Coding for System-on-Chip Networks: A Unified Framework

Srinivasa R. Sridhara, *Student Member, IEEE,* and Naresh R. Shanbhag, *Senior Member, IEEE*

***Abstract*—Global buses in deep-submicron (DSM) system-on-chip designs consume significant amounts of power, have large propagation delays, and are susceptible to errors due to DSM noise. Coding schemes exist that tackle these problems individually. In this paper, we present a coding framework derived from a communication-theoretic view of a DSM bus to jointly address power, delay, and reliability. In this framework, the data is first passed through a nonlinear source coder that reduces self and coupling transition activity and imposes a constraint on the peak coupling transitions on the bus. Next, a linear error control coder adds redundancy to enable error detection and correction. The framework is employed to efficiently combine existing codes and to derive novel codes that span a wide range of tradeoffs between bus delay, codec latency, power, area, and reliability. Using simulation results in 0.13-μm CMOS technology, we show that coding is a better alternative to repeater insertion for delay reduction as it reduces power dissipation at the same time. For a 10-mm 4-bit bus, we show that a bus employing the proposed codes achieves up to 2.17$\times$ speed-up and 33% energy savings over a bus employing Hamming code. For a 10-mm 32-bit bus, we show that 1.7$\times$ speed-up and 27% reduction in energy are achievable over an uncoded bus by employing low-swing signaling without any loss in reliability.**

***Index Terms*—Bus coding, bus delay, crosstalk avoidance, interconnection networks, low-power, on-chip buses, reliability, system-on-chip.**

I. Introduction

With shrinking of feature sizes, increasing die sizes, scaling of supply voltage, increasing interconnect density, and faster clock rates, global system-on-chip buses suffer from three major problems: 1) large propagation delay due to capacitive crosstalk [1]–[4] (*delay problem*); 2) high power consumption due to both parasitic and coupling capacitance [3], [5], [6] (*power problem*); and 3) increased susceptibility to errors due to deep-submicron (DSM) noise [7], [8] (*reliability problem*). Fig. 1 from the 2003 International Technology Roadmap of Semiconductors (ITRS) [1] shows the delay scaling trend with technology. While gate delay reduces with scaling, global wire delay increases. Therefore, delay of global buses will act as the performance bottleneck in many high-performance system-on-chip (SOC) designs. Interconnection networks consume 20%–36% of total system power in many large SOCs [9]. Future SOCs will follow the network-on-chip (NOC) paradigm [10], where high-speed energy-efficient reliable communication between various SOC components is vital. Thus, delay, power, and reliability are the three problems that need to addressed in the design of on-chip buses. Coding techniques have been proposed to alleviate each of these problems individually as described next.

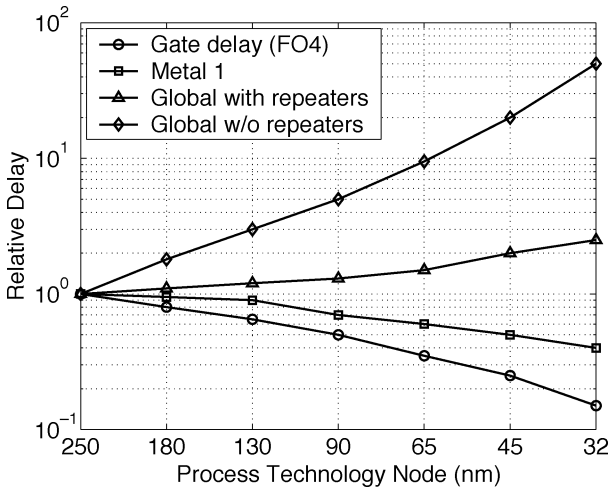


Fig. 1. Gate and wiring delay versus feature size [1].

For on-chip buses, low-power codes (LPC) were first employed to reduce transition activity resulting in low-power buses [11], [12]. However, these schemes ignored coupling capacitances, which result in delay and power penalty. Codes that reduce both self and coupling transitions were then proposed [5], [6], [13]. These codes addressed the power problem but did not address the delay problem. Crosstalk avoidance codes (CAC) that reduce the delay by forbidding specific transitions were recently proposed [13]–[16]. Though crosstalk between adjacent wires is addressed by CAC, other forms of DSM noise such as power grid fluctuations, crosstalk from other interconnects, electromagnetic interference makes buses susceptible to errors. Further, the use of low-swing signaling aggravates the reliability problem. Error control coding (ECC) was proposed in [7] as a way to achieve energy efficiency in I/O signaling in the presence of DSM noise. This idea was applied to on-chip buses in [7].

Though, solutions based on techniques other than coding do exist for crosstalk prevention [4] and power reduction [13], [18], they are usually technology- and implementation-dependent. Coding provides an elegant alternative that is technology-independent. Further, coding can provide a common framework for jointly solving delay, power, and reliability problems. However, no solution exists today that addresses all three problems

Manuscript received July 16, 2004; revised November 30, 2004. This work was supported by the MARCO-sponsored Gigascale Systems Research Center and Intel Corporation.

The authors are with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: sridhara@uiuc.edu; shanbhag@uiuc.edu).

Digital Object Identifier 10.1109/TVLSI.2005.848816

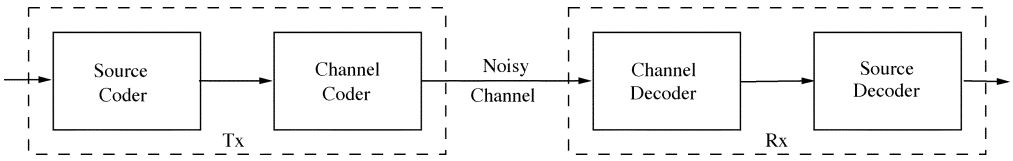


Fig. 2. Generic communication system.

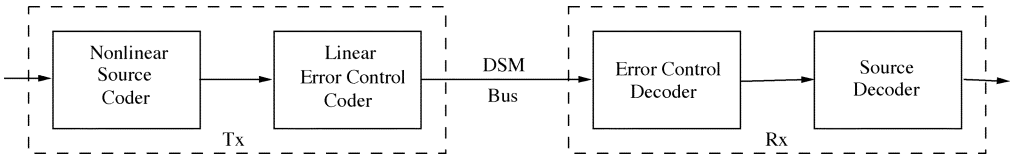


Fig. 3. Generic coding system for an on-chip DSM bus.

jointly. In this paper, we present a unified coding framework that provides practical codes to solve all three problems jointly.

Consider the generic communication system for data transmission over a noisy channel shown in Fig. 2. The source coder compresses the input data such that the number of bits required in the transmission is minimized. The channel coder adds redundancy to combat errors that may occur due to noise in the channel. In practical communication systems, channel coding is employed to reduce the signal power needed to achieve a required level of reliability.

The source coding framework in [11] is in fact based on the idea of viewing an on-chip bus as a communication channel. However, the channel was assumed to be noiseless. A DSM bus can be viewed as a noisy channel. A generic coding system for DSM buses is shown in Fig. 3. In [11], a source coder is employed to reduce self transition activity in buses. For closely coupled DSM buses, we can envisage the use of a source coder to not only reduce transition activity but also reduce average and/or peak coupling transitions. The reduction of average coupling transitions reduces power dissipation and reduction of peak coupling transitions reduces delay. Further, we can employ a error control scheme to combat errors that arise due to DSM noise. As in communication systems, error control coding allows for a tradeoff between the supply voltage (power consumption) and reliability.

We can employ this generic system to design optimum codes that achieve the best possible performance for a given amount of redundancy (additional wires in our case) following either graph-theoretic [15], [19] or information-theoretic approaches [7], [13]. However, such approaches provide us with bounds on the achievable results and are not, in general, useful for designing practical schemes.

Our contributions in this paper are as follows.

1) We propose a unified framework that employs practical components and enables seamless tradeoff between delay, power, and reliability.
2) We derive a wide variety of practical joint codes from the proposed framework.
3) We demonstrate the tradeoffs between delay, codec latency, power, reliability, and area achieved by the proposed codes.

Preliminary results from this work were presented in [20]. The rest of this paper is organized as follows. We begin with a review of the DSM bus model and existing codes in Section II. In Section III, we construct the framework and derive new codes from it. In Section IV, we make a detailed comparison of the codes by designing global buses in a 0.13-μm CMOS technology. We summarize and conclude in Section V.

II. Background

In this section, we review models for delay, energy, and reliability in DSM buses. Then, we present an overview of existing coding schemes and define our notation and terminology.

A. Bus Models

In this paper, we assume an n-bit parallel bus in a single metal layer. Further, we assume that rise time of the drivers and the loss in the interconnects are such that the inductance can be safely ignored [21]. Such DSM buses can be modeled as distributed *RC* networks with coupling capacitance between adjacent wires.

1) Delay Model: The delay of wire l of the bus is given by [13]

$$T_l = \begin{cases} \tau_0\left[(1+\lambda)\Delta_1^2 - \lambda\Delta_1\Delta_2\right], & l=1 \\ \tau_0\left[(1+2\lambda)\Delta_l^2 - \lambda\Delta_l(\Delta_{l-1}+\Delta_{l+1})\right], & 1<l<n \\ \tau_0\left[(1+\lambda)\Delta_n^2 - \lambda\Delta_n\Delta_{n-1}\right], & l=n \end{cases} \tag{1}$$

where τ_0 is the delay of a crosstalk-free wire, λ is the ratio of the coupling capacitance to the bulk capacitance, and Δ_l is the transition occurring on wire l, where Δ_l is equal to 1 for 0-to-1 transition, -1 for 1-to-0 transition, and 0 for no transition.

