Optimal Sequencing Energy Allocation for CMOS Integrated Systems

Martin Saint-Laurent[*], Vojin G. Oklobdzija[**], Simon S. Singh[†], and Madhavan Swaminathan[‡]

[*] Intel Corporation, Austin, Texas, United States of America
[**] University of California, Davis, California, United States of America
[†] National Semiconductor Corporation, Santa Clara, California, United States of America
[‡] Georgia Institute of Technology, Atlanta, Georgia, United States of America

Abstract

All synchronous CMOS integrated systems have to pay some sequencing overhead. This overhead includes the skew and the jitter of the clock. It also includes the setup time and the clock-to-output delay of the flip-flops. This paper discusses how much energy should be allocated for sequencing in these systems. It is pointed out that providing too little energy is just as bad as providing too much. It is also argued that directly trying to minimize the energy-delay product of the sequencing subsystem is practically not the right thing to do. A model for the relationship between supply voltage, clock frequency, and power dissipation is developed and empirically verified for a SPARC V9 microprocessor. An expression for the optimal energy allocation in a system is derived. Then, based on this optimum, a methodology to design energy-efficient systems is proposed.

Introduction

All digital systems have to pay some sequencing overhead. For synchronous systems, this overhead includes the skew and the jitter of the clock. It also includes the setup time and the clock-to-output delay of the flip-flops. The typical elements required for sequencing are shown in Fig. 1. The clock distribution network is one of them.

Choosing the right amount of sequencing overhead is non-trivial. For instance, how should the clock inaccuracy targets be set? Should the clock skew between adjacent functional units be 10% of the clock cycle? Or should it be 15% to reduce the clock distribution power? For sequential elements, maybe the data-to-output delay can be improved by making the clocked transistors bigger. If so, should it and by how much? The problem addressed here is to define a systematic way of making these kinds of tradeoffs.

In the clock distribution literature, the problem of determining the right amount of skew and jitter is typically ignored [1]. Instead, the focus is on meeting certain clock inaccuracy targets while dissipating as little power as possible [2]. Sometimes, the focus is instead on meeting a certain power target while minimizing the clock inaccuracy [3]. In both cases however, these design targets are set using *ad-hoc* methodologies, which may or may not be energy-efficient.

For sequential elements, a power-delay product minimization strategy is often used to choose the setup time and the clock-to-output delay [4]. This approach is reasonable in practice because it considers the additional power consumed by the clock distribution network when the clocked transistors increase in size. But the size of the clocked transistors also influences the clock inaccuracy [3]. The problem is that this dependency, which is typically strong, is ignored.

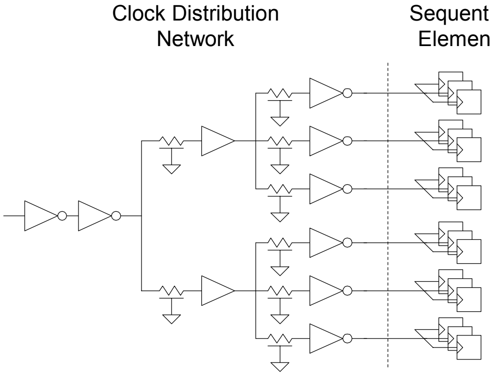

Fig. 1: Sequencing in CMOS integrated circuits.

This paper proposes a systematic method for choosing the right amount of sequencing overhead in a system from an energy-efficiency standpoint. The value of this method is that it can be used to jointly optimize the clock distribution network and the sequential elements. First, the relationship between supply voltage, clock frequency, and power dissipation is examined. It is argued that reducing the sequencing overhead is equivalent to saving power. The equivalence is used to derive an expression for the optimal allocation of the energy used for sequencing. Then, based on this optimum, a methodology to design energy-efficient systems is proposed.

Relationship Between Voltage, Power Dissipation, and Frequency

Most CMOS integrated circuits can be classified as either frequency-limited or power-limited. A chip is frequency-limited if its maximum operating frequency is determined by its slowest timing path at the maximum supply voltage allowed. Conversely, a chip is power-limited if it reaches the maximum power dissipation allowed when its supply voltage, its clock frequency at this voltage, or both, are below their upper limit.

