[3] X. D. Yang, C. K. Cheng, W. H. Ku, and R. J. Carragher, "Hurwitz stable reduced order modeling for RLC interconnect trees," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, Nov. 2000, pp. 222–228.
[4] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid interconnect circuit evaluator," in *Proc. IEEE/ACM Des. Autom. Conf.*, Jun. 1991, pp. 555–560.
[5] N. Gopal, D. P. Neikirk, and L. T. Pillage, "Evaluating RC-interconnect using moment-matching approximations," in *Proc. Tech. Dig. IEEE/ACM Int. Conf. Comput.-Aided Des.*, Nov. 1991, pp. 74–77.
[6] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximations via the Lanczos process," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639–649, May 1995.
[7] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 8, pp. 645–654, Aug. 1998.
[8] K. J. Kerns and A. T. Yang, "Preservation of passivity during RLC network reduction via split congruence transformation," in *Proc. 34th IEEE/ACM Des. Autom. Conf.*, Jun. 1997, pp. 34–39.
[9] Z. Qi, H. Yu, P. Liu, S. X.-D. Tan, and L. He, "Wideband passive multiport model order reduction and realization of $RLCM$ circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 8, pp. 1496–1509, Aug. 2006.
[10] G. Chen and E. G. Fridman, "An RLC interconnect model based on Fourier analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 2, pp. 170–183, Feb. 2005.
[11] M. Kamon, M. J. Ttsuk, and J. White, "FASTHENRY: A multipole-accelerated 3-D inductance extraction program," *IEEE Trans. Microw. Theory Tech.*, vol. 42, no. 9, pp. 1750–1758, Sep. 1994.
[12] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 10, no. 11, pp. 1447–1459, Nov. 1991.
[13] O. Brune, "Synthesis of a finite two-terminal network whose driving point impedance is a prescribed function of frequency," *J. Math. Phys.*, vol. 10, no. 3, pp. 191–236, Oct. 1931.
[14] A. B. Kahng and S. Muddu, "Efficient gate delay modeling for large interconnect loads," in *Proc. IEEE Multi-Chip Module Conf.*, 1996, pp. 202–207.
[15] G. Miano and A. Mafucci, *Transmission Lines and Lumped Circuits*. San Diego, CA: Academic, 2001.
[16] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-22, no. 6, pp. 504–509, Jun. 1975.
[17] K. Agarwal, D. Sylvester, and D. Blaauw, "An effective capacitance based driver output model for on-chip RLC interconnects," in *Proc. Des. Autom. Conf.*, 2003, pp. 376–381.
[18] L. K. Vakati and J. Wang, "A new multi-ramp driver model with RLC interconnect load," in *Proc. Int. Symp. Circuits and Syst.*, 2004, pp. 269–272.
[19] Y. I. Ismail and E. G. Friedman, "Effects of inductance on the propagation delay and repeater insertion in VLSI circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 2, pp. 195–206, Apr. 2000.
[20] A. B. Kahng and S. Muddu, "An analytical delay model for RLC interconnects," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 12, pp. 1507–1514, Dec. 1997.

Coding for Reliable On-Chip Buses: A Class of Fundamental Bounds and Practical Codes

Srinivasa R. Sridhara and Naresh R. Shanbhag

***Abstract*—A reliable high-speed bus employing low-swing signaling can be designed by encoding the bus to prevent crosstalk and provide error correction. Coding for on-chip buses requires additional bus wires and codec circuits. In this paper, fundamental bounds on the number of wires required to provide joint crosstalk avoidance and error correction using memoryless codes are presented. The authors propose a code construction that results in practical codec circuits with the number of wires being within 35% of the fundamental bounds. When applied to a 10-mm 32-bit bus in a 0.13-μm CMOS technology with low-swing signaling, one of the proposed codes provides 2.14× speedup and 27.5% energy savings at the cost of 2.1× area overhead, but without any loss in reliability.**

***Index Terms*—Coding, crosstalk, error correction, interconnect, low power, on-chip bus, reliability.**

I. Introduction

Interconnects such as global buses in deep submicrometer system-on-chip (SOC) designs consume significant amounts of power [1]–[4] and have large propagation delays [1], [2], [5], [6]. Future SOCs are expected to follow the network-on-chip paradigm [7], where high-speed energy-efficient communication between various SOC components is vital. Power consumption and delay in interconnects are expected to increase in future technologies due to increasing interconnect densities and die sizes. According to the International Technology Roadmap of Semiconductors (ITRS) [1], gate delay reduces with scaling, while global wire delay increases. Therefore, the delay of global buses will act as the performance bottleneck in many high-performance SOC designs. Repeater insertion mitigates the delay to some extent at the cost of additional power dissipation and chip area [1]. Therefore, design of high-speed low-power buses is a critical problem in the design of high-performance and/or low-power SOCs.