2) Energy Model: The average dissipated energy per bus transfer depends on data statistics and is given by [13]

$$\mathcal{E} = tr(C_T\mathcal{A})V_{dd}^2 \tag{2}$$

where $tr(X)$ is the trace of the matrix X, V_{dd} is the supply voltage, and C_T is a $n \times n$ capacitance matrix given by

$$C_T = \begin{bmatrix} 1+\lambda & -\lambda & 0 & \cdots & 0 \\ -\lambda & 1+2\lambda & -\lambda & \cdots & 0 \\ 0 & -\lambda & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & 1+2\lambda & -\lambda \\ 0 & 0 & \cdots & -\lambda & 1+\lambda \end{bmatrix} \cdot C \tag{3}$$

where the C is the total bulk capacitance of a wire, and $\mathcal{A}$ is the transition activity matrix whose elements are given by [13]

$$a_{i,j} = E\left[u_i^b u_j^b\right] - \frac{E\left[u_i^b u_j^a\right] + E\left[u_j^b u_i^a\right]}{2} \tag{4}$$

where $E[\cdot]$ is the expectation operator and u^b and u^a are data vectors on the bus before and after a transition, respectively.

3) Error Model: Errors occur in DSM buses due to power grid fluctuations, electromagnetic interference, crosstalk from other interconnects, or particle hits [8]. An accurate characterization of the error phenomena due to DSM noise is difficult as it requires knowledge of various noise sources and their dependence on physical and electrical parameters. Therefore, we assume that a Gaussian distributed noise voltage V_N with variance σ_N^2 is added to the signal waveform to represent the cumulative effect of all noise sources [7], [17]. Then, the probability of bit error is given by

$$\varepsilon = Q\left(\frac{V_{dd}}{2\sigma_N}\right) \tag{5}$$

where $Q(\cdot)$ is given by

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}} dy. \tag{6}$$

Assuming independent bit errors and low probability of bit error, the probability of word error for k-bit uncoded bus is given by

$$P_{\text{unc}}(\varepsilon) = k\varepsilon. \tag{7}$$

B. Low-Power Coding

The power dissipation in the bus depends on data transition activity. Early low-power coding schemes minimized the self transition activity. A general framework for low-power coding through self transition activity reduction was presented in [11]. We refer to codes that reduce the average transition activity as low-power codes (LPCs). A simple but effective LPC is the bus-invert code [12] in which the data is inverted and an invert bit is sent to the decoder if the current data word differs from the previous data word in more than half the number of bits. The effectiveness of bus-invert coding decreases with increase in the bus width. Therefore, for wide buses, the bus is partitioned into several sub-buses each with its own invert bit. In this paper, we denote bus-invert codes as $\text{BI}(i)$, where i is the number of sub-buses.

In DSM buses, both self transitions and coupling transitions contribute to the power dissipation. The idea of bus-invert coding has been extended for DSM buses [5], [6] by conditionally inverting the bus based on a metric that accounts for both self and coupling transitions. These codes require significant increase in complexity and overhead. Transition pattern code proposed in [13] employs nonlinear mapping from data to codeword to achieve significant self and coupling activity reduction in DSM buses but requires very complex encoders and decoders limiting its application.

Note that bus-invert coding is nonlinear. It has been shown in [13] that linear codes do not reduce transition activity.

C. Crosstalk Avoidance Coding

The delay of a wire in the bus depends on the transitions on the wire and wires adjacent to it. From (1), the worst-case delay of a wire is $(1+4\lambda)\tau_0$. The purpose of the crosstalk avoidance coding is to limit the worst-case delay to $(1+2\lambda)\tau_0$.

Crosstalk avoidance codes (CACs) proposed in [15] reduce the worst-case delay by ensuring that a transition from one codeword to another codeword does not cause adjacent wires to transition in opposite directions. We refer to this condition as forbidden transition (FT) condition. Shielding the wires of a bus by inserting grounded wires between adjacent wires is the simplest way to satisfy this condition. A forbidden transition code (FTC) that requires fewer wires that shielding has been proposed in [15]. In Appendix I, we show that there is no linear code that satisfies the FT condition while requiring fewer wires than shielding.

The worst-case delay can also be reduced to $(1+2\lambda)\tau_0$ by avoiding bit patterns "010" and "101" from every codeword [14]. We refer to this condition as forbidden pattern (FP) condition. The simplest method to satisfy the FP condition is to duplicate every data wire. In Appendix I, we show that there is no linear forbidden pattern code (FPC) that satisfies the FP condition while requiring fewer wires than duplication.

D. Error Control Coding

Error control is possible if the Hamming distance between any two codewords in the codebook is greater than one [22]. If the minimum Hamming distance between any two code words is two, then all single errors appearing on the bus can be detected. If the minimum Hamming distance is three, then all single errors can be corrected. Error detection is simpler to implement than error correction but requires retransmission of the data when an error occurs.

In this paper, we focus on linear and systematic error correcting codes (ECCs). In systematic codes, a few redundant bits are appended to the input bits to generate the codeword. Hamming code [22] is an example of a linear systematic error correcting code. Hamming code for k bits involves adding m parity bits such that $k \leq 2^m - m - 1$ [22]. Thus, the wiring overhead increases as $\log_2 k$. For low probability of bit error ε, the residual probability of word error P_{ham} can be approximated by

$$P_{\text{ham}}(\varepsilon) = \binom{k+m}{2}\varepsilon^2. \tag{8}$$

It is clear that LPC, CAC, and ECC address power, delay, and reliability individually. In Section III, we propose a unified framework from which codes are derived that jointly optimize power, delay, and reliability.

III. Unified Coding Framework

LPC, CAC, and ECC can be combined into a system as shown in Fig. 3 if the following conditions are satisfied.

1) CAC needs to be the outermost code as, in general, it involves nonlinear and disruptive mapping from data to codeword.
2) LPC can follow CAC as long as LPC does not destroy the peak coupling transition constraint of CAC.

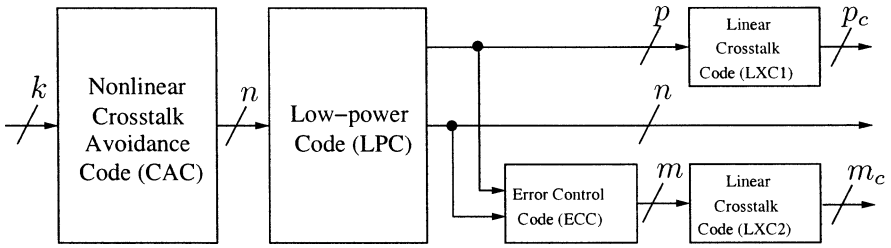


Fig. 4. Unified coding framework.

TABLE I
CODES DERIVED FROM THE PROPOSED FRAMEWORK

Category	Coding Scheme	CAC	LPC	ECC	LXC1	LXC2
LPC	BI(1)	–	BI(1)	–	–	–
	BI(8)	–	BI(8)	–	–	–
CAC	Shielding	Shielding	–	–	–	–
	FTC	FTC	–	–	–	–
ECC	Hamming	–	–	Hamming	–	
	HammingX	–	–	Hamming	–	Half-shielding
LPC+ECC	**BIH**	–	BI(1)	Hamming	–	–
CAC+ECC	**FTC+HC**	FTC	–	Hamming	–	Shielding
	DAP	Duplication	–	Parity	–	–
	DAPX	Duplication	–	Parity	–	Duplication
All	**DAPBI**	Duplication	BI(1)	Parity	Duplication	–

3) The additional information bits generated by LPC need to be encoded through a linear CAC to ensure that they do not suffer from crosstalk delay.
4) ECC needs to be systematic to ensure that the reduction in transition activity and the peak coupling transition constraint are maintained.
5) The additional parity bits generated by ECC need to be encoded through a linear CAC to ensure that they do not suffer from crosstalk delay.

A framework satisfying the above conditions is shown in Fig. 4. LXC1 and LXC2 are linear crosstalk avoidance codes based on either shielding or duplication. Nonlinear CACs can not be used because error correction has to be done prior to any other decoding at the receiver. In Fig. 4, a k-bit input is coded using CAC to get an n-bit codeword. The n-bit codeword is further encoded to reduce the average transitions through LPC resulting in p additional low-power information bits. ECC generates m parity bits for the $n+p$ code bits. The m parity bits and p low-power bits are further encoded for crosstalk avoidance to obtain m_c and p_c bits, respectively, that are sent over the bus along with n code bits. The total number of wires required to encode a k-bit bus is $(n+p_c+m_c)$, resulting in a code rate of $k/(n+p_c+m_c)$. In this paper, we assume k to be even.

In the remainder of this section, we develop a variety of codes based on the proposed unified framework that allow for tradeoff between delay, power, area, and reliability. The codes and their components are listed in Table I. Some of the known codes are also listed for comparison. The new codes are shown in bold in Table I and the remainder of the paper.

A. Joint LPC and CAC

Combining LPC and CAC codes is a hard problem as both are nonlinear codes and, even when such a combination is possible, the resulting code is complex. For example, it is not possible to combine bus-invert coding with FTC as inverting an FTC codeword destroys its crosstalk avoidance property. However, we show in Section IV-B that FTC reduces the average coupling power dissipation as it avoids the high power-consuming opposing transitions on adjacent wires. Thus, FTC codes can independently be used for crosstalk avoidance and low-power.

FPC can be combined with bus-invert based LPC schemes. This is because inverting an FPC codeword maintains the FP condition. In a joint code with FPC and bus-invert based LPC, the invert bits are encoded using a duplication (LXC1) code to avoid crosstalk delay in the invert bits. The joint code is a concatenation of the two component codes and no further optimization is possible. The codec overhead of the joint code is sum of codec overheads of the component codes and, hence, the joint code is complex.

