

A Robust Self-Calibrating Transmission Scheme for On-Chip Networks

Frédéric Worm, Paolo Ienne, *Member, IEEE*, Patrick Thiran, *Member, IEEE*, and Giovanni De Micheli, *Fellow, IEEE*

Abstract—*Systems-on-Chip* (SoC) design involves several challenges, stemming from the extreme miniaturization of the physical features and from the large number of devices and wires on a chip. Since most SoCs are used within embedded systems, specific concerns are increasingly related to correct, reliable, and robust operation. We believe that in the future most SoCs will be assembled by using large-scale macro-cells and interconnected by means of on-chip networks. In this paper, we examine some physical properties of on-chip interconnect busses, with the goal of achieving fast, reliable, and low-energy communication. These objectives are reached by dynamically scaling down the voltage swing, while ensuring data integrity—in spite of the decreased signal to noise ratio—by means of encoding and retransmission schemes. In particular, we describe a closed-loop voltage swing controller that samples the error retransmission rate to determine the operational voltage swing. We present a control policy which achieves our goals with minimal complexity; such simplicity is demonstrated by implementing the policy in a synthesizable controller. Such a controller is an embodiment of a self-calibrating circuit that compensates for significant manufacturing parameter deviations and environmental variations. Experimental results show that energy savings amount up to 42%, while at the same time meeting performance requirements.

Index Terms—Electrical parameter variations, interconnect for networks-on-chip, low-power systems-on-chip (SoC), self-calibrating designs, VLSI design methodology.

I. INTRODUCTION

THE successful design of highly-complex *Systems-on-Chips* (SoC) depends on the availability of robust methodologies that allow designers to cope with two major challenges: the extreme miniaturization of device and wire features, and the extremely large scale of integration. Since most SoCs will find their application within embedded systems, traditional figures of merit, such as performance, energy consumption and cost, will be as important as the first-design correct/reliable operation and robustness.

The main design challenge of future SoCs will be to connect efficiently many heterogeneous components into an effective network which implements the desired functionality. On-chip *micronetworks* [4] 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-based communication.

Micronetwork design has been subject of recent research activities in different directions. This article focuses on reliable low-energy point-to-point communication, achieved by *dynamic voltage-swing scaling* (DVSS) on communication links, and on control policies for DVSS. Without loss of generality, we restrict our attention to bus-based on-chip communication. We concentrate on the physical and data link layers, and we show the essential features that are needed to provide a hardware-based implementation of a DVSS scheme that can trade off energy consumption for latency, while satisfying a given reliability bound.

Specifically, robust and flexible communication design requires reaching three objectives:

- 1) **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 at design time and it may change drastically and dynamically during operation. Obviously, 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.
- 2) **Energy consumption.** Studies have shown that wires charging/discharging account for a significant fraction of the total energy consumption (up to 40%–50% [24]). 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 fraction of power consumed in the communication is likely to grow. This calls for techniques to reduce the energy consumed in on-chip communication.
- 3) **Reliability and robustness.** Communication reliability is the probability (as a function of time) that no uncorrected error occurs. Many technological factors challenge the traditional reliability 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 of increasing concern that can cause communication malfunctions include *electromagnetic interference* (EMI) as well as capacitive and inductive *crosstalk*.

Design robustness relates to the sensitivity of the design objectives (e.g., performance and energy) with respect to manufacturing variations (e.g., gate-oxide thickness) and environmental

Manuscript received November 27, 2003; revised April 8, 2004. This work was supported in part by Gigascale System Research Center (MARCO/GSRC).

F. Worm and P. Ienne are with the Processor Architecture Laboratory, Swiss Federal Institute of Technology Lausanne, 1015 Lausanne, Switzerland (e-mail: Frederic.Worm@epfl.ch; Paolo.Ienne@epfl.ch).

P. Thiran is with the Laboratory for Computer Communications and Applications, Swiss Federal Institute of Technology, 1015 Lausanne, Switzerland (e-mail: Patrick.Thiran@epfl.ch).

G. De Micheli is with the Computer Systems Laboratory, Stanford University, CA 94305 USA (e-mail: nanni@stanford.edu).

Digital Object Identifier 10.1109/TVLSI.2004.834241

changes (e.g., temperature). Since *deep submicron* (DSM) technologies push design to the very limits of the operational envelope, robust design techniques must take into account unforeseeable variations (i.e., beyond specs) of manufacturing and operation. Whereas robustness can be improved after manufacturing by tight die grading, on-chip dynamic techniques to enhance circuit robustness may significantly raise the die yield.

This paper has two objectives. The first is to address an important aspect of the *physical design of micronetworks*, namely design and policies for DVSS on communication busses. We will describe control policies that address both dynamic frequency and voltage swing scaling. We will report data—based on detailed simulation of performance and power—which support the evidence that DVSS is both a practical and an efficient solution to the physical implementation of communication schemes. The second and broader goal of our methodology is to show DVSS as a specific example of *self-calibration* for SoCs. In a self-calibrating design, the operating point is chosen dynamically without any prior knowledge of the relation between voltage-swing and actual circuit delay. We believe that self-calibrating designs will provide an important means of increasing reliability and robustness in future DSM technology nodes.

A. Self-Calibrating Design

Current design methodologies are typically based on conservative (worst-case) design approaches. Examples include physical design rules for layout, and delay estimates in static timing analysis. We will be mainly concerned about the latter case, since the performance/energy tradeoff is an outstanding issue in system design. Some recent work on statistical timing analysis [1] focuses on better tolerating process variations. Nonetheless, clock frequencies are almost always set on the basis of the longest propagation delay in the worst-case environmental situation (e.g., temperature, voltage, and local charge).

We believe that, within current design methodologies, the observed trends in worst-case analysis may invalidate the benefits of faster, scaled-down semiconductor technologies. Thus, large capital investment for achieving deep submicron silicon fabrication may not return competitive chips. Worst-case design will show diminishing return in speed as devices and supply voltages are scaled-down: the complex interaction of several physical factors will be harder and harder to model accurately, and thus, will push designers toward increasingly conservative assumptions. Whereas some research is ongoing to improve the accuracy of worst-case static timing estimations (e.g., [27]), we think that in some cases a more radical approach is needed. Otherwise, a heavy price will be paid mostly in terms of the feature whose containment is becoming the dominant success factor in very many SoC applications: energy consumption.

Self-calibration addresses the designers' uncertainty on the value of some physical parameters, such as design manufacturing parameters. It is conjectured that the spread of physical parameters is deemed to increase, as design features scale down, due to the ever increasing difficulty of controlling the lithographic and patterning process. At the same time, self-calibration makes design robust against other variations, such as environmental parameters (e.g., temperature). In this

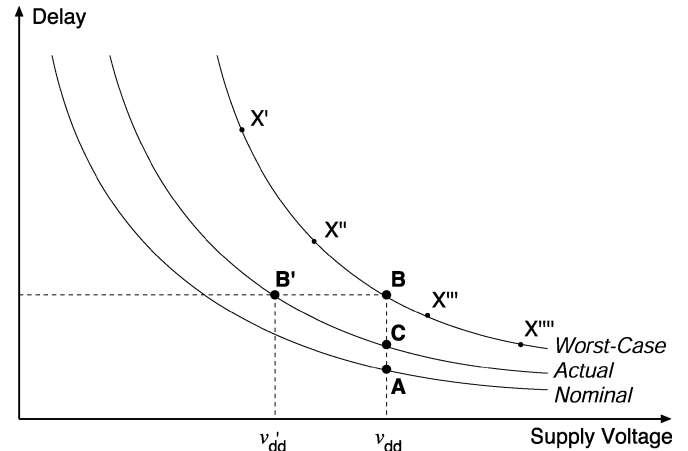


Fig. 1. Worst-case design typically results in a waste of resources—usually silicon area and, more critically, energy. Points X' to X'''' would be used by DVS techniques.

paper, we focus on self-calibration of voltage swings on busses, to cope with energy reduction and information reliability simultaneously.

Example 1: Fig. 1 illustrates the point with a simple qualitative example. Recall that the accurate knowledge of the delay/voltage relation is key for many optimization techniques, such as transistor sizing and dynamic voltage scaling (DVS). The *nominal* relation between delay and supply voltage might be worsened by the deviation of a number of physical phenomena whose cumulative effect is expressed in the *worst-case* relation. Therefore, at a given supply voltage v_{dd} , a designer will assume the most conservative delay—that is, that the operating point is not, for instance, A but B—and implement the design accordingly. In fact, the actual device at a particular instant is very likely to be operating in much more favorable conditions and, for instance, have the actual lower delay indicated by operating point C. This implies the following energy waste: operation at the reduced voltage v'_{dd} (B') would yield the same performance the system has been designed for. A less conservative operation in B' rather than B would achieve the very same user function in the same time but would save a potentially significant amount of energy—roughly proportional to the difference of the square of the supply voltages v_{dd} and v'_{dd} .

B. Related Work

1) *Self-Calibration:* The use of adaptive design techniques in extremely aggressive designs is not new—in some specific situations, it is commonplace. Researchers and practitioners have gone a long way in some cases; for instance, in a state-of-the-art processor, the regional clock skew is adaptively tuned at power-up using relatively complex controllers to compensate for local process variations across a single die [35]. Nonetheless, it is rather uncommon to use powerful digital controllers, such as complex finite-state machines, when the operating point of the transistors is involved, when the overall design functionality is potentially jeopardised, and/or while the circuit is operating: tight analogue feedback loops are generally employed (e.g., phase-locked loops and delay-locked loops).