Coding [8]–[23] has emerged as a promising solution to both power and delay problems in global buses. Past work in this area includes coding for: 1) low-power buses through self [8]–[10] and coupling [11], [13], [14] transition activity reduction (low-power codes); 2) delay reduction [12], [15], [16], [18] [crosstalk-avoidance codes (CACs)]; and 3) improved reliability in low-swing buses [error-control codes (ECCs)] [19], [20].

Coding involves mapping k data/information bits to n code bits resulting in an (n, k) code having a code rate of k/n. This mapping can involve memory. However, codes with memory, in general, suffer from error propagation at the decoder. Complex techniques, such as those employed in communication systems [24], are needed to ensure that error propagation is not catastrophic. We believe that such techniques are prohibitively complex to be used for on-chip buses in the foreseeable future. Further, even when the error propagation is not a concern, codes with memory tend to have significantly more complex

Manuscript received October 27, 2004; revised June 16, 2005 and December 23, 2005. This work was supported by the MARCO-sponsored Gigascale Systems Research Center and in part by Intel Corporation. This paper was recommended by Associate Editor M. Poncino.

S. R. Sridhara was with the University of Illinois, Urbana–Champaign, IL 61801 USA. He is now with DSP Solutions R&D Center, Texas Instruments, Dallas, TX 75243 USA (e-mail: sridhara@ti.com).

N. R. Shanbhag is with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois, Urbana–Champaign, IL 61801 USA (e-mail: shanbhag@uiuc.edu).

Digital Object Identifier 10.1109/TCAD.2006.884418

encoders and decoders [18]. Therefore, we focus on memoryless codes in this paper.

The design of memoryless codes boils down to determining a subset $\mathcal{C}$ of size/cardinality 2^k consisting of n-bit vectors derived from the set $\mathcal{S}$ of all possible 2^n n-bit vectors. The codewords in $\mathcal{C}$, referred to as the codebook, provide delay, power, or reliability benefits by satisfying specific constraints. For example, a (n, k, p) CAC achieves delay reduction by reducing the worst case delay of a bus from $(1 + 4\lambda)\tau_0$ to $(1 + p\lambda)\tau_0$, where τ_0 is the delay of a crosstalk-free bus line, λ is the ratio of the coupling capacitance to the bulk capacitance, and $p = 1, 2,$ or 3 is the maximum coupling. Similarly, a (n, k, d) ECC improves the reliability of buses by increasing the minimum distance d [24]. Codes with minimum distance $d = 3$ are of specific interest as they achieve a single error correction. Codes have been proposed [21]–[23] that reduce the delay and improve the reliability/power simultaneously. Such joint codes, denoted as (n, k, p, d) codes, need to satisfy dual constraints of maximum coupling p and minimum distance d. We use the notations $(n, k, p, -)$ and $(n, k, -, d)$ to signify that there are no constraints on d and p, respectively.

Coding involves area, codec, and latency overheads. The area overhead is due to the additional $(n - k)$ bus lines that have the same line geometry and spacing as original k wires. Therefore, it is important to determine the minimum value of n ($n_{\min}$) necessary for a specific k, in order to satisfy constraints on maximum coupling and/or minimum distance. Specific variations of this problem have been solved. For example, in [13], asymptotic bounds on k/n were derived for the three types of $(n, k, p, -)$ CACs. However, these bounds do not provide us with the minimum number of wires needed to encode a given k-bit bus with memoryless codes. In [15] and [16], $n_{\min}$ has been derived for the specific case of an $(n, k, 2, -)$ CAC. Error-control coding theory [24] provides bounds on $n_{\min}$ for $(n, k, -, 3)$ ECCs.

In this paper, we first derive $n_{\min}$ for $(n, k, 1, -)$ and $(n, k, 3, -)$ CACs, thereby filling the gap in the existing theory. Employing these results, we derive bounds on $n_{\min}$ for the general case of an (n, k, p, d) code, thereby solving this problem in its entirety for memoryless codes.

From an implementation point of view, two challenges in approaching $n_{\min}$ are the lack of a suitable code construction and the complexity of the codec circuits. In this paper, we present a code construction and derive several practical (n, k, p, d) codes that can be used in the design of high-speed reliable low-swing buses. The performance of these practical codes is evaluated using a standard 0.13-μm CMOS technology.

