Open access • Journal Article • DOI:10.1049/IET-CDT.2013.0113

Partial coding algorithm for area and energy efficient crosstalk avoidance codes implementation — Source link

Basel Halak

Institutions: University of Southampton

Published on: 01 Mar 2014 - Iet Computers and Digital Techniques (The Institution of Engineering and Technology)

Topics: Repeater insertion

Partial Coding Algorithm for Area and Energy Efficient Crosstalk Avoidance Codes Implementation

Basel Halak, Southampton University, UK

Abstract. Modern interconnect performance is greatly affected by crosstalk noise due continuous decrease in wire separation and increase in its aspect ratio with technology scaling. Such noise is highly dependent on data transition patterns, coding techniques have been proposed to alleviate crosstalk delay by controlling these patterns. The complexity of available crosstalk avoidance codes, along with their associated overheads, increase rapidly with bus width. The lack of energy and area efficient method to implement such cods has so far prevented their use in practical designs. This paper presents a generic framework which allows efficient implementations of crosstalk avoidance codes; the essence of the proposed approach is based on the partial coding concept. Quantitative analysis performed in 32nm technology shows that substantial savings in area and energy costs can be obtained using the proposed technique compared with both existing coding solutions and conventional methods as shielding and repeater insertion.

1. Introduction

Crosstalk noise is defined as undesirable electromagnetic coupling between switching and non-switching signal lines. Such coupling can cause logic failures and timing degradation in digital systems. The interconnect performance and power consumption in current deep sub-micron technologies is greatly affected by crosstalk noise due to the decreasing wire separation and increased wire aspect ratio [1]. This trend is anticipated to worsen in the future due to the increase in systems complexity and the sharp rise in the number of metal layers in modern fabrication technologies, necessitating crosstalk-aware design methodologies for on-chip interconnects. The international technology roadmap for semiconductor (ITRS) report on interconnect indicated that crosstalk noise will become the dominant problem affecting propagation delays, hence degrading overall system performance [1]. Techniques to avoid the crosstalk delay have been proposed by many researchers [2-9]. These methods are applied on different levels of the design; physical level solutions include wire sizing optimization, shield insertion and repeater insertion [6, 10]. Other techniques are implemented on the data link layer, this consists of data encoding to avoid crosstalk [7]. The use of coding to combat crosstalk has been first reported in [2], the principle of this technique is to encode information bits such that certain data transition on the bus are avoided, which helps reduce or remove crosstalk noise, this method assume a regular structure of the communication link which consists of parallel wires such as those present in networks on chips architectures[11]. The use of coding is a very attractive method as it is said to require less area overheads compared to conventional techniques such as shield insertion [2, 5]. Recent work in [12] has also show that crosstalk avoidance codes can be employed to maximising the throughput of communication links at relatively low energy and area costs. Another advantage of coding is that it may remove the need for using power hungry repeaters in order to meet the design timing requirements [13]. However, the complexity of the codec circuitry, especially for wide buses, has so far prevented the adoption of crosstalk avoidance codes in mainstream designs. Extensive researches have been carried out in order to develop energy and area efficient codes [7, 9, 14-18]. Most recently, the authors of [19, 20] have devised crosstalk avoidance codes based on numeral system, where in the complexity of their proposed codecs, in terms number of gates, increase quadratically with the width of the bus compared the exponential increase seen by other coding strategies. Although this is a significant improvement, the area and energy consumption reported for the numeral-based coding logic for buses wider than 8 bit is still prohibitive. This in practice may render these coding approaches impractical in modern microprocessor designs where in the width of the buses is usually 32 bits or more. One solution to harness the benefits of coding methods for larger buses at relatively low cost is to employ partial coding techniques. This effectively means dividing large links into sub-channels and separately encoding the data on each sub-channel. The use of partial coding has been mentioned by several papers [2, 3, 12, 20], however, to the best of our knowledge, no formal method have been devised to achieve area and power efficient partial coding of wide buses. This paper develops for the first time a new algorithm which allows cost effective implementation of crosstalk avoidance codes on wide buses. A quantitative comparison of overheads is made between the proposed solution and conventional techniques (existing coding schemes, shielding, repeater insertion). The results show the proposed approach achieves the same bus speed up with lower area and energy consumption in most cases. The organisation of this paper is as follows. The principle of crosstalk avoidance codes are reviewed in section 2. Section 3 explains the rational of the proposed algorithms. Section 4 outlines the derivations of area and energy cost function of crosstalk avoidance codes. In section 5, partial coding algorithms are presented. Section 6 summarises area and energy results obtained by applying the proposed algorithm on various networks on chips architectures, Conclusions are drawn in section 7.