We believe digital controllers can be used effectively in some particular but practically important situations, namely when it is possible to trade robustness for energy savings, when it is feasible to check—with low overhead—whether the system is operating correctly or not, and when the application has some intrinsic tolerance to limited latency deviations.

It is worth mentioning that the whole idea of operating CMOS devices at voltages below the worst-case characterization point, and thus, in subcritical regions where errors might occur, has seldom been investigated. In a recent paper [18], the possibility of exploiting devices in subcritical regions for digital signal processing (DSP) was presented; in that case, errors arising from the subcritical voltages are compensated by the DSP algorithms. Recently, a whole processor has been designed without noise margin [2], giving an increased momentum to self-calibration. Our goal is similar in the intents, but applies to a different domain (communication instead of computation): it can thus exploit classic techniques to achieve correct behavior despite occasional errors.

2) *Bus Encoding*: One common technique used to minimize power consumption on busses, is the choice of appropriate encoding schemes that reduce the switching activity without affecting the signal information content [5], [32]. This approach has been extended to account for interwire capacitances [21], [31] and reliability issues [8], [9]. Bus encoding techniques have shown effectiveness in reducing power consumption, although the best results are generally achieved in specific environments such as address busses. Researchers [22] have recently proposed to dynamically adapt the redundancy of the encoding scheme. The amount of redundancy is changed in function of the number of errors observed. In fact, encoding is complementary to the scheme described in this paper.

3) *DVS*: The classic silver bullet of power consumption reduction consists in using a lower supply voltage and, specifically for interconnects and busses, using low-swing signalling techniques [34], [42]. 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 complicating 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.

DVS is a now a well established and effective technique to reduce the consumption of systems under given performance constraints [3], [14], [28]. It is usually applied to adapt dynamically the speed of processors in PDAs to current computational requirements; it is now supported by several commercial processors (e.g., Intel XScale, Mobile Pentium, and Transmeta Crusoe). The technique is based on the characterization 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—as, for example, points X' to X'''' in Fig. 1. A transmission scheme applying DVS to chip-to-chip interconnection networks has been recently introduced [23]. 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.

4) *On-Chip Micronetworks*: The notion of transmitting packetized information on chips goes back to the seminal works of Dally [11], [12]. In the recent years, there has been a renewed interest in developing novel network architectures as well as tools and methodologies for protocol design. Full surveys [4], [6], [39] described related work. Among the activities in this domain, it is worth recalling: 1) the introduction of new architectures for on-chip networking, such as SPIN [15] and Octagon [20]; 2) the study of encoding schemes for error-resilient communication [8], [9]; 3) various attempts to use packet routing strategies [19], [26], [41]; and 4) efforts to understand the impact of system software and middleware [7], [10].

Despite a large body of work related to signaling [13], little work has been done to support the specific design of micronetworks at the physical layer. The aforementioned work by Shang [23] is decoupled from error control. In our work, we consider DVSS in relation to the three major goals for micro-network design: adaptive bandwidth, low-energy consumption, and error resiliency.

C. Structure of the Paper

Section II introduces the main ideas of the paper and describes the architecture of the on-chip transmission system which we propose to achieve the goals mentioned above. Section III is devoted to the operation point policy and studies some of its properties. Section IV describes relevant models. Among others, we explain our abstraction for the communication channel and its bit error rate. Simulation results, based on both artificial and real workloads, are discussed in Section V. We point out the ability our system shows to exploit dynamic bandwidth requirements, accommodate technology variations, as well as design uncertainties. Lastly, Section VI concludes with some remarks on the potentials of the presented scheme.

II. DVSS FOR ON-CHIP INTERCONNECTS

We consider a typical unidirectional point-to-point interconnect between subsystems. Fig. 2(a) shows a qualitative view of the classic interconnect: at the producer end, a first-in-first-out (FIFO) or a similar buffer is used to decouple 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 Fig. 2(b). A first extension of Fig. 2(a) in this direction is discussed in [40]. To reduce the energy consumed per bit, we apply DVSS 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 [42]. Of course, the variable voltage swing impacts the speed at

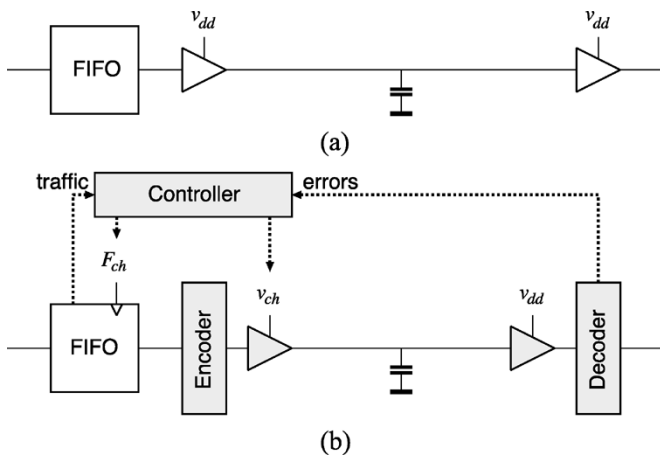


Fig. 2. The basic idea of a self-calibrating point-to-point unidirectional on-chip interconnect. (a) The classic static scheme, with a FIFO to decouple two subsystems. (b) The proposed self-calibrating scheme with the different elements needed to achieve the desired goals.

which the interconnect driver is able to 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 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 Go-Back-N [37]. In its simplest embodiment, error detection is provided by a code (e.g., 8-bit CRC) which is transmitted in parallel with the data. DVSS is also applied to the additional (redundant) lines, which consume additional energy. Nevertheless, energy savings achieved by the lower swing provided by our DVSS scheme outweigh the energy required by driving the additional lines.

We point out that our architecture can seamlessly be applied to segmented busses. In this case, the same voltage swing is used along all segments—which is possible as every repeater only consists of an inverter supplied at v_{ch} . As matter of fact, we will report in Section V conservative results which will only consider the energy spent on the interconnect wires; in reality, the repeaters will draw additional energy which also scales down with our technique. We have modeled a segmented bus, and found that the energy difference compared to a nonsegmented bus amounts to only a few percents.

Such a controller must choose, as a function of bandwidth requirements, safe voltage-frequency pairs from a set of possible 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, by monitoring the error rate.

In essence, the controller will provide the minimum voltage swing such that communication is achieved at the requested bandwidth and that the correction/retransmission rate is limited. In the ideal case that all transmission errors could be detected, our scheme would trade off transmission energy for additional communication latency (i.e., tardiness). In practice, codes detect only a subset of the possible transmission errors and undetected malfunctions cause data loss or corruption: the residual error rate can be seen as the failure rate of the communication channel. Thus, the overall objective of DVSS is to trade off transmission energy for additional communication latency under a transmission reliability bound.

A. Possible DVSS Architecture

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 [42]) 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 [16], [33]).

A more practical view of our system is represented in Fig. 3. It represents in greater detail the idea of Fig. 2(b), adding some necessary components that will be described in the following sections. As the figure suggests, the data path is pipelined into an encoding stage, a synchronizing stage, and a decoding stage. Although, the choice of pipelining belongs to implementation issues, our architecture relies on this fact to recover from metastability. We have chosen in this case to perform error detection and retransmission on a per word basis (a word is what is read from the FIFO). One could very well imagine that the ARQ strategy deals with packets of several words.

B. Self-Synchronising Encoding

There are several challenges to make the system robust under the extreme conditions planned. The main problem is that we are not trying to screen out and remove some relatively infrequent errors (as in most cases where error detecting codes and ARQ protocols are used). Instead, we operate the system within a small margin from where it becomes no longer operational. In a sense, we will push our system to explore the operating space and, thus, to become at times nonoperational. In this section, we will analyze some of the related challenges and suggest possible solutions.

The use of a simple spatial encoding (such as adding some parity bits to the data word) is not sufficient. This encoding would be effective to detect, for instance, that one bit has not yet made a transition, due to crosstalk. Yet, if our clock is so fast that the complete previous word is still present on the interconnect [for example, when the sampling process is like Fig. 4(b)], a pure spatial encoding would diagnose the result to be correct and would not detect that the new word is simply not ready. In other words, the bit-error probability which we will study

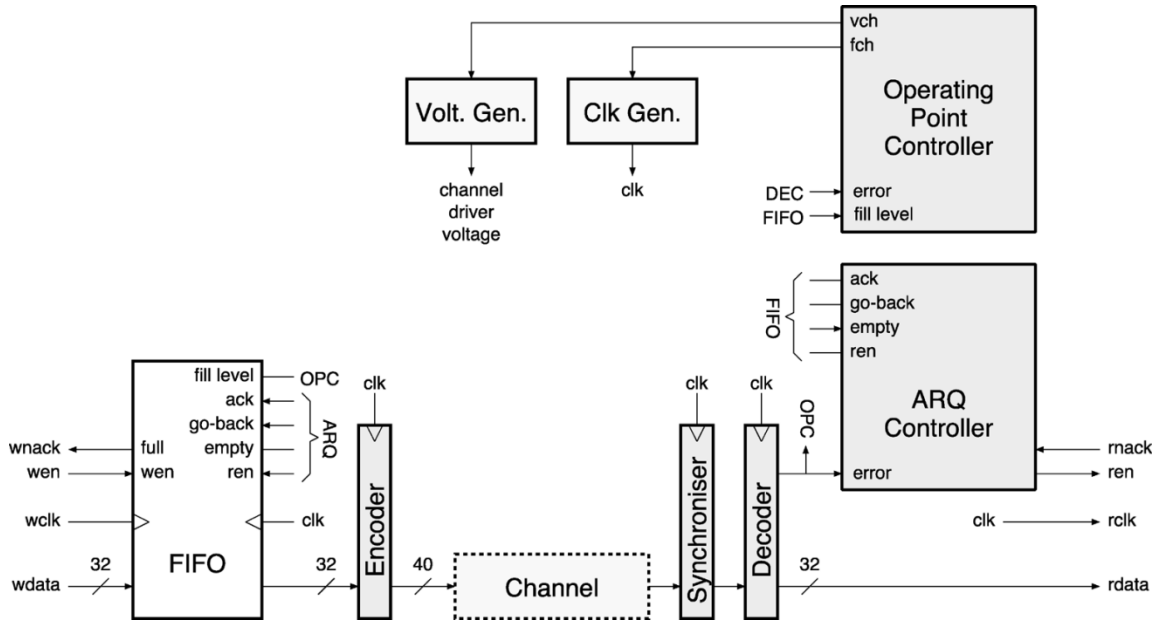


Fig. 3. A possible architecture for the self-calibrating point-to-point unidirectional on-chip interconnect.

