

# Coding for Reliable On-Chip Buses: Fundamental Limits and Practical Codes

Srinivasa R. Sridhara and Naresh R. Shanbhag  
Coordinated Science Laboratory/ECE Department  
University of Illinois at Urbana-Champaign  
1308 W Main St., Urbana IL 61801  
E-mail:[sridhara,shanbhag]@uiuc.edu

## Abstract

*A reliable high-speed bus employing low-swing signaling can be designed by encoding the bus to prevent crosstalk and provide error correction. In this paper, we present fundamental limits on the number of wires required to achieve joint crosstalk avoidance and error correction in on-chip buses. We propose a code construction that results in practical encoding and decoding schemes with the number of wires being within 35% of the fundamental limits. The proposed codes, when applied to a 10-mm 32-bit bus in a 0.13- $\mu\text{m}$  CMOS technology with low-swing signaling, provide  $2.14\times$  speed-up and 27.5% energy savings without any loss in reliability.*

## 1 Introduction

Interconnects such as global buses in deep submicron (DSM) system-on-chip (SOC) designs consume significant amounts of power [1–3] and have large propagation delays [2, 4]. Power consumption and delay in buses are expected to increase in future technologies due to increasing interconnect densities and die sizes. Therefore, design of high-speed low-power buses is a critical problem in the design of high-performance and/or low-power SOCs.

Coding has emerged as a promising solution to power, delay, and reliability problems in global buses. Past work in this area includes coding for: 1.) low-power buses through self [5, 6] and coupling [7–9] transition activity reduction (*low-power codes* (LPCs)), 2.) delay reduction [9–12] (*crosstalk avoidance codes* (CACs)), and 3.) improved reliability in low-swing buses (*error-control codes* (ECCs)) [13, 14].

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. 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  $\tau_0$  is the delay of a crosstalk-free bus line,  $\lambda$  is the ratio of the coupling capacitance to the bulk capacitance, and  $p = 1, 2$ , or  $3$  is referred to as the maximum coupling. Similarly, a  $(n, k, d)$  ECC improves the reliability of buses by increasing the minimum distance  $d$  [15]. Codes with minimum distance  $d = 3$  are of specific interest as they achieve single error correction. Codes have been proposed [16, 17] that reduce delay and improve reliability/power simultaneously. Such joint codes, denoted as a  $(n, k, p, d)$  codes, need to satisfy dual constraints of maximum coupling  $p$  and minimum distance  $d$ .

A key drawback of coding is the area overhead due to the additional  $(n - k)$  bus wires. Therefore, it is important to determine the theoretically minimum number of lines  $n_{min}$  needed to encode  $k$  bits while satisfying the maximum coupling  $p$  and/or minimum distance  $d$  constraints. Specific variations of this problem have been solved. For example, in [9], 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. In [10, 11],  $n_{min}$  has been derived for the specific case of an  $(n, k, 2, -)$  CAC. Error-control coding theory [15] provides bounds on  $n_{min}$  for  $(n, k, -, 3)$  ECCs.

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

## 2 Background

The minimum number of wires  $n_{min}$  for a given  $k$ -bit bus is determined by first finding the largest codebook

$C(n, p, d)$  of  $n$ -bit vectors that satisfies the constraints on maximum coupling  $p$  and minimum distance  $d$ . Since  $C(n, p, d)$  needs to have at least  $2^k$  codewords to encode a  $k$ -bit bus,  $n_{min}$  is given by

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

In this section, we review past work on the minimum number of wires  $n_{min}$  for  $(n, k, 2, -)$  CAC and  $(n, k, -, 3)$  ECC.