2. The Principles of Crosstalk Avoidance Codes (CACs)

Wire delay is a function of the wire geometry (length, width), separation from its adjacent wires, its electrical characteristic and the data transition activity on the bus. RC model is usually used to estimate wire delay using the formulae[13]:

$$D = 0.4\, R.C_g\, (1+ p.\lambda) \tag{1}$$

Where: R is wire resistance, C_g the loading capacitance between the wire and to the ground, lambda (λ) is the ratio between is the ratio of the coupling capacitance between adjacent wires and the ground capacitance (C_g/C_c). *p* is the coupling factor which depends on the transition activity on the wire and its adjacent lines. Table 1 summarise the values of *p* on the middle lines in three wire bus for different transition patterns. Note that ↑, ↓, and −, denote 0-to-1, 1-to-0, and no transitions, respectively. *d0* is the delay of a crosstalk-free line.

Table 1: Wire Delay for Different Crosstalk Cases

Crosstalk Case	Transitions	The Coupling Factor p	The Middle Wire Delay
1	↑↑↑, ↓↓↓	0	d0
2	↑↑- ,↓↓-, -↑↑, -↓↓	1	$(1+\lambda)d0$
3	-↑- , -↓-, ↓↑↑ , ↑↓↓ , ↑↑↓, ↓↓↑,	2	$(1+2\lambda)d0$
4	↓↑-, -↓↑, -↑↓, ↑↓-	3	$(1+3\lambda)d0$
5	↓↑↓, ↑↓↑	4	$(1+4\lambda)d0$

Coding is the process of mapping *k* information bits into *n bits* codewords such that the codewords exhibit certain desired properties. To reduce the impact of crosstalk on delay, noise and power, any two codewords following one another on the bus should not have transitions that lead to a large effective coupling capacitance ($p.C_c$). This can be achieved by either avoiding specific data patterns [21] or avoiding opposing transitions on adjacent wires [2]. Crosstalk avoidance codes are written as *C (N, K, p)*, where K is the number of information bits to be encoded onto N wire such that worst case crosstalk capacitance is ($p.C_c$). These codes are classified into three main categories based on the worst case coupling factor they achieve into:

2.1. CACs (p=2): *C (N, K, 3)*

These codes reduce the worst case coupling capacitance to 3Cc *(p=3)* by ensuring that transitions ↓↑↓ and ↑↓↑ are avoided. This can be done if and only if a codeword having the bit pattern *010* does not transition to another codewords having the pattern 101 at the same bit position and vice versa. Example of such codes is forbidden overlap codes (FOC)[21].

2.2. CACs (p=2): *C (N, K, 2)*

These codes reduced the worst case coupling capacitance to 2Cc *(p=2)* in one of the following methods [3]:

- 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).
- By avoiding bit patterns *010* and *101* from every codeword. This condition is referred to as forbidden pattern (FP) condition and codes that satisfy the FP condition are called forbidden pattern codes (FPC).

a. CACs (p=1): *C (N, K, 1)*

These codes reduce the worst case coupling capacitance to Cc *(p=1)* by ensuring that transitions ↓↑↓. The maximum capacitive coupling factor can be reduced to *(p=1)* if and only if the transitions ↓↑×, −↑−, and ↑−↑ (and their complements) are avoided. In order to achieve, the codebook should satisfies the following three conditions [3, 13]:

- The forbidden transition (FT) condition explained in the previous section
- The forbidden pattern (FP) condition explained in the previous section
- Forbidden adjacent boundary pattern (FABP) condition which states that any 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).

3. Partial Coding Principle

The problem of finding a valid solution to partially encode a set of binary data can be formulated as follows:

Requirements: Let M be the number of information bits to be transmitted simultaneously over a link such that the maximum crosstalk coupling factor, caused by data transition, does not exceed a target value (α) .

Problem: Let C_M be the set of available crosstalk avoidance codes written as follows:

$$C_M = \{C(N,K,p): 1 \leq K \leq M,\ p = \alpha\} \quad (2)$$

Find a subset $s_i \in C_M$ such that the above requirements are satisfied.

