Design of an Energy-Efficient 32-bit Adder Operating at Subthreshold Voltages in 45-nm CMOS

Anh T. Tran and Bevan M. Baas

Department of Electrical and Computer Engineering
University of California - Davis, USA
{*anhtr, bbaas*}*@ucdavis.edu*

***Abstract*—Low-power circuits have been quickly increasing their importance due to the high cost in design of cooling systems with complex chip packaging techniques, and also due to the low energy consumption requirement of portable devices powered by a limited battery capacity. This paper presents the design of a low-power 32-bit adder that is a basic functional unit in most computational platforms. Its energy efficiency is highly achieved while operating in the subthreshold regime. Simulation results in 45-nm PTM CMOS show the adder consumes only 22 fJ per computation at 0.2 V with maximum operating frequency of 3.8 MHz. While targeting an acceptable frequency of 100 MHz, it consumes only 3.4 µW or 34 fJ per computation at 0.37 V. At 1.0 V, it can operate at up to 2.85 GHz while consuming only 735 µW or 257 fJ per computation. It also stably operates at only 0.1 V with throughput of 400 kHz and consumes 18 nW.**

I. Introduction

The number of transistors integrated on a single chip still dramatically increases with Moore's Law. The chip temperature is becoming higher reaching a level that can even destroy the chip itself. High power dissipation per an area unit makes the design for chip cooling systems more complicated and expensive. Furthermore, portable devices require their hardware to consume as low energy as possible in order for they can operate longer with a limited battery capacity. Therefore, low-power circuit design has become more important or even been a mandatory requirement.

Many low-power digital circuit design techniques were addressed by Chandrakasan in his classic paper [1]. For many battery-based portable devices, however, energy-efficient computation while offering acceptable throughputs is becoming a popular research topic. A recent paper by Wang *et al.* reported a circuit design methodology for minimizing energy consumption by exploring its operation in the subthreshold region [2]. At the 180-nm CMOS technology process, a FFT processor designed by those authors can operate at a very low voltage of only 0.18 V.

In this paper, we report design of a 32-bit adder that is a basic element of most digital computational systems. The design methodology emphasizes the robust operation of the adder at the subthreshold region in 45-nm PTM CMOS node [3]. Robustness requirement is evaluated by measuring its output voltages representing high logic V_{OH} and low logic V_{OL} while driving fan-out of 4 (FO4) equivalent loads. Given a supply voltage V_{DD}, the adder is robust if V_{OH} is larger than $0.9V_{DD}$ and V_{OL} is smaller than $0.1V_{DD}$. This requirement guarantees the adder to be strong enough for driving its downstream loads in a complete computational system instead of only evaluating the adder alone using $0.5V_{DD}$ as a switching threshold for determining logic 1 or 0 as seen in previous work.

For the same design, scaling supply voltage down would make lower consumption but also increases the latency. Energy dissipation, that is product of the consumed power and latency, is expected to be minimal at a point in the subthreshold regime [2]. However, at this point, the circuit latency may so high that does not satisfy the requirement of some applications. Therefore, design to achieve fast computational speed for offering acceptable throughputs even at the subthreshold voltages is also an important requirement. In this work, we target a throughput of 100 million addition operations per second in the subthreshold region. At the 45-nm PTM CMOS, the typical supply voltage is 1.0 V [4] with the assumed threshold voltage is at around 0.4 V. Therefore, the target computation frequency is 100 MHz at 0.4 V; higher achievable frequency is better.

The rest of this paper is organized as follows: Section II presents the proposed 32-bit adder architecture. Section III shows in details the design of all circuit elements of the adder. Simulation and analysis results are shown in Section IV; and, finally, Section V concludes this paper.