Proceedings of the International Symposium on Quality Electronic Design (ISQED'02)
0-7695-1561-4/02 $17.00 © 2002 IEEE

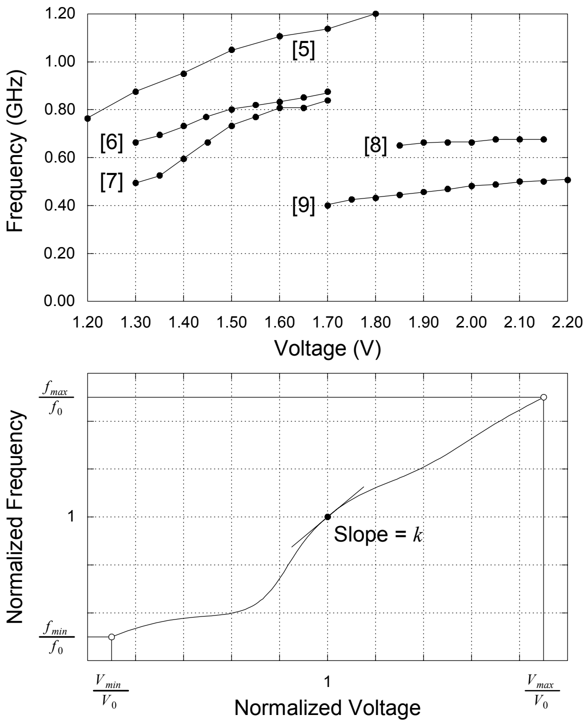

Fig. 2: Measured (top) and general (bottom) frequency-voltage relationships.

From a sequencing energy allocation standpoint, frequency-limited chips are not very interesting. There is no power tradeoff to make: they should simply be made as fast as possible. The power dissipated is unimportant because it does not exceed the budget. Power-limited chips however are more interesting and also more common. They may still be designed for high frequency, but not for high frequency at any price. The rest of this paper only considers power-limited chips. Frequency-limited chips are increasingly rare.

Fig. 2 shows the measured frequency-voltage relationship for five recent microprocessors [5-9]. These chips are all power-limited based on the definition given above.

A. Power Dissipation

The power dissipation P of any of these microprocessors as a function of the supply voltage V and the operating frequency f can be approximated by:

$$\begin{aligned} P &= P_d + P_s + P_{leak} \\ &= C_d V^2 f + Q_s V f + I_{leak} V \end{aligned} \tag{1}$$

The first term is the dynamic power. C_d represents the effective capacitance switched per cycle and is assumed independent of the supply voltage. The second term is the short-circuit power. Q_s is the average charge flowing directly from the positive to the negative power rail during a clock period. According to the model proposed in [10], Q_s is proportional to V. The last term is the subthreshold leakage power. I_{leak}, the subthreshold leakage current, is [11]:

$$I_{leak} = \kappa(1 - e^{-\beta V_{ds}})e^{\frac{\beta}{\eta}(V_{gs} - V_t)} \tag{2}$$

where κ is a technology and design dependent constant, β is the inverse of the thermal voltage, η is the subthreshold swing coefficient, and V_t is the threshold voltage. When V_{ds} is much larger than the thermal voltage and when V_{gs} is zero,

$$I_{leak} \approx \kappa\, e^{-\frac{\beta}{\eta}V_t} \tag{3}$$

As the supply voltage increases, the drain-induced barrier lowering (DIBL) effect linearly lowers V_t. Thus, it introduces an exponential relationship between I_{leak} and V. However, it is assumed here that the DIBL effect is small and that I_{leak} is independent of the supply voltage.