To solve the above problem, the following lemma is devised:

Lemma 1: Let $s \in C_M$ be a group of CACs that which can collectively encode M bits over a link such that crosstalk coupling factor does not exceed a target value α, s can be written as follows:

$$s = \left\{C_i(N_i, K_i, p_i\ ); \sum_{i=1}^{T} k_i = M\ ; p_i = \alpha\right\} \quad (3)$$

Where T is the number of codes in the subset s, k_i is the number of information bits of each code.

Example 1: Consider an 8 information bit ($M = 8$) to be transmitted simultaneously over a bus such that the worst case crosstalk coupling factor ($p = 3$). The available crosstalk codes in this case are[20]:

$$C_8 = \{C(N,K,p): 1 \leq K \leq 8,\ p = 3\}$$

$$C_8 = \{C(4,3,3), C(5,4,3), C(6,5,3), C(7,6,3), C(8,7,3), C(9,8,3)\}$$

By applying lemma 1 we can find all possible ways we can construct an 8 bit bus using codes from C_8 . In this case we only have three possibilities:

$s_1 = \{C(4,3,3), C(6,5,3)\}$

$s_2 = \{C(5,4,3), C(5,4,3)\}$

$s_3 = \{C(9,8,3)\}$

Choosing s_1 effectively means the division of the channel into two parts ($T_1 = 2$), one has 3 bits and the other five bits, and encoding each part separately.

Definition 1: Let $S = \{s_j: 1 \leq j \leq E(S)\}$ be the design space which includes all possible solutions as defined by lemma 1. E(S) denotes the number of element in the solution space S and j is an integer index. For example the solution space for example (1) is as follows:

$S = \{s_1, s_2, s_3\}$, E(S) =3

It should be noted here that the problem of finding the elements the space S is a special case of well-known mathematical problem of finding partitions for an integer number M [22].

4. Derivation of Cost Functions for CACs

Lemma 1 can be used to identify all elements of the design space S, however; in order find the optimum solution; there needs to be a cost function associated with each of these elements. Two cases have been considered, namely: area constrained and energy constrained buses. Based on this classification, the following cost functions have been devised.

4.1. Area Cost Functions

Consider a link with M information bits to be sent over wires with a length (*l*) and a width (*w*). The area overheads incurred by CACs include logic area for the coding/decoding circuitry and the extra wires needed to encode the data. When partial coding is used, additional lines are needed to separate sub-channels in order to prevent forbidden data transition on the channels edges. The overheads of these extra wires should also be taken into consideration.

Let $s = \{C_i(N_i, K_i, p_i); \sum_{i=1}^{T} k_i = M; p_i = \alpha\}$ be an element of the design space:

$\boldsymbol{S} = \{\boldsymbol{s_j}: 1 \leq j \leq E(S)\}$

The area overhead cost (AO) of each of these elements is defined below:

$$\boldsymbol{AO}(\boldsymbol{s_j}) = \frac{A_C + A_{BC}}{A_{BU}} \qquad (4)$$

Where:

A_{BU} : is the area of the un-coded link given as follows:

$$A_{BU} = (2M - 1) * w * l \qquad (5)$$

A_C : is the total area of the coding/decoding circuits calculated as the sum of the T codecs:

$$A_C = \sum_{i=1}^{T} AC_i \tag{6}$$

A_{BC} : is the area of the coded link

$$A_{BC} = (2(N + T - 1) - 1) * w * l \tag{7}$$

Where:

$$N = \sum_{i=1}^{T} N_i \tag{8}$$

(*T-1*) extra wires will need to be inserted between sub-links in order to prevent any forbidden transitions on the edges of sub-channels.

4.2. Energy Cost Functions

Let us consider a link with M information bits transmitted over a set of wires, which has a ground capacitance (C_g), an inter-wire capacitance (C_c), and voltage swing (V_{dd}). The application of CACs on such a link may lead to a reduction in its energy dissipation by lowering the effective crosstalk capacitance between adjacent wires ($p.C_c$). However, while computing the energy cost function, the overheads caused by the coding schemes must also be taken into account. The coding schemes require extra wires, which will increase the overall capacitance of the bus and hence lead to more energy dissipation. The encoders and decoders circuitry also dissipate energy, which should also be included in the computation of the energy costs.

