architecture using near threshold operation. In *PACT 2007*, pages 175–188.
- [7] D. Ernst, N. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: A low-power pipeline based on circuit-level timing speculation. In *Proc. Int’l Symp. on Microarchitecture*, pages 7–18, 2003.
- [8] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. Harris, D. Blaauw, and D. Sylvester. Bubble Razor: An architecture-independent approach to timing-error detection and correction. In *Proc. IEEE Int’l Solid-State Circuits Conf.*, pages 488–490, Feb. 2012.
- [9] L. Heller, W. Griffin, J. Davis, and N. Thoma. Cascode voltage switch logic: a differential CMOS logic family. In *Proc. Int’l Conf. on Solid-State Circuits*, pages 16–17, 1984.
- [10] J. Hu and X. Yu. Near-threshold full adders for ultra low-power applications. In *Pacific-Asia Conference on Circuits, Communications, and System*, pages 300–303, Aug. 2010.
- [11] International Technology Roadmap for Semiconductors. <http://public.itrs.net/>.
- [12] D. Kang and Y. Kim. Design of enhanced differential cascode voltage switch logic (EDCVSL) circuits for high fan-in gate. In *IEEE Int’l ASIC/SoC Conf.*, pages 30–313, Sept. 2002.
- [13] U. R. Karpuzcu, K. B. Kolluru, N. S. Kim, and J. Torrellas. Varius-ntv: A microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*, pages 1–11. IEEE, 2012.
- [14] M. Kishor and J. de Gyvez. Threshold voltage and power-supply tolerance of cmos logic design families. In *IEEE Int’l Symp. on Defect and Fault Tolerance in VLSI Systems*, pages 349–257, 2000.
- [15] M. Kontiala, M. Kuulusa, and J. Nurmi. Comparison of static logic styles for low-voltage digital design. In *IEEE Int’l Conf. on Electronics, Circuits, and Systems*, pages 1421–1424, 2001.
- [16] G. Maderbacher, T. Jackum, W. Pribyl, S. Michaelis, and C. Sandner. Fast and robust level shifters in 65 nm CMOS. In *Proc. of the ESSCIRC*, pages 195–198, Sept. 2011.
- [17] D. Markovic, C. Wang, L. Alarcon, T. Liu, and J. Rabaey. Ultralow-power design in near-threshold region. *Proc. of the IEEE*, 98(2):237–252, Feb. 2010.
- [18] N. Masoumi, J. Ghasemi, M. Ahmadian, F. Raissi, and M. Masoumi. Enhancing performance and saving energy in CMOS DCVSL gates by using a new transistor sizing algorithm. In *Workshop on SoC for Real-Time Applications*, pages 283–288, July 2005.
- [19] S. Mathew and R. Sridhar. A data-driven micropipeline structure using DSDCVSL. In *Proceedings of the IEEE Custom Integrated Circuits*, pages 295–298, May 1999.
- [20] T. Miller, X. Pan, R. Thomas, N. Sedaghati, and R. Teodorescu. Booster: Reactive core acceleration for mitigating the effects of process variation and application imbalance in low-voltage chips. In *HPCA 2012*, pages 1–12.
- [21] T. Miller, R. Thomas, J. Dinan, B. Adcock, and R. Teodorescu. Parichute: Generalized turbocode-based error correction for near-threshold caches. In *Proc. Int’l Symp. on Microarchitecture*, pages 351–362, 2010.
- [22] T. Miller, R. Thomas, X. Pan, and R. Teodorescu. VRSync: Characterizing and elimination synchronization-induced voltage emergencies in many-core processors. In *ISCA 2012*, pages 249–260.
- [23] P. Ng, P. Balsara, and D. Steiss. Performance of CMOS differential circuits. *IEEE Journal of Solid-State Circuits*, 31(6):841–846, June 1996.
- [24] V. Oklobdzija. Differential and pass-transistor CMOS logic for high performance systems. *Microelectronics Journal*, 29(10):679–688, 1998.
- [25] S. Purohit and M. Margala. Data driven DCVSL: A clockless approach to dynamic differential circuit design. In *Int’l Midwest Symp. on Circuits and Systems*, pages 640–643, Aug 2010.
- [26] J. Rocha, M. Santos, J. Costa, and F. Lima. High voltage tolerant level shifters and DCVSL in standard low voltage CMOS technologies. In *Int’l Symp. on Industrial Electronics*, pages 775–780, June 2007.
- [27] M. Shams. A unified delay model for CMOS logic styles. In *Int’l Conf. on Electronics, Circuits, and Systems*, pages 874–877, Dec. 2003.
- [28] J. Won and K. Choi. Self-timed statistical carry lookahead adder using multiple-output DCVSL. In *Int’l Conf. on VLSI and CAD*, pages 560–563, 1999.
- [29] A. Wu. High performance adder cell for low power pipelined multiplier. In *IEEE Int’l Symp. on Circuits and Systems*, pages 57–60, May 1996.