B. Joint LPC and ECC

A joint low-power and error-correcting code can be obtained by adding parity information to the low-power coded data and low-power information bits. These codes are suitable for long DSM buses where reduction of power consumption is important but voltage scaling is not possible due to the presence of DSM noise. In joint low-power and error-correction coding, LPC reduces the transition activity on the bus while ECC allows for voltage scaling without lowering the reliability requirement. While it is possible to combine any low-power code with an error correcting code according to the framework, the total coding delay will equal the sum of the individual coding delays resulting in a large delay or, in case of pipelined systems, increased latency. For example, the encoder delay of the concatenation of BI and Hamming codes in Fig. 5(a) will equal the sum of the individual coding delays. Here, we propose a way of

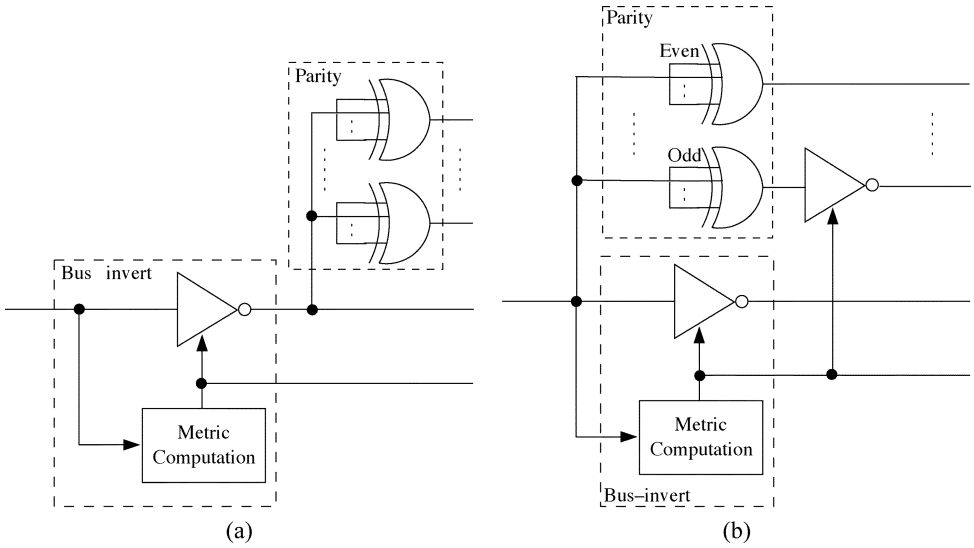


Fig. 5. Joint LPC and ECC: (a) Concatenation of BI and Hamming codes and (b) bus-invert Hamming (**BIH**) code with reduced encoder delay.

reducing this delay for the important class of BI-based LPC and parity-based ECC.

In BI-based LPC, the data bits are conditionally inverted based on a metric. Therefore, the inputs to ECC in Fig. 4 are either the original data bits or their complement. In parity-based ECC schemes, different parity bits are generated through XORing different subsets of the input bits. These subsets may contain even or odd number of input bits. We use the following property of XOR operation to reduce the total delay of the joint code.

Property of XOR: If an odd (even) number of the inputs of an XOR gate are inverted, then the output is inverted (unchanged).

In the proposed scheme shown in Fig. 5, we determine the parity bits of the ECC using the original data bits, instead of waiting for invert bits of the LPC to be computed. Once the invert bits are computed, the parity bits resulting from odd number of input bits are conditionally inverted using the invert bits. Thus, parity generation and invert bit computation can occur in parallel reducing the total delay to the maximum of the two and the delay of an inverter. Though decoding still occurs serially, the decoding delay of the joint code is not significantly higher as bus-invert decoding involves just conditionally inverting the received bits using the invert bits. The joint code that results from such a combination of bus-invert code BI(1) and Hamming code is referred to as bus-invert Hamming (**BIH**) code as listed in Table I. Encoder delays of both Hamming and BI codes are proportional to $\log_2 k$. Therefore, BIH achieves significant reduction in encoder delay. Estimates from gate level netlists indicate 21%–33% reduction in encoder delay.

C. Joint CAC and ECC

A joint crosstalk avoidance and error-correction code can be obtained by combining a crosstalk avoidance code with an error-correcting code. The parity bits of the ECC are further encoded for crosstalk avoidance using a linear crosstalk avoidance code LXC2. The LXC2 code is shielding if CAC satisfies FT condition and it is duplication if CAC satisfies FP condition. For global buses, the reduction in bus delay due to

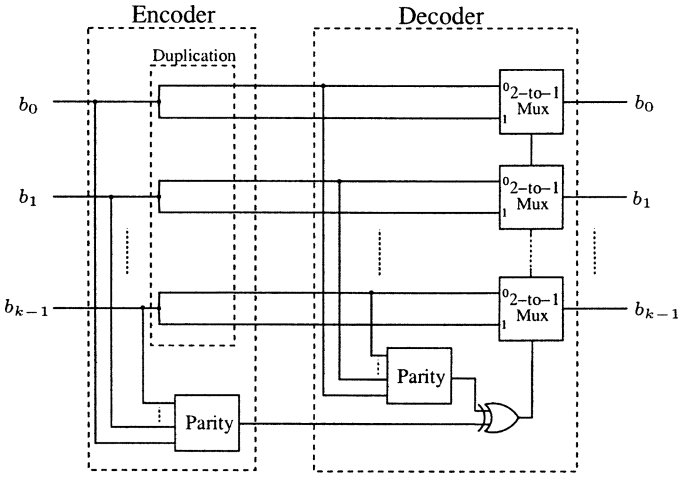


Fig. 6. Joint CAC and ECC: Duplicate-add-parity (DAP).

CAC can be greater than codec latency of the joint code. Thus, the resulting buses can have lower latency than an uncoded bus. Therefore, these codes can be used as *zero or negative latency* ECCs. Note that the achieved latency depends on several factors including the process technology, bus length, bus width, etc. The increasing gap between logic delay and interconnect delay as seen in Fig. 1 will enable several of the proposed codes to act as *zero or negative latency* ECCs in future technologies.

For example, **FTC+HC** combines FTC(4,3) and Hamming codes with shielding as LXC2 as listed in Table I. However, the coding overhead will be significant as the joint code is a concatenation of the two individual codes. Here, we propose a code that has significantly lower overhead.

Consider the duplication scheme for avoiding crosstalk delay. This code has a Hamming distance of two as any two distinct codewords differ in at least two bits. We can increase the Hamming distance to three by appending a single parity bit. This code referred to as duplicate-add-parity (**DAP**) is shown in Fig. 6.

To decode, we recreate the parity bit by using one set of the received data bits and compare that with received parity bit. If

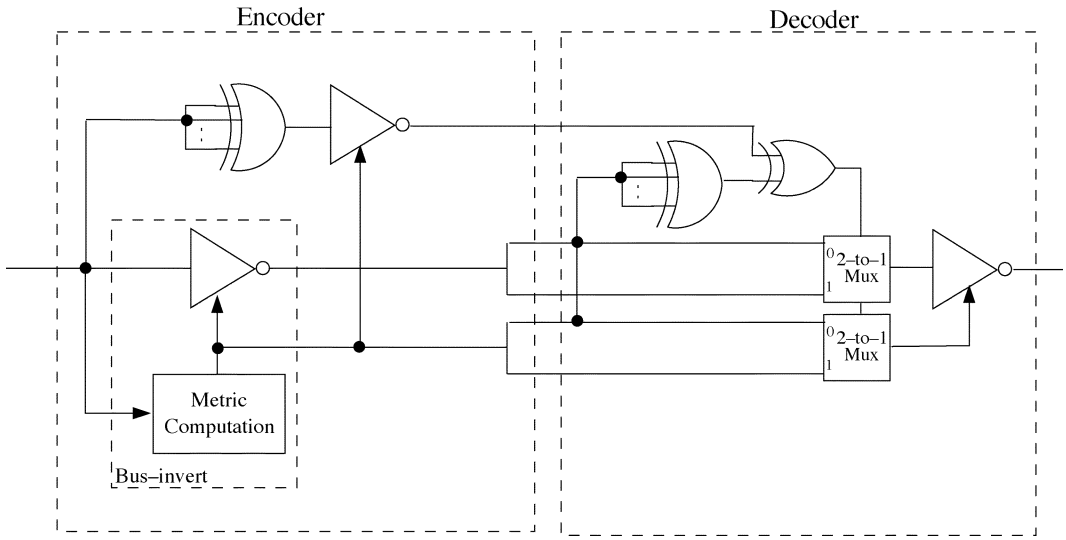


Fig. 7. Joint LPC, CAC, and ECC: Duplicate-add-parity bus-invert (DAPBI).

the two match, the set of bits used to recreate the parity bit is chosen as the output, else the other set is chosen as shown in Fig. 6. Since a single error will at most affect one of the sets or the parity bit, it is correctable. Note that the **DAP** code is similar to boundary shift code (BSC) [19], which is based on the FT condition, but has better performance as shown in Section IV.

For low probability of bit error ε, the residual probability of word error $P_{\rm dap}$ of a k-bit bus is given by (see Appendix II)

$$P_{\rm dap}(\varepsilon) = \frac{3k(k+1)}{2}\varepsilon^2. \tag{9}$$

D. Joint LPC, CAC, and ECC

We can combine all three component codes to arrive at a joint code that has low-power, crosstalk avoidance and error correction properties. However, only FPCn based CAC and bus-invert based LPC can be used. Further, the overhead due to the combination will be significant. Here, we consider the combination of the **DAP** code with bus-invert based LPC. The joint code, referred to as duplicate-add-parity bus-invert (**DAPBI**) code, is shown in Fig. 7. As this code uses bus-invert based LPC and parity based ECC, we employ the technique described in Section III-B to reduce the encoder delay. Further, the invert bit is duplicated (LXC1) to ensure error-correction and crosstalk avoidance for the bit.