II. 32-bit Adder Architecture

Many techniques for designing a fast adder can easily be found in the literature [5], [6]. However, design for an adder to operate at very-low subthreshold voltages comes with some critical trade-offs among performance and power as well as its robustness. In this section, we discuss which adder architecture be mostly suitable in context of simplicity with low-area (so potentially low power), high robustness ($V_{OH} > 0.9V_{DD}$ and $V_{OL} < 0.1V_{DD}$) and acceptable speed at subthreshold voltages (larger than 100 MHz at 0.4 V) as described in Section I.

The Ripple-Carry Adder (RCA) is very slow because in the worst-case the carry can propagate through all 32 Full Adders (FA) from C_{in} to C_{out}. It would be so difficult to get an adder meeting the throughput requirement of 100MHz at around 0.4V. The Carry Look-Ahead Adder (CLA) has good speed; however, it requires high fan-in circuits for computing the group-generated carry (*GG*) and group-propagated carry (*PG*) signals. High fan-in circuits should be avoided at subthreshold voltages due to their low I_{on}/I_{off} ratio [2]. If we

978-1-4244-7057-0/10/$26.00 ©2010 IEEE

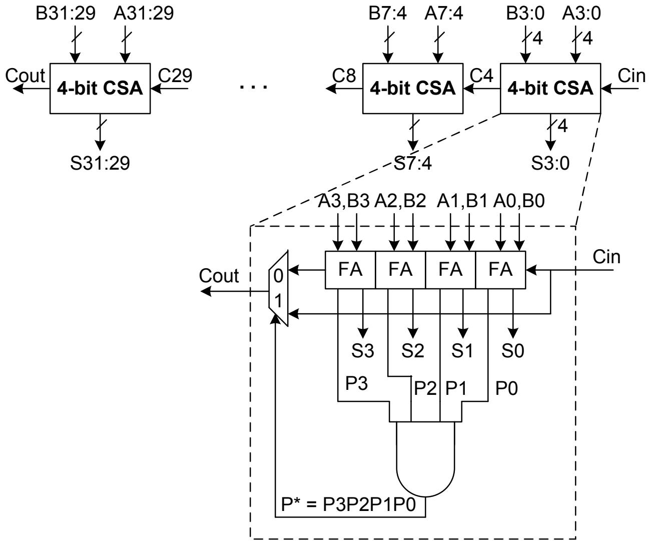

Fig. 1. Block diagram of a 32-bit Carry-Skip Adder using eight 4-bit blocks

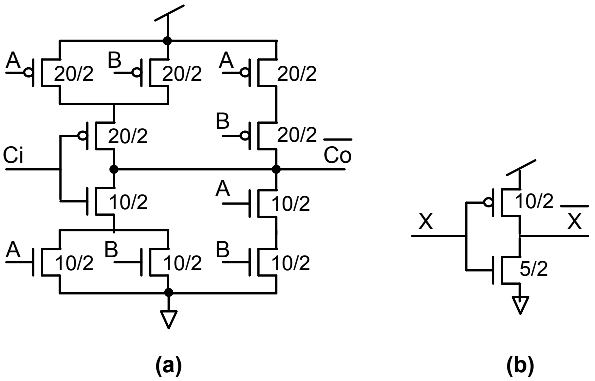

Fig. 2. a) The mirror circuit to compute $\overline{Co}$ with sizing information; b) a minimum size balanced inverter.

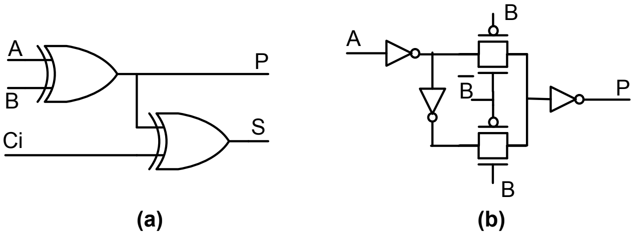

Fig. 3. a) Logic gates for P and S; b) the circuit of a XOR

size up transistors of these circuits to have good I_{on}/I_{off}, their capacitances go up and therefore consume more power.

