

Optimizing addition for sub-threshold logic

David Blaauw

Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109, United States
Email: blaauw@umich.edu

James Kitchener and Braden Phillips

School of Electrical and Electronic Engineering
The University of Adelaide, Adelaide 5005, Australia
Email: jkitch,phillips@eleceng.adelaide.edu.au

Abstract—Digital circuits operating at subthreshold-voltage levels can achieve extremely low energy consumption. Typical applications include sensor processors with modest processing requirements that must run for long intervals on a low energy supply. The design goal is to minimise the total energy required for a processing task. Optimal architectures strike a balance between leakage and dynamic dissipation: if a unit is too slow, leakage energy is wasted throughout the system; however increasing the unit's speed may cost increased dynamic dissipation and leakage within the unit. We examine this trade-off through the simulation of a variety of adder architectures. The results show that for a 180 nm process, system leakage dominates adder switching energy. For all but the smallest systems, when the adder is on the critical timing path, overall energy consumption is minimized by choosing a fast tree adder. The results also show that high valency tree adders perform well at subthreshold levels in this process.

I. INTRODUCTION

Static CMOS circuits can be designed to operate at remarkably low supply voltage. As the supply voltage, V_{dd} , drops below the transistor threshold voltage, V_t , the current available from a gate decreases and an exponential increase in circuit delay is observed; however there is also a dramatic reduction in switching energy due to the quadratic dependence of energy on voltage through $E_{sw} \propto CV^2$. Systems designed to take advantage of this low-power behaviour and operate in the subthreshold region have been demonstrated. These include the *Phoenix* sensor processor [1], a $915 \mu\text{m}^2$ device fabricated in a 180 nm CMOS process which consumes only 2.8 pJ/cycle at $V_{dd} = 0.5$ V and an operating frequency of 106kHz. Another notable recent example of subthreshold design appears in [2].

How does subthreshold operation influence the design of arithmetic circuits? This paper presents the results of an empirical study of adders undertaken to address this question. Addition was chosen because it is a crucial element of other arithmetic circuits as well as a fundamental arithmetic operation in its own right. Moreover, adders are well understood by arithmetic designers who, it is hoped, will be able to apply the lessons learned other arithmetic circuits.

Our approach was inspired by the survey of CMOS adders by Zimmermann and Fitchner [3]. The goal in that case was to compare static CMOS and pass-transistor logic styles at nominal (superthreshold) supply voltage, but two aspects of the study strongly influenced the present work. The first was

the observation that full adders are not the only important gate for adders and, in fact, most high-performance adders do not use full adders at all. Secondly, [3] debunked many published claims concerning pass-transistor logic made on the basis of erroneous or unrealistic simulation scenarios. Hence, in the present work, care has been taken to ensure good simulation practices are observed.

In Section II we examine important adder gates including full adders, inverters, NAND, NOR, XOR and XNOR gates, and the and-or-invert and or-and-invert gates used for prefix cells at valency 2 and beyond. Complete adder architectures are considered in Section III. Prior to this, further background on subthreshold logic is presented in Section I-A. Conclusions appear in Section IV along with a discussion of open questions for future research.

A. Challenges in Subthreshold Logic Design

Transistor variations in the subthreshold region can have a dramatic effect on signal propagation times [4]. Threshold voltage variations due to random dopant fluctuations are a particular problem because they change the behaviour of a transistor relative to its immediate neighbors. These signal delay variations can lead to timing violations. Approaches to this problem include: the design of robust sequencing elements [5]; the use of shallow pipelines so that random effects average out over the length of the logic path [6]; and compensation using transistor body bias [4]. In this work we side-step this issue by considering only combinatorial adder circuits. Some observations of the effect of transistor variations on gate performance are made in Section II-B4.

With the supply voltage lower than the threshold voltage, the absolute noise margin is necessarily small. That one cannot afford a single threshold voltage drop excludes the use of most pass-transistor logic families. This study, therefore, is confined to static CMOS gates and the question of transmission-gate and dynamic logic will be left for future work.