The rest of this paper is organized as follows. In Section II, we present an overview of the coding schemes and define our notation and terminology. In Section III, we derive fundamental limits on the number of wires required for $(n, k, 1, -)$ and $(n, k, 3, -)$ CACs. In Section IV, we derive bounds on the number wires required for joint error correction and crosstalk avoidance, i.e., (n, k, p, d) codes. In Section V, we describe the code construction for practical encoding and decoding and derive several novel codes from the construction. We demonstrate the speedup and the energy savings achievable by the proposed codes using a standard 0.13-μm CMOS technology.

II. Background

The minimum number of wires $n_{\min}$ for a given k-bit bus is found by first determining the largest codebook $\mathcal{C}(n, p, d)$ of n-bit vectors that satisfies the constraints on maximum coupling p and minimum distance d. Since $\mathcal{C}(n, p, d)$ needs to have at least 2^k codewords to encode a k-bit bus, $n_{\min}$ is given by

$$n_{\min} = n : \quad |\mathcal{C}(n, p, d)| \geq 2^k. \tag{1}$$

For example, $\mathcal{C}(n, -, -)$ has all 2^n codewords because there are no constraints on p and d and, hence, represents the uncoded case, where $n_{\min} = k$.

In this section, we review past work on determining the minimum number of wires $n_{\min}$ for $(n, k, 2, -)$ CAC and $(n, k, -, 3)$ ECC. We also describe the corresponding practical codes.

A. CAC With $p = 2$

The delay of a wire is not a function of bits but of transitions occurring on the wire and its neighbors [12]. Therefore, crosstalk avoidance for delay reduction is a spatio-temporal problem. CACs can be designed by restricting the type of transitions that can occur on a wire and its neighbors.

CACs proposed in [16] reduce the maximum coupling to $p = 2$ 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 a forbidden-transition (FT) condition. Codes that satisfy the FT condition are referred to as FT codes (FTC). It has been shown in [16] that the size of the largest n-bit codebook satisfying the FT condition is

$$|\mathcal{C}\,(n, 2\,(\text{FT}), -)| = F(n + 2) \tag{2}$$

where $F(n)$ is the Fibonacci sequence satisfying

$$F(n) = F(n - 1) + F(n - 2), \qquad \text{for } n \geq 3 \tag{3}$$

with initial conditions $F(1) = F(2) = 1$. Therefore, $n_{\min}$ satisfies

$$n_{\min} = n : \quad F(n + 2) \geq 2^k, \qquad \text{for } p = 2\ (\text{FT}). \tag{4}$$

Worst case delay of $(1 + 2\lambda)\tau_0$ can also be achieved by avoiding bit patterns 010 and 101 from every codeword [15]. This condition is referred to as a forbidden-pattern (FP) condition and codes that satisfy the FP condition are called FP codes (FPC). The size of the largest codebook satisfying the FP condition is $|\mathcal{C}(n, 2(\text{FP}), -)| = 2F(n + 1)$ [15] and, therefore, $n_{\min}$ satisfies

$$n_{\min} = n : \quad 2F(n + 1) \geq 2^k, \qquad \text{for } p = 2\ (\text{FP}). \tag{5}$$

Note that $n_{\min}$ is smaller for FPC. However, this translates into saving of at most one wire for a given k.

For $(n, k, 2, -)$ CACs, code constructions exist that can be used to achieve $n_{\min}$. However, encoding all bits at once using CAC is infeasible for large buses due to a prohibitive complexity of the codec circuits. Therefore, partial coding [15], [16] is employed, in which the bus is broken into subbuses of smaller width which are encoded into subchannels. These subchannels are then combined in such a way so as to avoid the crosstalk delay at their boundaries.

FTC and FPC are nonlinear codes. It has been shown that there is no linear code that satisfies the FT (FP) condition while requiring fewer wires than shielding (duplication) [25].

B. Error-Control Code (ECC)

A $(n, k, -, 3)$ code provides a single error correction by increasing the minimum distance to $d = 3$. The problem of finding the largest codebook $\mathcal{C}(n, -, 3)$ satisfying $d = 3$ is unsolved [24]. Instead, an upper bound on the value of $|\mathcal{C}(n, -, 3)|$ can be obtained. The upper bound, referred to as the Hamming bound, is computed by analytical methods and is given by [24]

$$|\mathcal{C}(n, -, 3)| \leq \frac{2^n}{n + 1}. \tag{6}$$

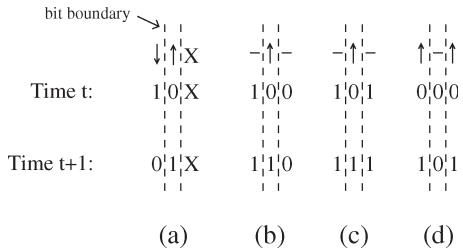

Fig. 1. Data patterns causing FTs for $p = 1$. The dotted lines indicate bit boundaries.