E. Encoder Delay Masking in Systematic Codes

Based on the unified framework, we describe a technique to eliminate encoder delay of systematic codes at the expense of additional wires. In Fig. 4, n bits from the output of LPC are transmitted without any modification and p_c parity bits generated from the concatenation of ECC and LXC2 are also transmitted. Therefore, the ECC encoder delay only slows down the parity bits but still has an impact on the overall delay of the bus. In the joint codes proposed in previous sections, LXC2 is employed to ensure that p_c parity bits have bus delay identical to n output bits. Here, we observe that the ECC encoder delay can be eliminated by using LXC2 to make bus delay of

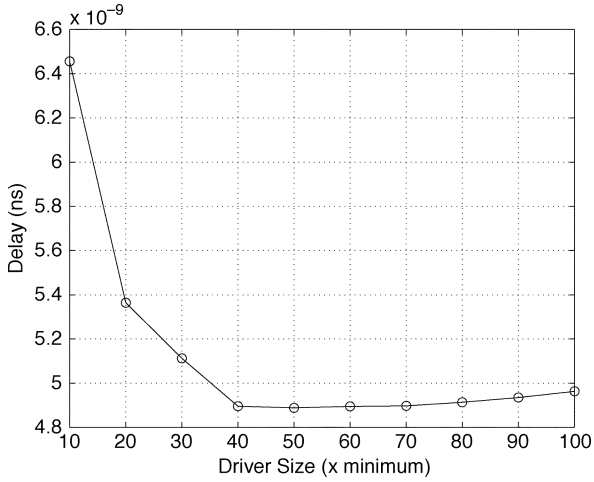


Fig. 8. Worst-case delay of a 10-mm 3-bit bus as a function of the driver size.

the p_c parity bits smaller than the n output bits. Since, parity bits are few in number, we can employ a wide variety LXC2 codes such as half-shielding, shielding, duplication, and duplication & shielding.

In this paper, we propose two such codes. First, we consider the Hamming code. The data bits in a Hamming code do not have crosstalk avoidance and, therefore, have a bus delay of $(1+4\lambda)\tau_0$. The parity bits obtained from the Hamming encoder also have $(1+4\lambda)\tau_0$ bus delay but, in addition, have Hamming encoder delay. Therefore, we can eliminate the encoder delay for long buses by using half-shielding as LXC2 and reducing the bus delay for parity bits to $(1+3\lambda)\tau_0$. We refer to such a code as **HammingX**.

Next, we consider the **DAP** code proposed in Section III-C. From Fig. 6, we observe that the delay from the parity generator of the encoder affects only the parity bit. Therefore, we can eliminate this delay by using duplication as LXC2. Since there is a single parity bit, this results is only one additional wire. LXC2 reduces the bus delay of the parity bit to $(1+\lambda)\tau_0$. For long buses, the reduction in bus delay can completely mask the encoder delay. We refer to such a code as **DAPX**.

TABLE II
CODE COMPARISON FOR A RELIABLE 4-BIT BUS

Coding Scheme	Bus			Codec			Area overhead (%)
	No. of wires	Delay ($\times\tau_0$)	Average energy ($\times CV_{dd}^2$)	Area (μm^2)	Delay (ps)	Average energy (pJ)	
Hamming	7	$1+4\lambda$	$1.75+3.00\lambda$	792	448	1.16	0.00
HammingX	8	$1+4\lambda$	$1.75+3.00\lambda$	792	306	1.16	14.9
BIH	9	$1+4\lambda$	$1.78+3.25\lambda$	1764	748	2.45	33.5
FTC+HC	14	$1+2\lambda$	$2.09+3.20\lambda$	1390	636	1.10	107
BSC	9	$1+2\lambda$	$2.25+2.00\lambda$	725	567	0.83	29.6
DAP	9	$1+2\lambda$	$2.25+2.00\lambda$	412	465	0.50	28.4
DAPX	10	$1+2\lambda$	$2.50+2.00\lambda$	412	311	0.50	43.4
DAPBI	11	$1+2\lambda$	$1.81+1.75\lambda$	1390	709	1.86	61.9

IV. SIMULATION RESULTS

In this section, we present simulation results to demonstrate the improvement in delay, power, and reliability achieved by employing the proposed codes. The achieved improvements vary with bus length L, ratio of coupling capacitance to bulk capacitance λ, bus width k, and the process technology. We quantify the improvements in a standard 0.13-μm CMOS technology for various values of L, λ, and k.

The codecs, i.e., encoders and decoders, are synthesized using a 0.13-μm CMOS standard cell library and optimized for speed. The overhead required in terms of area, delay, and energy dissipation is obtained from synthesized gate level netlists. The codecs employ a nominal supply voltage $V_{dd} = 1.2$ in order to ensure reliable coding and decoding operations.

We consider a metal 4 bus of length L with minimum width of 0.2 μm and minimum spacing of 0.2 μm. Note that each code has a minimum bus length L_{crit} below which the codec overhead nullifies the benefits of employing the code. In this paper, we consider global buses and point out codes for which L_{crit} is in the range of 6 mm to 14 mm.

For a given bus geometry, the value of λ depends on the metal coverage in upper and lower metal layers [2], [3]. We vary λ between the following two extreme scenarios. First, 100% metal coverage is assumed in metal layers 3 and 5, resulting in $\lambda = 0.95$. Second, all the bulk capacitance is assumed to be from metal 4 to the substrate, resulting in $\lambda = 4.6$.

The driver size is chosen to be 50$\times$ minimum size as it minimizes the worst-case delay of the middle wire of a 10-mm 3-bit bus as shown in Fig. 8. Delay and energy dissipation for various bus transitions are obtained using HSPICE [23]. The average energy per bus transfer is computed assuming that the data is spatially and temporally uncorrelated and that "0" and "1" are equally likely to appear.

We use speed-up, energy savings, and area overhead as the metrics of comparison. Speed-up of code 1 over code 2 is defined as

$$\text{speed}-\text{up} = \frac{T_{b2}+T_{c2}}{T_{b1}+T_{c1}} \tag{10}$$

where T_{bi} is the bus delay with code i and T_{ci} is the codec delay of code i. The energy savings and area overhead include codec energy dissipation and area, respectively.

We consider the design of two bus types. First, we design a reliable bus where ECC is required at nominal supply voltage in order to meet the reliability requirement. Next, we design a bus where reliability is traded off with energy efficiency by employing low-swing signaling and ECC. Signal swing is reduced such that there is no loss in reliability compared to an uncoded bus operating at the nominal signal swing of V_{dd}.

A. Codes for Reliable Buses

We consider the design of a bus that is prone to DSM noise and, hence, susceptible to errors. A reliable bus can be designed by employing a Hamming code to detect and correct occasional errors. However, Hamming code increases bus energy dissipation and adds codec latency to bus delay. Thus, the joint codes can be employed.

We begin by considering the design of a reliable 4-bit bus. Table II lists number of wires, delay, and average energy of the 4-bit bus employing codes that provide reliability. The corresponding codec overheads in 0.13-μm technology are also listed. The total area overhead over Hamming coded bus for $L = 10$ mm is also shown.

DAP and **DAPX** have the least codec area and energy overhead among all codes. **HammingX** has the least codec delay. **DAPBI** has the least bus energy dissipation as it leads to substantial reduction in the coupling component of energy. We observe that **BIH** and **FTC+HC** perform worse than Hamming code and **DAP**, respectively, in all metrics. **BIH** performs worse than Hamming as benefits of activity reduction are nullified by the addition of extra parity bit required to encode the data and invert bits. **FTC+HC** has significantly large codec and area overhead compared to **DAP**.Therefore, we exclude **BIH** and **FTC+HC** from further comparison.

Note that **DAPX** has lower codec delay than Hamming code, while **DAP** has slightly larger codec delay. Further, both these codes reduce bus delay to $(1+2\lambda)\tau_0$. As we show next, these codes achieve significant speed-up over Hamming code for long buses. In fact, they can be used as reliable codes with zero or negative latency compared to the uncoded bus.

Fig. 9(a) plots the speed-up as a function of λ for a 10-mm bus. We see that **HammingX** provides a constant speed-up of 1.03 as it has no impact on bus delay but reduces the coding delay of Hamming code by 32% by masking the encoder delay using half-shielding. The other codes combine CAC with ECC and, hence, reduce bus delay and achieve significant speed-up that increases as a function of λ. Though **DAP** and BSC [19] have the same bus delay, **DAP** provides higher speed-up than BSC as it has 18% lower codec delay as listed in Table II. **DAPX** provides a further 33% reduction in codec delay over **DAP** by masking the encoder delay of **DAP**. **DAPX** achieves the largest speed-up in the range of 1.79–2.13.