Some existing subthreshold designs (e.g. [6]) use a cell library with gates limited to fanin 2 or less. One reason put forward for this is an expected increased influence of the body effect which degrades performance when more than 2 FETs are stacked in series. Gates of fanin 3 and greater are examined in Section II of the present study.

The overarching goal of subthreshold design is usually to minimise the total energy to complete an operation for applications where latency is not critical. The total energy consists of

This research was supported by the South Australian Government under the Premier's Science and Research Fund.

the switching energy and the leakage energy. The latter makes a significant contribution at subthreshold voltage levels. In the context of addition, one might consider minimizing energy by using the smallest, simplest adder available: a ripple-carry adder. This would, most likely, minimise both the switching and leakage energy in the adder; however it would be slow, and while waiting for the adder to complete, the rest of the system would waste leakage energy. To increase the speed of the adder, a more complex, more power-hungry architecture could be used. Hence there is an interesting trade between speed and switching energy. Examination of this trade is the primary motivation for this paper. Results are presented in Section III.

II. LOGIC GATES

This section examines the subthreshold performance of the static CMOS logic gates which are important for adder designs.

A. Method

A TSMC 180 nm, 1.8 V general logic CMOS process was used. This has $V_t \approx 0.4$ V. Its fanout 4 inverter delay, FO4, is approximately 100 ps.

Layout was generated for all of the gates using Magic¹, DEEP SCMOS rules and a technology file from MOSIS². SPICE decks were extracted from the layout using Magic with the threshold for interconnect capacitance and resistance set to zero so that all parasitics extracted were included in the decks.

Simulations were performed with HSPICE using the BSIM V3.3 (Lv 49) models derived from a MOSIS test lot³. Circuits were all simulated at 70°C and the typical process corner (except when noted otherwise).

Fig. 1 shows a schematic of the testbed used for the simulations. Two stages of input shaping logic, and two of output loading logic were used. The multiplier parameter (M) in HSPICE was used to simulate the effect of fanout. All input transitions were observed for the 2-input gates. For the 3-input gates the rising and falling edges of the worst-case transition (based on examination of the layout) were observed. A separate supply was used for the device under test to observe its switching energy. Leakage energy was measured using an independent instance of the device under test at steady state conditions with constant inputs and no output load.

B. Results

1) *Inverter Characteristics and P:N Ratio*: Fig. 2 shows the average of rising and falling propagation delays for an inverter as a function of V_{dd} and fanout, H . Two variations of the inverter are shown: one with minimum width, 4λ , pMOS and nMOS FETs; the other with a 4λ nMOS, and a 10λ pMOS. The latter gives approximately equal rise and fall times.

¹Magic Version 7.5 Revision 145, from <http://opencircuitdesign.com/>
²<ftp://ftp.mosis.edu/pub/sondeen/magic/new/beta/current.tar.gz> downloaded on 29 September 2008

³http://www.mosis.com/cgi-bin/params/tsmc-018/t28m_lo_epi-params.txt

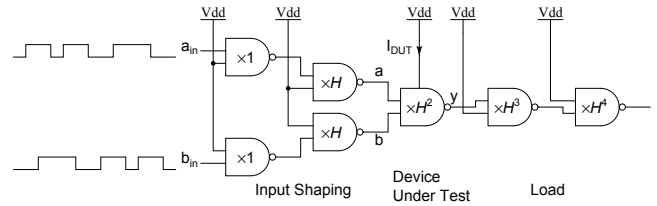


Fig. 1: The testbed used for gate simulations. A 2-input NAND gate is shown.

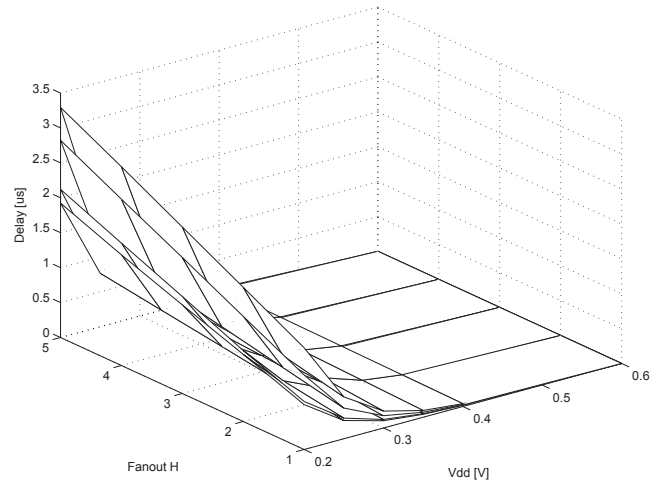


Fig. 2: Average propagation delay for inverters and NAND gates. From the bottom the curves are: inverter with P:N = 2.5:1; inverter with P:N = 1:1; 2-input NAND; 3-input NAND.