Therefore, $n_{\min}$ satisfies

$$n_{\min} = n: \quad \frac{2^n}{n+1} \geq 2^k, \qquad \text{for } d = 3. \tag{7}$$

A Hamming code [24] is an example of single error correcting codes. Hamming codes are linear and systematic. In systematic codes, redundant/parity bits are added to the input bits, which are unchanged, to generate the codeword.

III. Crosstalk-Avoidance Coding

In this section, we first present the conditions on the codewords required to achieve $p = 1$ and 3 and, then, derive $n_{\min}$ for $(n, k, 1, -)$ and $(n, k, 3, -)$ codes. From Section II, we see that $n_{\min}$ for $(n, k, 2, -)$ codes already exists in literature [15], [16]. Thus, the results of this section solve the two remaining cases.

A. CAC With $p = 1$

The analysis [13] of all possible transitions on a wire and its two adjacent wires shows that maximum coupling $p = 1$ can be achieved if and only if the transitions $\downarrow\uparrow \times$, $-\uparrow -$, and $\uparrow -\uparrow$ (and their complements) are avoided. Here, $\uparrow$, $\downarrow$, $-$, and $\times$ denote 0-to-1, 1-to-0, no, and don't-care transitions, respectively.

Examples of bit patterns resulting in such transitions are shown in Fig. 1. From Fig. 1(a), the transition $\downarrow\uparrow \times$ can be avoided if a codeword with 10 pattern does not transition to a codeword 01 pattern at the same bit boundary. In other words, the codebook cannot have both 01 and 10 at the same boundary (FT condition). We refer to a bit boundary as 01 type if the codebook has 01 patterns in some codewords at that boundary but has no 10 patterns in any of the codewords. Similarly, a bit boundary is 10 type if 10 appears at that boundary in some codewords, but 01 does not appear in any of the codewords.

Fig. 1(b) imposes the additional constraint that two adjacent bit boundaries in the codebook cannot both be 01 type or 10 type (Forbidden adjacent boundary pattern condition). Fig. 1(c) and (d) indicates the requirement of avoiding patterns 010 and 101 (FP condition).

Codes satisfying the above necessary and sufficient conditions are referred as one lambda codes (OLC). The simplest OLC is duplication and shielding, where every bit is duplicated and shield wires are inserted between adjacent pairs of duplicated bits. We show that there is no linear code that satisfies all three conditions while requiring fewer wires than duplication and shielding [18].

Theorem 1: A $(n, k, 1, -)$ code has an $n_{\min}$ given by

$$n_{\min} = n: \quad G(n) \geq 2^k, \qquad \text{for } p = 1 \tag{8}$$

where $G(n)$ is a sequence satisfying

$$G(n) = G(n-1) + G(n-5), \qquad \text{for } n \geq 6 \tag{9}$$

with initial conditions $G(1) = 2$, $G(2) = 3$, $G(3) = 4$, $G(4) = 5$, and $G(5) = 7$.

Proof: See [18]. ■

B. CAC With $p = 3$

A wire has delay $(1 + 4\lambda)\tau_0$ if and only if it has a rising (falling) transition when both its neighboring wires have falling (rising) transitions. Therefore, the maximum coupling can be reduced to $p = 3$ if transitions $\downarrow\uparrow\downarrow$ and $\uparrow\downarrow\uparrow$ are avoided. This can be done if and only if a codeword having the bit pattern 010 does not transition to another codeword having the pattern 101 at the same bit position and vice versa. Thus, the codebook $\mathcal{C}(n, 3, -)$ needs to satisfy the following necessary and sufficient condition.

Forbidden-overlap (FO) condition: The codebook cannot have both 010 and 101 appearing centered around any bit position.

Codes that satisfy the above condition are referred to as FO codes (FOC). The simplest FOC is half-shielding, where a shield wire is inserted after every two wires. We show that there is no linear code that satisfies FO condition while requiring fewer wires than half-shielding [18].

Theorem 2: A $(n, k, 3, -)$ code has an $n_{\min}$ given by

$$n_{\min} = n: \quad T(n+2) \geq 2^k, \qquad \text{for } p = 3 \tag{10}$$

where $T(n)$ is the tribonacci number sequence satisfying

$$T(n) = T(n-1) + T(n-2) + T(n-3), \qquad \text{for } n \geq 4 \tag{11}$$

with initial conditions $T(1) = 1$, $T(2) = 1$, and $T(3) = 2$.

Proof: See [18]. ■

IV. Joint CAC and ECC Codes

A (n, k, p, d) code with $p = 1, 2,$ or 3 and $d = 3$ provides joint crosstalk avoidance and single error correction. In this section, we bound the value of $n_{\min}$ for joint CAC-ECC codes by computing the bounds on the codebook size.