B. Frequency

To analyze how the frequency of operation is related to the supply voltage, it is useful to consider the logic path limiting the frequency. Let its delay be D. Its sequencing overhead S can be written as the sum of the latency L of its sequential elements and the clock inaccuracy y:

$$S = L + y \tag{4}$$

For flip-flops, L is the setup time plus the data-to-output delay. It is worth noting that certain skew-tolerant design techniques can partially hide the clock inaccuracy [12]. These techniques are very important. They are treated here as a mean of making the effective clock inaccuracy smaller. The clock inaccuracy is the sum of the skew and the jitter.

The supply voltage affects D and S in a complicated way. But based on the empirical data shown in Fig. 2 and on the short-channel transistor model proposed in [13], it is reasonable to assume a simple linear relationship between f and V when V is near some nominal value V_0:

$$f = \frac{1}{D+S}\left(1 + k\frac{V - V_0}{V_0}\right) \tag{5}$$

In (5), k is a frequency-scaling factor with no units. As shown at the bottom of Fig. 2, k measures the ease of increasing the clock frequency by increasing the supply voltage. Definitions for the nominal voltage and the nominal sequencing overhead are given later. When the supply voltage and the sequencing overhead are nominal, so is the frequency.

Benefit of Reducing the Sequencing Overhead

Increasing the supply voltage of a chip and reducing its sequencing overhead are two equivalent ways of improving its frequency. In each case, the power dissipation increases. But generally, the power increase associated with a supply voltage increase is greater than the power increase associated with a sequencing overhead reduction. Because of this, the

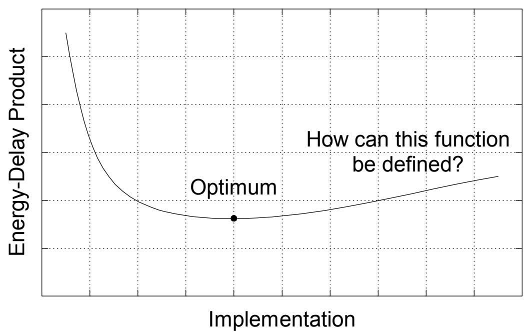

Fig. 3: The problem with energy-delay minimization.

sequencing overhead reduction is equivalent to a power savings at the new frequency.

To quantify this benefit, it is necessary to compare the cost of increasing frequency by increasing the supply voltage to the cost of increasing frequency by reducing the sequencing overhead. The two are generally not equal. For the supply voltage increase ΔV and the sequencing overhead reduction ΔS to produce the same given frequency improvement Δf, the following equality must hold:

$$f(V_0 + \Delta V, S_0) = f(V_0, S_0 - \Delta S) = f_0 + \Delta f \qquad (6)$$

Solving for ΔV and ΔS yields:

$$\begin{aligned} \Delta V &= \frac{V_0 \Delta f}{k f_0} \\ \Delta S &= \frac{1}{f_0} - \frac{1}{f_0 + \Delta f} \end{aligned} \qquad (7)$$

If the supply voltage is increased, the power dissipated at the new frequency is $P(V_0 + \Delta V, S_0)$. But if the sequencing overhead is decreased, the power dissipated at the new frequency is $P(V_0, S_0 - \Delta S)$. In both cases, the power dissipation increases because the new frequency is higher. However, the voltage is not increased when the sequencing overhead is reduced. Therefore, $P(V_0 + \Delta V, S_0)$ is greater than $P(V_0, S_0 - \Delta S)$ and the sequencing overhead improvement is equivalent to a savings in power. Mathematically, the difference is:

$$\begin{aligned} \Delta P &= P(V_0 + \Delta V, S_0) - P(V_0, S_0 - \Delta S) \\ &= \frac{1}{k}(2C_d V_0^2 f_0 + 2Q_s V_0 f_0 + I_{leak} V_0)\left(\frac{f_0 \Delta S}{1 - f_0 \Delta S}\right) + \\ &\frac{C_d V_0^2 f_0 + Q_s V_0 f_0}{k^2} \frac{1 + 2k(1 - f_0 \Delta S)}{1 - f_0 \Delta S} \left(\frac{f_0 \Delta S}{1 - f_0 \Delta S}\right)^2 \end{aligned} \qquad (8)$$

From (7), ΔV is a function of Δf, which can be considered a function of ΔS. Therefore, ΔP is also a function of ΔS. It is worth noting that ΔP is always positive. Equation (8) quantifies the benefit associated with a sequencing overhead reduction.