At constant fanout the delay increases exponentially as V_{dd} decreases. For constant V_{dd} the delay is a straight line as one typically expects for superthreshold operation. Adopting the terminology of *logical effort* [7], we observe that both the delay of an unloaded gate, or *parasitic delay*, and the increase of delay with fanout, or *logical effort*, increase as V_{dd} decreases.

Fig. 3 shows the static noise margin for the two inverters as a function of V_{dd} . The noise margin for both devices degrades gracefully as V_{dd} falls. Resizing the pMOS transistor improves the minimum static noise margin by 17 mV; however it does not improve average propagation delay and will only increase switching and leakage energy. Thus, for the remainder of the experiments we use equal width pMOS and nMOS. They are all minimum-width, except when noted otherwise.

2) *NAND Characteristics and Fanin*: Fig. 2 also shows the average of rising and falling propagation delays for 2 and 3-input NAND gates. While the parasitic delay for the 3-input gate is larger, the logical efforts of the two gates are approximately equal. Fanin is affecting capacity to drive a load less than expected for superthreshold circuits. The increased fanin does not affect noise margin with both gates having 0.091 V of margin at $V_{dd} = 0.3$ V. This is considered further in Section II-B5.

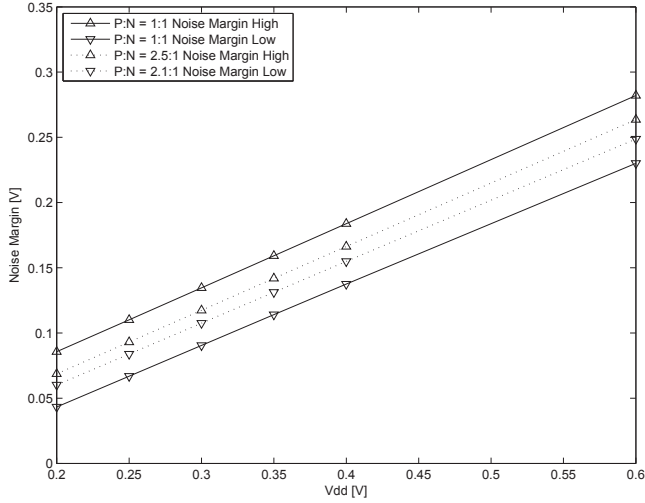


Fig. 3: Static noise margins for inverters with P:N ratios of 1:1 and 2.5:1.

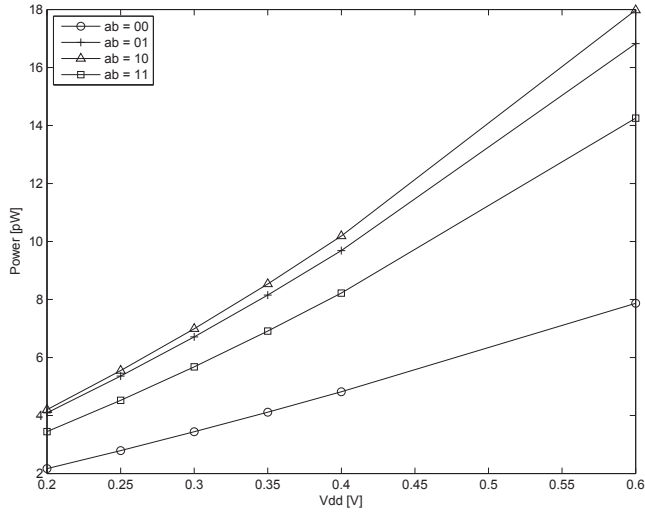


Fig. 4: Leakage power in a 2-input NAND gate.