A. Exact Value of $n_{\min}$ for Small k

Similar to standard ECC, the largest codebook $\mathcal{C}(n, p, d)$ for $p = 1, 2,$ or 3 and $d = 3$ is not known. We can obtain $\mathcal{C}(n, p, d)$ by finding the largest clique in a graph representing maximum coupling and minimum distance conditions [21]. The graph has all 2^n codewords as nodes and edges between all pairs of codewords that satisfy both conditions. The value of $n_{\min}$ can be obtained by using (1). However, the problem of finding the largest clique in a graph is NP-complete, and an algorithm such as the reactive local search [26] requires excessive time and memory to compute $\mathcal{C}(n, p, d)$ for large n.

The value of $n_{\min}$ for $n \leq 11$ computed using the Max Clique Solver [26] is shown in Table I. As expected, there is a tradeoff between the bus delay and the number of wires n required to encode a given bus. For example, to encode a 4-bit bus with a joint code, 11 wires are required for $p = 1$, ten wires for $p = 2$ (FT), nine wires for $p = 2$ (FP), and eight wires for $p = 3$.

B. Bound on the Value of $n_{\min}$ for Large k

For $n > 11$, we present the following bound on the value of $n_{\min}$.

Theorem 3: The value of $n_{\min}$ for $(n, k, p, 3)$ code satisfies

$$n_{\min} = n: \quad 2^k \leq \begin{cases} \frac{G(n)}{2} & p = 1 \quad d = 3 \\ \frac{F(n+2)}{\lfloor n/2 \rfloor + 1} & p = 2 \text{ (FT) } d = 3 \\ \frac{2F(n+1)}{3} & p = 2 \text{ (FP) } d = 3 \\ \frac{T(n+2)}{\lfloor n/2 \rfloor + 2} & p = 3 \quad d = 3 \end{cases} \tag{12}$$

TABLE I
MINIMUM NUMBER OF WIRES $n_{\min}$ REQUIRED FOR ENCODING A k-BIT BUS WITH VARIOUS JOINT CODES

k	n_{min}			
	$p=1$ $d=3$	$p=2$ (FT) $d=3$	$p=2$ (FP) $d=3$	$p=3$ $d=3$
1	3	3	3	3
2	6	6	5	5
3	9	8	7	7
4	11	10	9	8
5	–	–	11	10
6	–	–	–	11

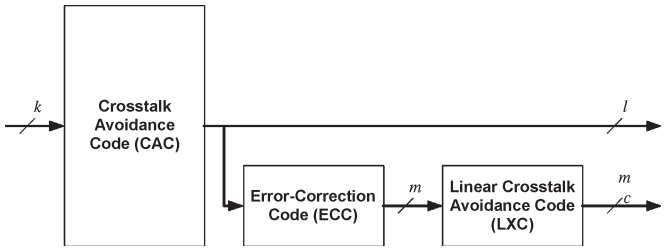

Fig. 2. Code construction for joint crosstalk avoidance and error correction.

where $G(n)$ is given by (9) and $F(n)$ and $T(n)$ are Fibonacci and tribonacci sequences, respectively.

Proof: See [18]. ■

V. PRACTICAL CODES

The fundamental bounds do not provide us with an implementation for the joint codes. We employ a code construction derived from the coding framework proposed in [22] to design practical joint codes. The code construction is shown in Fig. 2. A k-bit input is first encoded using CAC. Then, ECC generates m parity bits for the l bits at the output of CAC. Here, we assume that ECC is a systematic code. The parity bits generated by ECC do not satisfy the constraint on coupling. Therefore, we need to encode them using a CAC. However, CAC for the parity bits must be a code that does not modify the parity bits in any way as decoding of ECC has to occur before any other decoding in the receiver. Therefore, a linear crosstalk code (LXC), which does not modify its inputs, is employed to generate $m_{\rm c}$ encoded parity bits. The LXC employed is half-shielding for $p=3$, shielding for $p=2$ (FT), duplication for $p=2$ (FP), and duplication and shielding for $p=1$.

A practical joint CAC-ECC code can be obtained by combining any of the practical CACs with a Hamming code using the construction in Fig. 2. Table II lists the coding schemes FOC+HC, FTC+HC, and OLC+HC that combine a Hamming code with FOC(5,4), FTC(4,3), and OLC(8,4) based CACs [23], respectively. Alternative constructions referred to as duplicate-add-parity (DAP) [25] and duplicate-shield-add-parity (DSAP) [23] are also listed in the table.

Table II also lists the number of wires required for a 32-bit bus employing the practical code and compares it with the fundamental bounds derived in Section IV. The number of wires required for the practical codes is within 35% of $n_{\min}$. We see that DSAP requires lower number of wires than OLC+HC. For a 32-bit bus, DSAP requires 97 wires compared to 106 wires for OLC+HC.