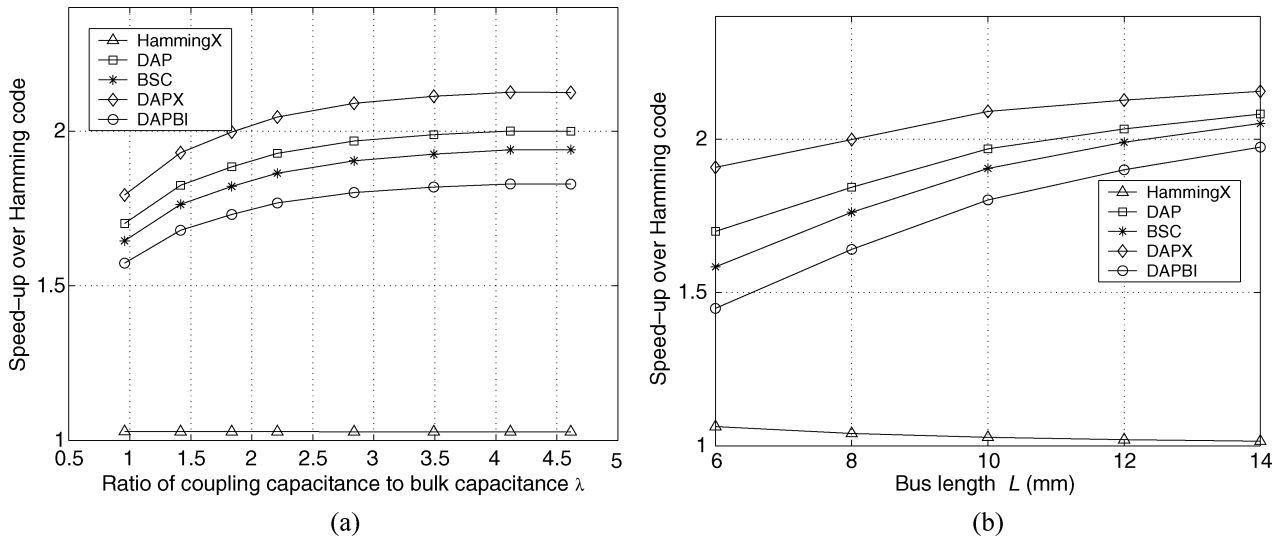


Fig. 9. Speed-up over Hamming code for a 4-bit bus: (a) as a function of λ at $L = 10$ mm and (b) as a function of L at $\lambda = 2.8$.

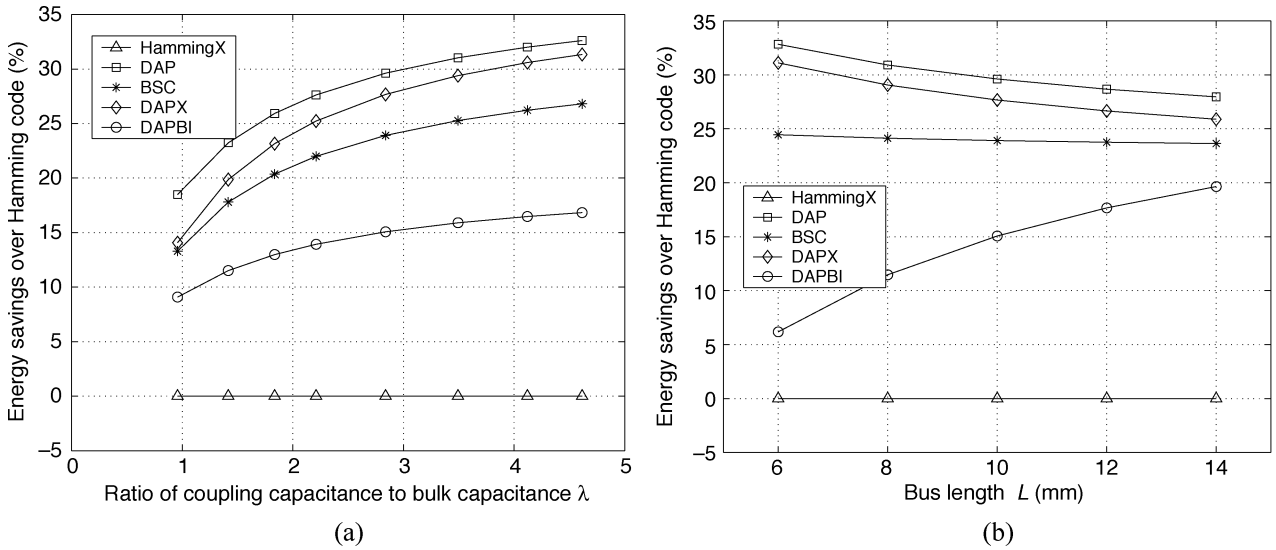


Fig. 10. Energy savings over Hamming code for a 4-bit bus: (a) as a function of λ at $L = 10$ mm and (b) as a function of L at $\lambda = 2.8$.

The achieved speed-up over Hamming code improves with increasing L for all codes except **HammingX** as seen in Fig. 9(b). This is because codec delay accounts for a smaller portion of the total delay for longer bus lengths and, hence, codes that reduce bus delay achieve higher speed-ups. However, **HammingX** does not change the bus delay but simply masks the encoder delay. Therefore, benefits of delay masking reduce with increasing L.

Fig. 10(a) plots the energy savings achieved over Hamming code as a function of λ for $L = 10$ mm. **DAP** provides higher savings than BSC as it has 40% lower codec energy dissipation. A bus coded with **DAP** has 18%–33% energy savings over the Hamming coded bus. Though **DAPBI** has significantly lower bus energy dissipation than all other codes as shown in Table II, the effective energy savings after accounting for codec overhead are lower than **DAP** and **DAPX** because of high codec overhead in the current technology. We also observe that **HammingX** has the same energy dissipation as Hamming code.

Fig. 10(b) plots energy savings as a function of L at $\lambda = 2.8$. For longer L, codec energy dissipation accounts for a smaller portion of the total energy dissipation. Since **DAP** has lower codec energy dissipation and **DAPBI** has lower bus energy dissipation, the gap between **DAP** and **DAPBI** reduces with increasing L. This indicates that **DAPBI** will be more effective in future as codec overhead will account for smaller portion of total delay and energy dissipation due to technology scaling.

Fig. 11 compares the codes across bus widths at $L = 10$ mm and $\lambda = 2.8$. The codec delay increases with k and accounts for a larger portion of the total delay. Therefore, in Fig. 11(a), we observe a trend opposite to Fig. 9(b). **HammingX** provides higher speed-ups at larger bus widths, while the other codes have smaller speed-ups. Increasing bus width has a more dramatic impact on energy savings as shown in Fig. 11(b). This is because the self-transition component of energy increases linearly with k for **DAP**-based codes while it increases logarithmically for Hamming codes. At $k = 64$, **DAPX** still

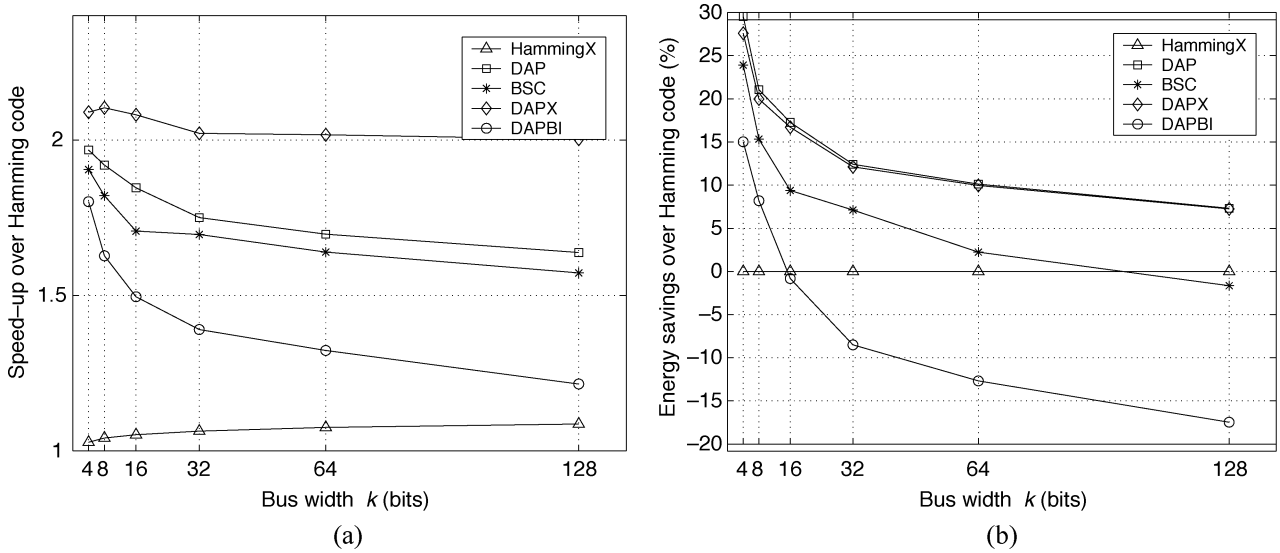


Fig. 11. Comparison across bus widths at $L = 10$ mm and $\lambda = 2.8$: (a) speed-up and (b) energy savings over Hamming code.