3) *Leakage Power and Switching Energy*: Fig. 4 shows the leakage power for a 2-input NAND gate against V_{dd} for the different input logic states. The leakage does differ markedly with logic state. For a 2-input NOR gate the leakage power for the state $ab = 00$ is almost 10 times greater than for $ab = 11$. This suggests a strategy for minimizing power consumption in idle circuits by avoiding the leaky states.

The energy per switch for a 2-input NAND gate is shown in Fig. 5.

4) *Threshold Voltage Variations*: To explore the effect of threshold voltage variations, fast and slow design corners were simulated with a 0.04 V decrease or increase in $|V_t|$ respectively. Temperature was 70°C throughout. Fig. 6 shows the effect on the average delay of a 2-input NAND gate. The minimum noise margin at $V_{dd} = 0.3$ V fell from 0.091 V at the TT corner to only 0.051 V at the FS corner. Maximum

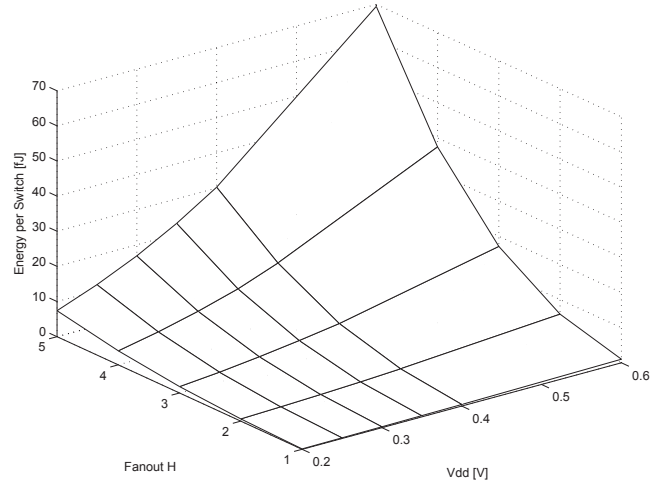


Fig. 5: Switching energy for the slow input in a 2-input NAND gate.

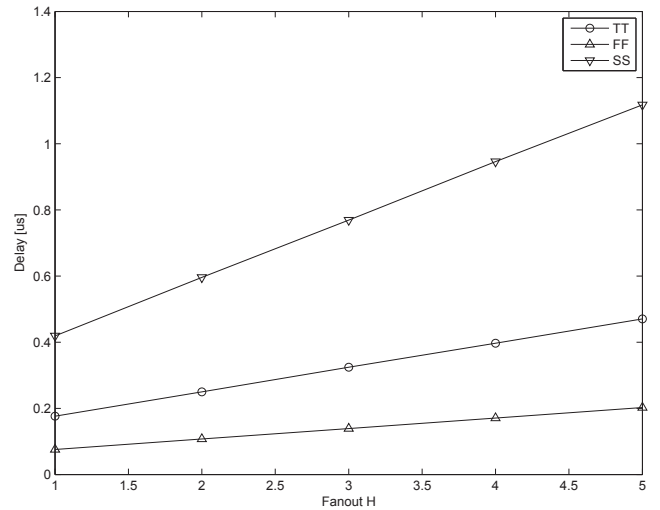


Fig. 6: Average 2-input NAND propagation delay at design corners corresponding to ± 0.04 V changes in V_t .

leakage power increased from 6.7 pW at the TT corner to 16.4 pW at the FF corner. These results confirm the dramatic influence of V_t variations.

5) *Logic Gate Summary*: A summary of logic gate performance at $V_{dd} = 0.3$ V and 1.8 V is given in Table I. The parasitic delay has been given in units of τ , where τ is one fifth of the FO4 inverter delay at $V_{dd} = 0.3$ V or 1.8 V as appropriate. Logical effort is given in units of τ per fanout. These values were obtained from the average of rising and falling edges of the slowest input transition except for the full adder and gray cells for which the the carry-in to carry-out transitions were observed.

The switching energy was measured as the average for the rising and falling transitions of the input with the worst-case switching energy. The leakage power was recorded for the input state with the highest leakage.