### 2.1 CAC with $p = 2$

Crosstalk avoidance codes proposed in [11] 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 forbidden transition (FT) condition. Codes that satisfy the FT condition are referred to as forbidden transition codes (FTC). It has been shown in [11] that the size of the largest  $n$ -bit codebook satisfying the FT condition is

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

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

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

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

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

Worst-case delay of  $(1 + 2\lambda)\tau_0$  can also be achieved by avoiding bit patterns 010 and 101 from every codeword [10]. This condition is referred to as forbidden pattern (FP) condition and codes that satisfy the FP condition are called forbidden pattern codes (FPC). The size of the largest codebook satisfying the FP condition is  $|C(n, 2 \text{ (FP)}, -)| = 2F(n + 1)$  [10] and, therefore,  $n_{min}$  satisfies

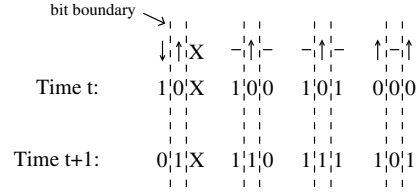
$$n_{min} = n : 2F(n + 1) \geq 2^k \quad \text{for } p = 2 \text{ (FP)}. \quad (5)$$

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 prohibitive complexity of the codec circuits. Therefore, partial coding [10, 11] is employed, in which the bus is broken into sub-buses of smaller width which are encoded into sub-channels. These sub-channels are then combined in such a way so as to avoid crosstalk delay at their boundaries.

### 2.2 Error correction coding (ECC)

A  $(n, k, -, 3)$  code provides single error correction by increasing the minimum distance to  $d = 3$ . The problem of finding the largest codebook  $C(n, -, 3)$  satisfying  $d = 3$  is unsolved [15]. Instead, an upper bound on the value of  $|C(n, -, 3)|$  can be obtained. The upper bound, referred to as Hamming bound, is computed by analytical methods and is given by [15]

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



**Figure 1. Forbidden transitions for  $p = 1$ .**

Therefore,  $n_{min}$  satisfies

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

A Hamming code [15] 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.

## 3 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.

### 3.1 CAC with $p = 1$

The analysis [9] 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 transition, 1-to-0 transition, no transition, and don't-care transition, respectively.

Examples of bit patterns resulting in such transitions are shown in Figure 1. Clearly, the transition  $\downarrow\uparrow\times$  can be avoided if a codeword with 01 pattern does not transition to a codeword 10 pattern at the same bit boundary. Thus, 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 codewords. Similarly, a bit boundary is of 10-type if 10 appears at that boundary in some codewords but 01 does not appear in any of the codewords.

The second example in Figure 1 imposes the additional constraint that two adjacent bit boundaries in the codebook cannot both be of 01-type or 10-type. The last two examples in Figure 1 require avoiding patterns 010 and 101 (FP condition). Thus, the codebook  $C(n, 1, -)$  satisfies FT, FP, and the following condition.

**Forbidden adjacent boundary pattern (FABP) condition:** Two adjacent bit boundaries in the codebook cannot both be of 01-type or 10-type.

Codes satisfying the above 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.

Here, we state a theorem on the fundamental limit  $n_{min}$  for  $(n, k, 1, -)$  CACs without proof. A proof of the theorem is presented in [18].

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

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

where  $G(n)$  is a sequence satisfying

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

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

### 3.2 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 forbidden overlap codes (FOC). The simplest FOC is half-shielding, where a shield wire is inserted after every two wires.

**Theorem 2.** A  $(n, k, 3, -)$  code has an  $n_{min}$  given by [18]

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

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

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

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

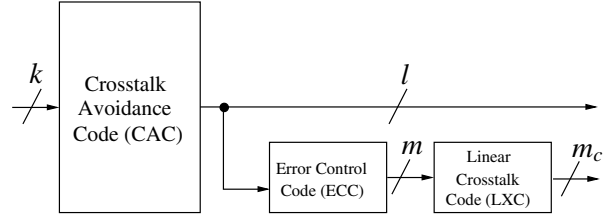
## 4 Joint Codes

A  $(n, k, p, d)$  code for  $p = 1, 2$ , or  $3$  and  $d = 3$  provides joint crosstalk avoidance and single error correction. Similar to  $(n, k, -, 3)$  ECC, the largest codebook  $\mathcal{C}(n, p, d)$  for joint codes 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 [17]. 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. Excessive time and memory are required to compute  $\mathcal{C}(n, p, d)$  for large  $n$ .