The Conditional Sum Adder and Carry-Select Adder require two redundant subcircuits to compute the results corresponding to $C_{in} = 0$ and $C_{in} = 1$ before choosing the correct result once C_{in} is available. Their large circuit areas consume high power. Some Prefix Adders such as Brent-Kung, Ladner-Fischer, etc. require long interconnect wires between stages. In advanced deep submicron CMOS processes, interconnect wires are no longer scaled as well as transistors [7]; therefore, long wires can contribute much more power and latency to the adder than CMOS gates themselves, especially at very low subthreshold voltages.

From analysis above, we choose the Carry-Skip Adder (CSA) architecture because it has good performance compared to a CLA adder with simple and regular design that has low area (also potentially has low power) and high robustness in the subthreshold regime. The CSA adder can divided into many small blocks with carry-skip circuit inside each block. We use 4-bit fixed-size blocks and connect eight these blocks to form a 32-bit adder as depicted in Fig. 1. Some techniques for speeding up CSA adders by using variable-size blocks and/or inside-block carry skipping circuits were proposed in [8], [9]; however, their complexity and non-modularity would consume high power and, therefore, have high energy (in spite of low delay) at low voltages.

In Fig. 1, each 4-bit CSA block is a sequence of four 1-bit Full Adders (FA). The critical path of each block is the carry propagation path through all 4 FAs. Therefore, design to get a fast carry-out bit for each FA is mostly important. The carry is skipped if all $P_i = 1$, where $i = 0..3$ and $P_i = XOR(A_i, B_i)$.

III. Circuit Design

In this section, we describe the design and optimization of all elements shown in Fig. 1. These elements are 1-bit full-adders (FA), 4-bit CSA blocks with their carry-skip signals and also the overall 32-bit adder circuit formed from these 4-bit blocks.

A. Design of a Full-Adder (FA)

The logic equations of a 1-bit FA with its carry-skip signal are:

$$S = A \oplus B \oplus C_i \quad (1)$$

$$C_o = AB + C_i(A + B) \quad (2)$$

$$P = A \oplus B \quad (3)$$

Because C_o will be an input of the next FA (in the worst-case, the carry would propagate through all FAs), so it is on the critical path that is needed to be quickly computed. We adopt the mirror circuit for computing C_o [6] as shown in Fig. 2(a). This circuit calculates the inverted value $\overline{C_o}$. Its transistors are sized to have the worst-case pullup and pulldown times equivalent to a minimum-size inverter with switching threshold at $0.5V_{DD}$ as shown in Fig. 2(b)[1].

This mirror circuit is fast because the inverted $\overline{C_o}$ is available as soon as A, B and C_i occur. Its delay is equal to only one minimum-size inverter. Besides that, the pullup and pulldown networks only have stack of two transistors that should give a good I_{on}/I_{off} ratio [2].

The logic circuits for S and P are shown in Fig. 3(a) using XOR gates (Fig. 3(b)). Delay of P and S are one and two XOR gate delays, respectively. Since logic S is not on the critical path, its delay is negligible in compared to the delay of whole 32-bit adder (which is mainly determined by the carry propagation path).

[1] Minimum sizes of CMOS and NMOS transistors of the balanced inverter are obtained through simulation using 45-nm PTM CMOS technology card [3].

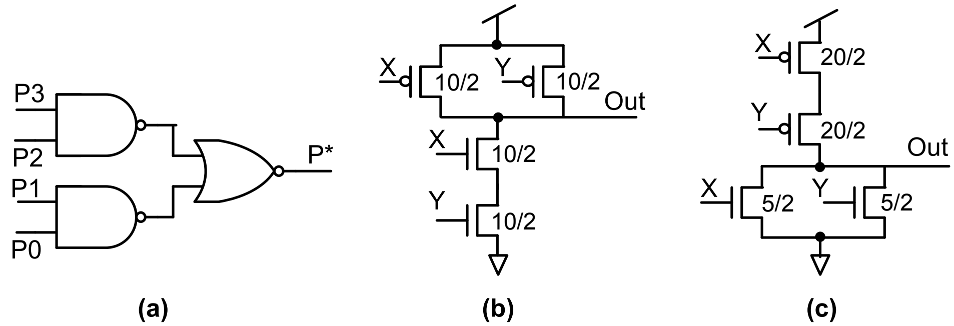

Fig. 4. a) P^* logic; b) 2-input NAND gate; c) 2-input NOR gate.