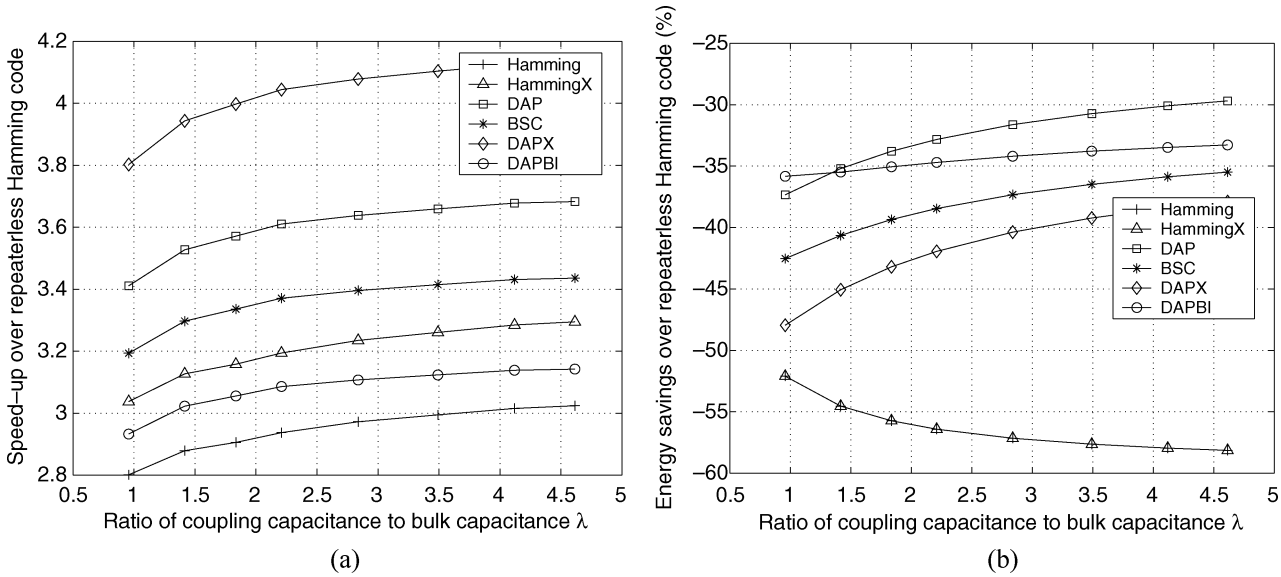


Fig. 12. Joint repeater insertion and coding: (a) speed-up and (b) energy savings over 4-bit 10-mm repeater-less Hamming coded bus.

provides 2.02× speed-up and 10% energy savings over Hamming code.

The above improvements are obtained at the cost of area overhead as shown in Table II. Clearly, Hamming code provides reliability with the least area requirement. However, the proposed codes **DAP** and **DAPX** provide speed-up and energy savings along with reliability but require area overheads of 28% and 43%, respectively.

Repeater insertion [24] is a well-known technique for delay reduction in long on-chip buses. We evaluate the performance of codes for 10-mm bus with repeaters inserted every 2 mm. The repeaters are sized so as to optimize the bus delay. The energy dissipation of repeaters is obtained from HSPICE. We employ repeater-less Hamming coded bus as the reference and compare the effect of joint repeater insertion and coding for the various codes.

Fig. 12 plots speed-up and energy savings over repeater-less Hamming code for 4-bit 10-mm bus as a function of λ. Note that we have also included repeater-inserted Hamming code in the plot to illustrate the effect of repeater insertion alone. Repeater insertion provides 2.97× speed-up for Hamming coded bus. However, this comes at the cost of 57% energy overhead as repeaters consume significant power to drive high bus capacitances. In contrast, **DAPX** provides 2.09× speed-up *and* 27% energy savings as shown in Fig. 11. Therefore, coding is a better alternative to repeater insertion for delay reduction as it reduces power dissipation at the same time. Though repeater insertion provides slightly higher speed-up in the current technology, the diverging trend between gate and interconnect delays seen in Fig. 1 will make coding superior to repeater insertion in the future.

Repeater-inserted **DAPX** has a speed-up of 4.08× compared to repeater-less Hamming coded bus. Clearly, coding and repeater insertion are complementary. However, the achieved speed-up comes at the increased energy dissipation as shown in Fig. 12(b). Energy overhead decreases for **DAP**-based codes for

TABLE III
CODE COMPARISON FOR A 32-BIT BUS

Coding Scheme	Bus				Codec			Area overhead (%)
	No. of wires	Delay ($\times \tau_0$)	Average energy ($\times CV_{dd}^2$)	V_{dd} (V)	Area (μm^2)	Delay (ps)	Average energy (pJ)	
Uncoded	32	$1+4\lambda$	$8.00+15.0\lambda$	1.2	0	0	0	0.00
BI(1)	33	$1+4\lambda$	$7.53+13.8\lambda$	1.2	7164	1233	10.9	8.8
BI(8)	40	$1+4\lambda$	$6.25+12.2\lambda$	1.2	7392	337	10.4	31.2
Shielding	63	$1+2\lambda$	$8.00+15.5\lambda$	1.2	0	0	0	98.4
FTC	53	$1+2\lambda$	$8.90+12.4\lambda$	1.2	1830	107	1.61	68.1
Hamming	38	$1+4\lambda$	$9.5+18.5\lambda$	0.884	6427	1005	9.9	24.1
HammingX	41	$1+4\lambda$	$9.5+18.5\lambda$	0.884	6427	660	9.9	33.7
BIH	39	$1+4\lambda$	$9.03+16.8\lambda$	0.884	13734	1952	21.0	33.1
FTC+HC	65	$1+2\lambda$	$10.4+15.4\lambda$	0.896	11180	1275	15.4	113.6
BSC	65	$1+2\lambda$	$16.3+16.0\lambda$	0.860	5978	1230	7.40	109.5
DAP	65	$1+2\lambda$	$16.3+16.0\lambda$	0.860	4016	1111	5.36	108.0
DAPX	66	$1+2\lambda$	$16.5+16.0\lambda$	0.860	4016	671	5.36	111.1
DAPBI	67	$1+2\lambda$	$14.4+14.2\lambda$	0.862	11228	1963	16.4	120.0

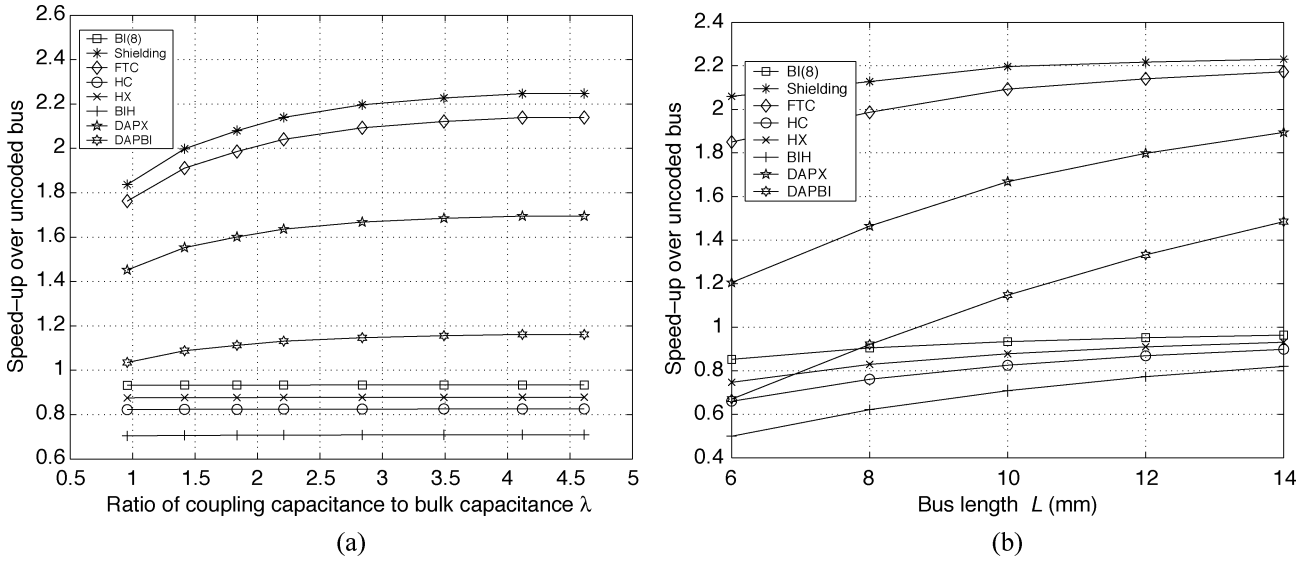


Fig. 13. Speed-up over 32-bit uncoded bus: (a) as a function of λ at $L = 10$ mm and (b) as a function of L at $\lambda = 2.8$.

large λ. Therefore, even for repeater-inserted buses, proposed codes provide reduction in energy dissipation.

In summary, Hamming code provides reliability with least area overhead but increases delay due to codec latency. **HammingX** eliminates encoder delay through encoder delay masking. Joint CAC and ECC codes provide speed-up and energy savings while providing the same level of reliability. Specifically, **DAP** and **DAPX** achieve significant improvements with manageable area overheads due to their efficient codecs. **DAPBI** does not provide significant benefits in current technology due to high codec overhead. However, it can be a suitable alternative in the future. Unlike repeater-insertion that provides speed-up at the cost of energy overhead, coding provides speed-up and energy savings. Further, coding provides improvements for both repeater-less and repeater-inserted buses.

B. Tradeoff Between Reliability and Energy Efficiency

In order to quantify the tradeoff between power dissipation and reliability, we assume that, due to DSM noise, voltage scaling below the nominal supply voltage $V_{dd} = 1.2$ V is not possible without lowering the reliability requirement. In this paper, we employ the error model described in Section II and assume that the probability of word error $P_{\text{unc}}(\varepsilon) = 10^{-20}$. If the residual probability of word error with ECC is $P_{\text{ecc}}(\varepsilon)$, then $P_{\text{ecc}}(\varepsilon) < P_{\text{unc}}(\varepsilon)$. Using (5), we can reduce the supply voltage to

$$\hat{V}_{dd} = V_{dd}\frac{Q^{-1}(\hat{\varepsilon})}{Q^{-1}(\varepsilon)} \tag{11}$$

such that $P_{\text{ecc}}(\hat{\varepsilon}) = P_{\text{unc}}(\varepsilon)$. P_{ecc} for Hamming and **DAP** codes are given by (8) and (9), respectively.

We begin by designing a 32-bit bus employing such a tradeoff. Table III lists delay, energy, and area required for various codes. Note that the bus energy dissipation is a function of $\hat{V}_{dd}$, which is lower than the nominal supply voltage $V_{dd} = 1.2$ V for codes employing ECC as shown in the table.

