An Adaptive Low-power Transmission Scheme for On-chip Networks

Frédéric Worm
Processor Architecture Laboratory
EPFL, Lausanne, Switzerland
Frederic.Worm@epfl.ch

Paolo Ienne
Processor Architecture Laboratory
EPFL, Lausanne, Switzerland
Paolo.Ienne@epfl.ch

Patrick Thiran
Lab. for Computer Communications and Appl.
EPFL, Lausanne, Switzerland
Patrick.Thiran@epfl.ch

Giovanni De Micheli
Computer Systems Laboratory
Stanford University, Calif., USA
nanni@stanford.edu

ABSTRACT

Systems-on-Chip (SoC) are evolving toward complex heterogeneous multiprocessors made of many predesigned macrocells or subsystems with application-specific interconnections. Intra-chip interconnects are thus becoming one of the central elements of SoC design and pose conflicting goals in terms of low energy per transmitted bit, guaranteed signal integrity, and ease of design. This work introduces and shows first results on a novel interconnect system which uses low-swing signalling, error detection codes, and a retransmission scheme; it minimises the interconnect voltage swing and frequency subject to workload requirements and S/N conditions. Simulation results show that tangible savings in energy can be attained while achieving at the same time more robustness to large variations in actual workload, noise, and technology quality (all quantities easily mispredicted in very complex systems and advanced technologies). It can be argued that traditional worst-case correct-by-design paradigm will be less and less applicable in future multibillion transistor SoC and deep sub-micron technologies; this work represents a first example towards robust adaptive designs.

Categories and Subject Descriptors

B.4.3 [**Input/Output and Data Communications**]: Interconnections (Subsystems)

General Terms

Design

Keywords

Networks-on-Chip, Systems-on-Chip, Low-Power.

1. INTRODUCTION

The successful design of highly-complex *Systems-on-Chips* (SoC) depends on the availability of robust design methodologies that allow short times-to-market with low risk. The pressure on design technologies is rapidly increasing with the future need of integrating billions of transistors on a single chip now in sight.

Design of such SoCs will be made possible by the use of complex components (e.g., full subsystems with processors, controllers, DSPs) as major building blocks. Therefore SoCs can be seen as heterogeneous multi-processing systems, with multiple (possibly asynchronous) timing references, because of the difficulty of global synchronization. Hence, given a library of modular components, the main design challenge of future SoCs will be to connect efficiently such components into an effective network which implements the desired functionality. On-chip micronetworks [2] will become the central focus of the design process and will inherit a number of techniques and methodologies from today's macronetworks, such as layered design and/or packet communications.

Long distance on-chip VLSI interconnect schemes for *Networks-on-Chip* [2] pose several challenges, including, but not limited to, the choice of network topology to implement the communication scheme, the related choice of protocols, and the achievement of multiple interrelated objectives such as providing high performance, low energy consumption and high reliability.

Without loss of generality, in this paper we consider bus based communication on chip and we focus on three specific objectives:

- **Performance requirements.** A bus implementing a communication channel should provide enough bandwidth to support the required communication demand. Such demand may not be precisely known in the early stages of the design. Additionally, it is important to realize that the workload on a bus may change dynamically, and thus that the bus bandwidth needs not be kept at its peak at all times. Therefore, the design versatility may greatly benefit from a dynamic adjustment of the bus bandwidth.

- **Energy consumption.** Studies have shown that wires account for a significant fraction of the total en-

ergy consumption (up to 40–50%) [11]. A large share of this consumption is due to long high-capacity wires crossing the die and connecting different subsystems. With larger die sizes and more subsystems on chip, the proportion of power consumed in the communication can only grow. This calls for techniques to reduce the energy consumed in on-chip communication.

- **Reliability and noise sensitivity.** Many technological factors challenge the traditional robustness of digital CMOS design. Low-energy communication can be achieved using small voltage swings but, together with supply voltage reduction, this factor contributes to the decrease of noise immunity of the communication implementation. Other factors affecting noise sensitivity and communication reliability include higher manufacturing tolerances at very deep sub-micron feature sizes, increasing importance of capacitive and inductive crosstalk due to the evolution of technological geometries, etc. Design methodologies for interconnects must take into account the growing noise sensitivity.

The economic importance of an extremely high reliability of SoC components and interconnect has lead to the classic very conservative VLSI design style. In practice, the delay of a circuit is characterized by adding up the negative contributions due to all possible "noise" factors: deviation of many parameters from their nominal values yielded by the variability of the manufacturing process, effects of temperature over the complete operating range, extreme temperature, crosstalk noise, acceptable power supply noise, electromagnetic (EM) noise, etc. The result is a worst-case design which, if every source of disruption is well characterized, works robustly, but which, in typical conditions, is heavily suboptimal in terms of achieved speed and consumed energy.

We believe that conservative worst-case static design cannot leverage deep sub-micron technologies at best and achieve the mandated performance, energy, and reliability goals. The objectives of this paper are two-fold:

- Minimise the energy used for reliable communication while satisfying a Quality-of-Service constraint such as the average delay of the transmission. We employ a dynamically-varying voltage swing on busses, that adjust to the performance and error rate requirements.

- The second (and broader) goal of our methodology is to show an approach where design parameters (voltage swing and frequency) are not set on the basis of predicted *a priori* worst-case characteristics of the used technology. Conversely, these parameters are set *a posteriori*, based on the fulfilment of the primary communication constraints (performance and reliability) and on the effects, as measured *in situ*, of environmental factors (temperature, external EM noise) and of the technological parameters spread.

We believe that this design paradigm, based on dynamic run-time parameter adjustment, can make both SoC design easier—by avoiding some worst-case analyses—and more effective—by obtaining design objectives that directly relate to the characteristics of silicon and of the environment.

1.1 Related Work

One common technique used to minimise power consumption on buses is the choice of appropriate encoding schemes which reduce the switching activity without affecting the signal information content [14, 3]. This approach has been recently extended to account for interwire capacitances [13, 9] and reliability issues [4]. Bus encoding techniques have shown effectiveness in reducing power consumption although the best results are generally achieved in specific environments such as address buses. In fact, encoding is complementary to the scheme described in this paper.

The classic silver bullet of power consumption reduction is a lower supply voltage and, specifically for interconnects and buses, the use of low-swing signalling techniques [19, 16]. Although very effective on the power side, these techniques alone compromise significantly the robustness of the design and, instead of helping designers to address new deep submicron effects, further complexify the design process. The proposed scheme makes judicious use of low-swing communication while ensuring that the overall reliability of the system is not decreased but, on the contrary, raised.