The SPARC V9 microprocessor described in [5] is useful to illustrate the concept. Given its technology, the leakage power is assumed negligible. A nominal dynamic power dissipation of 80.0 W is reported when the supply voltage is 1.60 V and the frequency is 1105 MHz. Measurements indicate that increasing the supply voltage to 1.70 V increases the frequency to 1135 MHz. This is equivalent to decreasing the clock cycle by 24 ps. This increase in voltage results in a power increase of 12.8 W. If the sequencing overhead for this chip was somehow reduced by 24 ps, the frequency would still be 1135 MHz. But then, the power dissipation would only be 82.2 W. Reducing the sequencing overhead by 24 ps is thus equivalent to a power savings of 12.8 – 2.2 = 10.6 W. This is the benefit of reducing the sequencing overhead.

Practical Difficulties with Energy-Delay Product Minimization

In principle, the sequencing energy allocation is easy to optimize. For a microprocessor, the runtime R of a program having n instructions is $R = n / f$ and the energy required to execute it is $E = PR$. Like before, f is the operating frequency of the chip and P is its power dissipation. Minimizing the energy-delay product ER is equivalent to minimizing P / f^2 because n is fixed.

The problem is that directly minimizing ER is impractical. There exist an optimal balance between the energy allocated for clock distribution and the energy allocated for the sequential elements. Finding this optimum requires a joint optimization of the two. Unfortunately, this joint optimization is impractical because ER is a complicated function of the supply voltage and an intractable function of the microprocessor implementation. This is shown in Fig. 3.

It has been suggested in [14] that energy efficiency is more accurately measured using an energy-delay^2 metric rather than an energy-delay metric. But clearly, minimizing ER^2 is generally not easier than minimizing ER. A more practical way of optimally allocating the energy for sequencing is discussed in the following section.

Optimal Energy Allocation

The proposed optimal energy allocation strategy is based on the marginal benefit B (measured in mW / ps) associated with one unit of sequencing overhead reduction. The marginal benefit is, by definition, the savings in power normalized to the reduction:

$$\begin{aligned} B &\equiv \lim_{\Delta S \to 0} \frac{\Delta P}{\Delta S} \\ &= \frac{1}{k}(2C_d V_0^2 f_0 + 2Q_s V_0 f_0 + I_{leak} V_0) f_0 \\ &= \frac{1}{k}(2P_{d0} + 2P_{s0} + P_{leak0}) f_0 \end{aligned} \qquad (9)$$

0-7695-1561-4/02 $17.00 © 2002 IEEE

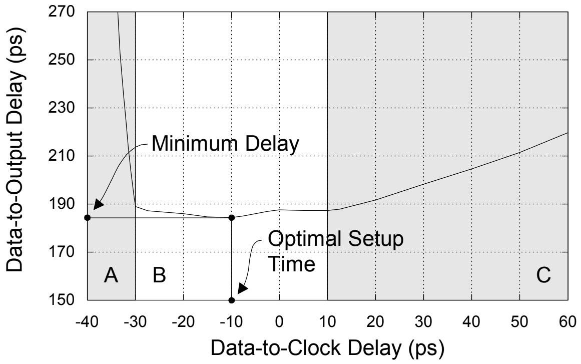

Fig. 4: Data-to-output characteristic.

A design change aimed at reducing the sequencing overhead should be implemented only if its marginal cost is below *B*. If its marginal cost is above *B*, the change requires too much power and it is cheaper to increase the supply voltage to achieve the same frequency improvement.

For the SPARC V9 microprocessor considered earlier, allocating more power for clock distribution could hypothetically reduce the sequencing overhead by 24 ps. This extra power could be used to drive more global clock grid wires for instance. If the additional power does not exceed 10.6 W, then it is cheap to reduce the sequencing overhead. The design change should be implemented. If the cost of reducing the sequencing overhead exceeds 10.6 W, increasing the supply voltage costs less and achieves the same frequency.