Fig. 13(a) plots the speed-up over the uncoded bus as a function of λ at $L = 10$ mm. Codes without a CAC have speed-up less than 1, indicating an effective slow-down due to codec delay. Codes with CAC have significant speed-up. **DAPX** code reduces the codec delay of **DAP** by 40% and achieves speed-up of 1.46–1.70. Shielding has highest speed-up as it has no codec

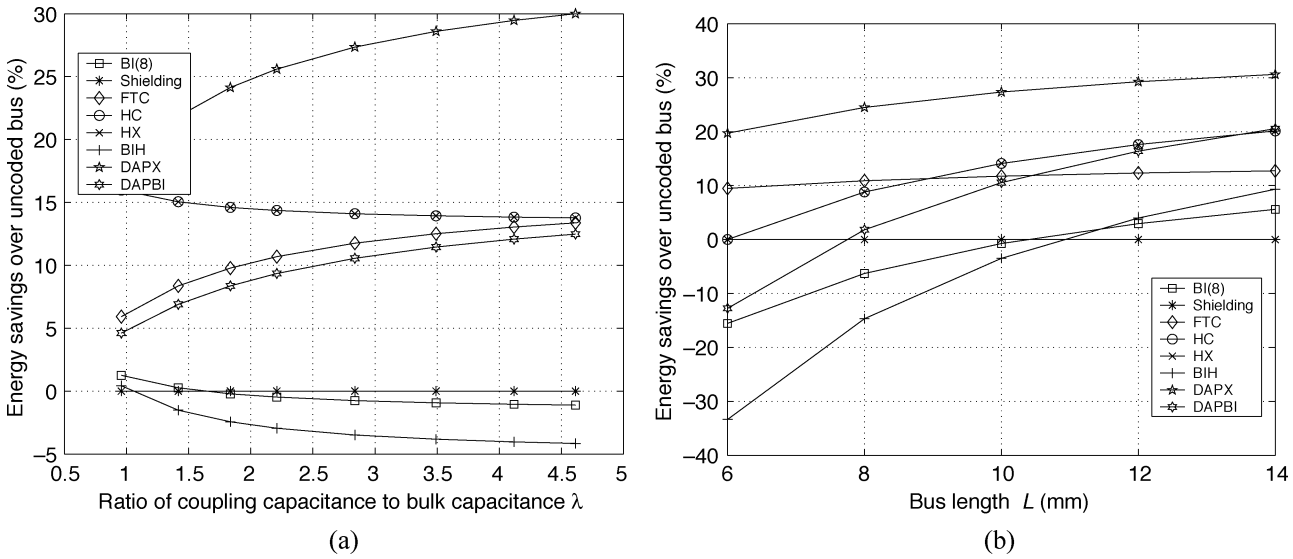


Fig. 14. Energy savings over 32-bit uncoded bus: (a) as a function of λ at $L = 10$ mm and (b) as a function of L at $\lambda = 2.8$.

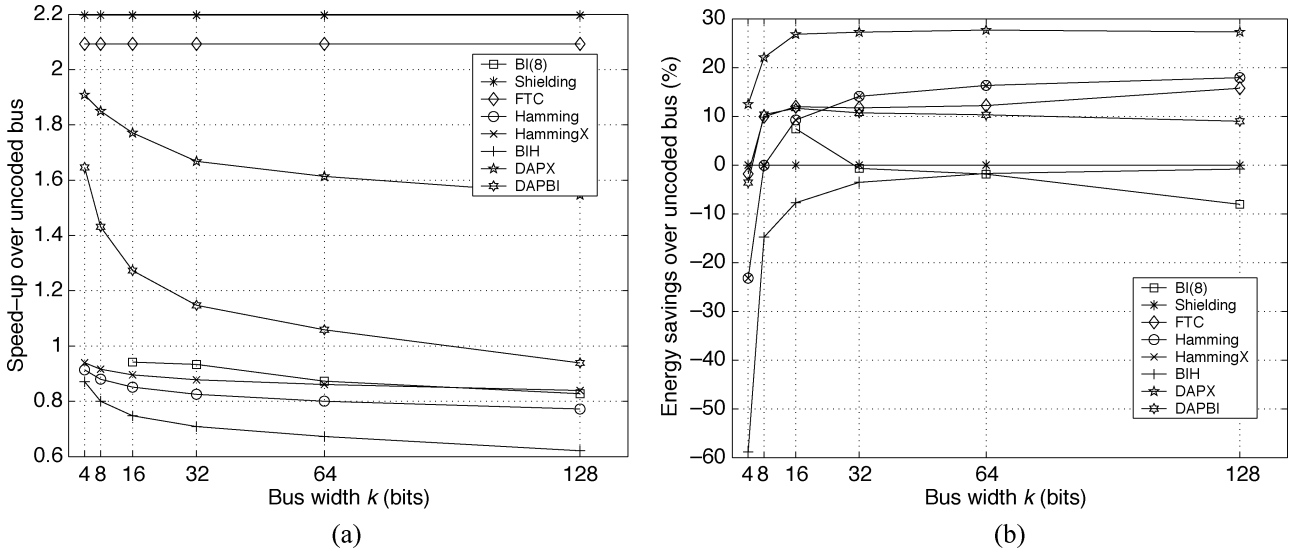


Fig. 15. Comparison across bus widths at $L = 10$ mm and $\lambda = 2.8$: (a) speed-up and (b) energy savings over uncoded bus.

delay. However, the proposed codes offer significant energy savings while shielding does not provide any energy savings.

Fig. 13(b) plots the speed-up over the uncoded bus as a function of L at $\lambda = 2.8$. For longer bus lengths, the coding delay accounts for a smaller portion of the total delay and, hence, speed-ups achieved by the codes improve. We also observe several codes with CAC have speed-up of less than 1 at $L = 6$ mm. This indicates that, in the current technology, $L_{\text{crit}} > 6$ mm for these codes.

Fig. 14(a) compares the energy savings over uncoded bus as function of λ at $L = 10$ mm. We observe that BI-based codes do not provide any energy savings due to high codec energy dissipation. However, codes with ECC provide significant energy savings by allowing a tradeoff between supply voltage and reliability. Further, **DAPX** code achieves energy savings 18%–30% due to its low codec overhead. Note that **DAP** and **DAPX** have higher energy savings than Hamming code and also provide significant speed-up as discussed earlier. From Table III, we see that **DAPBI** has lower bus energy dissipation than **DAPX**. However, in 0.13-μm technology, **DAPBI** has high codec energy dissipation and, hence, lower energy savings than **DAPX**. The energy savings improve with L as seen in Fig. 14(b).

Fig. 15(a) compares the speed-up across bus widths at $L = 10$ mm and $\lambda = 2.8$. Codec delay of a code with ECC increases logarithmically with k. Therefore, the achieved speed-ups for codes with ECC decrease with increasing k. Fig. 15(b) compares the energy savings for various k. **DAP**-based codes have no significant change in energy savings for $k \geq 16$ as both codec and bus energy dissipation increases linearly with k. However, codes with Hamming code have higher energy savings for larger bus width. This is because the number of parity wires increase logarithmically with k resulting lower bus energy overhead over uncoded bus at large k.

In summary, the tradeoff between reliability and energy efficiency provides significant energy savings. While Hamming code achieves this tradeoff with minimum area overhead,

the proposed **DAP**-based codes provide benefits in terms of speed-up and lower codec energy overhead at the cost of higher area overhead. Compared to shielding, **DAPX** code provides lower speed-up but allows the tradeoff between reliability and energy efficiency. **DAPX** also has higher speed-up and energy savings than BI-based codes. Thus, the proposed **DAP**-based codes provide the best tradeoff between bus delay, codec latency, power, area, and reliability.

V. Conclusions

We have derived a framework for joint design of codes for low-power, crosstalk avoidance, and error-correction in DSM buses. We have shown that schemes based on Hamming codes are more suitable for low-power than schemes based on bus-invert code for buses with DSM noise. We have shown that crosstalk avoidance codes based on forbidden transitions achieve power savings as well as bus speed-up. We have proposed the duplicate-add-parity code that combines crosstalk avoidance with error correction resulting in codes that are low-power, high-speed and noise-tolerant.

The proposed codes tradeoff delay and power dissipation in the bus with delay and power dissipation in the codec. This tradeoff will be increasingly favorable in future technologies due to the increasing gap between gate delay and interconnect delay brought about by shrinking feature sizes and due to the longer bus lengths brought about by bigger die sizes. Therefore, coding schemes that result in low bus delay and energy such as **BIH**, **DAPBI**, and **FTC+HC** will become more effective in the future.

All error correction codes considered in this paper have single error capability. With aggressive supply scaling and increase in DSM noise, more powerful error correction schemes may be needed to satisfy the reliability requirement. Multiple error correction codes such as Bose–Chaudhuri–Hocquenghem (BCH) [22] can be employed in such situations. BCH codes have more complex codecs than Hamming code and codec overhead will be a concern for implementation. Further, efficient joint codes that provide crosstalk avoidance, activity reduction, and multiple error correction need to be discovered in order to meet the need for high-speed energy-efficient reliable on-chip communication.

Appendix I
Nonlinearity of Crosstalk Avoidance Codes

Theorem 1: There is no linear crosstalk avoidance code that satisfies the FT (FP) condition while requiring fewer wires than shielding (duplication).