The full adder in the table is a 28 transistor static CMOS device [8] and the inverting full adder is the same with the output inverters removed.

The superthreshold numbers for logical effort and parasitic delay in this table correspond well with nominal values often used for hand estimation [8]. When normalized against τ at 0.3 V, the subthreshold numbers differ in some interesting ways. The subthreshold parasitic delays are generally worse; however the logical effort for the inverter, NAND gates and full adder improve. The 3-input NAND has logical effort almost equal to the 2-input NAND. Hence for these gates, fanin and fanout have less influence on delay at subthreshold voltage but the no-load delay per stage is increased. This suggests architectures with fewer stages of gates with higher fanin and fanout may be faster for subthreshold designs. The NOR gates do not do as well indicating the stacked, minimum-sized pMOS transistors have a more negative impact at subthreshold than superthreshold voltage.

III. ADDERS

In this section, different 8, 16 and 32-bit adder architectures are compared at subthreshold voltage.

A. Method

The simulation methodology described in Section II has been used. Spice decks were extracted from layout with interconnect parasitics included. The testbed shown in Fig. 7 was used to observe the rising and falling transitions at C_{out} due to a change in C_{in} . The exact transitions were from $\{A, B, C_{in}\} = \{0 \dots 00, 1 \dots 11, 0\}$ to $\{1 \dots 11, 0 \dots 00, 1\}$ and then to $\{0 \dots 00, 1 \dots 11, 0\}$. Thus all of the input bits and sum bits were toggled to obtain some indication of worst-case switching energy. A transient analysis was used to measure the average propagation delay, switching energy and leakage power for these 2 transitions.

Various ripple-carry adders were tested as they were expected to use little switching or leakage energy at the cost of high delay. They were: a chain of full adders; a chain of inverting full adders with inverters on even inputs and odd outputs; generate and propagate signals passed to a chain of gray cells with sums evaluated by XOR gates; a chain of gray cells, generate, propagate and sum logic with alternating columns of true and inverted gates. Sklansky adders [9] were selected to represent high-energy, low-delay adders. It has been shown that Sklansky adders can be energy-efficient at superthreshold voltage; and that to optimize their performance it is usually sufficient to place minimum-width transistors everywhere, except for the few high-fanout nodes [10]. Valency 2, inverting valency 2, and valency 3 Sklansky adders were tested [8].

B. Results

Fig. 8, 9 and 10 show the delay, switching energy and leakage power for the adders. All of the adders use minimum-width transistors except for the resized Sklansky adders which

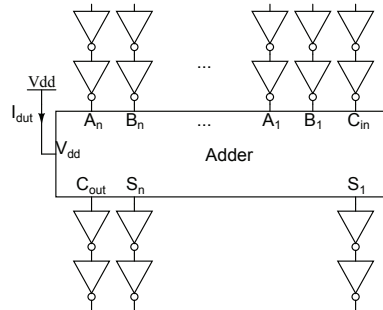


Fig. 7: The testbed used for the adder simulations.

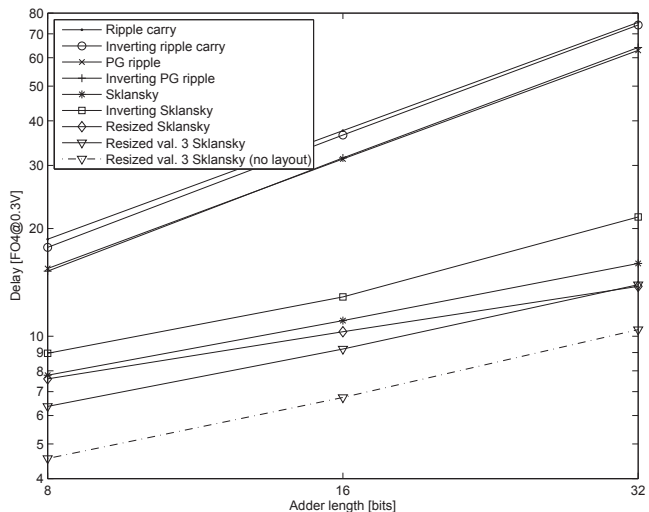














Fig. 8: Adders: average propagation delay from C_{in} to C_{out} at $V_{dd} = 0.3$ V.

use either 2-times or 4-times minimum-width transistors in the inverters driving the high-fanout nodes on the critical path.