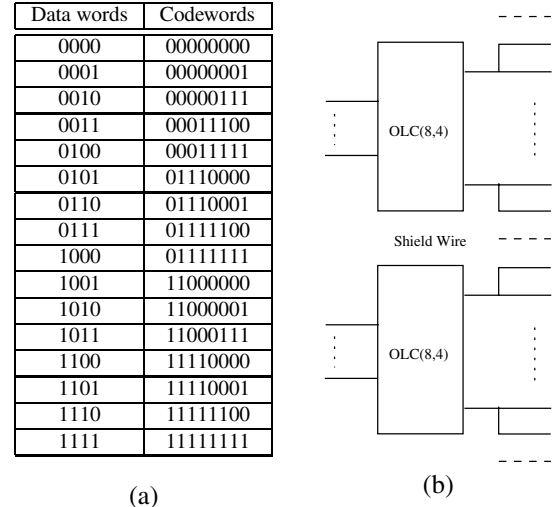
For large  $n$ , we present the following bound on the value of  $n_{min}$  [18].

**Theorem 3.** A  $(n, k, p, d)$  code has an  $n_{min}$  given by

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



**Figure 2.** A code construction for joint crosstalk avoidance and error correction.



**Figure 3.** One lambda codes: (a) codewords for OLC(8,4) and (b) partial coding for large buses with duplication and shielding at the boundaries of sub-channels.

## 5 Practical Codes

In this section, we first present practical  $(n, k, 1, -)$  and  $(n, k, 3, -)$  CACs. In case of CAC, we cannot encode all  $k$  bits at once for large buses. Therefore, we employ partial coding to reduce the complexity of the CAC codec circuits.

Next, in order to implement a joint code, we employ a code construction derived from the coding framework proposed in [16]. The code construction is shown in Figure 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  $m_c$  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  $p = 1$ .

Data words $b_{3:0}$	Codewords $a_{4:0}$
0000	00000
0001	00100
0010	00001
0011	00101
0100	00011
0101	00111
0110	10011
0111	10111
1000	10000
1001	10100
1010	10001
1011	10101
1100	11000
1101	11100
1110	11001
1111	11101

$$\begin{aligned}
a_4 &= b_1 b_2 + b_3 \\
a_3 &= b_2 b_3 \\
a_2 &= b_0 \\
a_1 &= b_2 b'_3 \\
a_0 &= b_1 + b_2 b'_3 \\
b_3 &= a'_1 a_4 \\
b_2 &= a_1 + a_3 \\
b_1 &= a_0(a'_1 + a_4) \\
b_0 &= a_2
\end{aligned}$$

**Figure 4. Codewords and boolean expressions for FOC(5,4).**

### 5.1 One lambda codes

For  $k < 10$ , the highest code rate ( $k/n_{min}$ ) of 50% is achieved for  $k = 4$ . The data words and codewords corresponding to  $k = 4$  are shown in Figure 3(a). We refer to this code as OLC(8,4). Partial coding for large buses using OLC(8,4) requires duplicating the boundary bits of the sub-channel and inserting a shield wire between sub-channels as shown in Figure 3(b).

To encode a  $k$ -bit bus using OLC(8,4),  $l = 11k/4 - 3$  wires are required, where  $k$  is assumed to be a multiple of 4. For example, 85 wires are required for  $k = 32$  compared to  $n_{min} = 78$  given by (10).