Proof: A binary code is linear if and only if the modulo-2 sum of two codewords is also a codeword [22]. Consider a linear CAC that satisfies the FT condition. In such a code, a codeword having 01 pattern at a given position cannot transition to another codeword with 10 at that position. Therefore, the codeword with 01 pattern can only transition to codewords with either 00 or 11 at that position. However, modulo-2 sum of 01 and 11 is 10, which is not allowed at that position as it violates the FT condition. Therefore, only other pattern allowed is 00. Similarly, a codeword with 10 pattern can only transition to a codeword with 00 pattern. Hence, bits on either side of a transitioning bit in a linear CAC must remain at 0. In other words, a transitioning wire must be shielded (grounded wires on either side). Hence, there is no linear CAC that satisfies the FT condition while requiring fewer wires than shielding.

Consider a linear CAC satisfying the FP condition. Codewords in the codebook cannot have 010 and 101 patterns. Thus, codewords are restricted to 000, 001, 011, 100, 110, and 111 patterns. Now, some of these patterns cannot overlap as their modulo-2 sum results in either 010 or 101 patterns. Specifically, 001 and 110 cannot transition to 011 and 100 and vice versa. Therefore, every bit location in the codebook has either {000, 001, 110, 111} or {000, 011, 100, 111} patterns. The first set has the two identical bits in the beginning in all its elements and the second set has two identical bits in the end. Therefore, every bit has to be duplicated. Hence, there is no linear CAC that satisfies the FP condition while requiring fewer wires than duplication. ■

Appendix II
Residual Probability of Error for DAP

From Fig. 6, **DAP** decoder receives two sets of inputs. Let the two sets be $\mathcal{A}$ and $\mathcal{B}$. The decoder recreates the party bit using $\mathcal{A}$ and compares the regenerated parity bit with the received parity bit to obtain the selector of multiplexer. If the selector evaluates to 0, $\mathcal{A}$ is chosen as the final output. Otherwise, $\mathcal{B}$ is chosen.

We compute the residual probability of error P_{dap} by first computing the probability of error-free decoding. This is possible if at least one of the sets has no errors. If $\mathcal{A}$ is error-free, then the received parity bit also needs to be error-free to ensure that $\mathcal{A}$ is selected as the decoded output. In such a case, error-free decoding is possible in the presence of zero or more errors in $\mathcal{B}$. For k-bit bus, the probability of error-free decoding, with no errors in $\mathcal{A}$, is

$$P_A = \sum_{i=0}^{k} \binom{k}{i} \varepsilon^i (1-\varepsilon)^{2k+1-i} \tag{12}$$

where ε is the probability of bit error.

If $\mathcal{B}$ is error-free, then the selector of multiplexers has to evaluate to 1 in order to achieve error-free decoding. The selector is the result of XOR between the received parity bit and the regenerated parity bit. Therefore, it would evaluate to 1 if and only if the number of errors occurring in the $k+1$ bits (k bits of $\mathcal{A}$ and the received parity bit) is odd. Therefore, the probability of error-free decoding, with no errors in $\mathcal{B}$, is

$$P_B = \sum_{i=0}^{\frac{k}{2}} \binom{k+1}{2i+1} \varepsilon^{2i+1} (1-\varepsilon)^{2k-2i}. \tag{13}$$

Now, the probability of error is $1-(P_A+P_B)$. Therefore,

$$P_{\text{dap}} = 1 - \sum_{i=0}^{k} \binom{k}{i} \varepsilon^i (1-\varepsilon)^{2k+1-i} - \sum_{i=0}^{\frac{k}{2}} \binom{k+1}{2i+1} \varepsilon^{2i+1} (1-\varepsilon)^{2k-2i}. \tag{14}$$

For small probability of bit error ε, (14) simplifies to

$$P_{\text{dap}} \approx \frac{3k(k+1)}{2} \varepsilon^2. \tag{15}$$

REFERENCES

[1] (2003) International Technology Roadmap for Semiconductors. Semiconductor Industry Assoc. [Online]. Available: http://public.itrs.net

[2] F. Caignet, S. Delmas-Bendhia, and E. Sicard, "The challenge of signal integrity in deep-submicrometer CMOS technology," *Proc. IEEE*, vol. 89, no. 4, pp. 556–573, Apr. 2001.

[3] D. Sylvester and C. Hu, "Analytical modeling and characterization of deep-submicrometer interconnect," *Proc. IEEE*, vol. 89, no. 5, pp. 634–664, May 2001.

[4] J. Yim and C. Kung, "Reducing cross-coupling among interconnect wires in deep-submicron datapath design," in *Proc. DAC*, 1999, pp. 485–490.

[5] K. Kim, K. Baek, N. Shanbhag, C. Liu, and S. Kang, "Coupling-driven signal encoding scheme for low-power interface design," in *Proc. ICCAD*, 2000, pp. 318–321.

[6] Y. Zhang, J. Lach, K. Skadron, and M. R. Stan, "Odd/even bus invert with two-phase transfer for buses with coupling," in *Proc. ISLPED*, 2002, pp. 80–83.

[7] R. Hegde and N. R. Shanbhag, "Toward achieving energy efficiency in the presence of deep submicron noise," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 4, pp. 379–391, Aug. 2000.

[8] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital design," in *Proc. ICCAD*, 1996, pp. 147–151.

[9] V. Soteriou and L.-S. Peh, "Design-space exploration of power-aware on/off interconnection networks," in *Proc. ICCD*, 2004, pp. 510–517.

[10] L. Benini and G. D. Micheli, "Networks on chips: a new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[11] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A coding framework for low-power address and data busses," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 2, pp. 212–221, Jun. 1999.

[12] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 1, pp. 49–58, Mar. 1995.

[13] P. P. Sotiriadis, "Interconnect modeling and optimization in deep submicron technologies," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, May 2002.

[14] C. Duan, A. Tirumala, and S. P. Khatri, "Analysis and avoidance of crosstalk in on-chip buses," in *Proc. Hot Interconnects*, 2001, pp. 133–138.

[15] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in *Proc. ICCAD*, 2001, pp. 57–63.

[16] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag, "Area and energy-efficient crosstalk avoidance codes for on-chip buses," in *Proc. ICCD*, 2004, pp. 12–17.

[17] D. Bertozzi, L. Benini, and G. D. Micheli, "Low power error resilient encoding for on-chip data buses," in *Proc. DATE*, 2002, pp. 102–109.

[18] H. Zhang, V. George, and J. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 3, pp. 264–272, Jun. 2000.

[19] K. Patel and I. Markov, "Error-correction and crosstalk avoidance in DSM busses," in *Proc. SLIP*, 2003, pp. 9–14.

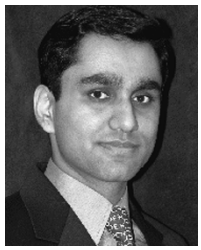
[20] S. R. Sridhara and N. R. Shanbhag, "Coding for system-on-chip networks: a unified framework," in *Proc. DAC*, 2004, pp. 103–106.

[21] D. Pamunuwa, L.-R. Zheng, and H. Tenhunen, "Maximizing throughput over parallel wire structures in the deep submicrometer regime," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 2, pp. 224–243, Apr. 2003.

[22] R. E. Blahut, *Algebraic Codes for Data Transmission*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[23] *HSPICE Simulation and Analysis Manual*, Synopsys, Mountain View, CA, 2003.

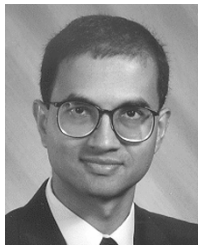
[24] H. Bakoglu, *Circuits, Interconnects and Packing for VLSI*. Reading, MA: Addison-Wesley, 1990.



Srinivasa R. Sridhara (S'01) received the B.Tech. degree in electronics and electrical communications engineering from the Indian Institute of Technology, Kharagpur, India, in 1999. He is currently pursuing the Ph. D. degree in electrical engineering at the University of Illinois at Urbana-Champaign.

From 1999 to 2000, he was with Synopsys, Bangalore, India. In summer of 2001, he interned at Lucent Technologies, Whippany, NJ. In fall of 2003, he interned at Intel Corporation, Hillsboro, OR. His research interests include low-power VLSI design, on-chip bus coding, and high-speed interconnects. He has co-authored ten conference and journal papers in these areas.

Mr. Sridhara received the Outstanding Student Designer Award from Analog Devices Inc. in 2001.



Naresh R. Shanbhag (S'87–M'93–SM'98) received the B.Tech. degree from the Indian Institute of Technology, New Delhi, India, in 1988, the M.S. degree from Wright State University, Dayton, OH, in 1990, and the Ph.D. degree from the University of Minnesota, Minneapolis, in 1993, all in electrical engineering.

From 1993 to 1995, he was with AT&T Bell Laboratories, Murray Hill, NJ, where he was the lead chip architect for AT&T's 51.84 Mb/s transceiver chips over twisted-pair wiring for Asynchronous Transfer Mode (ATM)-LAN and very high-speed digital subscriber line (VDSL) chipsets. Since August 1995, he has been with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, where he is presently a Professor. His research interests are in the design of integrated circuits and systems for broad-band communications including low-power/high-performance VLSI architectures for error-control coding, equalization, as well as digital integrated circuit design. He has published more than 90 journal articles, book chapters, and conference publications in this area and holds three US patents. He is also a coauthor of the research monograph *Pipelined Adaptive Digital Filters* (Boston, MA: Kluwer, 1994).

Dr. Shanbhag received the 2001 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS Best Paper Award, the 1999 IEEE Leon K. Kirchmayer Best Paper Award, the 1999 Xerox Faculty Award, the National Science Foundation CAREER Award in 1996, and the 1994 Darlington Best Paper Award from the IEEE Circuits and Systems Society. Since July 1997, he is a Distinguished Lecturer for the IEEE Circuits and Systems Society. From 1997 to 1999 and from 1999 to 2002, he served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: ANALOG AND DIGITAL SIGNAL PROCESSING and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, respectively. He has served on the technical program committees of various international conferences.