VI. SIMULATION RESULTS

We quantify the benefits of the crosstalk avoidance and error correction by computing the reduction in delay and energy dissipation for a 32-bit, metal 4 bus with minimum width and spacing in a standard 0.13-μm CMOS technology for various values of bus length L. 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 $V_{\rm dd}=1.2$ in order to ensure reliable coding and decoding operations.

Speedup of code 1 over code 2 is defined as

$$\text{speedup} = \frac{T_{\rm b2}+T_{\rm c2}}{T_{\rm b1}+T_{\rm c1}} \tag{13}$$

where $T_{{\rm b}i}$ is the bus delay with code i and $T_{{\rm c}i}$ is the codec delay (sum of encoder and decoder delays) of code i. The energy savings and area overhead include codec energy dissipation and area, respectively.

In Table III, we compare codes for a 32-bit bus. Bus delay and energy dissipation are listed as function of λ, τ_0, $\hat{V}_{\rm dd}$, and C, where C is total bulk capacitance of a wire. Note that $\hat{V}_{\rm dd}$ is the supply swing at which the residual word-error rate is equal to 10^{-20}. The corresponding codec overheads in 0.13-μm technology are listed. The total area overheads over a 32-bit uncoded bus for $L=10$ mm are also shown.

Fig. 3(a) plots the speedup over the uncoded bus as a function of λ at $L=10$ mm. Hamming code has a speedup of less than one, indicating an effective slow down due to the codec delay. Along with the single error correction, joint codes offer a crosstalk avoidance and, hence, have significant speedup. FOC+HC reduces the bus delay to $(1+3\lambda)\tau_0$ and achieves a speedup of close to one for $L=10$ mm. Therefore, FOC+HC can be used as a "zero-latency" error correction code, in the sense that the reduction in the bus delay completely masks coding latency for global buses. DAP and DSAP reduce maximum coupling to two and one, respectively, and achieve speedups of 1.44 and 2.14, respectively, at $L=10$ mm and $\lambda=2.8$. Speedup is an increasing function of L and λ and, therefore, technology scaling leading to reduced codec delay, larger L, and larger λ will improve the achieved speedup.

Along with speedup, joint codes achieve energy savings over the uncoded bus as shown in Fig. 3(b). Energy savings over uncoded bus are achieved mainly due to the reduced swing. Further, many of the joint codes also reduce the coupling component of energy. Therefore, energy savings improve with λ. Although OLC+HC has the lowest coupling transition component of energy, its high self-transition component of bus energy and codec energy dissipation makes it unsuitable as an energy-efficient code in current technology. DAP and DSAP provide 27.5% energy savings at $L=10$ mm and $\lambda=2.8$. These codes have the lowest codec energy dissipation as shown in Table III. As shown in Fig. 3(b), the achieved energy savings improve with L because the codec energy overhead accounts for a smaller portion of the total energy for long buses.

A. Alternatives to Coding

An alternative to coding for delay reduction is to increase the interwire spacing. In Fig. 4, we plot the relative delay as a function of relative area for FOC, FTOC [18], and OLC and compare that with the relative delay achieved via increasing spacing. While FOC provides no benefits over increasing spacing, FTOC and OLC provide a significant delay reduction beyond what is achievable by just increasing spacing. Specifically, OLC provides 74% reduction in delay compared to 57% achieved by spacing with the same area overhead.

Increasing the interwire spacing also reduces the power dissipation by reducing the coupling capacitance. However, it does not provide reliability against external noise sources. Here, we compare a Hamming-coded 32-bit bus with spacing increased, such that it occupies the same area as a DAP-coded bus. Note that both schemes employ low-swing

TABLE II
PRACTICAL JOINT CODES AND COMPARISON WITH FUNDAMENTAL LIMITS ON MINIMUM NUMBER OF WIRES $n_{\min}$

Coding Scheme	Maximum coupling p	Minimum distance d	Components			Number of wires for 32-bit bus	
			CAC	ECC	LXC	Practical code	n_{min}
FOC+HC	3	3	FOC(5,4)	Hamming	Half-shielding	49	42
FTC+HC	2 (FT)	3	FTC(4,3)	Hamming	Shielding	65	53
DAP	2 (FP)	3	Duplication	Parity	–	65	48
OLC+HC	1	3	OLC(8,4)	Hamming	Duplication+Shielding	106	80
DSAP	1	3	Duplication+Shielding	Parity	Shielding	97	80