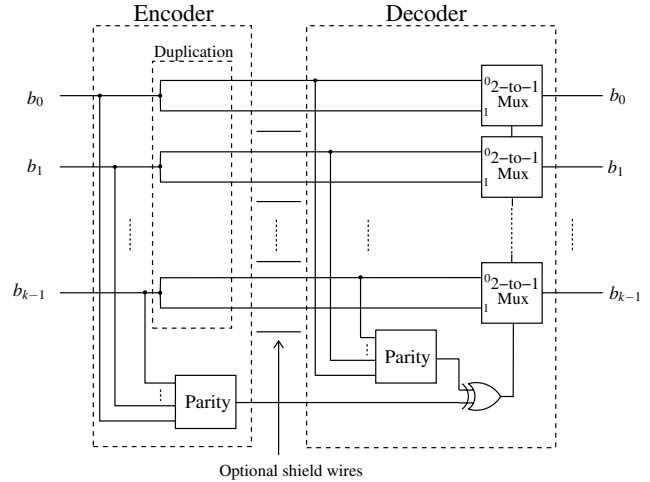
### 5.2 Forbidden overlap codes

We cannot place two sub-channels encoded with FOC next to each other without violating the FO condition at the boundary. We need shield wires between sub-channels to satisfy the FO condition, thereby increasing the number of wires. But, a closer inspection reveals that if there are no opposing transitions between the boundary bit and its neighbor within a sub-channel, then two sub-channels can be placed next to each other without any shielding. However, eliminating opposing transitions at the boundaries increases  $n_{min}$  [18].

**Theorem 4.** A  $(n, k, 3, -)$  code without opposing transitions at the boundaries has an  $n_{min}$  given by

$$n_{min} = n : T(n+2) - T(n) \geq 2^k \quad \text{for } p = 3. \quad (13)$$

For  $k < 10$ , the largest code rate ( $k/n_{min}$ ) of 80% is achieved for  $k = 4$ . Though a higher coding rate is possible for  $k > 10$ , we choose  $k = 4$  as the codec complexity grows exponentially with  $k$ . Larger buses are broken into 4-bit sub-buses and encoded into 5-bit sub-channels. We refer to this code as FOC(5,4). The codewords and the corresponding boolean expressions for FOC(5,4) are shown in Figure 4. Encoder and decoder are simple combinational



**Figure 5. Duplicate-(shield)-add-parity: A joint code for FPC and OLC. Shield wires required for OLC.**

circuits and complexity for large buses grows linearly with bus width.

To encode a  $k$ -bit bus using FOC(5,4),  $l = 5k/4$  wires are required, where  $k$  is assumed to be a multiple of 4. For example 40 wires are required for  $k = 32$  compared to  $n_{min} = 37$  given by (10).

### 5.3 Joint codes

A practical joint code can be obtained by combining any of the practical CACs with a Hamming code using the construction in Figure 2. Table 1 lists 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, respectively. An appropriate LXC scheme is also employed as listed in the table.

An alternative construction that does not employ Hamming codes is possible for FPC and OLC based joint codes. Consider the duplication scheme for satisfying the FP condition. This code has a minimum distance of two as any two distinct codewords differ in at least two bits. We can increase the minimum distance to three by appending a single parity bit. This code referred to as duplicate-add-parity (DAP) is shown in Figure 5. DAP achieves  $(1 + 2\lambda)\tau_0$  delay and single error correction. We can also satisfy the FABP condition and reduce the delay to  $(1 + \lambda)\tau_0$  by adding shield wires to obtain duplicate-shield-add-parity (DSAP) code as shown in Figure 5.

To decode, we recreate the parity bit by using one set of the received data bits and compare that with received parity bit. If 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 Figure 5. Since a single error will at most affect one of the sets or the parity bit, it is correctable.

Table 1 also lists the number of wires required for a 32-bit bus employing the practical code and compares it with the fundamental limits. The number of wires required for

**Table 1. Practical joint codes and comparison with 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

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.

## 6 Simulation Results

We quantify the benefits of crosstalk avoidance and error correction by computing the reduction in delay and energy dissipation. The achieved improvements vary with bus length  $L$ , ratio of coupling capacitance to bulk capacitance  $\lambda$ , bus width  $k$ , and the process technology. Here, we present the improvements for 10-mm 32-bit in a standard 0.13- $\mu\text{m}$  CMOS technology.