Dynamic Voltage Scaling (DVS) is a now well established and effective technique to reduce the consumption of systems under given performance constraints [12, 1, 5]. It is usually applied to adapt dynamically the speed of processors in PDAs to current computational requirements; it is now supported by several commercial processor (e.g., Intel XScale and Mobile Pentium, Transmeta Crusoe). The technique is based on the characterisation of the devices at a number of different working points (pairs of supply voltage and maximal operating frequency): they correspond to a set of safe operating conditions computed or measured taking into account all worst-case parameters. A transmission scheme applying DVS to chip-to-chip interconnection networks has been recently introduced [10]. Such a system is a direct extension of processor voltage-scaling and assumes the knowledge of a fixed relation between the voltage and frequency for safe operation. Our communication scheme similarly extends the idea of DVS to on-chip communications in the form of variable voltage-swing signalling but does not rely on *a priori* knowledge of robust working points.

The idea of operating CMOS devices at voltages below the worst-case characterisation point—and thus in subcritical regions where errors might occur—has seldom been investigated. In a recent paper [8], the possibility of exploiting devices in subcritical regions for *Digital Signal Processing* (DSP) was discussed; in that case, errors arising from the subcritical voltages are compensated by the DSP algorithms. The present work is similar in the intents, but applies to a different domain (communication instead of computation) and can thus exploit classic techniques to achieve fully correct behaviour despite occasional errors.

1.2 Structure of the Paper

Section 2 introduces the main idea of the paper and describes the architecture of the on-chip transmission system which we propose to achieve the goals mentioned above. Section 3 models formally the system and formulates the problem to solve to control the system and achieve minimal energy consumption. Several policies are described in Section 4. Section 5 shows some simulation results on artificial and real workloads and contrasts the exact policies with a simpler implementable control algorithm. Section 6 concludes with some remarks on the potentials of the presented scheme.
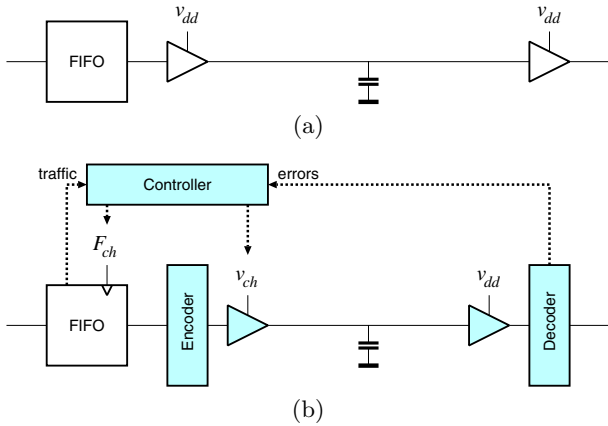
Figure 1: A simplified view of a typical point-to-point unidirectional on-chip interconnect. (a) The classic scheme, with a FIFO to decouple the two subsystems. (b) The proposed scheme with the different elements needed to achieve the required goals.

2. DVS FOR ON-CHIP INTERCONNECTS

For simplicity, we focus on a typical unidirectional point-to-point interconnect between subsystems. Figure 1(a) shows a qualitative view of the classic interconnect: at the producer end, a FIFO or a similar buffer is used to decouple the two subsystems which may operate at different frequencies, and a large driver (typically a chain of appropriately sized inverters) charges or discharges the large capacitance represented by the interconnecting wires. A receiver (typically a CMOS gate) compares the level of the line to a threshold and delivers the resulting information to the consumer.

We add a few elements to the classic scheme, as indicated in Figure 1(b). To reduce the energy consumed per bit, we apply a form of DVS to the interconnect by controlling dynamically the driver swing and the corresponding receiver threshold. Electrical schemes to reduce the voltage swing of the interconnect are known and well studied [19]. Of course, the variable voltage swing impacts the speed at which the interconnect driver is able of charge or discharge the load capacitance, and thus the maximal reliable operating frequency is reduced with lower swings: hence, we need to adapt the communication speed too, as in traditional DVS techniques.

The operation with lower swings makes our communication more sensitive to several noise sources; to cancel this effect, we introduce an error detection encoding at word level on the source side and we implement a typical *Automatic Repeat reQuest* (ARQ) strategy, such as Stop-and-Go [17].

Finally, our scheme requires a controller which decides of the operating frequency and voltage swing: such a controller must be able not only to choose voltage-frequency pairs from a set of safe operating-points and as a function of the requested bandwidth; but also to explore the design space for safe operating points. Therefore, it needs as input some information on both bandwidth requirements and channel reliability.

In summary, our system (1) uses a variable frequency and swing to trade off speed for energy, (2) implements error detection and ARQ to guarantee reliable communication, and

(3) exploits a variable relation between operating frequency and voltage swing to find the best safe operating point in the current environmental conditions, based on monitoring the error rate. We will discuss more formally the different elements in the next section and we will describe the control policy needed to achieve the desired goals.

In the present paper, we will focus on the feasibility and potential advantages of such adaptive transmission scheme and on the challenges of abandoning a conservative worst-case design style. We will not detail some circuit design aspects such as the implementation of the variable supply transmitters and receivers (which we suppose achievable as a derivation from known techniques [19]) nor the availability of on-chip efficient controllable power supply sources (a key component for any DVS technique and the object of many research efforts [6, 15]).

3. SYSTEM MODELLING

To implement the ideas of Section 2, we need to create a model of the communication system and of the noise associated to it; we do this in Section 3.1. In Section 3.2 we use such models to derive a relation between our controlled variables (communication frequency and voltage swing) and the probability of errors in the transmission. Since this probability is not readily available, it will be estimated from the number of retransmissions of a packet until it is received correctly. This estimate will be needed to constrain our optimisation policy and achieve reliable communication.

Similarly, our scheduling policy will have to guarantee some *Quality-of-Service* (QoS), defined here in terms of average delay. This value, which includes the propagation time and the queueing delay in the FIFO buffer, which needs therefore to be estimated: this is done in Section 3.4.

Finally, in Section 3.5 we discuss the different sources of energy consumption in our system and develop an approximate expression for the effective energy consumed to send a useful bit of data across the interconnect. This is the objective function that our system should minimise.