Let $s = \{C_i(N_i, K_i, p_i\ ); \sum_{i=1}^{T} k_i = M\ ; p_i = \alpha\}$ be an element of the design space $\{s_j : 1 \leq j \leq E(S)\}$.

The energy consumption overhead (EO) of for each of these elements is defined below:

$$EO(\boldsymbol{s_j}) = \frac{E_{coded} + E_{BC}}{E_{BU}} \tag{9}$$

Where:

E_{BU} : is the average energy consumed on an un-coded bus, which depends on the statistical distribution of data and is given as follows [23]:

$$E_{BU} = tr\ (C_T\ A) * {V_{dd}}^2 \tag{10}$$

Where *A* is the transition activity matrix, *tr(X)* is the trace of matrix *X* and C_T is a (*M x M)* capacitance matrix given by:

$$C_T = \begin{bmatrix} 1+\lambda & -\lambda & 0 & .. & 0 \\ -\lambda & 1+2\lambda & -\lambda & .. & 0 \\ 0 & -\lambda & .. & .. & : \\ : & : & : & 1+2\lambda & -\lambda \\ 0 & 0 & .. & -\lambda & 1+\lambda \end{bmatrix} * C_g \tag{11}$$

E_{BC} : Average energy consumed on a on coded bus, this can also be calculated using equation (10), but ins this case $C_T$is a *(N x N) matrix,*

Where:

$$N = \sum_{i=1}^{T} N_i \qquad (12)$$

E_{codec} : is the total power consumed by the coding/decoding logic, calculated as the sum of energy dissipated by of all encoders/decoders used.

$$E_{codec} = \sum_{i=1}^{T} E_{codec[i]} \qquad (13)$$

Where T is the number of codes in the subset s

$E_{codec[i]}$: is the power consumed by the coding/decoding circuits of each partial code

5. Partial Coding Algorithms

This section outlines the proposed algorithms as follows.

5.1. Area Cost Minimisation Algorithm:

Requirement*:* find a set of codes ($s = \{C_i(N_i, K_i, p_i); \sum_{i=1}^{T} k_i = M ; p_i = \alpha\}$) which collectively encode M information bits and avoid all data transition which results in($p > \alpha$), such that area cost $AO(\boldsymbol{s_j})$ is minimised.

The algorithm should carry out two main tasks, the first is to find all valid partial coding sets as described in equation 3, and the second is to find the set with the minimum area overheads. To achieve the first task the algorithm employs lemma 1 described above, this is done using a set of variables $\{i_k: 1 \leq k \leq M\}$ where i_k represent the number of codec of a crosstalk avoidance code $C(N, k, \alpha)$. For example considering CACs with (p=3), than i_4 represent the number of the crosstalk avoidance code ($C(5,4,3)$) used in a partial coding set. It should be noted here that according to lemma 1, the sum of all variable i_k is equal to M in all valid partial coding sets. It is also worth noting that the value range of the index of variable i vary depending on the type of crosstalk avoidance codes used. For example there is no crosstalk avoidance codes which can encode 1 bit only to achieve (p=3), in this case the minimum width of the sub-channel is 2, which correspond to using half-shield insertion, whereas it is possible to devise a crosstalk avoidance codes with (p=2) to encode one bit only, this effectively means using shield insertion.

The total number of valid solutions is stored in a variable called E(S); the second task of the algorithm is achieved through a numerical search for the set with the minimum area cost. Finally the initial value for the area cost is chosen to be very large compared to a typical area cost based on our estimations in section 6. The pseudo code of the optimization algorithm is presented below. An example code for (M=10, p=3) is shown in appendix 1.

Algorithm 1:

```
Begin
    **find valid solutions (S)
   j=0
   For ($i_1 = 1; i_1 \leq \binom{M}{i_1}; i_2 + +$); ($i_2 = 1; i_2 \leq \binom{M}{i_2}; i_2 + +$)); …….($i_m = 1; i_m \leq 1; i_m + +$))
   Begin
          $if\ \left((i_1 + 2i_2 + 3i_3 + \cdots + Mi_M) = M\right)$
          Begin
          $j + +$
          $s[j] = \{i_1, i_2, i_3, \dots \dots, i_M\}$
         End
         $E(S) = j$
   End
  **find the optimum partial coding set ($s_{OPT}$)
  $AO_{OPT} = 100$
  $s_{opt} = s[1]$
 For ( $j = 1; j \leq E(S); j + +$);
 Begin
        If ($AO(s[j]) \leq AO_{OPT}$) then:
        $s_{OPT} = s[j]$
 End
Output $s_{OPT}$

End
```

5.2. Energy Cost Minimisation Algorithm:

Requirement*:* find a set of codes ($s = \left\{C_i(N_i, K_i, p_i\ ); \sum_{i=1}^{T} k_i = M\ ; p_i = \alpha\right\}$) which collectively encode M information bits and avoid all data transition which results in($p > \alpha$), such that area cost $EO(\boldsymbol{s_j})$ is minimised.