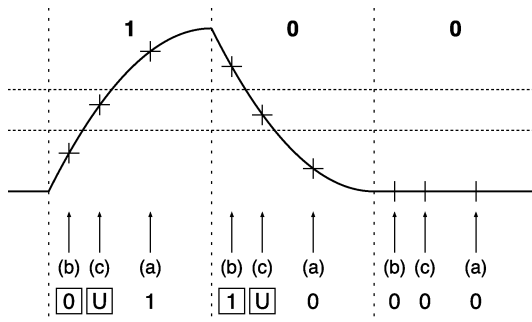


Fig. 4. A qualitative view of the sources of error in a self-calibrated interconnect operating in too aggressive delay/voltage conditions. (a) Correct operation after a sufficient delay. (b) Bit-errors due to the sampling after a largely insufficient delay. (c) Risk of metastability in the receiver for slightly too aggressive sampling times. (Note that the figure is simplistic in that a new symbol would be emitted at the same time the line is sampled.)

in detail in Section IV-B does not express a bit-flipping probability: since we model the charging and discharging of interconnect bit-lines—including timing errors such as those induced by crosstalk—the bit-error probability models approximately the probability that a line is sampled before it had the time to change to its new state (see Fig. 4).

Instead of more classical self-synchronising codes [36], such as 1-of- N schemes, we use the simpler scheme shown in Fig. 5. Our error detection scheme works by 1) generating one additional bit, alternatively a 0 and a 1, which is not transmitted but produced independently at the source and destination, and 2) computing and transmitting eight CRC-8 bits using the generator polynomial $x^8 + x^2 + x + 1$ [37] on the data word (e.g., 32 bits) padded with the generated bit. This bit ensures that any two successive identical data words cannot have the same encoding—hence, two successive 40-bit encoded words on the channel may be identical only if an error has occurred. To correct possible desynchronization errors between the toggling

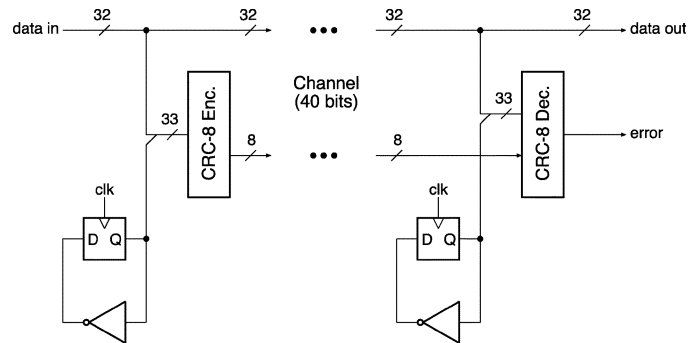


Fig. 5. A possible self-synchronising encoding scheme. The error signal detects also when the sampled word is still the last one correctly sent across the channel.

flip-flops, they can be reset with the ARQ signal triggering a resend.

The analytic assessment of the robustness of this scheme, which combines a flipping bit and a CRC-8, is beyond the scope of this paper. Some simulations were performed in VHDL with a functional model of the channel that approximates an analytical bit-error rate model to be introduced in Section IV-B. No residual undetected error could be observed over $0.32 \cdot 10^9$ random bits for raw bit-error rates up to 10^{-3} . Fig. 6 shows the residual bit errors as a function of high raw bit-error rates.

Although, by no mean specific to this encoding, one point is worth mentioning: as the bit-error rate approaches unity, the absolute number of undetected errors increases dramatically. While this is no concern in typical application of error correcting codes, where the error rate is assumed to be always small, self-calibrating systems might operate briefly in regions of extremely high bit-error rate: thanks to the flipping bit, our encoding scheme has the important feature of detecting errors when the raw bit error rate approaches unity. This feature is essential to prevent the operation point controller using too aggressive points where the bit error rate would be close to unity.

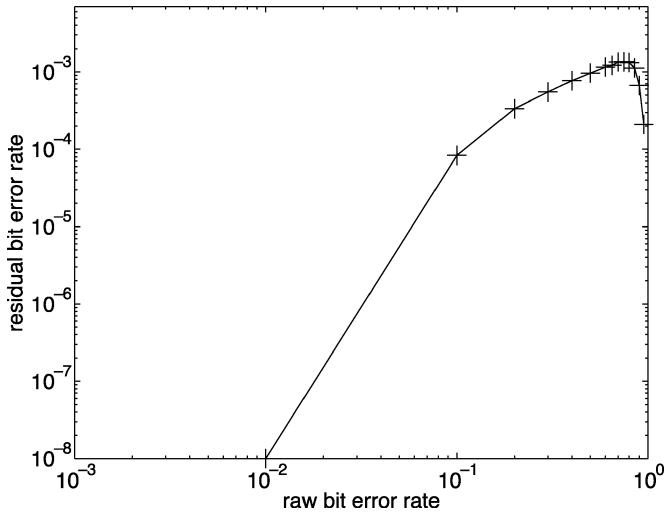


Fig. 6. Residual bit error rate as a function of the raw bit error rate.

Other error-resilient encoding schemes can be applied. For example, codes with stronger detection probability and/or under different error models. It is also possible to avoid the additional lines required to transmit the redundant codes, and to insert error detecting codes in the data stream. These implies trading off latency for area. A more subtle tradeoff involves the energy consumed and error-detecting probability of various coding styles.

In this paper, we consider the parallel coding scheme of Fig. 5, for the sake of simplicity. Nevertheless, we want to stress that our approach is general, and can be combined with different data formats, including packet encapsulation.

III. OPERATION-POINT CONTROL POLICY

This section deals with the operation-point control policy. The problem is stated; we discuss the solution we bring from the algorithmic point of view down to the implementation issues. Let us now express the control scheme first mentioned in Section II as a constrained optimization problem. Each time the control policy is applied, the controller has to find the pair (v_{ch}, F_{ch}) which

- 1) minimizes the energy consumption,
- 2) meets a performance constraint, and
- 3) meets a reliability constraint.

In our system, minimizing the energy consumption is equivalent to using the lowest possible voltage swing which does not cause transfer errors (retransmissions). Namely, the controller periodically reduces the voltage swing until retransmissions occur to finally settle on the lowest voltage swing where no retransmissions were recorded. The policy therefore minimizes energy, while meeting a reliability constraint, since unsafe operating points are avoided. As a matter of fact, operation points causing retransmissions are neither energy efficient nor do they meet tight reliability constraints (such as residual word error rate of the order 10^{-10}). While the performance constraint has to be set by an upper layer to reflect performance changes during system operation, the reliability constraint is assumed to be constant. For example, the controller has to guarantee that no residual word error appears for, at least,

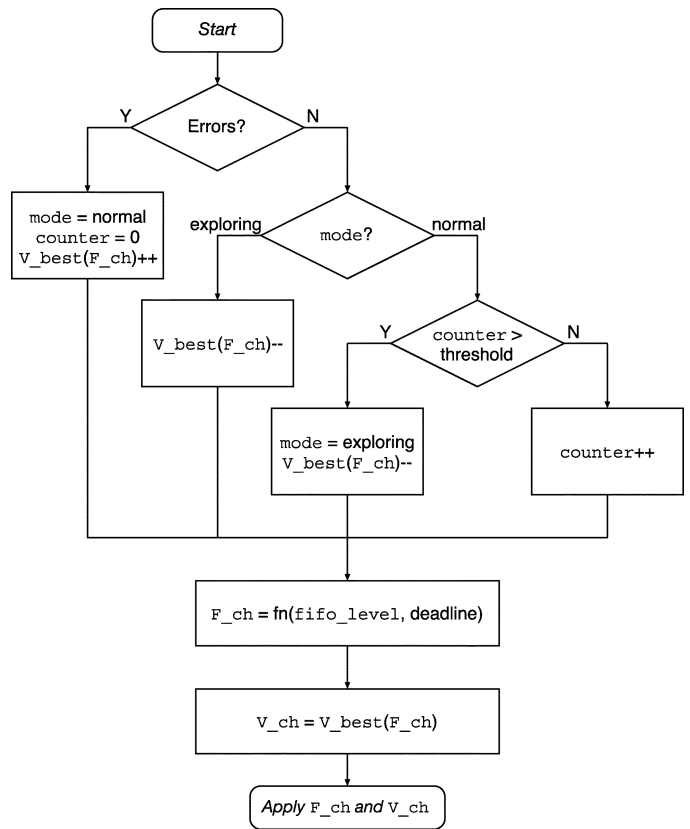


Fig. 7. Simplified operation-point control policy.