3.1 Channel and Noise

We model the physical communication channel (from the FIFO of Figure 1 to the destination subsystem) as a system made of four elements: a driver or encoder, additive noise, a low-pass filter, and a receiver. The driver encodes the binary signal on two voltage levels whose difference is the desired *voltage swing* v_{ch}. The low-pass filter accounts for the delay through the physical interconnect, which is customarily approximated with the time to charge the load capacitance—i.e., the interconnect—to half of the swing voltage. The latter is used as the threshold by the receiver, which compares thus the received input to this threshold to decide the value of received binary symbol. The *channel cut-off frequency* F_{cut} can be expressed as [18]

$$F_{cut} = \frac{k_m}{C_L} \cdot \frac{(v_{ch} - v_{th})^2}{v_{ch}}, \tag{1}$$

where:

- k_m is the transistor transconductance, which depends on the driver transistor dimensions and on some technological parameters,
- C_L is the interconnect capacitance, and
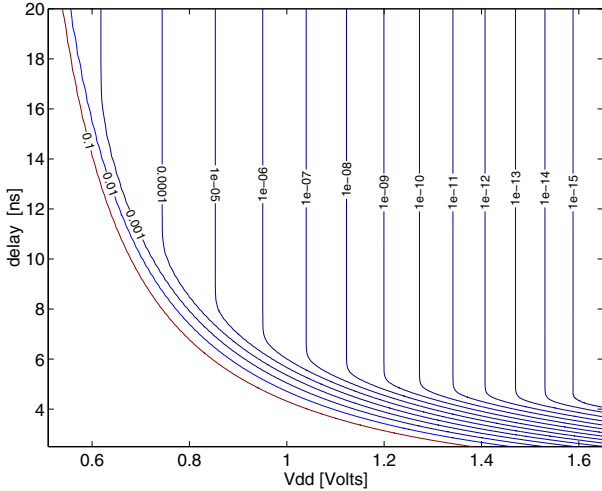- v_{th} is the threshold voltage of the devices.

Figure 2: Contour plot of the bit error rate in the v_{ch} and $1/F_{ch}$ plan.

For the sake of simplicity, we reduce all sources of noise to two: (i) a voltage noise v_{noise}, which is added to the signal decoded by the receiver as mentioned above, and (ii) a deviation of the cut-off frequency F_{cut} from its nominal value in Eq. (1). These perturbations are the most important ones in digital design. For instance, the additive noise can represent EM and power supply noise. The variation of F_{cut} represents many stationary and nonstationary effects. The stationary noise essentially represents the process variation during manufacturing, which affects the geometries and the characteristics of the devices, and thus k_m. A slowly varying source of noise represents the effects of temperature. Crosstalk is another nonstationary source of noise that changes the effective capacitance of the channel C_L, depending on the transitions taking place on neighbouring wires.

We approximate the variability of F_{cut} and the additive noise v_{noise} by two independent Gaussians. More precisely, we assume that the ratio k_m/C_L, which we denote by α, is a Gaussian random variable, with mean μ_α and standard deviation σ_α:

$$\alpha \sim N(\mu_\alpha, \sigma_\alpha),$$

and that the additive noise v_{noise} is a white Gaussian noise, with standard deviation $\sigma_{v_{noise}}$:

$$v_{noise} \sim N(0, \sigma_{v_{noise}}).$$

3.2 Bit Error Rate

With the model of the channel and of the noise sources developed in the previous section, it is now possible to express the *bit error rate* ϵ_b as a function of v_{ch} and F_{ch}. Whether a bit will be received correctly, for given values of v_{ch} and F_{ch}, depends on the realization of the two random variables α (which in turn impacts F_{cut}) and v_{noise}.

For the sake of simplicity, we make the following approximation: an error occurs if either the operating frequency F_{ch} exceeds the channel cut-off frequency F_{cut}, or the additive noise v_{noise} exceeds half the voltage swing v_{ch}. This implies that the contribution of intersymbol interference to the bit error rate is neglected for $F_{ch} \leq F_{cut}$, and is approximated

to one otherwise. After some computations, one obtains

$$\epsilon_b = 1 - (P(F_{ch} < F_{cut}) \cdot P(v_{noise} < v_{ch}/2)) =$$
$$= 1 - \left(Q\left(\frac{F_{ch} - \mu_{F_{cut}}}{\sigma_{F_{cut}}} \right) \right) \cdot \left[1 - Q\left(\frac{v_{ch}}{2\sigma_{v_{noise}}} \right) \right], \quad (2)$$

where $Q(\cdot)$ is the complementary cumulative Gaussian distribution function:

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy.$$

Eq. (2) is a generalisation of the relation introduced in [7]. The new relation takes into account some other, important sources of errors. Figure 2 shows a typical plot of ϵ_b in the (v_{ch}, F_{ch}) plane. One recognises the critical zone where the circuit passes from a faulty to a functionally correct state: for delay values sufficiently above the critical value the probability of error is almost zero, whereas the same probability is 1 for in regions where the circuit is overconstrained.

In a classic system, which does not use ARQ for error detection, a word is wrong if any of its N bits is wrong. Assuming bit errors to be independent, identically distributed (i.i.d.), one has therefore that the word error rate in a classic system is $1 - (1 - \epsilon_b)^N$. In our adaptive system, the introduction of the error detection code and of the ARQ policy reduces the number of actual errors on the words exchanged between the two subsystems. The error detection codes adds n bits per word; we indicate as ϵ_w^{raw} the *raw word error rate before detection*, whose value is $\epsilon_w^{raw} = 1 - (1 - \epsilon_b)^{N+n}$; it is of course larger than in the classic system. However, the *residual word error rate* ϵ_w^{res}, which is the probability that an erroneous word is not detected *after* having performed error detection, is much smaller. Its expression is a function $f_{code}(\cdot)$ of the encoding scheme and the raw bit error rate ϵ_b:

$$\epsilon_w^{res} = f_{code}(\epsilon_w^{raw}) = f_{code}\left(1 - (1 - \epsilon_b)^{N+n} \right). \quad (3)$$

For instance, assume that a system uses a Hamming code ($n = 8$ redundant bits added to each $N = 32$-bit word) to detect errors. To achieve a residual word error rate 10^{-10}, a classic system must be designed to have a bit error rate of approximately $3 \cdot 10^{-12}$. On the other hand, the adaptive system with ARQ can tolerate bit error rates up to approximately 10^{-5} for the same residual word error rate.