The pseudo code of the optimization algorithm is presented below.

Algorithm 2:

```
Begin
    **find valid solutions (S)
   j=0
   For ($i_1 = 1; i_1 \leq \binom{M}{i_1}; i_2 + +$); ($i_2 = 1; i_2 \leq \binom{M}{i_2}; i_2 + +$)); …….($i_m = 1; i_m \leq 1; i_m + +$))
   Begin
          $if\ \left((i_1 + 2i_2 + 3i_3 + \cdots + Mi_M) = M\right)$
          Begin
          $j + +$
          $s[j] = \{i_1, i_2, i_3, \dots \dots, i_M\}$
         End
         $E(S) = j$
   End
  **find the optimum partial coding set ($s_{OPT}$)
```

```
        $EO_{OPT} = 100$
        $s_{opt} = s[j]$
        $For\ (j = 1; j \leq E(S); j++);$
        $Begin$
              $If\ (EO(s_j) \leq EO_{OPT})\ then:$
              $s_{OPT} = s[j]$
        $End$
      $Output\ \ s_{OPT}$

  $End$
```

5.3. Algorithm Modification for Massively Wide Buses

The algorithm presented above perform search on all valid sets S= $\{s_j: 1 \leq j \leq E(S)\}$, in order to find the best option, this, however, may not be computationally efficient or even feasible for massive buses as the number of valid sets $(E(S))$ rises very sharply as the width of the bus increases. This is explained in lemma 2 below.

Lemma 2: $\boldsymbol{S} = \{\boldsymbol{s_j}: 1 \leq j \leq E(S)\}$ be the design space associated with a bus with M number of information bits . Let par(M) denotes the number of partitions of the an integer number M [22]. Then the upper bound of $E(S)$ is $par(M)$ (i.e.$E(S) \leq pAR(M)$)

Proof: See appendix 2.

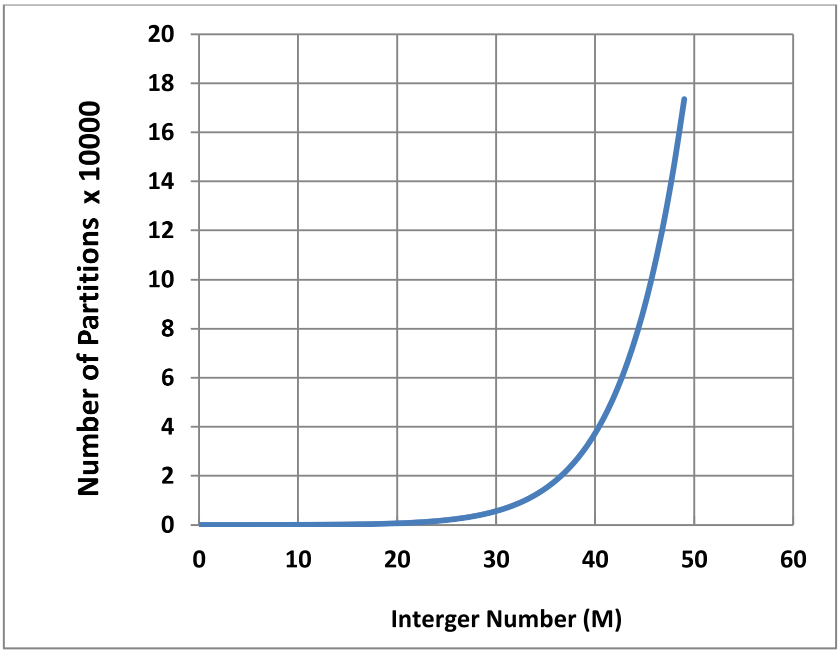


Figure 1: Valid Partial Coding Solutions Upper Bound

Figure 1 shows the upper bound on E(S) for different bus widths (M). Although, the actual size of possible solutions E(S) may be slightly less than par(M), but both figures will have the same order of magnitude. This trend makes the task of searching through this space computationally very intensive, for example for a 32 bits bus, there are more than 10000 possible valid partial coding sets.