1 million transfers. Without loss of generality we will assume both constraint constant in the sequel.

As Fig. 3 shows, in practice one can separate completely an ARQ controller and a controller devoted to the choice of the operating point. The former has the sole task to push all words of data through the channel until they are communicated without error, ignoring the channel parameters. In other words, the ARQ controller only decides *which words* to push through the channel. On the other hand, the operating-point controller is in charge of picking the lowest frequency and voltage swing required to meet some communication constraint—such as an average delay: it decides *how* to communicate, checks if the choice is appropriate, but ignores *what* is going through the channel.

Fig. 7 shows a very simplified control algorithm for the operating-point controller which memorises the best operating point for each possible frequency. The controller performs independently three tasks:

- 1) it records where are the best voltage/frequency points (that is, for each possible frequency, it finds out the lowest voltage swing usable). It does so from the experienced errors and from periodic attempts to explore more aggressive operating regions;
- 2) it chooses a frequency based on the delay constraint and buffer fill level;
- 3) it chooses the voltage swing of the estimated best point at the selected frequency.

Fig. 8(a) shows graphically how the operating point is selected among a set of possibilities (one point per frequency)

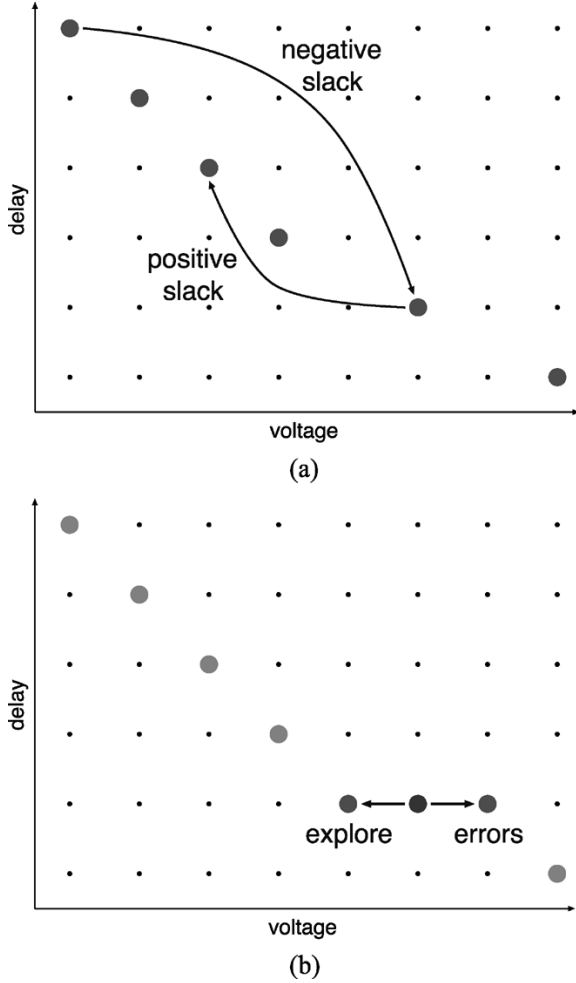


Fig. 8. Use and estimation of best operating points. (a) The control policy fixes the operating frequency in function of the delay constraint; it sets the operating voltage to the minimum value which has experienced error-free transmission. (b) The controller raises the best voltage for a given frequency when experiencing errors; otherwise, every several cycles, it attempts tentatively to reduce it to ensure most aggressive operation.

which are approximately an estimation of Pareto operating points. The most appropriate is chosen as a function of the observed traffic and the delay constraint. Fig. 8(b) illustrates the effect of the estimation process: Errors push immediately the system to become more conservative (that is, to increase the voltage swing associated with a given frequency). To ensure the most aggressive operation, whenever the system works fine for a given number of cycles (e.g., 500) the controller briefly attempts to reduce the voltage at constant frequency; if there are no observed errors for a few cycles (e.g., 50), the new point is recorded as the best point at that frequency. Tuning these counters allows to parametrise the controller behavior and ensures the stability of the control policy.

A. Estimated Delay

In the sequel, we denote by *packet* a data chunk that is buffered, encoded, transmitted, decoded and retransmitted if needed. Our controller needs to estimate the expected packet delay through the system. A comparison with the given delay

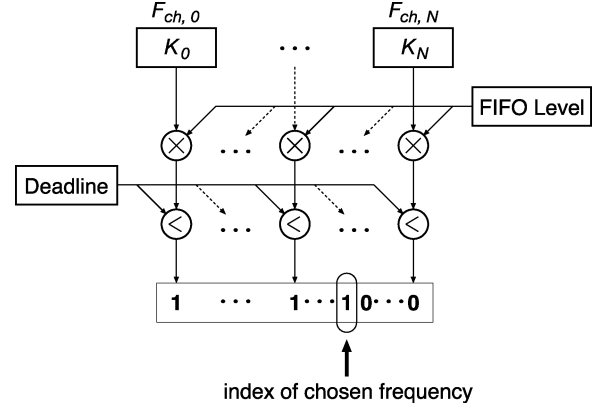


Fig. 9. Choice of the slowest frequency making possible to meet the delay constraint. By convention, frequency index 0 corresponds to the fastest frequency. K_0, \dots, K_N are constants that are hardwired for every frequency.

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 queuing. We estimate it as follows:

$$\Delta_{est}(F_{ch}) = \frac{l \cdot \Xi_W}{F_{ch}} \quad (1)$$

where l represents the queue size, that is, the number of packets of W words currently present in the FIFO buffer, and Ξ_W is the expected number of cycles needed to send a packet of size W . 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. Actually, if we model the system as a M/D/1 queue, finding out the most slack critical element turns out to be a tough problem. As a result, tracking the most critical element is likely to cause a logic overhead that will not be amortized by the additional savings it allows. Moreover, due to the extremely low residual error rates which are acceptable, we assume that in most cases, no retransmission occurs—i.e., that the packet transfer succeeds at the first attempt. Therefore, our controller assumes that Ξ_W does not depend on the state of the channel (F_{ch} and v_{ch}) and represents the latency of the system from the FIFO to the output (1 if the system was not pipelined). As a result, Δ_{est} is not a function of v_{ch} , and this independence is key in our policy: it allows to split the concerns and choose the frequency appropriate for the deadline, first, and choose the voltage swing appropriate to the frequency, next. As far as the implementation is concerned, the controller has to find, among a discrete set of frequencies, the slowest one that still meets the delay. Fig. 9 illustrates this selection process.

B. Properties

After having introduced the control policy, we informally discuss natural questions that arise as to its *optimality*, *stability*, and *sensitivity* to design parameters. We consider two design parameters: the first one is the threshold mentioned in Fig. 7, the second one is how slower the controller runs compared to

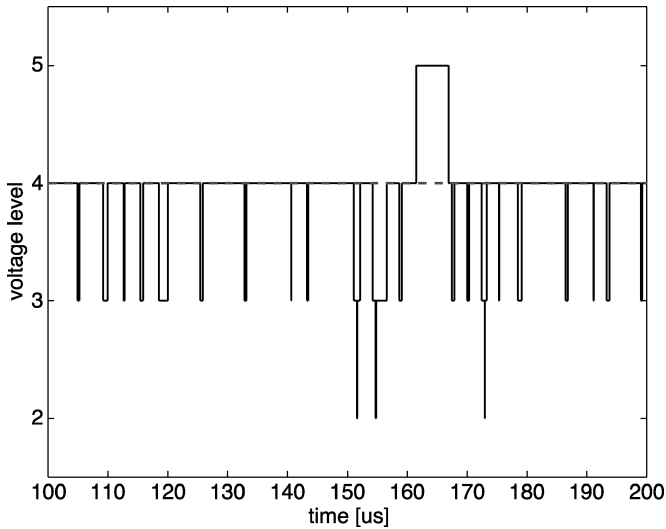


Fig. 10. Evolution of one entry in the Pareto-voltage table. Level 4 is the optimal Pareto voltage. The controller periodically attempts to reduce the Pareto voltage. The controller mostly operates on the correct Pareto-voltage levels.

the datapath. The next three sections discuss the impact of these parameters on optimality, stability, and sensitivity.

1) *Optimality*: The algorithm presented in Fig. 7 contains two deviations from an optimal control approach:

- the slower controller clock, that results in the frequency being adjusted not every cycle;
- the tracking of the optimal voltage (Pareto-voltage) associated to every operating frequency.

We stress that the choice of F_{ch} is not approximated; performance is only affected by how slower the controller runs. Therefore, the first deviation from optimal control stems from implementation reasons, while the second is intrinsic to the algorithm. We study the impact of the slower controller clock in Section III-B3, and show here that the way we track the Pareto-voltage hardly affects the control quality. To do so, we simulate the transfer of 100 000 words and *a-posteriori* determine the Pareto-voltage of every frequency. Then, we compare the performance of our controller with another *optimal* controller that knows which Pareto-voltage to use for every frequency. First, Fig. 10 illustrates how the voltage level (given a frequency) changes as time passes. The difference in performance is minute: our controller saves 25% of energy in this case, while the optimal controller saves one more percent. Both controllers performs the same in term of average transfer delay and residual word errors (none in both cases).