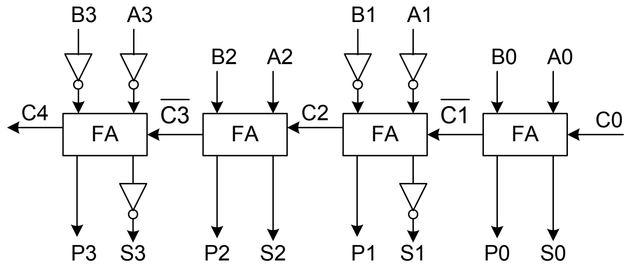

Fig. 5. A 4-bit CSA block connected from four FAs

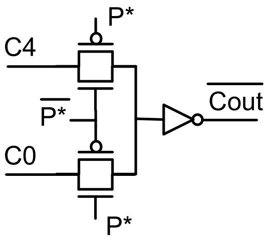

Fig. 6. The 2-input MUX at output of each 4-bit CSA block

Because keeping the I_{on}/I_{off} ratio high in the subthreshold region is very important, all XOR gates are output buffered (by an inverter) as shown in Fig. 3(b). In our simulation results, if without using these output buffers, at 0.2 V the output S has V_{OH} (logic 1) less than $0.9V_{DD}$. Furthermore, without using buffers for XOR gates, the weak driving current also makes the circuit become much slow.

B. 4-bit CSA Blocks

Each 4-bit CSA block is formed by cascading four FAs. As shown in Fig. 2(a), the mirror circuit only produces $\overline{C_o}$. So, if we want to connect two FAs, we need to invert it back to C_o. However, this can slow down the whole circuit because it will add more inverters to the carry propagation path.

Fortunately, if we invert all A, B and C_i signals, from the equations (1, 2, 3), value of P does not change while values of both S and C_o are inverted. From this observation, the 4-bit CSA block can be formed as depicted in Fig. 5. As shown in the figure, no inverter is needed at the carry outputs of each FA.

C. Carry-Skip Circuit

The carry is skipped if $P^* = P_3P_2P_1P_0 = 1$ (as shown in Fig. 1). To avoid a high fan-in circuit that can make the I_{on}/I_{off} ratio low at subthreshold voltages, the logic P^* is formed from only 2-input NAND and NOR gates as shown in Fig. 4(a) (instead of an 4-input AND gates). The 2-input NAND and NOR gates are shown in Fig. 4(b) and (c) with transistor sizing in order that they have worst-case pullup and pulldown times equivalent to a minimum size balanced inverter (which was shown in Fig. 2(b)).

The carry out of a 4-bit CSA block is gotten from a 2-input MUX. This MUX uses a transmission gate with buffer (by an inverter) at output as shown in Fig. 6. Benefit of the output buffer here is two-fold. First, it makes $\overline{C_{out}}$ stronger at output of each 4-bit CSA block. Second, it avoids the case when C_0 can be skipped and travels through all eight 4-bit blocks without any intermediate buffer. This will seriously degrade the final carry output signal due to weak driving current (so it will not meet the robustness requirement of $V_{OL} \leq 0.1V_{DD}$ and $V_{OH} \geq 0.9V_{DD}$ even at high supply voltages), and also slows down the overall speed of the adder.