The ripple-carry adders are slower than the Sklansky adders, but consume less switching or leakage energy. The inverting Sklansky adder is slower than the non-inverting version suggesting that fanout has become a problem. There may be scope to improve the former with careful buffer insertion and sizing. Resizing the transistors at the critical nodes of the non-inverting Sklansky adder improves its delay, especially at 32-bits, with little cost in switching energy. The valency 3 Sklansky adder is faster than the valency 2 device at 8 and 16-bits. The simulation results for the valency 3 Sklansky adder without interconnect parasitics is also shown in Fig. 8 and 9.

Given these results, which adder gives the lowest energy for a particular application? If the adder is not on the critical timing path of the system, then it would be best to choose one of the ripple-carry adders. If the adder is on the critical timing path, then it may be most efficient to use a faster adder. For the following analysis we assume the clock is set by the adder delay, t_{add} , and sequence overhead t_{seq} . Furthermore, we assume $t_{seq} = 10$ FO4 where FO4 is the fanout-4 inverter delay at the V_{dd} of the adder.

TABLE I: Logic Cells Commonly Used in Addition at $V_{dd} = 0.3$ V (and $V_{dd} = 1.8$ V). All transistors are minimum width.

Cell	Icon	Parasitic Delay p [τ]	Logical Effort g [τ/fanout]	Energy per Switch vs. inv	Leakage Power vs. inv	Min. Noise Margin [V]
inv		1.134 (0.676)	0.961 (1.079)	1.000 (1.000)	1.000 (1.000)	0.090 (0.454)
nor2		2.327 (2.188)	1.734 (1.662)	1.860 (1.853)	1.999 (2.000)	0.087 (0.413)
nand2		1.526 (1.481)	1.337 (1.401)	1.785 (1.880)	0.999 (1.000)	0.091 (0.454)
xnor2		6.145 (6.012)	3.817 (3.436)	4.746 (4.991)	3.957 (3.875)	0.087 (0.412)
xor2		6.268 (6.008)	3.713 (3.468)	4.728 (4.924)	3.363 (3.230)	0.131 (0.768)
nand3		2.825 (2.110)	1.322 (1.470)	2.122 (2.270)	1.217 (1.065)	0.091 (0.438)
nor3		4.191 (3.842)	2.137 (2.013)	2.024 (2.209)	2.997 (3.000)	0.084 (0.384)
oai21		3.409 (3.523)	1.913 (1.775)	2.101 (2.278)	1.856 (1.678)	0.090 (0.412)
aoi21		3.311 (3.328)	1.906 (1.780)	2.182 (2.265)	1.998 (2.000)	0.087 (0.412)
fulladd (C_{out})		9.394 (9.365)	3.467 (3.732)	9.685 (10.636)	5.040 (4.101)	N/A
fulladdi (C_{in})		6.078 (6.591)	7.474 (7.867)	6.583 (7.636)	4.218 (3.348)	0.087 (0.412)
gray cell		0.000 (6.544)	0.000 (1.141)	3.811 (4.045)	2.408 (2.355)	N/A

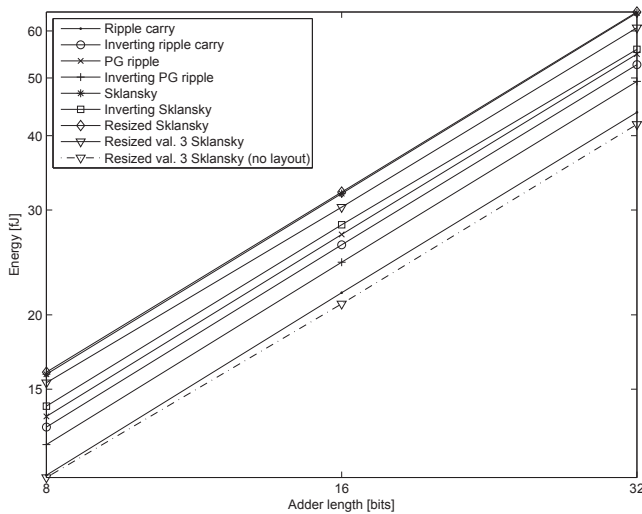


Fig. 9: Adders: average switching energy at $V_{dd} = 0.3$ V.