We assume that reducing the supply voltage  $V_{dd}$  will cause errors with probability [13, 14]

$$\varepsilon = Q\left(\frac{V_{dd}}{2\sigma_N}\right), \quad (14)$$

where  $Q(\cdot)$  is the Gaussian probability of error function and  $\sigma_N^2$  is variance of the additive Gaussian noise. For a  $k$ -bit uncoded bus, the probability of word error is  $P_{unc}(\varepsilon) = k\varepsilon$ . If the residual probability of word error with ECC is  $P_{ecc}(\varepsilon)$ , then  $P_{ecc}(\varepsilon) < P_{unc}(\varepsilon)$ . For a given reliability requirement, we can reduce the supply voltage to

$$\hat{V}_{dd} = V_{dd} \frac{Q^{-1}(\hat{\varepsilon})}{Q^{-1}(\varepsilon)} \quad (15)$$

such that  $P_{ecc}(\hat{\varepsilon}) = P_{unc}(\varepsilon)$ .  $P_{ecc}$  for Hamming and DAP-based codes are given by

$$P_{ham}(\varepsilon) = \binom{l+m}{2} \varepsilon^2, \quad P_{dap} = \frac{3k(k+1)}{2} \varepsilon^2. \quad (16)$$

respectively. In this paper, we assume a word-error rate requirement of  $10^{-20}$ .

We consider a metal 4 bus with minimum width of 0.2  $\mu\text{m}$  and minimum spacing of 0.2  $\mu\text{m}$ . For a given bus geometry, the value of  $\lambda$  depends on the metal coverage in upper and lower metal layers [2]. We vary  $\lambda$  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 of the bulk capacitance is assumed to be from metal 4 to the substrate, resulting in  $\lambda = 4.6$ .

We assume  $50\times$  minimum-sized drivers and obtain bus delay and energy dissipation for various bus transitions using HSPICE. The average energy per bus transfer is computed using the energy model in [9]. We assume that the

data is spatially and temporally uncorrelated and that 0 and 1 are equally likely to appear. In case of ECC, the bus energy is also computed at the reduced supply voltage  $\hat{V}_{dd}$ . The codes are synthesized using a 0.13- $\mu\text{m}$  CMOS standard cell library and optimized for speed.

Figure 6(a) plots the speed-up over the uncoded bus as a function of  $\lambda$ . Hamming code has a speed-up of less than 1, indicating an effective slow-down due to codec delay. Along with single error correction, joint codes offer crosstalk avoidance and, hence, have significant speed-up. FOC+HC reduces the bus delay to  $(1 + 3\lambda)\tau_0$  and achieve a speed-up of close to 1. Therefore, FOC+HC can be used as a “zero-latency” error correction code, in the sense that the reduction in bus delay completely masks coding latency for global buses. DAP and DSAP reduce maximum coupling to 2 and 1, respectively, and achieve speed-ups of 1.44 and 2.14, respectively, at  $\lambda = 2.8$ . Speed-up is an increasing function of  $\lambda$  and, therefore, technology scaling leading to reduced codec delay, and larger  $\lambda$  will improve the achieved speed-up.

Along with speed-up, joint codes achieve energy savings over uncoded bus as shown in Figure 6(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  $\lambda$  as shown in the figure. DAP and DSAP have high energy savings as they achieve reduced swing with low codec power overhead. DAP and DSAP provide 27.5% energy savings at  $\lambda = 2.8$ .

## 7 Conclusions

Bus coding for crosstalk avoidance and error correction requires additional wires to satisfy the constraints on the code. In this paper, we have derived fundamental limits on the number of wires required for memoryless codes. We have addressed the two challenges in approaching the fundamental limits viz. code construction and codec complexity. Future work will be directed at finding code constructions that approach the fundamental limit without substantially increasing the codec complexity.

All joint codes trade-off delay and power dissipation in the bus with delay and power dissipation in the codec. This trade-off 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, the codes presented in this paper will result in greater reduction in energy dissipation and delay in SOCs of the future.