TABLE III
CODE COMPARISON FOR A 32-BIT BUS

Coding Scheme	Bus				Codec			Area overhead (%)
	No. of wires	Delay ($\times\tau_0$)	Average energy ($\times C\hat{V}_{dd}^2$)	$\hat{V}_{dd}$ (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
Hamming	38	$1+4\lambda$	$9.5+18.5\lambda$	0.884	6427	1005	9.90	24.1
FOC+HC	49	$1+3\lambda$	$10.3+17.0\lambda$	0.889	11069	1209	15.5	62.8
BSC	65	$1+2\lambda$	$16.3+16.0\lambda$	0.860	5978	1230	7.40	109.5
FTC+HC	65	$1+2\lambda$	$10.4+15.4\lambda$	0.896	11180	1275	15.4	113.6
DAP	65	$1+2\lambda$	$16.3+16.0\lambda$	0.860	4016	1111	5.36	108.0
OLC+HC	106	$1+\lambda$	$18.6+12.2\lambda$	0.900	25494	1639	31.17	255.2
DSAP	97	$1+\lambda$	$16.3+16.0\lambda$	0.860	4016	1111	5.36	209.5

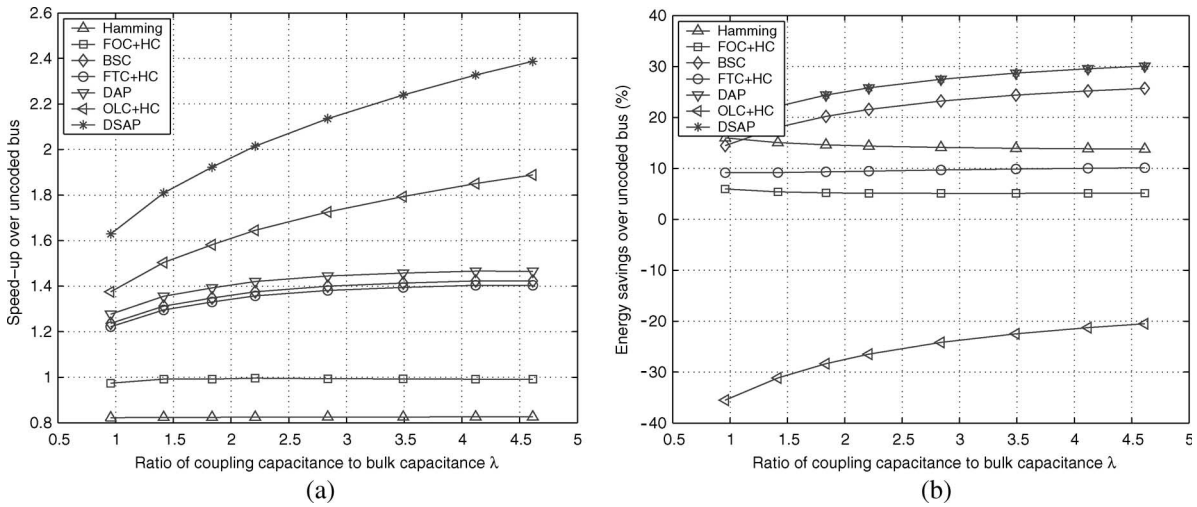

Fig. 3. Comparison across λ for 32-bit bus with $L = 10$ mm: (a) speedup and (b) energy savings over uncoded bus.

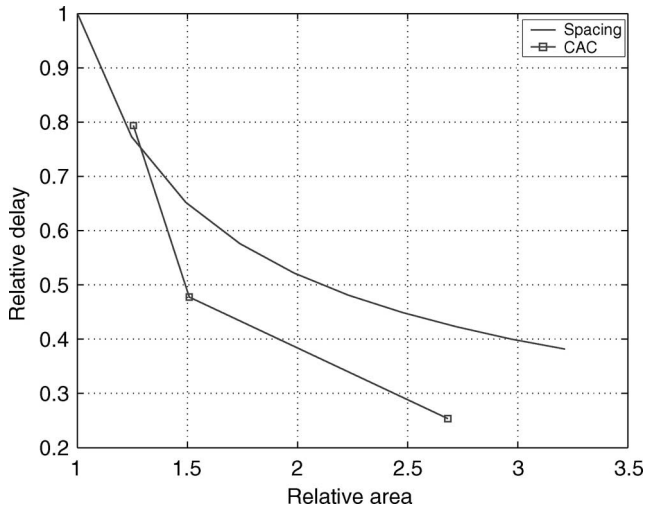

Fig. 4. Coding versus spacing: Delay reduction for $L = 10$ mm and $k = 32$.

signaling to reduce the power dissipation. Simulation results show that increasing spacing of a Hamming-coded bus reduces delay by 22% and power by 23% with respect to the uncoded minimum-spaced bus. From Fig. 3, DAP with minimum spacing reduces delay by 32% and power by 30%. Clearly, coding is a better alternative to increasing interwire spacing in terms of both delay and power dissipation. Further, these benefits are expected to increase in future technologies, as codec overheads will decrease with technology scaling.