To make this task easier, the search space should be limited to a subset of all valid sets, this should be done without leading to a less optimum solution. One way to achieve this is to specify an upper limit on the size of the codebook used. This method can be very effective in eliminating valid sets with large size codebooks, which are usually associated with high area and power costs. In fact our results presented in section 6 showed that valid partial coding sets which include large size codes (> 20) do not in any case present an optimum solution.

It should be noted here that the upper limit on the size of code will be different form one code to another depending on the type of implementation. A pseudo code is shown below for a modified area cost minimisation algorithm for an M bit bus with a maximum codebook size (U< M)

Algorithm 3:

```
Begin
     **find valid solutions (S)
    j=0
    For ($i_1 = 1; i_1 \le \left(\frac{M}{i_1}\right); i_2 + +$); ($i_2 = 1; i_2 \le \left(\frac{M}{i_2}\right); i_2 + +$)); .......($i_U = 1; i_U \le 1; i_U + +$))
    Begin
            if $\left((i_1 + 2i_2 + 3i_3 + \cdots + Ui_U) = M\right)$
            Begin
            $j + +$
            $s[j] = \{i_1, i_2, i_3, \dots \dots, i_U\}$
          End
          $E(S) = j$
   End
   **find the optimum partial coding set ($s_{OPT}$)
  $AO_{OPT} = 100$
  $s_{opt} = s[j]$
 For ( $j = 1; i_1 \le E(S); J + +$);
 Begin
         If ($AO(s[j]) \le AO_{OPT}$) then:
         $s_{OPT} = s[j]$
 End
Output $s_{OPT}$

End
```