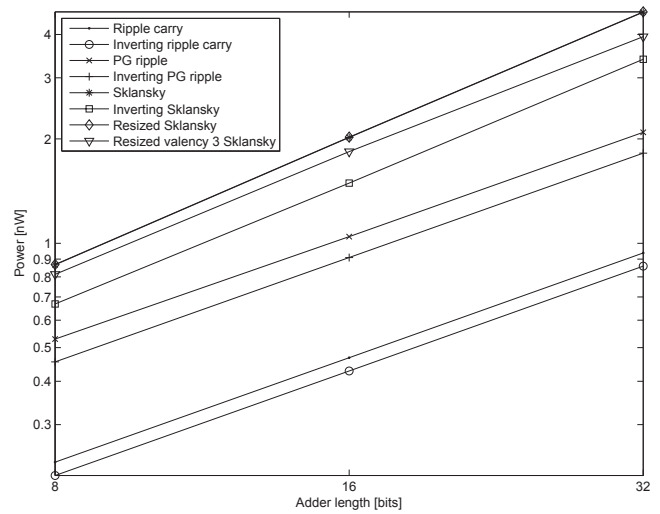


Fig. 10: Adders: leakage power at $V_{dd} = 0.3$ V.

The energy consumed per cycle for a system is

$$E = E_{sw_add} + E_{sw_sys} + (t_{add} + t_{seq})(P_{leak_add} + P_{leak_sys})$$

where E_{sw_add} is the switching energy per cycle for the adder, E_{sw_sys} is the switching energy per cycle for the rest of the system, P_{leak_add} is the leakage power for the adder, and P_{leak_sys} is the leakage power in the rest of the system. If we consider the energy saved by using one adder relative to

another, the result is independent of the switching energy in the system. The remaining unknown factor, the leakage energy in the system, can be taken as a parameter. Fig. 11 shows the energy saved by the different 32-bit adders relative to a ripple-carry adder. The leakage power of the rest of the system is normalized against the leakage from an 8-bit ripple-carry adder. The ripple-carry is most efficient for very simple systems, but the more complex tree adders demonstrate an

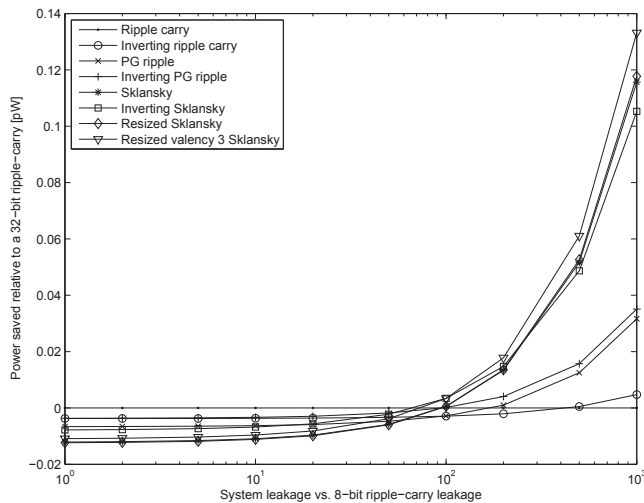


Fig. 11: The energy saved by the different 32-bit adders relative to a 32-bit ripple-carry adder, assuming the adder delay sets the cycle time.

energy saving for systems of very modest complexity with the valency 3 adder being the most efficient.