Furthermore, a design change should always be implemented if its marginal cost is below *B*, even if the higher frequency is not desired. Although implementing the change costs some power, it increases frequency. But because its marginal cost is below *B*, even more power can be regained by reducing the supply voltage to return to the initial frequency.

The argument for a design change aimed at reducing power is analogous. Its cost in frequency should not exceed its benefit. If it does not, it should be implemented. For a given system, the sequencing energy allocation is optimal if and only if the marginal cost *C* of any design change aimed at reducing the sequencing overhead or the power dissipation is equal to its benefit:

$$B = C \tag{10}$$

It is worth noting that in (9), *B* is roughly equal to $2P_0 f_0$ if the leakage power is small and the frequency-scaling factor is close to unity.

Impact on Clocked Storage Elements

The clocked storage elements play a major role in determining the sequencing overhead. It is observed in [15] that some of them exhibit a desirable property: under certain circumstances, the clock inaccuracy has very little effect on their delay.

A. Clock Inaccuracy Absorption

Fig. 4 shows the data-to-output delay of the hybrid latch-flip-flop (HLFF) used on the AMD K6 microprocessor [16]. The horizontal axis represents the time difference between the data and the clock arrival. A negative value means the data arrives after the clock (the HLFF has a negative setup time). Fig. 4 has three distinct regions. If the data arrives later than 30 ps after the clock, the HLFF fails. This corresponds to region A. If the data arrives much earlier than the clock, it is blocked. This time spent waiting for the clock effectively makes the data-to-output delay larger. This behavior occurs in region C. It is worth noting that in region B, the data-to-output characteristic is fairly flat. When the data arrives from 10 ps to –30 ps before the clock, the data-to-output delay is almost constant. This flatness has two interpretations. The first is data-centric. It assumes that the delay of the pipeline stage producing the data received by the HLFF exceeds one clock cycle. The flatness of region B allows the extra delay to be passed to the next pipeline stage. In the design community, this is often called slack passing, time borrowing, or cycle stealing. The second interpretation is clock-centric: in the flat region, the clock can move relatively freely without affecting the data-to-output delay. The size of the flat region defines how much clock inaccuracy can be absorbed.

Reducing the sequencing overhead by absorbing clock inaccuracy is equivalent to saving power. In the SPARC V9 example presented earlier, increasing the ability of the clocked storage elements to absorb skew and jitter by 24 ps would still be equivalent to a 10.6-W power reduction.

B. Generalized Optimal Setup Time

When there is no clock inaccuracy, the best arrival time for the data is the one minimizing the data-to-output delay [4, 17]. This arrival time is called the optimal setup time and is widely used in practice. It is called here the traditional optimal setup time.

Determining the best arrival time for the data in the presence of clock inaccuracy is a more general problem. Obviously, it is dangerous to allow data to arrive close to the failure region. There, a small amount of clock inaccuracy could make the clocked storage element unreliable. A generalized optimal setup time is introduced here for a given worst-case clock inaccuracy *y*. It is assumed that the data-to-output characteristic T_{DQ} of the clocked storage element is known. The generalized optimal setup time is the setup time T_{DC} that minimizes the following delay:

$$\max[T_{DQ}(T_{DC} + t)] \tag{11}$$

where $t \in [-y, y]$. Clearly, when the clock inaccuracy is zero, this generalized setup time becomes equal to the traditional optimal setup time, i.e. the one that minimizes $T_{DQ}(T_{DC})$. This generalized setup time can be interpreted as follows. It is the setup time that minimizes the worst-case data-to-output delay