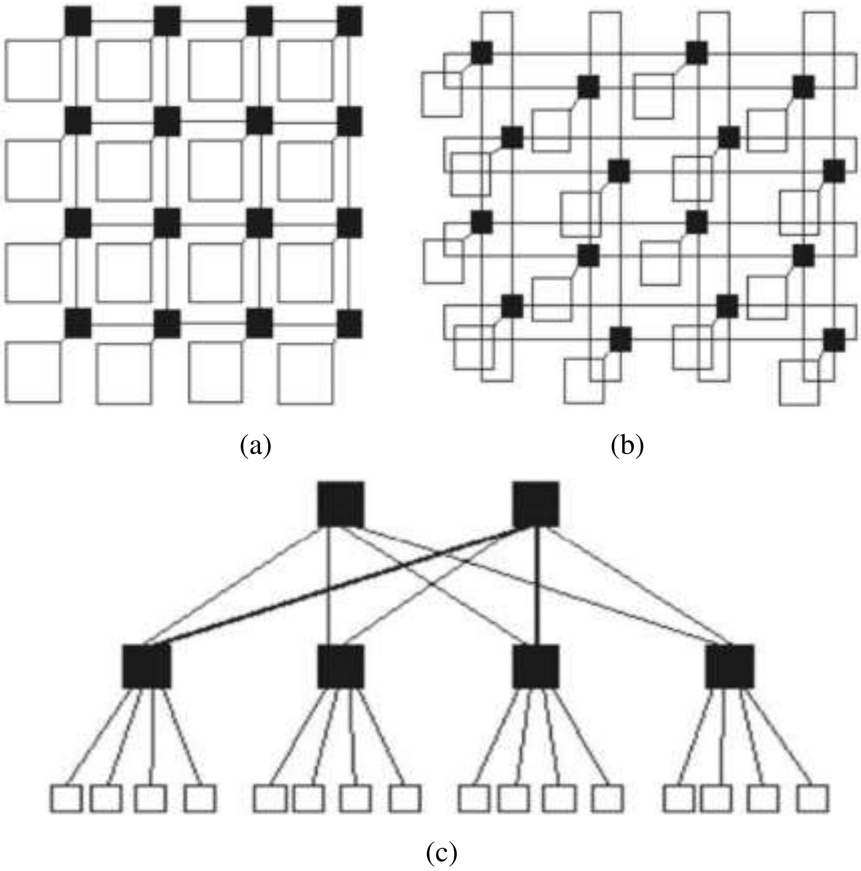
---

6. Results and Discussion

6.1. Experimental and Simulation Setups

This sections presents the energy and area savings obtained by employing the proposed partial coding. The achieved improvements depends on wires length (*L*), bus width(*K*) and the ratio of crosstalk capacitance to ground capacitance(λ), and the process technology. The results are obtained using a standard 32 nm CMOS technology for various values of (*L*) and (*K*). A bus laid out on Metal 8 layer was considered, with minimum wire width and spacing (0.5μm), the value of (λ) is this case was calculated to be (1.75). The coding circuitry (i.e. encoders and decoders), are synthesized using a 32nm CMOS standard cell library from Synopsys, The power and area overheads are obtained from synthesized gate level netlists. The supply voltage used is (1v).

Crosstalk avoidance codes is intended for use with straight buses with parallel wires [2], therefore it is especially relevant to networks on chips (NoC) architectures which are based on parallel inter-switch links [24]. Therefore the proposed algorithms are applied to three of the most widely used NoC topologies as shown in figure 2. Shaded blocks are network routers, and transparent ones are the IP nodes.



(a) (b)

(c)

Figure 2: Network on Chip Topologies (a) mesh; (b) folded torus; (c) butterfly-fat-tree

For each case under consideration the wire lengths of the inter-switch links have been calculated. For mesh-based topology, this length is given by the equation below [24]:

$$\boldsymbol{l} = \frac{\sqrt{\boldsymbol{Area}}}{\sqrt{\boldsymbol{P}}-\mathbf{1}} \qquad (14)$$

Where ''Area'' is the area of the silicon die used, and P is the number of intellectual property (IP) blocks in the SOC. The wire length of inter-switch links in folded-torus architecture is twice that of the mesh [24]

The inters-witch wire length for the BFT architecture between level a and a+1 is given by equation below [24]:

$$\boldsymbol{l}_{\boldsymbol{a},\boldsymbol{a}+\mathbf{1}} = \frac{\sqrt{\boldsymbol{Area}}}{\mathbf{2}^{\boldsymbol{levels}-\mathbf{1}}} \qquad (15)$$

Where *levels* is the total number of levels in the BFT architecture given by $\log_4 P$

In order to characterize the performance of the proposed partial coding schemes in NOC communication infrastructures, a system consisting of 64 IP blocks has been considered, and mapped them onto mesh, folded torus. The network is assumed be spread over a die size of (10 mm x10 mm). The wire length of each network topology was calculated accordingly. In the case of BFT network, we have considered links between level 2 and 3.

For this analysis, crosstalk avoidance codes based on numeral systems have been considered [2, 18, 19, 25]. Finally, algorithms 1 and 2 have applied to find the optimum solutions. The area and cost functions of partial coding methods are compared to those of full coding techniques (.i.e. encoding all bits at once). They are also compared to the overheads of an equivalent shielding method which achieves the same reduction in the effective crosstalk capacitance. For FOC CACs code, the equivalent shielding technique is half-shield insertion, for FPC CACs, the equivalent shielding technique is shield insertion, and finally for OLC CACs, the equivalent shielding technique is duplication and shield insertion. The results are summarised in Tables 2, 3, 4 &5. The first column of the table includes the different types of coding classified according to the reduction they can achieve in the effective crosstalk capacitance (see section 2). The second column includes the network types, which differ in the wire length (*l*) (1.42mm for mesh, 2.84 for folded torus, 5mm for BFT). The third column shows the optimum partial coding solutions obtained. And the fourth column includes overhead comparison.

Table 2: Area- Efficient Partial Crosstalk Avoidance Coding Solutions for 16 bits Links

Crosstalk Coding Type	Network Type	Optimum Partial Coding Solutions For Numeral Based CACs (S_{OPT})	Area Cost		
			Partial Coding $AO(S_{OPT})$	Full Coding	Equivalent Shielding Method
FOC	Mesh	{4C(5,4,3)}	1.48	1.64	1.5
	Folded Torus	{C(9,7,3),C(12,9,3)	1.44	1.47	
	BFT	{C(21,16, 3)}	1.4	1.4	
FPC	Mesh	{4C(5,4,2)}	1.53	2.1	2
	Folded Torus	{4C(5,4,2)}	1.48	1.8	
	BFT	{4C(5,4,2)}	1.46	1.61	
OLC	Mesh	{4C(6,3,1), C(8,4,1)}	2.45	4.5	3
	Folded Torus	{4C(6,3,1), C(8,4,1)}	2.3	3.5	
	BFT	{4C(8,4,1)}	2.2	3	