3.3 Estimated Word Error Rate

The actual word error rate ϵ_w^{raw} is not directly available and is needed by some versions of our controller. We assume that our system can monitor the total number n_{rtx} of transmissions of a packet (including both successful and unsuccessful attempts). The expected value of this variable is $W/(1 - \epsilon_w^{raw})$, with W the size of packets in words, and to smooth this estimated value, we filter it with an *Exponentially Weighted Moving Average* (EWMA) to obtain

$$\epsilon_w^{raw}(v_{ch}, F_{ch}) \leftarrow \epsilon_w^{raw}(v_{ch}, F_{ch}) \cdot (1 - \beta) + \beta \left(1 - \frac{W}{n_{rtx}} \right), \quad (4)$$

for some filtering parameter $0 < \beta < 1$.

3.4 Estimated Delay

Our controller needs to estimate the expected packet delay through the system. A comparison with the given aver-

age delay constraint will then result in decisions such as to increase or reduce the channel operating frequency.

We denote by Δ_{est} the expected delay that the last packet in the FIFO queue will experience, which includes transmission and queueing. We estimate it as follows:

$$\Delta_{\text{est}} (v_{\text{ch}}, F_{\text{ch}}, l) = \frac{l \cdot \Xi_{\text{tot}} (\epsilon_{\text{w}}^{\text{raw}} (v_{\text{ch}}, F_{\text{ch}}))}{F_{\text{ch}}}, \qquad (5)$$

where l represents the queue size, that is, the number of packets of W words currently present in the FIFO buffer, and Ξ_{tot} is the expected number of cycles needed to send a packet of size W, including cycles spent on attempted transmissions that failed. Note that for simplicity we choose to base our estimation on the last packet in the queue, although this is not necessarily the most likely to miss its transfer delay constraint.

The number of cycles needed to send a packet is the sum of the cycles spent for a successful packet transfer with, possibly, the cycles accounting for previously failed transmission attempts. Assuming the attempts to be i.i.d., this number has thus a geometric distribution, with parameter $(1 - \epsilon_{\text{w}}^{\text{raw}})^W$, and an average equal to

$$\Xi_{\text{tot}} = \Xi_W + \left(\frac{1}{(1 - \epsilon_{\text{w}}^{\text{raw}})^W} - 1 \right) (\Xi_W + \Xi_{\text{rtx}}), \qquad (6)$$

where:

- Ξ_W is the number of cycles to transfer a packet of W words and
- Ξ_{rtx} is the number of dummy cycles imposed by the transmission protocol to retransmit W words—besides the cycles for the retransmission themselves (could be zero).

Note that Eq. (6) assumes that the detection of errors is performed per-word, that an ARQ strategy like Stop-and-Go is used [17], and that the bus has a one-word width.

Inserting Eq. (6) into Eq. (5), one obtains the required delay estimation:

$$\Delta_{\text{est}} = \frac{l}{F_{\text{ch}}} \left(\Xi_W + \left(\frac{1}{(1 - \epsilon_{\text{w}}^{\text{raw}})^W} - 1 \right) (\Xi_W + \Xi_{\text{rtx}}) \right). \qquad (7)$$

This value, or its moving average, can be compared with the design constraint; as a result, the control policy can decide the best action to make if the constraint appears likely to be violated or not.

3.5 Energy Consumption

Finally, we need to express the objective function which we want to minimise. The energy consumed in the transmission scheme is made of several contributions:

- the energy E_{cl} to transmit the N bits of each message word,
- the energy $E_{\text{redundant}}$ to transmit the additional n redundancy bits of the error detection code,
- the energy E_{codec} to generate the above mentioned redundant bits, and the energy to decode the received words,
- the energy E_{rtx} to resend words when previous trial(s) arrived corrupted to the receiver,
- the energy E_{ctrl} consumed by the controller to calculate the next state for the transmission,

- the energy E_{conv} lost in the voltage converter used to generate v_{ch}.

All but the first component are additional contributions of our versatile scheme, which have no counterpart in the classic system. In the sequel, we will assume E_{codec}, E_{ctrl}, and E_{conv} negligible.

The *energy per transmitted word* E_{w} is proportional, through constants such as C_{L} and the line flipping probability, to v_{ch}^2. The energy consumed per word by the classic, non adaptive scheme only comes from bit flipping activity:

$$E_{\text{cl}} = K \cdot N \cdot v_{\text{ch}}^2. \qquad (8)$$

The energy consumed per useful word by the adaptive scheme is similar, but is worsened by both the need of sending redundant information (n additional bits) and the need of resending some words through ARQ because of transmission errors:

$$\begin{aligned} E_{\text{adaptive}} &= E_{\text{cl}} + E_{\text{redundant}} + E_{\text{rtx}} \\ &= K \cdot \frac{N + n}{1 - \epsilon_{\text{w}}^{\text{raw}}} \cdot v_{\text{ch}}^2. \end{aligned} \qquad (9)$$

Eqs. (8) and (9) give a comparison basis for the energy consumed per transmitted word E_{w} in the classic and adaptive systems.

4. CONTROL POLICIES

Having described in the previous sections our models of energy consumption, channel error rates, and transmission delay, we can now express the control scheme first mentioned in Section 2 as a constrained optimization problem. Each time the control policy is applied, the controller has to find the pair $(v_{\text{ch}}, F_{\text{ch}})$ which:

1. minimises E_{adaptive} as expressed in Eq. (9),
2. meets the performance constraint ($\Delta_{\text{est}} \leq \overline{\Delta}$),
3. meets the reliability constraint ($\epsilon_{\text{w}}^{\text{res}} \leq \overline{\epsilon}_{\text{w}}^{\text{res}}$).

The values of $\overline{\Delta}$ and $\overline{\epsilon}_{\text{w}}^{\text{res}}$ are an input to the control policy and are specified by the user. Typically, $\overline{\epsilon}_{\text{w}}^{\text{res}}$ would be a constant whereas $\overline{\Delta}$ might change during the system operation to reflect changes in the desired performance. Without loss of generality we will assume both constant in the sequel.

We will present experiments with three different control policies which correspond to three different ways to approach the above optimisation problem. We call them *Exact Nonadaptive*, *Exact Adaptive*, and *Feedback Control* and we present them in the next three sections.