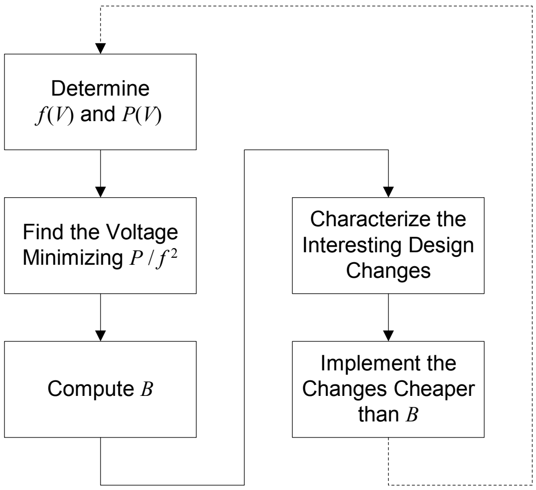
Determine $f(V)$ and $P(V)$
Find the Voltage Minimizing P / f^2
Compute B
Characterize the Interesting Design Changes
Implement the Changes Cheaper than B

Fig. 6: Methodology for energy-efficiency design.

of the clocked storage element under all the possible clock arrival times. The failure region of Fig. 4 is implicitly taken into account by (11). This is because the delay of a clocked storage element for which the setup constraint is not satisfied is infinite.

In Fig. 4, the generalized optimal setup time is equal to the traditional optimal time (i.e. –10 ps) when the clock inaccuracy is under 20 ps. Making the data nominally arrive in the middle of the flat region allows the clock to move without affecting the system performance. When the worst-case clock inaccuracy increases to 30 ps, the generalized optimal setup time becomes 0 ps to avoid the failure region.

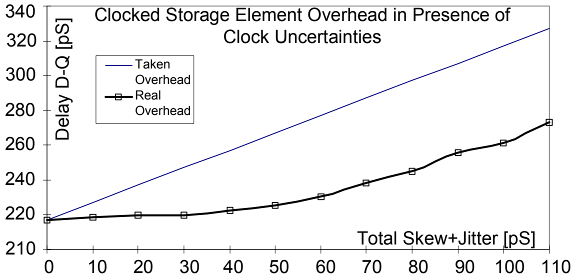

Fig. 5: Clocked storage element overhead in the presence of skew and jitter.

Fig. 5 shows the portion of the clock inaccuracy reflected on the data-to-output delay of the HLFF with and without clock inaccuracy absorption. For this example, about 40 ps of clock inaccuracy can be completely absorbed. About 20 ps more can be partially absorbed. This feature can be directly traded for power.

Methodology for Energy-Efficient Design

The condition for optimal energy efficiency derived in the previous section can be used to define a practical methodology to design energy-efficient systems. The methodology has five steps. It starts with an implementation having a suboptimal energy efficiency.

The first step is to characterize the power-versus-voltage and frequency-versus-voltage relationships of the system to optimize. This implies determining how the system behaves when the supply voltage changes. For a system that actually exists, measuring $f(V)$ and $P(V)$ is relatively easy. For a system being designed, the delay of the worst timing paths at different voltages can be simulated. Then, (1) can be used to determine the power consumed at these voltages. This assumes that C_d, Q_s and I_{leak} are known.

The second step is to find the supply voltage that yields the best energy-delay product, i.e. the one minimizing P / f^2. This voltage is the nominal voltage in (5). Since $P(V)$ and $f(V)$ are known from the first step, the best energy-delay point is obtained by computing $P(V) / f^2(V)$ for each supply voltage. The frequency-versus-voltage relationship also directly gives the frequency-scaling factor k at the optimal supply voltage.

The third step is to compute the marginal benefit of reducing the sequencing overhead using (9). For this computation, the supply voltage minimizing the energy-delay product of the system should be used.

The fourth step is to evaluate all the design changes aimed either at increasing the frequency of the system or reducing its power consumption. The cost of these interesting changes (in mW / ps) must be quantified through simulation.

The last step is to implement the changes having a cost below B. The methodology is summarized in Fig. 6. The procedure can be repeated if the implemented design changes significantly modify the power-versus-voltage and frequency-versus-voltage relationships of the system.