2) *Stability*: In our context, stability will translate into the following properties:

- the FIFO level does not grow to infinity,
- after an uncontrolled disturbance (traffic variation or errors), the controller eventually settles on an(other) operating point.

Because it tries to obey a delay constraint, the controller will not let the FIFO fill level grow to infinity. It does only so in pathological cases where the classical system's FIFO would fill too. This is even more evident as the self-calibrating system has a maximum frequency larger than the classic one which suffers

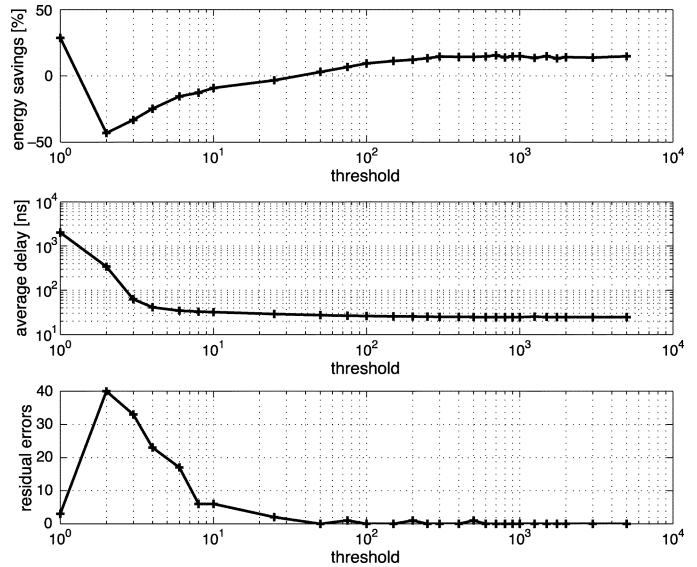


Fig. 11. Sensitivity of various metrics to the threshold appearing in the description of the controller algorithm.

worst-case limitations. The second bullet is granted by the algorithm itself. The only point to discuss stems from the threshold that governs the controller aggressivity. A too low value will lead to undesirable voltage level oscillations. The next paragraph shows that a wide range of values enable the system to exhibit a sound behavior.

3) *Sensitivity*: We show here how relevant metrics such as the energy, transfer delay, and residual word errors are affected by the choice of the controller threshold and by how much slower the controller runs compared to the data path. Recall that the threshold determines how often the controller attempts to set more aggressively the voltage level used for a certain frequency. This threshold is henceforth related with the speed of reaction to noise level changes. In addition to illustrating the sensitivity to these parameters, the results presented guide us toward a wide range of acceptable values. We have simulated the transfer of 100 000 words (refer to Section V for more experimental details). Fig. 11 shows that energy saving, words transfer delay and residual word errors remain in desired intervals over a wide range of threshold value. Fig. 12 illustrates the impact of the ratio between the controller clock and the data path clock. The controller is only allowed to run slower. If the controller clock is too slow, the system behaves poorly both in terms of energy saving and average delay. Because of the violated delay constraint, energy efficient operating points are excluded. On the contrary, if the controller clock is too fast, the delay constraint is already met. At the same time, the controller overhead in terms of gates becomes tangible and decreases energy savings. The top two graphs of Fig. 11 do not show any tradeoff—i.e., the larger the threshold, the better the performance. The reason is that we did not modify during the simulation the statistics of errors. As matter of fact, even large threshold values do not hurt energy and delay figures since the Pareto-voltage are constant over the whole simulation.

The next section briefly discusses the hardware cost of such a controller.

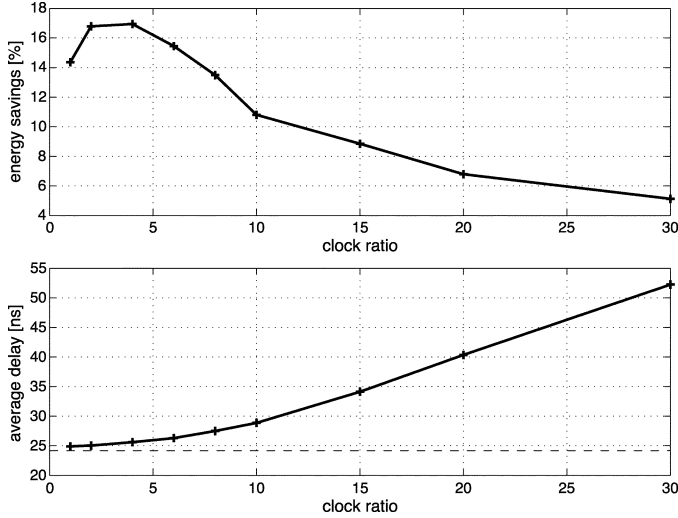


Fig. 12. Impact of the ratio between controller and data path clock on: (top) energy saving and (bottom) average transfer delay. The dashed line represents the target delay of the classic system.

C. Hardware Complexity

The implementation of the algorithm described in Fig. 7—although considerably more detailed than what is shown—has still a relatively low hardware complexity and requires an area equivalent to a few thousand NAND-2 gates. The whole system consists actually of 770 cells (230 sequential and 540 combinatorial). An accurate estimation of the energy consumed is given in Fig. 15, assuming a discrete operation set of four voltages and four frequency levels.

IV. COMMUNICATION CHANNEL AND ENERGY MODELS

This section describes the underlying models for the communication channel, noise, bit error rate, and energy.

A. Channel and Noise

We model the physical communication channel as a lumped capacitive load, that is charged/discharged by the driver through MOS transistors. For the sake of simplicity, we ignore distributed effects, and we assume that resistive and inductive effects are negligible.

We consider two possible sources of noise. The first one is an additive white Gaussian noise, modeling external disturbances. The second error source captures the variability of the channel cutoff frequency around its nominal value, representing the effects of temperature, manufacturing conditions, etc., on the propagation delay through the interconnect. We assume these two noise sources to be uncorrelated. We further assume that an error occurs if either the operating frequency exceeds the channel cutoff frequency, or if the additive noise exceeds half the voltage swing.

The *channel cutoff frequency* F_{cut} is defined as inverse of the delay through the transmission wires. A simple expression can be derived by assuming that delays are measured as the time for the lumped capacitance to effect half a swing (i.e., $v_{\text{ch}}/2$) and

by neglecting velocity saturation and channel length modulation [38]:

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

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

A more complex expression can be used when considering velocity saturation and channel length modulation [29] as well as sense amplifiers at the receiving end. Our control methodology does not depend on the delay modeling—this is only used for simulating our design and can only have a limited numerical impact on the results. We therefore refrain from using more complex delay-models in our experiments.

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 $\sigma_{v_{\text{noise}}}$

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

Although on-chip disturbances are more accurately modeled as burst noise, the white noise model developed here suffices to prove our concept.

B. 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 $F_{\text{ch}} \leq F_{\text{cut}}$, and is approximated to one otherwise. After some computations, one obtains

$$\begin{aligned} \epsilon_b &= 1 - \left(P(F_{\text{ch}} < F_{\text{cut}}) \cdot P\left(v_{\text{noise}} < \frac{v_{\text{ch}}}{2}\right) \right) \\ &= 1 - \left(Q\left(\frac{F_{\text{ch}} - \mu_{F_{\text{cut}}}}{\sigma_{F_{\text{cut}}}}\right) \right) \cdot \left[1 - Q\left(\frac{v_{\text{ch}}}{2\sigma_{v_{\text{noise}}}}\right) \right] \end{aligned} \quad (3)$$

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

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

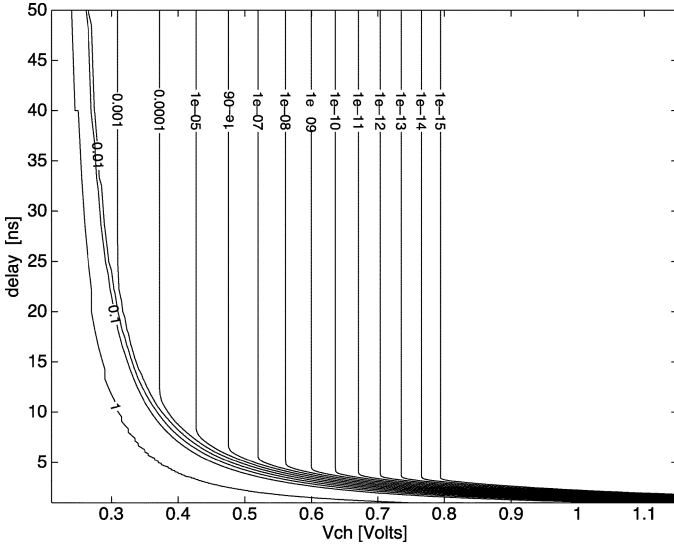


Fig. 13. Contour plot of the bit error rate in the v_{ch} and $1/F_{ch}$ plan.