4.1 Exact Nonadaptive Policy

The first policy we consider is called Exact Nonadaptive. It consists in solving exactly the above energy minimisation problem under both the delay and the residual error constraints. In practice, we restrict the operation to a discrete set of points is the $(v_{\text{ch}}, F_{\text{ch}})$ space to make the decision easier. To guarantee that the reliability constraint is satisfied, in this policy (a) we use the *a priori* knowledge of the residual word error rate expressed through Eqs. (3) and (2), and (b) we forbid operating points in the $(v_{\text{ch}}, F_{\text{ch}})$ space whose residual word error rate exceeds $\overline{\epsilon}_{\text{w}}^{\text{res}}$.

This policy is nonadaptive, as are essentially all DVS techniques [12, 1, 5], because working points are selected *a priori*, using traditional worst-case design considerations.

4.2 Exact Adaptive Policy

Our second policy, which we call Exact Adaptive Policy, removes this *a priori* knowledge by estimating *in situ* the effects of technological and environmental factors on the circuit possibilities. The estimated word error rate is initialised with the same *a priori* knowledge used for the Exact Nonadaptive policy, but then kept updated through Eq. (4). The latter estimate is needed for predicting the packet transfer delay—using Eq. (7).

With this and the following policy, we allow our design to explore any area of the (v_{ch}, F_{ch}) space: in other word we are no longer constrained by worst case assumptions. Unfortunately, with these policies we have no mean to guarantee the upper bound on ϵ_w^{res}. Hence, the Exact Adaptive policy minimises the energy and takes into account the delay constraint but completely disregards the reliability constraint.

One could argue that the constraint on the delay and the need to minimise the energy spent will implicitly force the system to avoid large error rates: indeed, we will show in Section 5 that the condition on ϵ_w^{res} is not violated in practice in our simulations.

4.3 Feedback Control Policy

The two policies presented above solve exactly the minimisation problem by selecting the (v_{ch}, F_{ch}) operating point which minimises the energy under the appropriate constraints. Such an exhaustive search is too complex to implement in hardware, both in terms of hardware resources and energy consumption. We show here an example of a simpler policy that could be used to design a real controller.

The Feedback Control policy represents a simple controller which makes changes to the operating point (v_{ch}, F_{ch}) depending on (a) the relation between Δ_{est}—estimated through Eq. (7)—and the user constraint $\overline{\Delta}$, and (b) the transmission error record over the (v_{ch}, F_{ch}) space. Figure 3 shows the possible changes in the operating point (v_{ch}, F_{ch}).

This policy estimates ϵ_w^{raw} the same way the Exact Adaptive policy does. However, it uses the estimated word error rate not only to predict a transfer delay value, but also to check the safety of a new working point selected by the rules Figure 3 illustrates. Therefore, candidates working points with high predicted word error rate are avoided. The basic features of the three policies are summarised as follows:

policy	minimis.	restricted (v_{ch}, F_{ch})	error est.
Exact Nonadaptive	exh. search	Yes	(3)–(2)
Exact Adaptive	exh. search	No	(4)
Feedback Control	heur.	No	(4)

5. SIMULATION RESULTS

We simulated 32-bit interconnect systems implementing our three policies, and we compared them with a classic fixed-swing system. We model a typical 0.13μm CMOS technology and noise sources as follows:

- Nominal supply voltage: $v_{dd} = 1.5$V.
- Threshold voltage: $v_{th} = 0.3$V.
- Additive noise: $\sigma_{v_{noise}} = 0.1$V.
- Cut frequency: $\mu_{F_{cut}} = 500$MHz and $\sigma_{F_{cut}} = 36$MHz.

Systems implementing retransmission use a Hamming code that adds $n = 8$ redundancy bits to each word. Systems implementing the Exact Nonadaptive policy guarantee
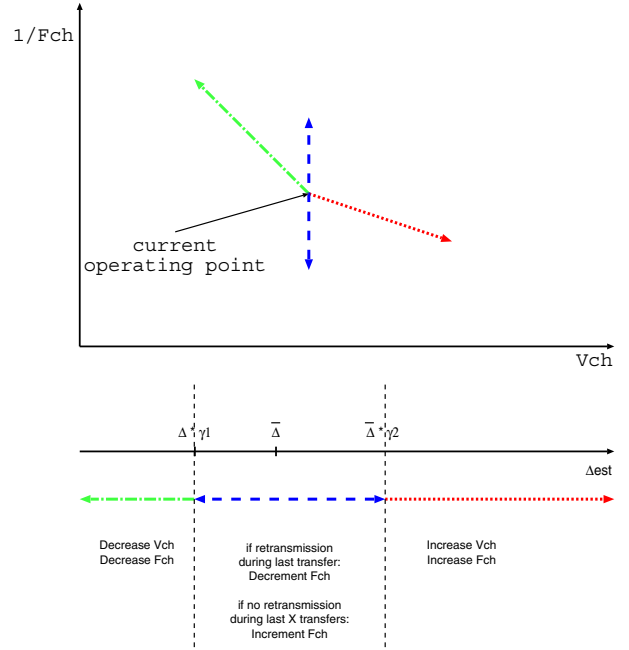

Figure 3: The Feedback Control policy makes simple changes to the operating point (v_{ch}, F_{ch}) depending on the relation between Δ_{est} vs. $\overline{\Delta}$ and on the transmission error record.

a residual word error rate $\overline{\epsilon}_w^{res} = 10^{-10}$, which is identical to that of the classic system. In fact, the classic system is designed using the normal worst-case methodology: the operating point (v_{ch}, F_{ch}) must satisfy the reliability constraint, the design space thus reduces to the normal line of worst-case Pareto points, and the operating frequency is approximately half of the typical value (e.g., 250MHz at 1.5V). As mentioned in Section 4.1, the Exact Nonadaptive policy limits the set of possible (v_{ch}, F_{ch}) pairs to areas where ϵ_w^{res} is below the constraint. The main features of each system are:

	classic	our schemes
v_{ch}[V]	1.5	$0.6 \leq v_{ch} \leq 1.6$
F_{ch}[MHz]	250	$50 \leq F_{ch} \leq 400$
ϵ_w^{res}	$\leq 1 \cdot 10^{-10}$	$\leq 1 \cdot 10^{-10}$
ϵ_b	$\leq 3 \cdot 10^{-12}$	$\leq 1.1 \cdot 10^{-5}$
n	0	8