IV. CONCLUSION

For the 180 nm CMOS process simulated, the behaviour of static CMOS logic gates at subthreshold voltage is not dramatically different to the behaviour one expects at superthreshold voltage – provided process and environment variations are ignored. Absolute switching delays increase exponentially as the supply drops, but switching energy falls quadratically. Static noise margins are well behaved and fall away linearly with the supply level. At a particular operating voltage the linear relationship between fanout and delay is maintained. When normalized against inverter delay at the operating voltage, the logical effort and parasitic delay are not dissimilar to values familiar from superthreshold design. In other words, the relative behaviour of gates is approximately maintained as V_{dd} drops below V_t . An interesting exception is fanin. It was observed that the logical effort of 3 and 2-input NAND gates was approximately equal. This improved tolerance to fanin at subthreshold voltage carried through to adder architectures with valency 3 Sklansky adders being the fastest of the adders tested.

Experiments with the P:N ratio in an inverter showed that static noise margin could be improved by a small amount by increasing the size of the pMOS FET; however this was at the cost of switching energy and leakage power and provided no significant benefit for average propagation delay. Increasing the width of all of the transistors in the critical high fanout gates of the Sklansky adders did improve delay and hence save leakage energy throughout the system.

The results in Fig. 11 show that when the adder is on the critical timing path of the system, leakage in the rest of the system quickly outweighs the switching and leakage cost of

a faster adder. A ripple-carry adder is most energy efficient for simple systems, but the valency 3 Sklansky adder became most efficient when system leakage exceeded around 100 times the leakage of an 8-bit ripple-carry adder. This effect can be expected to be even more pronounced for feature sizes smaller than 180 nm in which leakage increases relative to switching energy.

A. Future Work

Short (2 to 8-bit) carry chains are important building blocks in larger adders and should be examined at subthreshold voltage. While most pass-transistor logic styles will not work at subthreshold voltage, some transmission-gate styles will work, as will the bridge logic style of [11]. It would be interesting to compare these logic styles with static CMOS at subthreshold voltage. It would also be interesting to confront the challenges posed by dynamic logic at subthreshold voltage. A comparison between static CMOS and Manchester carry chains would be interesting.

The differences observed in this paper between circuits at superthreshold and subthreshold voltage should be explained in terms of the underlying mechanisms. In particular the tolerance of the subthreshold gates to fanin (Section II-B2) is worth deeper examination.

Despite the conclusion of this paper (fast adders are good), the broader question of what kinds of system architectures are best for subthreshold applications remains open. For example, is it better to choose a long-wordlength datapath to get the task done in a few cycles, or take a bit serial approach?

REFERENCES

- [1] M. Seok, S. Hanson, Y.-S. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw, "The phoenix processor: A 30pW platform for sensor applications," in *Proc. IEEE Symposium on VLSI Circuits (VLSI-Symp)*, June 2008, pp. 188–189.
- [2] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, Jan. 2005.
- [3] R. Zimmermann and W. Fichtner, "Low-power logic styles: Cmos versus pass-transistor logic," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 7, July 1997.
- [4] S. Hanson, B. Zhai, M. Seok, B. Cline, K. Zhou, M. Singhal, M. Minuth, J. Olson, L. Nazhandali, T. Austin, D. Sylvester, and D. Blaauw, "Exploring variability and performance in a sub-200-mV processor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 881–891, Apr. 2008.
- [5] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "Variation-tolerant sub-200mV 6-T subthreshold SRAM," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 10, pp. 2338–2348, Oct. 2008.
- [6] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, and T. Austin, "A 2.60pJ/inst subthreshold sensor processor for optimal energy efficiency," in *Proc. IEEE Symposium on VLSI Circuits (VLSI-Symp)*, June 2006, pp. 303–307.
- [7] I. Sutherland, B. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.
- [8] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison Wesley, 2005.
- [9] J. Sklansky, "Conditional-sum addition logic," *IRE Transactions on Electronic Computers*, vol. EC-9, pp. 226–231, 1960.
- [10] D. Patil, O. Azizi, M. Horowitz, R. Ho, and R. Ananthraman, "Robust energy-efficient adder topologies," in *Proc. 18th IEEE Symposium on Computer Arithmetic*, June 2007, pp. 16–25.
- [11] K. Navi, O. Kavehie, M. Rouholamini, A. Sahafi, and S. Mehrabi, "A novel CMOS full adder," in *Proc. 20th International Conference on VLSI Design (VISID'07)*, 2007, pp. 303–307.