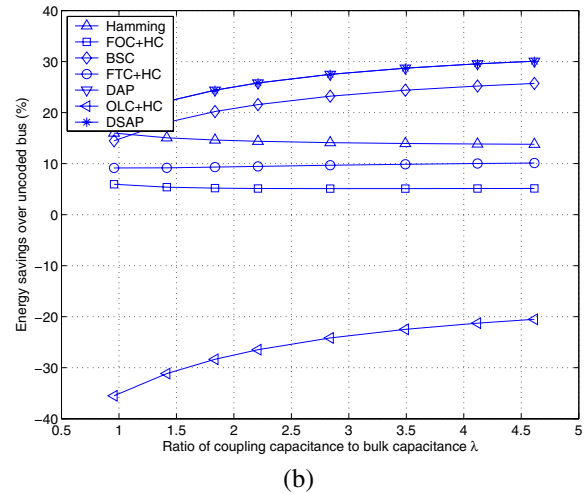
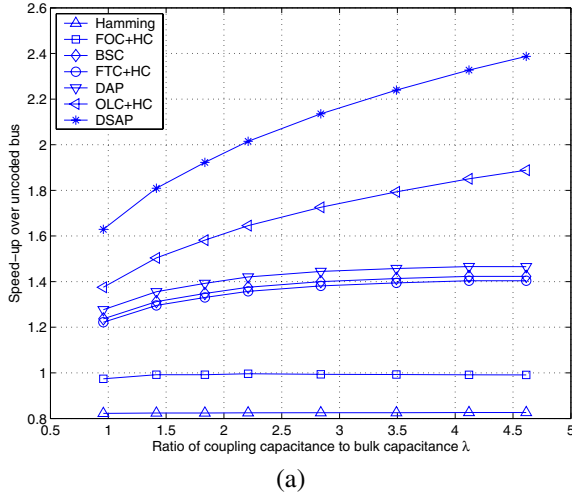


Figure 6. Improvements for 10-mm 32-bit bus as a function of  $\lambda$ : (a) speed-up and (b) energy savings.

## 8 Acknowledgments

This work was supported by the MARCO-sponsored Gigascale Systems Research Center and Intel Corporation.

## References

- [1] D. Liu and C. Svensson, "Power consumption estimation in CMOS VLSI chips," *IEEE J. Solid-State Circuits*, vol. 29, pp. 663–670, June 1994.
- [2] D. Sylvester and C. Hu, "Analytical modeling and characterization of deep-submicrometer interconnect," *Proc. IEEE*, vol. 89, pp. 634–664, May 2001.
- [3] H. Zhang, V. George, and J. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 264–272, June 2000.
- [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] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A coding framework for low-power address and data busses," *IEEE Trans. VLSI Syst.*, vol. 7, pp. 212–221, June 1999.
- [6] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. VLSI Syst.*, vol. 3, pp. 49–58, Mar. 1995.
- [7] 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.
- [8] 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.
- [9] P. P. Sotiriadis, "Interconnect modeling and optimization in deep sub-micron technologies," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, May 2002.
- [10] C. Duan, A. Tirumala, and S. P. Khatri, "Analysis and avoidance of cross-talk in on-chip buses," in *Proc. Hot Interconnects*, 2001, pp. 133–138.
- [11] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in *Proc. ICCAD*, 2001, pp. 57–63.
- [12] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag, "Area and energy-efficient crosstalk avoidance codes for on-chip buses," in *Proc. ICCD*, 2004.
- [13] D. Bertozzi, L. Benini, and G. D. Micheli, "Low power error resilient encoding for n-chip data buses," in *Proc. DATE*, 2002, pp. 102–109.
- [14] R. Hegde and N. R. Shanbhag, "Toward achieving energy efficiency in the presence of deep submicron noise," *IEEE Trans. VLSI Syst.*, vol. 8, pp. 379–391, Aug. 2000.
- [15] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York, NY: North-Holland, 1977.
- [16] S. R. Sridhara and N. R. Shanbhag, "Coding for system-on-chip networks: a unified framework," in *Proc. DAC*, 2004, pp. 103–106.
- [17] K. Patel and I. Markov, "Error-correction and crosstalk avoidance in DSM busses," in *Proc. SLIP*, 2003, pp. 9–14.
- [18] S. R. Sridhara and N. R. Shanbhag, "Coding for reliable on-chip buses: fundamental limits and practical codes," in preparation.