However, with this output MUX buffer, the carry out of 4-bit block is inverted. Again, instead of using one more inverter for inverting this carry out before connecting it to the next block, we can build the next 4-bit block with inputs inverted as depicted in Fig. 7. These two 4-bit CSA blocks form an 8-bit adder block. The final 32-bit adder is formed by cascading these four 8-bit blocks.

IV. Simulation Results

The worst-case delay of this 32-bit CSA adder (Fig. 1) occurs when the carry C_{in} propagates through the first 4-bit block, then skips six intermediate blocks and propagates again through the last block [5]. This happens when $A_{31:0}$ changes from ’0000 ... 0000’ to ’0000 ... 0001’ while $B_{31:0}$ moves from ’0000 ... 0000’ to ’0111 ... 1111’.

In order to measure the adder circuit on SPICE, we drive all inputs by inverters powered by V_{DD} (that is the same supply voltage for the adder), and all outputs are connected to FO4 load inverters powered by a different V_{CC} source (to avoid taking the load power into account). In simulation results, all outputs meet the robustness requirement of $V_{OL} \leq 0.1V_{DD}$ and $V_{OH} \geq 0.9V_{DD}$ for all supply voltages from 0.1V to 1.0V.

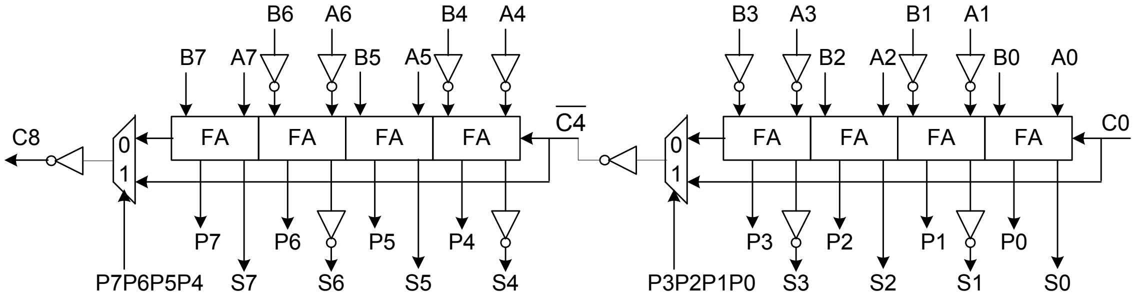

Fig. 7. Connection of two 4-bit CSA with carry out inverted to form a 8-bit adder.