Eq. (3) is a generalization of the relation introduced in [17]. The new relation takes into account some other, important sources of errors. Fig. 13 shows a typical plot of ϵ_b in the (v_{ch}, F_{ch}) plane. One recognizes 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 detecting code and of the ARQ policy reduces the number of actual errors on the words exchanged between the two subsystems. The error detecting 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

$$\epsilon_w^{res} = f_{code}(\epsilon_w^{raw}) = f_{code}(1 - (1 - \epsilon_b)^{N+n}). \quad (4)$$

For instance, consider a system that uses a CRC code with $n = 8$ redundant bits added to each $N = 32$ -bit word). 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^{-6} for the same residual word error rate. At last, let us point out that the CRC check-bits can as well be computed over a few words, building thus a multiwords packet. Doing so reduces the additional bit lines overhead whereas it increases the latency, and makes retransmissions more cost expensive. As a matter of fact, the packet size becomes another design parameter, enabling new tradeoffs.

TABLE I
OPERATING RANGE OF THE SELF-CALIBRATING INTERCONNECT

	Classic system	Self-calibrating system
Voltage swing	1.2V	0.6 . . . 1.2V
Frequency	500MHz	50 . . . 1000MHz

C. Energy Consumption

Finally, we need to express the objective function which we want to minimize. 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 $E_{redundant}$ to transmit the additional n redundancy bits of the error detecting code;
- the energy E_{rtx} to resend words when previous trial(s) arrived corrupted to the receiver;
- the energy E_{system} consumed by the logic of the ARQ and operation points controllers, the encoder, the decoder and the synchronizer;
- 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.

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, nonadaptive scheme only comes from bit flipping activity

$$E_{cl} = K \cdot N \cdot v_{ch}^2. \quad (5)$$

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

$$E_{adaptive} = E_{cl} + E_{redundant} + E_{rtx} + E_{system}. \quad (6)$$

Equations (5) and (6) give a comparison basis for the energy consumed per transmitted word E_w in the classic and adaptive systems. We have neglected E_{conv} : the efficiency of state-of-the-art voltage converters can be as high as (95%) [30]. In addition, we have neglected the energy overhead of three backward control lines due to their very low switching activity.

V. SIMULATION RESULTS

We synthesized and simulated a self-calibrating 32-bit interconnect system, and we compared it with two kind of classic fixed-swing systems. The first classic system transmits raw unencoded data on 32 bits. The second one represents the increasing concern with reliability of on-chip communication (see for instance the implementation of practically all internal buses of Itanium 2 [25]). This latter system uses an error detecting code such as the one described in Section II-B and retransmits corrupted words. We refer to the former system as the *classic* one and to the latter as the *classic with codec*.

We model a typical 90-nm CMOS technology and noise sources as follows.

- Nominal supply voltage: 1.0 V.
- Device threshold voltage: 0.2 V.
- Additive noise standard deviation: 0.04 V.

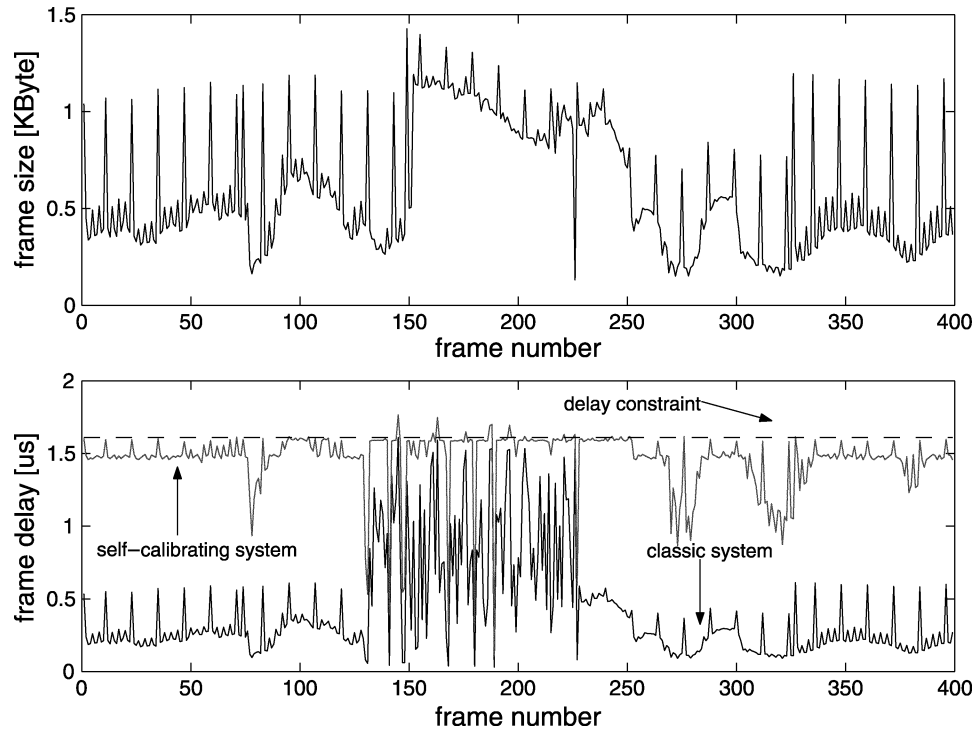


Fig. 14. Transmission of a variable workload. Top: workload variation in time. Bottom: incurred frame delay in the classic system (low delay) and in the self-calibrating interconnect (delay as close as possible to the imposed constraint—dashed line).

- Cut frequency average and standard deviation: 1000 and 70 MHz.

These technology data are needed for bit-error simulation while the controller is completely technology independent. We assume a 1-cm bus length. According to the technology manual, the corresponding bit line capacitance amounts to 2.73 pF. Table I summarizes the operating range of the system. We will present our results with delays and frequencies relative to the two classic system. To calculate the energy advantage of the self-calibrating system we will take into account all the contributions described in Section IV-C, except the overhead of the voltage converter. In particular, the energy overhead incurred by the additional redundant wires is accounted for.

We present three experiments.

- 1) The first example focuses on the energy advantage of dynamic bandwidth adaptation considering a realistic MPEG-based workload and an artificial one.
- 2) The second example shows the energy advantage of dynamically tuning the operating point to actual technology variations.
- 3) The last example illustrates the robustness of our system to unpredictable noise sources.

A. Dynamic Bandwidth Adaptation

Modern multimedia algorithms have dynamically varying requirements. Fig. 14 shows how the self-calibrating system takes advantage of a time-varying MPEG workload. In the bottom graph, one can observe that the adaptive system tries to exactly match the bandwidth to the current needs: it slows down the communication link to send every MPEG frame as slowly as possible in the allotted time and, ideally, not any faster. The operation at a lower frequency grants a tangible

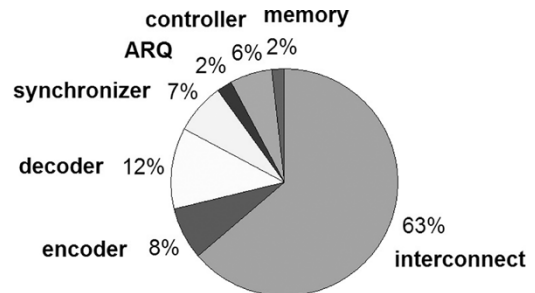


Fig. 15. Energy breakdown of the self-calibrating interconnect.

reduction in average energy: the whole trace, consisting of 400 frames of several kilobytes each, requires 42% less energy with a dynamically self-calibrating system compared to a classic system, and 47% less energy compared to the classic system with codec. Such a saving was achieved by letting the channel controller run at half the frequency than the other components; this reduces the energy consumption in the controller. Fig. 15 depicts how much each component of the self-calibrating system contributes to the energy budget. One can notice that the controller logic and the synchronizing registers, which are the only contributions added in our scheme compared to the classic system with codec, impose a relatively limited overhead (the operation-point controller and synchronising registers account for 16% of the total system energy). On the other hand, it is interesting to note that the energy cost of the codec is not fully negligible and about of the same order of magnitude.

As a last illustration of the energy saving resulting from dynamic bandwidth adaptation, we expose both the classic and the self-calibrating systems to various Poissonian workload of different traffic intensity. Each workload consists of 100 000 word generated according to Poisson arrivals. We require that, for each workload, the self-calibrating interconnect

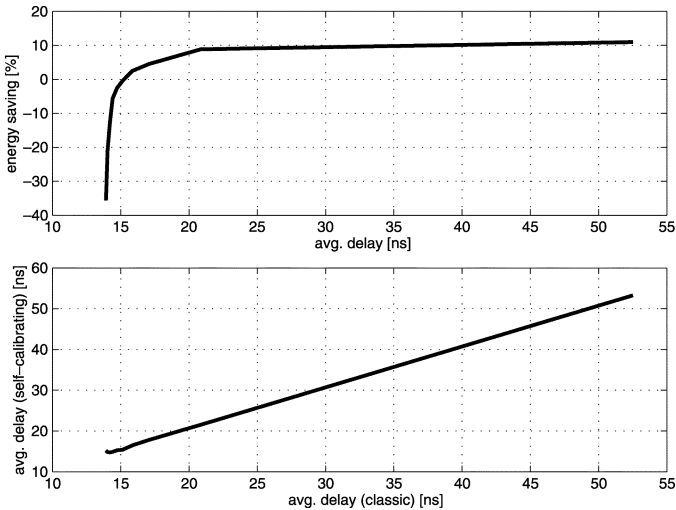


Fig. 16. Transmission of different Poissonian workloads. Top: energy saving (with respect to a classic system) for various average transfer delay constraints. The system becomes energy-inefficient only when the workload requires the interconnect to always work at full speed. Bottom: average transfer delay of the self-calibrating interconnect as a function of the classic system average transfer delay.

offers the same average transfer latency as the one experienced by the classic interconnect. Fig. 16 illustrates the energy saving granted in this case. The top graph shows that even in this more stringent situation, our controller manages to save energy consistently in all those cases where the average delay is not too constrained—only for tight constraints, the controller energy overhead is larger than the diminishing savings. The bottom graph shows the quality of the control policy which, in almost all cases manages to match the performance of the classic system.