In the simulations, we use both artificial and real workloads. Artificial workloads consists of 55,000 words with words arrival times following a Poisson process. As a real workload, we simulate the transfer over a bus of a 400 frames sample from an MPEG trace—a typical periodic workload with maximum bandwidth requirements different from the average value. Each MPEG frame consisting of several Kbytes, we have split each frame into packets of 64 Bytes. Since MPEG frames are generated periodically, we assume a delay constraint $\overline{\Delta}$ for all frames, which we take equal to the time taken by the classic system to transfer the largest frame. The control is applied once for each one-KByte transfer. For all systems we measure:
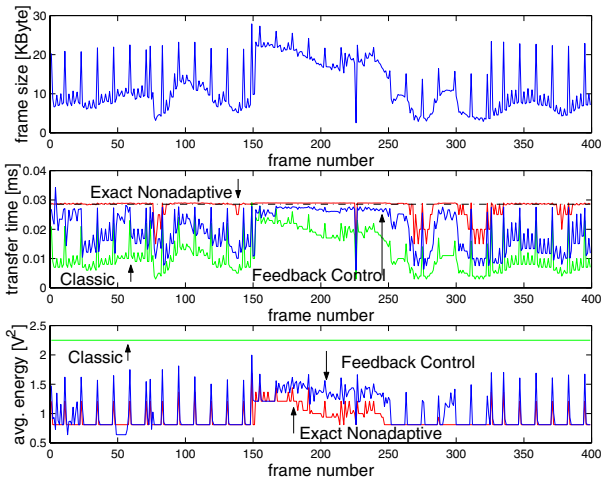
- average energy per word (E_w),

Figure 4: Transmission of a variable workload. Top: workload variation in time. Middle: incurred frame delay. Bottom: energy consumption metric.

	classic	Exact Nonadaptive	Feedback Control
$E_{\mathrm{w}}[V^2]$	2.25	0.90	1.02
$\overline{\Delta}[\mu s]$	0.29	0.29	0.29
$\Delta_{\max}[\mu s]$	0.29	0.29	0.34
$l_{\max}[\text{Kbytes}]$	29	29	33
$\epsilon_{\mathrm{w}}^{\mathrm{res}}$	$9.97 \cdot 10^{-11}$	$1.21 \cdot 10^{-11}$	$8.76 \cdot 10^{-12}$

Table 1: Comparison metrics for MPEG transfer.

- transfer delay, maximum/average (Δ_{avg}, $\Delta_{\max}$),
- buffer needs, maximum/average (l_{avg}, $l_{\max}$), and
- residual word error rate ($\epsilon_{\mathrm{w}}^{\mathrm{res}}$).

The residual word error rate is computed off-line, using Eq. (3) and averaging the bit error rate over the operating points chosen during the simulation. In Section 5.1, we quantify energy savings resulting from dynamic bandwidth adaptation to the workload requirement. Section 5.2 shows a situation where power consumption can be reduced in good operating conditions or on good wafers. Finally, Section 5.3 illustrates how our transmission scheme accommodates worse than expected noise conditions, without further design complications.

5.1 Dynamic Bandwidth Adaptation

Figure 4 shows how systems using the Exact Nonadaptive and the Feedback Control policies handle the MPEG workload. The former system corresponds roughly to the pure application of DVS techniques to on-chip interconnects. Comparison metrics are shown in Table 1. The results show how both policies are effective in saving a tangible fraction of the energy spent by the classic system; the simple Feedback Control policy reduces the consumption of 55%.

Since in this scenario we do not make any particular assumption about the process manufacturing quality, we have characterised the noise by $\mu_{F_{\mathrm{cut}}} = 500$MHz, $\sigma_{F_{\mathrm{cut}}} = 36$MHz and $\sigma_{v_{\mathrm{noise}}} = 0.15$V. Note that the Feedback Control policy, although unable to guarantee a given $\overline{\epsilon}_{\mathrm{w}}^{\mathrm{res}}$(see Section 4.2), achieves a residual word error rate comparable to the one of the Exact Nonadaptive policy. Nevertheless, the former system explores high error areas (high $\epsilon_{\mathrm{b,max}}$)
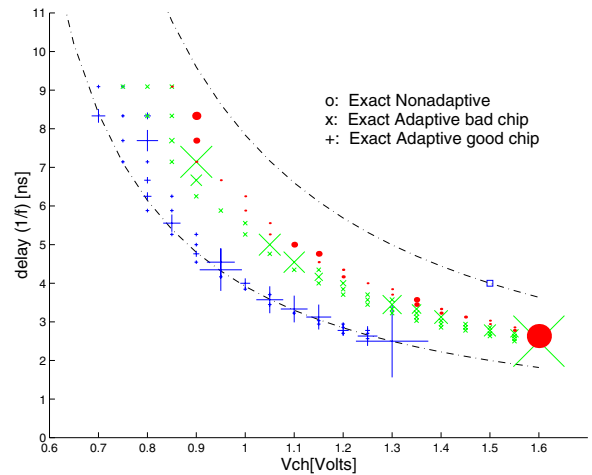

Figure 5: Operating points used by the Exact Nonadaptive and Exact Adaptive policies on good and poor wafers. $\square \rightarrow$ Classic; $\bullet \rightarrow$ Nonadaptive; $+ \rightarrow$ Adaptive good wafer; $\times \rightarrow$ Adaptive poor wafer.

but quickly learns to avoid them due to energy saving and delay constraints.

This section has illustrated the energy savings possible with DVS techniques on interconnects; the next two sections show additional benefits of our technique.

5.2 Exploiting Technology Variations

As mentioned, the Exact Nonadaptive policy, as all classic DVS policies, is unable to exploit technology variations to minimise energy. We contrast here the Exact Nonadaptive policy with both the Exact Adaptive and the Feedback Control policies when applied in different conditions. We use a Poisson workload leading to an utilisation of 0.75—i.e., the ratio between arrival and service rates, for the classic system, is 0.75. The delay constraint is $\overline{\Delta} = 20$ns—i.e., the average delay value of the classic system.

The Exact Nonadaptive policy assumes a residual word error rate based on Eq. (2) with $\mu_{F_{\mathrm{cut}}} = 500$MHz and $\sigma_{F_{\mathrm{cut}}} = 36$MHz (notice that the other policies do not rely on Eq. (2) at all). We account for good and poor wafers by simulating the real error rates using Eq. (2) with slightly shifted $\mu_{F_{\mathrm{cut}}}$ and a reduced standard deviation (to account for a lower indeterminacy once the process quality is fixed). A good wafer has a higher cut-off frequency: $\mu_{F_{\mathrm{cut}}} = 570$MHz and $\sigma_{F_{\mathrm{cut}}} = 15$MHz. Conversely, we simulate a poor manufacturing with $\mu_{F_{\mathrm{cut}}} = 430$MHz and $\sigma_{F_{\mathrm{cut}}} = 15$MHz. Approximately, 2% of the wafers have respectively better and worse cut-off frequency than these values in nominal conditions.