Conclusions

For most synchronous CMOS integrated systems the power dissipated for sequencing is a very large fraction of the total power. Therefore, making sure that the energy allocation is not too far from the optimum is important. The model proposed for the relationship between supply voltage, clock frequency, and power dissipation has been empirically verified for a SPARC V9 microprocessor. It has been argued that the energy is allocated optimally if the following condition holds: any design change aimed at reducing the sequencing overhead or the power dissipation of the system has to have a marginal cost equal to its benefit. The optimal energy allocation condition has been used to define a practical methodology to design energy-efficient systems.

Acknowledgements

The authors would like to thank K. Nowka, P. Hofstee, and G. Carpenter of the IBM Austin Research Laboratory for sharing their thoughts on power efficiency metrics.

References

[1] E. G. Friedman, “Clock Distribution Networks in Synchronous Digital Integrated Circuits”, *Proceedings of the IEEE*, May 2001, pp. 665-692.

[2] T. Xanthopoulos *et al.*, “The Design and Analysis of the Clock Distribution Network for a 1.2-GHz Alpha Microprocessor”, *IEEE International Conference on Solid-State Circuits*, 2001, pp. 402-403.

[3] P. J. Restle *et al.*, “A Clock Distribution Network for Microprocessors”, *IEEE Journal of Solid-State Circuits*, May 2001, pp. 792-799.

[4] V. Stojanovic and V. G. Oklobdzija, “Comparative Analysis of Master-Slave Latches and Flip-Flops for High-Performance and Low-Power Systems”, *IEEE Journal of Solid-State Circuits*, April 1999, pp. 536-548.

[5] R. Heald *et al.*, “A Third-Generation SPARC V9 64-b Microprocessor”, *IEEE Journal of Solid-State Circuits*, November 2000, pp. 1526-1538.

[6] S. Rusu and G. Singer, “The First IA-64 Microprocessor”, *IEEE Journal of Solid-State Circuits*, November 2000, pp. 1539-1544.

[7] C. Nicoletta *et al.*, “A 450-MHz RISC Microprocessor with Enhanced Instruction Set and Copper Interconnect”, *IEEE Journal of Solid-State Circuits*, November 1999, pp. 1478-1491.

[8] R. Senthinathan, S. Fischer, H. Rangchi, and H. Yazdanmehr, “A 650-MHz, IA-32 Microprocessor with Enhanced Data Streaming for Graphics and Video”, *IEEE Journal of Solid-State Circuits*, November 1999, pp. 1454-1465.

[9] D. R. Bearden *et al.*, “A 780-MHz PowerPC Microprocessor with Integrated L2 Cache”, *IEEE International Solid-State Circuits Conference*, 2000, pp. 90-91.

[10] K. Nose and T. Sakurai, “Analysis and Future Trend of Short-Circuit Power”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, September 2000, pp. 1023-1030.

[11] K. A. Bowman *et al.*, “A Physical Alpha-Power Law MOSFET Model”, *IEEE Journal of Solid-State Circuits*, October 1999, pp. 1410-1414.

[12] D. Harris and M. Horowitz, “Skew-Tolerant Domino Circuits”, *IEEE Journal of Solid-State Circuits*, November 1997, pp. 1702-1711.

[13] T. Sakurai and A. R. Newton, “Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas”, *IEEE Journal of Solid-State Circuits*, April 1990, pp. 584-594.

[14] D. H. Allen *et al.*, “Custom Circuit Design as a Driver for Microprocessor Performance”, *IBM Journal of Research and Development*, November 2000, pp. 799-822.

[15] N. Nedovic and V. G. Oklobdzija, “Dynamic Flip-Flop with Improved Power”, *Proceedings of the 26th European Solid-State Circuits Conference*, Stockholm, Sweden, 2000.

[16] H. Partovi *et al.*, “Flow-Through Latch and Edge-Triggered Flip-Flop Hybrid Elements”, *IEEE International Solid-State Circuits Conference*, 1996, pp. 138-139.

[17] V. G. Oklobdzija, “High-Performance System Design: Circuits and Logic”, *IEEE Press*, 1999.