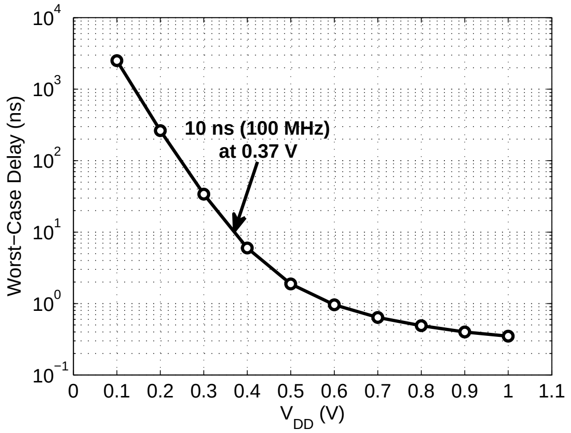

Fig. 8. Worst-case delay vs. supply voltage

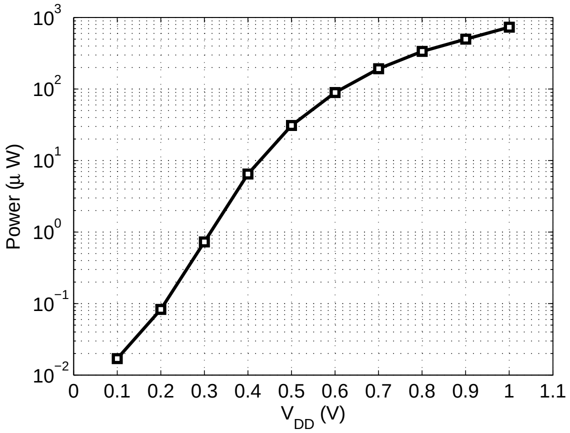

Fig. 9. Power vs. supply voltage

A. Worst-case Delay versus Supply Voltage

The maximum delay of the adder is measured from when the A_0 rising edge reaches $0.9V_{DD}$ until the S_{31} rising edge also reaches $0.9V_{DD}$ under the worst-case input pattern described above. The maximum delays of 32-bit adder corresponding to different supply voltages V_{DD} from 0.1 V to 1.0 V are shown in Fig. 8 (in log scale). At 1.0 V, the adder has maximum delay of 0.35 ns that allows it to compute at a throughput of 2.85 GHz. The delay linearly increases as supply voltage reduces from 1.0 V to 0.6 V; however, it begins to exponentially increase as supply voltage is below 0.5 V. This happens because the adder is now operating in the subthreshold region, in which the transistor threshold voltage becomes large compared to the supply voltage so that the circuit speed is much slow [10].

However, as shown in the figure, the adder has maximum delay of 10 ns (or 100 MHz computational frequency) at around 0.37 V that meets our speed requirement. At 0.4 V, the adder has maximum delay of 5.99 ns, so it can operate up to 167 MHz. At 0.1 V, it still stably operates with delay of 2512 ns or at the frequency of 400 kHz.

B. Average Power and Energy

For mostly accurate evaluation of the power consumption, random input patterns are used instead of the worst-case one. Since the delay depends on the supply voltage V_{DD}, so for each V_{DD} we have to regenerate inputs corresponding to the delay that was reported in Fig. 8. The number of random input vectors used for simulation at each V_{DD} is 100. The power consumption corresponding to supply voltages is shown in Fig. 9.

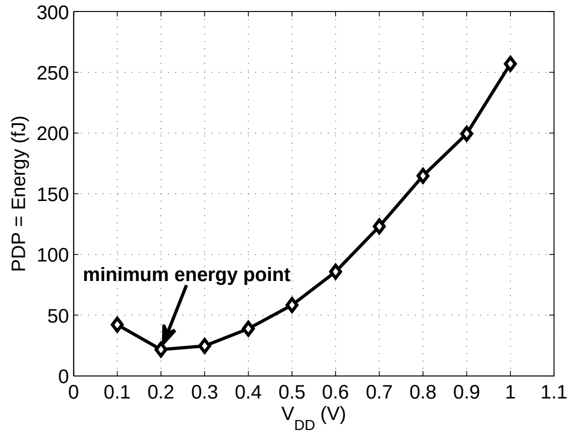

Fig. 10. Average energy (PDP) vs. supply voltage

The power exponentially reduces as the supply voltage reduces. Therefore, the average energy, that is calculated by the Power Delay Production (PDP), also reduces when the supply voltage downs as shown in Fig. 10. This is because the power reduction is much faster than the increase of the circuit delay.

However, interestingly, when supply voltage is less than 0.2 V, the average energy begins to increase. This happens because at these very low voltages, the leakage power becomes dominant that can be comparable with the dynamic power

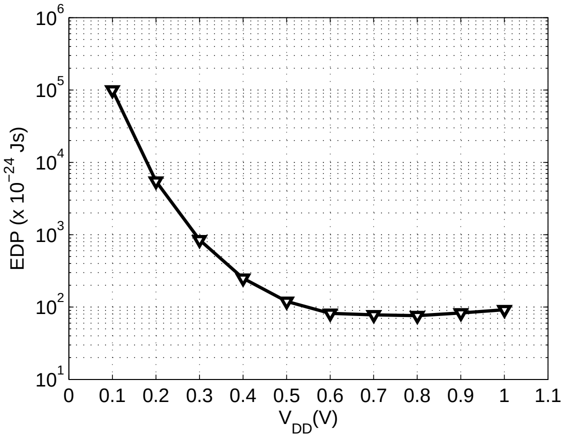

Fig. 11. Energy Delay Production (EDP) vs. supply voltage

itself. As a result, the minimum energy point occurs at around 0.2 V that is about 22 fJ per addition. This result is attractive for energy-efficient design of portable devices that require to dissipate low energy while operating with limited battery sources.