B. Exploiting Technology Variations

Fig. 17 illustrates the effect of technology on the choice of control points: On a *poor* wafer, simulated with a cut frequency average of 875 MHz, the controller chooses mainly Pareto points relatively close to the worst-case delay line. On a *good* wafer, simulated with a cut frequency average of 1125 MHz, the points chosen are mostly along a more aggressive delay-voltage line and reflect the lowest delays experienced by the system. In other words, our control policy is effective in “discovering” the real delay-voltage characteristic of the technology without making any assumptions on it. In both cases, the cut frequency standard deviation has been reduced to 55 MHz to reflect its lower indeterminacy on wafers of a defined quality. These hypotheses result in 1% of the wafers being classified as *good* or better than *good*, and *poor* or worse than *poor*. The simulated traffic consists of an artificial workload of 100 000 words with arrival times following a Poisson process. Table II summarizes energy savings and relative performance of the self-calibrating interconnect compared to the classic and classic with codec systems; Fig. 18 illustrates the impact of wafer quality on energy saving. As expected, for any type of wafer, the savings improve with the relaxation of the timing constraints and are dependent on the quality of the wafer. This fact can have a very interesting effect in products designed early after the introduction of a new technology node: at design time the

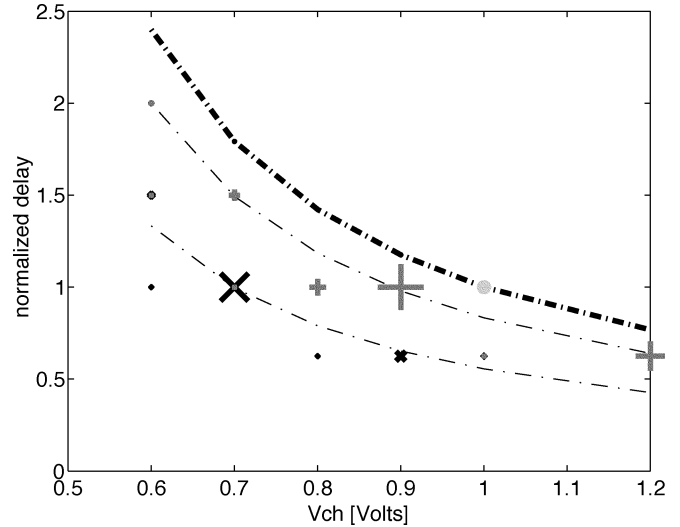


Fig. 17. Operating points used depending on technology variations. $\circ \rightarrow$ classic system; $+ \rightarrow$ self-calibrating system on a *poor* wafer; $\times \rightarrow$ self-calibrating system on a *good* wafer. The bold line represents the worst-case relation between delay and voltage.

TABLE II
ENERGY SAVINGS AND AVERAGE DELAY VARIATION, FOR DIFFERENT WAFERS QUALITY, AND COMPARED TO THE CLASSIC, AND CLASSIC WITH CODEC SYSTEMS

	Energy saving		Average delay variation	
	Wafer		Wafer	
	Good	Poor	Good	Poor
Classic	21%	-8%	$\leq 0.5\%$	$\leq 0.5\%$
Classic with codec	30%	5%	$\leq 0.5\%$	$\leq 0.5\%$

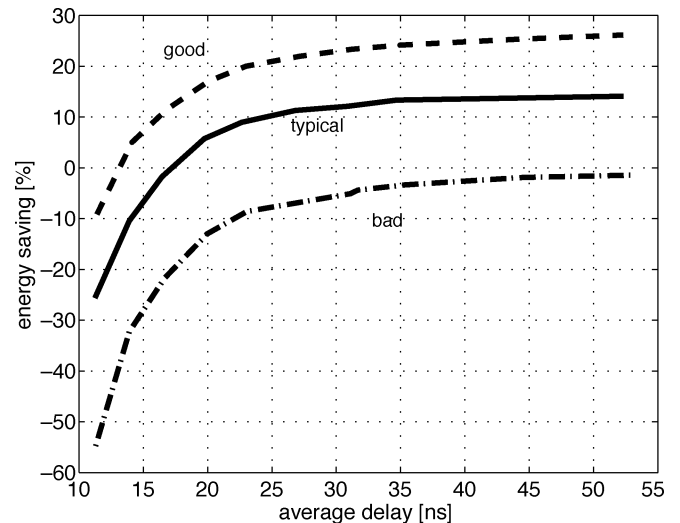


Fig. 18. Energy-latency tradeoff for different wafer quality. The better the wafer, the more important the energy savings. The savings are with respect to the classical system.

technology is poorly controlled and chances are that our system will not save significant amounts of energy. Yet, as products will go into production and the technology will mature, more significant savings would be possible. Classic systems would need a redesign, whereas our system will profit automatically of the technology improvements.

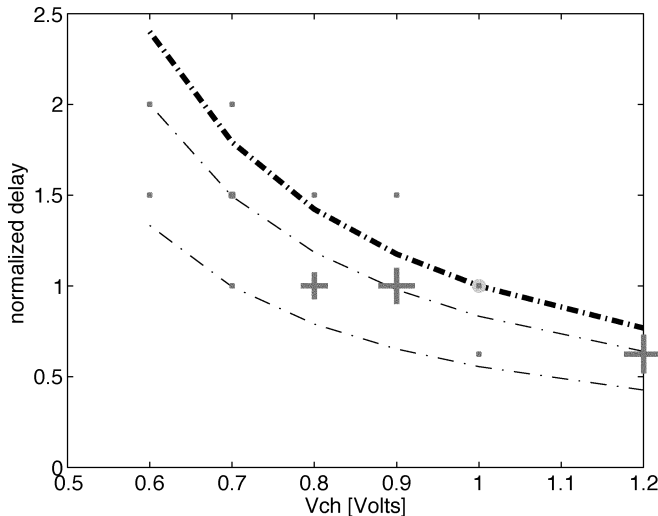


Fig. 19. Operating points used by the self-calibrating system in the presence of strong noise. $\circ \rightarrow$ classic system; $+$ \rightarrow self-calibrating system. The classic system has a reduced yield under these conditions, while the self-calibrating one moves to more energy-consuming, but safer operating points.

C. Robustness Toward Design Uncertainties

Fig. 19 simulates the effect of design hypotheses which have turned out to be too optimistic after manufacturing, for instance due to unexpected sources of on-chip noise. To simulate the self-calibrating system with a higher noise, we raise the standard deviation of the additive noise from 0.04 to 0.1 V and the cut-frequency standard deviation from 70 to 90 MHz. It should be stressed that the classic system is not expected to work any more under these conditions: 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. As the figure shows, the self-calibrating system adapts to the strong noise by choosing less aggressive operating points and by trading energy for robustness: energy savings shrink to 4% and 16% for the classic system and classic with codec respectively. As to the average latency, the increase amounts to 4% compared to the desired behavior of the classic system—but the interconnect operates correctly and avoids the yield reductions incurred by the classic system.

VI. CONCLUSIONS

Our contributions are two fold. We have first presented a general paradigm to cope with deep submicron design, where physical parameters are characterized with a certain level of uncertainty. We claim that achieving high-performance and low-power consumption requires circuits that adapt their operating point at run time, thus avoiding pessimistic design assumptions that may offset the advantages of downscaling silicon technology. Next, we have shown a specific embodiment of self-calibration: we have described a self-calibrating circuit that controls the voltage swing on busses, so that dynamic power consumption is minimal for a required data rate.

We have outlined the scheme of an online controller that we have synthesized and simulated extensively. Moreover, we thoroughly estimated the incurred energy overhead. Our experimental results show that self-calibrating circuits display

the features that, in our opinion, are important in future VLSI designs. Specifically, we show that a self-calibrating interconnect can save energy under some realistic workload and that the energy saving depends on the quality of the manufactured chips. We also show that, while the yield of a traditional system reduces when the design is done with optimistic assumptions on the noise sources, self-calibration allows a circuit to trade energy for robustness without reducing the yield.

We believe that this kind of dynamically self-calibrating techniques will be essential to exploit the potentials of future deep-submicron VLSI technologies.

ACKNOWLEDGMENT

The authors would like to thank Prof. Y. Leblebici for his precious help and advice.