Figure 5 and Table 2 compare the selected operating points in the $(v_{\mathrm{ch}}, 1/F_{\mathrm{ch}})$ space by the Exact Nonadaptive and Exact Adaptive policies for both a good and poor wafer. The classic system operates at 1.5V and 250MHz, whence an energy metric of 2.25. Of course, the Exact Nonadaptive policy is insensitive to technology variations and therefore only one set of points is shown. One can observe that the operating points of the Exact Nonadaptive policy are distributed over a shifted version of the worst-case classic line: indeed, this is also a worst-case line but the Exact

	Exact Nonadaptive		Exact Adaptive	
	process:		process:	
	good	bad	good	bad
$E_{\mathrm{w}}[V^2]$	1.43	1.43	0.98	1.33
$\overline{\Delta}[\mathrm{ns}]$	20	20	20	20
$\Delta_{\mathrm{avg}}[\mathrm{ns}]$	16.81	16.84	16.2	15.92
$l_{\mathrm{avg}}[\mathrm{words}]$	1.74	1.74	1.68	1.67
$l_{\mathrm{max}}[\mathrm{words}]$	11	11	12	10
$\epsilon_{\mathrm{w}}^{\mathrm{res}}$	$1.95 \cdot 10^{-15}$	$2.96 \cdot 10^{-15}$	$2.10 \cdot 10^{-11}$	$8.88 \cdot 10^{-11}$

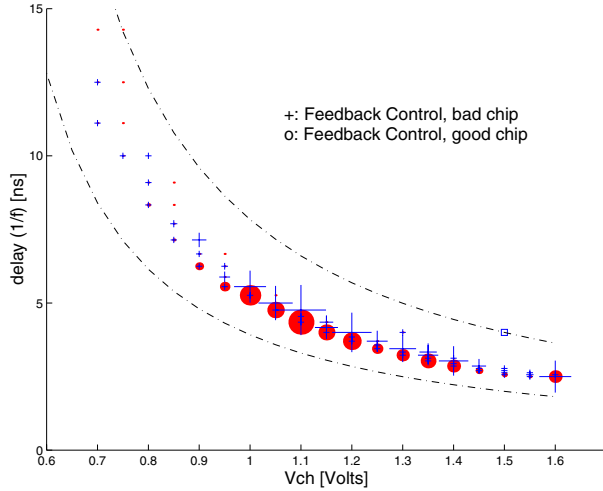**Table 2: Comparison metrics for Exact Nonadaptive and Exact Adaptive policies on good and poor wafers.**

	Feedback Control	
	process:	
	good	bad
$E_{\mathrm{w}}[V^2]$	1.26	1.34
$\overline{\Delta}[\mathrm{ns}]$	20	20
$\Delta_{\mathrm{avg}}[\mathrm{ns}]$	19.3	19.3
$l_{\mathrm{avg}}[\mathrm{words}]$	2.01	2.01
$l_{\mathrm{max}}[\mathrm{words}]$	14	13
$\epsilon_{\mathrm{w}}^{\mathrm{res}}$	$2.51 \cdot 10^{-13}$	$3.20 \cdot 10^{-11}$

Table 3: Comparison metrics for the Feedback Control policy on good and poor wafers.

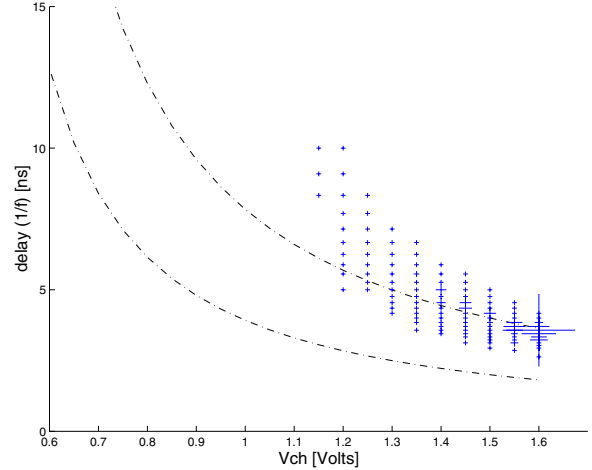

Figure 6: Operating points used by the Feedback Control policy, $+ \to$ poor wafer; $\bullet \to$ good wafer.

Figure 7: Operating points used by the Feedback Control policy with higher noise.

Nonadaptive policy operates the interconnect at higher frequencies because the ARQ policies makes higher error rates tolerable. As with classic DVS schemes, this generates an energy saving: approximately 35% in this case—less than in Section 5.1 due to the larger variance of the MPEG workload. More interesting is the case of the Exact Adaptive policy which clearly choses operating points in a different sets depending on the wafer quality or operating conditions. The better the wafer quality, the more the operating points become more aggressive. The result is an higher energy saving: for a good wafer, an additional 31% over the Exact Nonadaptive policy or a total of 56% over a classic interconnect. It is important to notice that, although the Exact Adaptive policy ignores completely the reliability constraint (see Section 4.1), the resulting error rates are higher than in the case of the Exact Nonadaptive policy but still below or almost below the limit for reliable operation (see Section 5).

We mentioned in Section 4.3 that the Exact Adaptive policy is too complex to implement in hardware. Figure 6 and Table 3 show the same simulations for the realistic Feedback Control policy. Compared to Figure 5, one notices that this policy is slightly less effective in exploiting better operating conditions but achieves the same qualitative result. Again, error rates are below the limit for reliable operation.

5.3 Robustness towards Design Uncertainties

The third example illustrates the tolerance of our transmission scheme to design uncertainties. We assume that the design hypotheses turned out to be too optimistic and that the noise level is actually higher than the situation described by $\sigma_{v_{\mathrm{noise}}} = 0.1$V and $\sigma_{F_{\mathrm{cut}}} = 36$MHz and we simulate the higher real error rates with $\sigma_{v_{\mathrm{noise}}} = 0.15$V and $\sigma_{F_{\mathrm{cut}}} = 55$MHz.