Another alternative to delay reduction is repeater insertion. However, repeater insertion increases the power dissipation. We have shown in [25] that joint codes are a better alternative to Hamming codes with repeater insertion as joint codes reduce the power dissipation at the same time. Specifically, we have shown that the repeater insertion provides 2.97× speedup 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, DAP provides 2.09× speedup and 27% energy savings. Therefore, coding is a better alternative to repeater insertion for delay reduction as it reduces the power dissipation at the same time.

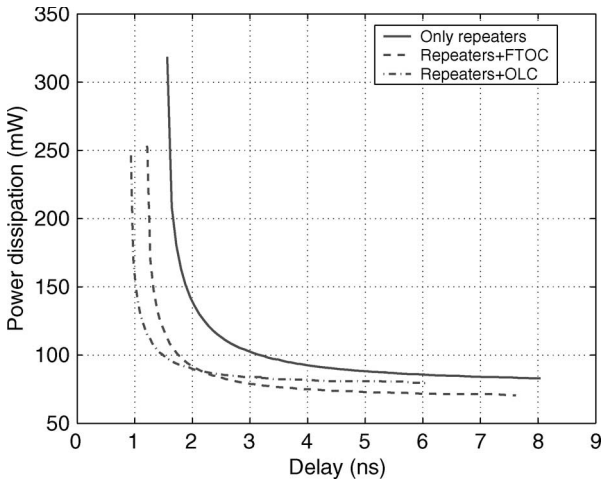

Fig. 5. Power versus delay tradeoff for 10-mm 32-bit bus using ITRS guidelines for 45-nm predictive technology model.

Although the repeater insertion provides slightly higher speedup in the current technology, the diverging trend between the gate and interconnect delays will make coding superior to repeater insertion in the future. We have used ITRS guidelines to show that this is indeed the case for 90-nm and smaller technologies [27]. In Fig. 5, we plot the power versus the delay tradeoff achieved using the repeater insertion, joint repeater insertion and FTOC [18], and joint repeater insertion and OLC for a 10-mm 32-bit bus in 45-nm technology. It is seen that the joint repeater insertion and OLC achieves 69.4% power savings while achieving the same delay as the optimally repeater inserted bus. Further, OLC, when used with optimal repeater insertion, achieves 21% speedup and 40% power savings over optimal repeater insertion.

VII. Conclusion

All joint codes presented in this paper trade off the 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 the gate delay and interconnect delay, and due to the longer bus lengths. Therefore, the codes presented in this paper will result in a greater reduction in energy dissipation and delay in SOCs of the future [27].

References

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

[2] D. Sylvester and C. Hu, "Analytical modeling and characterization of deep-submicrometer interconnect," in *Proc. IEEE*, May 2001, vol. 89, pp. 634–664.

[3] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE J. Solid-State Circuits*, vol. 29, no. 6, pp. 663–670, Jun. 1994.

[4] 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.

[5] 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.

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

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

[8] R. Fletcher, "Integrated circuit having outputs configured for reduced state changes," U.S. Patent 4 667 337, May 19, 1987.

[9] 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.

[10] 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.

[11] 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.

[12] P. P. Sotiriadis and A. Chandrakasan, "Reducing bus delay in submicron technology using coding," in *Proc. ASP-DAC*, 2001, pp. 109–114.

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

[14] 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.

[15] 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.

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

[17] W. Cheng and M. Pedram, "Memory bus encoding for low power: A tutorial," in *Proc. ISQED*, 2001, pp. 199–204.

[18] S. R. Sridhara, "Communication inspired design of on-chip buses," Ph.D. dissertation, Univ. Illinois, Urbana, IL, Jan. 2006.

[19] 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.

[20] 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.

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

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

[23] S. R. Sridhara and N. R. Shanbhag, "Coding for reliable on-chip buses: Fundamental limits and practical codes," in *Proc. VLSI Des.*, 2005, pp. 417–422.

[24] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York: North-Holland, 1977.

[25] S. R. Sridhara and N. R. Shanbhag, "Coding for system-on-chip networks: A unified framework," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 655–667, Jun. 2005.

[26] R. Battiti and M. Protasi. (2001, Apr.). "Reactive local search for the maximum clique problem, " *Algorithmica*, vol. 29, no. 4, pp. 610–637. [Online]. Available: http://rtm.science.unitn.it/intertools/clique

[27] S. R. Sridhara and N. R. Shanbhag, "A low-power bus design using joint repeater insertion and coding," in *Proc. ISLPED*, 2005, pp. 99–102.