C. Energy-Delay Product (EDP)

Another important metric to evaluate the design efficiency of a circuit is the product of its energy and delay (EDP). This metric enables a flexible trade-off between energy and circuit speed. Clearly, we want to run at low supply voltage to have small energy; however, the circuit is slow or has high delay that makes its EDP high. Therefore, a circuit with low EDP would be highly expected.

Fig. 11 shows the EDP of the adder over various supply voltages. Due to its large delay at subthreshold voltages, at less than 0.5 V the EDP increases near-exponentially with the decrease of supply voltage. However, when V_{DD} is larger than 0.6 V, the EDP looks like unchanged and slightly goes up beginning at 0.9 V. As a result, 0.6 V-0.9 V is the minimum EDP range of the adder.

V. Conclusion

We have presented the design of a low-power and energy-efficient 32-bit adder. The design is robust allowing it to efficiently operate in the subthreshold regime with very low supply voltages. The adder can achieve 167 MHz throughput at 0.4 V with only 6.5 μW and 39 fJ per computation. At 0.2 V, it achieves the minimum energy with only 22 fJ per computation. The adder is also able to stably operate at the voltage of only 0.1 V with 400 kHz throughput and 18 nW power consumption.

Even though we only reported the design of a 32-bit adder, the design methodology presented here should be applied well to a higher precision such as 64-bit because the carry still only propagates through maximum of only two 4-bit CSA blocks under the worst-case pattern. The total circuit delay is slightly added when the carry is skipped through more intermediate blocks (which consists of only a 2-input MUX and an inverter buffer as depicted in Fig. 7). However, as shown in Section IV-C, the adder will have minimum EDP point at a voltage higher than the transistor threshold. Design and evaluation of an 64-bit adder with minimum EDP point in the subthreshold region is our future work.

Acknowledgments

The authors would like to thank professor R. Amirtharajah for useful comments on low-power and subthreshold circuit design methods. This work was supported in part by a VEF Fellowship, SRC GRC Grant 1971, and NSF Grant 0903549.

References

[1] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low power CMOS digital design,” *JSSC*, vol. 27, pp. 473–484, 1992.

[2] A. Wang and A. P. Chandrakasan, “A 180-mv subthreshold fft processor using a minimum energy design methodology,” *IEEE JSSC*, vol. 40, no. 1, pp. 310–319, 2005.

[3] PTM, “Predictable technology model,” Online, http://ptm.asu.edu/.

[4] W. Zhao and Y. Cao, “New generation of predictive technology model for sub-45nm early design exploration,” *IEEE TED*, vol. 53, pp. 2816–2823, Nov. 2006.

[5] I. Koren, *Computer Arithmetic Algorithms,* A K Peters Ltd., second edition, 2002.

[6] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective,* Prentice-Hall, New Jersey, U.S.A, second edition, 2003.

[7] R. Ho, K. W. Mai, and M. A. Horowitz, “The future of wires,” *Proceedings of the IEEE*, vol. 89, pp. 490–504, Apr. 2001.

[8] P. K. Chan, M. D. F. Schlag, et al., “Delay optimization of carry-skip adders and block carry-look ahead adders using multidimentional dynamic programming,” *IEEE Trans. on Computers*, vol. 41, no. 8, pp. 920–930, 1992.

[9] V. Kantabutra, “Designing optimum one-level carry-skip adders,” *IEEE Trans. on Computers*, vol. 42, no. 6, pp. 759–764, 1993.

[10] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, *Sub-threshold Design for Ultra Low-Power Systems,* Springer Science, first edition, 2006.