First of all, we should note that the classic system is not expected to work any more under these conditions—that is, its residual word error rate largely exceeds the constraint. This situation corresponds to a very realistic design situation: if in the normal design flow any source of error is overlooked or underestimated, such as crosstalk or other deep submicron second-order effects, the manufactured chips may not work or have a very limited yield.

The simulation results under the same Poisson workload of the previous section are shown in Figure 7 and in Table 4. Indeed, the residual word error rate of the classic system is well above $1 \cdot 10^{-10}$ (see Section 5), the required level for correct functionality: the classic design is wrong and needs a redesign.

	classic	feedback control
$E_{\mathrm{w}}[V^2]$	2.25	2.26
$\Delta_{\mathrm{avg}}[\mathrm{ns}]$	N.A	17.1
$l_{\mathrm{avg}}[\mathrm{words}]$	N.A	1.82
$l_{\mathrm{max}}[\mathrm{words}]$	N.A	13
$\epsilon_{\mathrm{w}}^{\mathrm{res}}$	$9.69 \cdot 10^{-5}$	$6.50 \cdot 10^{-10}$

Table 4: Comparison metrics for the classic system and Feedback Control policy with higher noise.

On the other hand, the system implementing the Feedback Control policy still works fine: the bit error rate is only marginally above the constraint and the delay constraint is met. To meet these constraints, the sysytem shifts its working points to more energy expensive but safer regions. In fact, the energy consumption of the classic system and of the Feedback Control policy happen to be practically identical in this specific simulation: essentially, the Feedback Control policy has traded energy to restore the required reliability. The adaptive system accomodates the larger noise level without any redesign.

6. CONCLUSIONS

The success of future highly complex SoC designs will depend on the capability of creating networks of heterogeneous subsystems in an efficient and reliable manner. The availability of robust low-energy transmission schemes will be critical. In this paper, we have presented a scheme which combines three essential elements: (1) a variable low-swing transmission to reduce energy; (2) an error detection code and an ARQ protocol to guarantee reliable communication; and (3) a controller which tunes the voltage swing and the operating frequency to minimise the energy per useful bit under a QoS constraint, in a way similar in principle to DVS techniques but more aggressive.

We have discussed different control policies and shown that our transmission scheme achieves the desired goals: (a) Tangible energy savings are possible, especially with typical multimedia workloads (-55% for MPEG traffic) but also with other workloads (approximately -40% with a Poisson traffic). (b) Some control policies we discussed go beyond classic DVS approaches and do not rely on *a priori* characterisation of the process quality and noise sources; they manage to exploit the full capabilities of the silicon by estimating the working conditions *in situ*. This represents a revolution with respect to the ubiquitous worst-case design methodology. Finally, (c) we illustrate an additional advantage of the adaptiveness of our system by showing that a classic interconnect simply fails if the noise sources were underestimated in the design phase; our system, on the contrary, simply invests more energy to achieve reliable communication and therefore needs no redesign.

Although not all pieces of technology are immediately available to implement our scheme, we believe this work suggests that adaptive techniques will be essential to design successful multibillion-transistor SoCs: such techniques must guarantee correct operation with minimal energy over large variations in actual workload, noise, and technology quality.

7. REFERENCES

[1] L. Benini and G. De Micheli. *Dynamic Power Management: Design Techniques and CAD Tools.* Kluwer Academic, Boston, Mass., 2000.

[2] L. Benini and G. De Micheli. Networks on chips: A new SoC paradigm. *Computer*, 35(1):70–78, Jan. 2002.

[3] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano. Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems. In *Proceedings of the 7th Great Lakes Symposium on VLSI*, Urbana, Ill., Mar. 1997.

[4] D. Bertozzi, L. Benini, and G. De Micheli. Low power error resilient encoding for on-chip data buses. In *Proceedings of the Design, Automation and Test in Europe Conf. and Exhibition*, Paris, Mar. 2002.

[5] K. Flautner, S. Reinhardt, and T. Mudge. Automatic performance setting for dynamic voltage scaling. In *Proceedings of the 7th Conf. on Mobile Computing and Networking*, pages 260–71, Rome, July 2001.

[6] V. Gutnik and A. P. Chandrakasan. Embedded power supply for low-power DSP. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VLSI-5(4):425–35, Dec. 1997.

[7] R. Hegde and N. R. Shanbhag. Soft digital signal processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VLSI-8(4):379–91, Aug. 2000.

[8] R. Hegde and N. R. Shanbhag. Soft digital signal processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VLSI-9(6):813–23, Dec. 2001.

[9] H. Lekatsas and J. Henkel. ETAM++: Extended Transition Activity Measure for low power address bus designs. In *Proceedings of the Asia and South Pacific Design Automation Conf.*, Bangalore, India, Jan. 2002.

[10] N. K. J. Li Shang, Li-Shiuan Peh. Power-efficient interconnection networks: Dynamic voltage scaling with links. *IEEE Computer Architecture Letters*, 1(5), May 2002.

[11] D. Liu and C. Svensson. Power consumption estimation in CMOS VLSI chips. *IEEE Journal of Solid-State Circuits*, 29(6):663–70, June 1994.

[12] T. Pering, T. Burd, and R. Brodersen. The simulation and evaluation of dynamic voltage scaling algorithms. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 76–81, Monterey, Calif., Aug. 1998.

[13] P. P. Sotiriadis and A. Chandrakasan. Low power bus coding techniques considering inter-wire capacitances. In *Proceedings of the IEEE Custom Integrated Circuit Conf.*, pages 507–10, Orlando, Fla., May 2000.

[14] M. R. Stan and W. P. Burleson. Bus-invert coding for low-power I/O. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VLSI-3(1):49–58, Mar. 1995.

[15] A. J. Stratakos. *High-Efficiency Low-Voltage DC-DC Conversion for Portable Applications*. Ph.D. thesis, University of California, Berkeley, Calif., 1998.

[16] C. Svensson. Optimum voltage swing on on-chip and off-chip interconnect. *IEEE Journal of Solid-State Circuits*, 36(7):1108–12, July 2001.

[17] J. Walrand and P. Varaiya. *High-Performance Communication Networks*. Morgan Kaufmann, San Mateo, Calif., second edition, 2000.

[18] N. H. E. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. VLSI System Series. Addison-Wesley, Reading, Mass., second edition, 1993.

[19] H. Zhang, V. George, and J. M. Rabaey. Low-swing on-chip signaling techniques: Effectiveness and robustness. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, VLSI-8(3):264–72, June 2000.