REFERENCES

- [1] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay," in *Proc. 40th Design Automation Conf.*, Anaheim, CA, Jun. 2003, pp. 348–353.
- [2] T. Austin, D. Blaauw, T. Mudge, and K. Flautner, "Making typical silicon matter with razor," *Computer*, vol. 37, no. 3, pp. 57–65, Mar. 2004.
- [3] L. Benini and G. De Micheli, *Dynamic Power Management: Design Techniques and CAD Tools*. Norwell, MA: Kluwer, 2000.
- [4] —, "Networks on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.
- [5] 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 *Great Lakes Symp. VLSI*, Urbana, Ill., Mar. 1997.
- [6] L. Benini, T. Tao Ye, and G. De Micheli, "Networks-on-chips—efficient design of SoC interconnects," in *Low-Power Electronics Design*, C. Piguet, Ed. Boca Raton, FL: CRC Press, 2004, ch. 30.
- [7] D. Bertozzi, F. Poletti, L. Benini, and A. Bogliolo, *Design Automation of Embedded Systems*. Norwell, MA: Kluwer, 2003.
- [8] D. Bertozzi, L. Benini, and G. De Micheli, "Low-power error resilient encoding for on-chip data busses," in *Proc. Design, Automation, Test Conf. Exhibition*, Paris, France, Mar. 2002, pp. 102–109.
- [9] D. Bertozzi, L. Benini, and B. Riccò, "Energy-efficient and reliable low-switching signaling for on-chip busses based on redundant coding," in *IEEE Int. Symp. Circuits Systems*, Scottsdale, AZ, May 2002.
- [10] W. O. Cesário, D. Lyonnard, G. Nicolescu, Y. Paviot, S. Yoo, A. A. Jeraya, L. Gauthier, and M. Diaz-Nava, "Multiprocessor soc platforms: a component-based design approach," *IEEE Des. Test Comput.*, vol. 19, pp. 52–63, Nov.–Dec. 2002.
- [11] W. J. Dally and H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, pp. 466–755, Apr. 1993.
- [12] W. J. Dally and T. Brian, "Route packets, not wires: on-chip interconnection networks," in *Proc. 38th Design Automation Conf.*, Las Vegas, NV, Jun. 2001, pp. 684–689.
- [13] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
- [14] K. Flautner, S. Reinhardt, and T. Mudge, "Automatic performance setting for dynamic voltage scaling," in *Proc. 7th Conf. Mobile Computing Networking*, Rome, Italy, Jul. 2001, pp. 260–271.
- [15] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Proc. Design, Automation and Test Conf. and Exhibition*, Paris, France, Mar. 2000, pp. 250–256.
- [16] V. Gutnik and A. P. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 425–435, Dec. 1997.
- [17] R. Hegde and N. R. Shanbhag, "Toward achieving energy efficiency in presence of deep submicron noise," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 379–391, Aug. 2000.
- [18] —, "Soft digital signal processing," *IEEE Trans. VLSI Syst.*, vol. 9, pp. 813–823, Dec. 2001.
- [19] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," in *Proc. Asia and South Pacific Design Automation Conf.*, Kitakyushu, Japan, Jan. 2003, pp. 233–239.
- [20] F. Karim, A. Nguyen, S. Dey, and R. Rao, "On-chip communication architecture for oc-768 network processors," in *Proc. 38th Design Automation Conf.*, Las Vegas, NV, Jun. 2001, pp. 678–683.

- [21] H. Lekatsas and J. Henkel, "ETAM++: extended transition activity measure for low-power address bus designs," in *Asia and South Pacific Design Automation Conf.*, Bangalore, India, Jan. 2002.
- [22] L. Li, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Adaptive error protection for energy efficiency," in *Proc. Int. Conf. Computer-Aided Design*, San Jose, CA, Nov. 2003, pp. 2–7.
- [23] M. J. Jha Li Shang and L.-S. Peh, "Power-efficient interconnection networks: dynamic voltage scaling with links," *Comput. Architecture Lett.*, vol. 1, pp. 1–4, May 2002.
- [24] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE J. Solid-State Circuits*, vol. 29, pp. 663–670, Jun. 1994.
- [25] C. McNairy and D. Soltis, "Itanium 2 processor microarchitecture," *IEEE Micro*, vol. 23, pp. 44–55, Mar.–Apr. 2003.
- [26] E. Nilsson, "Design and implementation of a hot-potato switch in network on chip," Master's thesis, Laboratory of Electronics and Computer Systems, Royal Institute of Technology (KTH), Stockholm, Sweden, Jun. 2002.
- [27] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," in *Proc. 39th Design Automation Conf.*, New Orleans, LA, Jun. 2002, pp. 556–561.
- [28] T. Pering, T. Burd, and R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," in *Proc. Int. Symp. Low-Power Electronics and Design*, Monterey, CA, Aug. 1998, pp. 76–81.
- [29] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 2003.
- [30] S. Sakiyama, J. Kajiwara, M. Kinoshita, K. Satomi, K. Ohtani, and A. Matsuzawa, "An on-chip high-efficiency and low-noise DC/DC converter using divided switches with current control technique," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Paper*, San Francisco, CA, Feb. 1999, pp. 156–157.
- [31] P. P. Sotiriadis and A. Chandrakasan, "Low power bus coding techniques considering inter-wire capacitances," in *Proc. IEEE Custom Integrated Circuit Conf.*, Orlando, FL, May 2000, pp. 507–510.
- [32] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 49–58, Mar. 1995.
- [33] A. J. Stratakos, "High-Efficiency Low-Voltage DC-DC Conversion for Portable Applications," Ph.D. dissertation, Univ. of California at Berkeley, Berkeley, CA, 1998.
- [34] C. Svensson, "Optimum voltage swing on on-chip and off-chip interconnect," *IEEE J. Solid-State Circuits*, vol. 36, pp. 1108–1112, July 2001.
- [35] S. Tam, S. Rusu, U. Nagarji Desai, R. Kim, J. Zhang, and I. Young, "Clock generation and distribution for the first IA-64 microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1545–1552, Nov. 2000.
- [36] V. I. Varshavsky, Ed., *Self-Timed Control of Concurrent Processes*. Dordrecht, The Netherlands: Kluwer, 1990.
- [37] J. Walrand and P. Varaiya, *High-Performance Communication Networks*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2000.
- [38] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, 2nd ed. Reading, A: Addison-Wesley, 1993.
- [39] W. Wolf, *Computers as Components: Principles of Embedded Computer Systems Design*. San Mateo, CA: Morgan Kaufmann, 2001.
- [40] F. Worm, P. Jenne, P. Thiran, and G. De Micheli, "An adaptive low-power transmission scheme for on-chip networks," in *Proc. 15th Int. Symp. System Synthesis*, Kyoto, Japan, Oct. 2002, pp. 92–100.
- [41] T. Tao Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Proc. 39th Design Automation Conf.*, New Orleans, LA, Jun. 2002, pp. 524–529.
- [42] H. Zhang, V. George, and J. M. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 264–272, Jun. 2000.



Frédéric Worm received the M.S. degree from the School of Computer and Communication Sciences, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, where he is currently working toward the Ph.D. degree.

He is currently studying self-calibration techniques for networks-on-chips.



Paolo Jenne (S'90–M'96) received the Dottore in Ingegneria Elettronica degree from Politecnico di Milano, Milan, Italy, in 1991, and the Ph.D. degree from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland in 1996.

From 1990 to 1991, he was a Junior Researcher with Brunel University, Uxbridge, U.K. From 1992 to 1996, he was a Research Assistant at the Micro-computing Laboratory (LAMI) and at the MANTRA Center for Neuro-Mimetic Systems of the EPFL. In December 1996, he joined the Semiconductors

Group of Siemens AG in Munich, Germany (which is now Infineon Technologies AG). After working on datapath generation tools, he became Head of the embedded memory unit in the Design Libraries division. Since 2000, he has been a Professor at the EPFL and heads the Processor Architecture Lab (LAP). His research interests include various aspects of computer and processor architecture, reconfigurable computing, language-based VLSI design methodologies, and computer arithmetic.

Dr. Jenne was the recipient of the 40th Design Automation Conference Best Paper Award in 2003. He is also a member of the program committees of international workshops and conferences, including the International Conference on Computer Aided Design (ICCAD), Design Automation & Test in Europe (DATE), and the Workshop on Application Specific Processors (WASP).



Patrick Thiran (S'89–M'97) received the electrical engineering degree from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1989, the M.S. degree in electrical engineering from the University of California at Berkeley, CA, in 1990, and the Ph.D. degree from the Swiss Federal Institute of Technology Lausanne (EPFL) in 1996.

He became an Adjunct Professor in 1998, and an Assistant Professor in 2002, at the EPFL. From 2000 to 2001, he was with Sprint Advanced Technology Labs, Burlingame, CA. His research interests include

communication networks, in particular, performance evaluation and wireless networks, and in dynamical systems.

Dr. Thiran was an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I PART II: ANALOG AND DIGITAL SIGNAL PROCESSING from 1997 to 1999. In 1996, he was the recipient of the 1996 EPFL Ph.D. award.



Giovanni De Micheli (F'94) is a Professor in the Department of Electrical Engineering, and by courtesy, of The Department of Computer Science at Stanford University, Stanford, CA. He is author of *Synthesis and Optimization of Digital Circuits* (New York: McGraw-Hill, 1994), and co-author and/or co-editor of five books and over 270 technical articles. He has been a member of the technical advisory board of several companies, including Magma Design Automation, Coware, Aplus Design Technologies, Ambit Design Systems, and

STMicroelectronics. His research interests include several aspects of design technologies for integrated circuits and systems, with particular emphasis on synthesis, system-level design, hardware/software co-design and low-power design.

Dr. De Micheli was the recipient of the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems. He received the Golden Jubilee Medal for outstanding contributions to the IEEE Circuits and Systems Society in 2000. He received the 1987 D. Pederson Award for the best paper in IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN and two Best Paper Awards at the Design Automation Conference in 1983 and in 1993. He is a Past President of the IEEE Circuits and Systems Society. He was Editor in Chief of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN from 1987 to 2001. He was also the Program Chair and General Chair of the Design Automation Conference (DAC) in 1996–1997 and 2000, respectively. He is a Fellow of ACM.