Table 3: Energy- Efficient Partial Crosstalk Avoidance Coding Solutions for 16 bits Links

Crosstalk Coding Type	Network Type	Optimum Partial Coding Solutions For Numeral Based CACs (S_{OPT})	Energy Cost		
			Partial Coding $PO(S_{OPT})$	Full Coding	Equivalent Shielding Method
FOC	Mesh	{4C(5,4,3)}	1.48	5.2	0.99
	Folded Torus	{2C(7,5,3),C(8,6,3)}	1.18	3.1	
	BFT	{2C(7,5,3), C(8,6,3)}	1.01	2.11	
FPC	Mesh	{4C(5,4,2) }	1.5	8.6	1.04
	Folded Torus	{{4C(5,4,2) }	1.12	4.7	
	BFT	{4C(5,4,2) }	0.94	3.3	
OLC	Mesh	{4C(6,3,1), C(8,4,1)}	3.31	33	1.28
	Folded Torus	{4C(6,3,1), C(8,4,1)}	2	17.9	
	BFT	{4C(6,3,1), C(8,4,1)}	1.43	10	

Table 4: Area-Efficient Partial Crosstalk Avoidance Coding Solutions for 32 bits Links

Crosstalk Coding Type	Network Type	Optimum Partial Coding Solutions For Numeral Based CACs (S_{OPT})	Area Cost		
			Partial Coding $AO(S_{OPT})$	Full Coding	Equivalent Shielding Method
FOC	Mesh	{8C(5,4,3}	1.51	1.9	1.5
	Folded Torus	{C(25,20,3), C(15,12,3)}	1.43	1.58	
	BFT	C(25,20,3), C(15,12,3)}	1.37	1.43	
FPC	Mesh	{8C(5,4,2)}	1.51	2.1	2
	Folded Torus	{8C(5,4,2)}	1.49	1.77	
	BFT	{8C(5,4,2)}	1.48	1.62	
OLC	Mesh	8{C(8,4,1)}	2.48	7.5	3
	Folded Torus	8{C(8,4,1)}	2.35	5	
	BFT	8{C(8,4,1)}	2.3	3.9	

Table 5: Energy-Efficient Partial Crosstalk Avoidance Coding Solutions for 32 bits Links

Crosstalk Coding Type	Network Type	Optimum Partial Coding Solutions For Numeral Based CACs (S_{OPT})	Energy Cost		
			Partial Coding $PO(S_{OPT})$	Full Coding	Equivalent Shielding Method
FOC	Mesh	{4C(5,4,3)}	1.48	10.6	0.99
	Folded Torus	{4C(7,5,3), 2C(8,6,3)}	1.18	5.8	
	BFT	{4C(7,5,3), 2C(8,6,3)}	1.01	3.7	
FPC	Mesh	{8C(5,4,2)}	1.54	18.6	1.04
	Folded Torus	{8C(5,4,2)}	1.12	9.7	
	BFT	{8C(5,4,2)}	0.94	5.8	
OLC	Mesh	{8C(6,3,1),2 C(8,4,1)}	3.3	78	1.28
	Folded Torus	{8C(6,3,1),2 C(8,4,1)}	2	39.6	
	BFT	{8C(6,3,1),2 C(8,4,1)}	1.43	23	

6.2. Energy and Area Cost Analysis

First, the effect of applying partial coding algorithm on the area cost is examined.

Consider the case of FOC CACs, it can be seen, from tables (2 and 4), that up to 40% area saving can be obtained compared to a full coding approach, this however depends on the width of the bus and the type of the network. The wider the bus, the more costly the full coding approach becomes, as the codec area increases quadratically with number of bits. For partial coding increasing the width of the bus entails using more encoder, which means the increase in area overheads tend to be linear. The type of the network architectures affects the length of the buses. In general, the longer the bus, the less significant the overheads of the codec circuitry becomes compared to the wiring overheads, this effectively means that area saving obtained by partial coding compared to that of full coding is reduced. These trends are reflected in tables (2 & 4), with the maximum saving achieved for the shortest and widest bus (i.e. mesh based topology and 32 bit bus). The other extreme is no savings at all for the narrowest and longest bus (BFT based topology and 16 bit bus).

The same conclusions can be drawn for FPC and OLC CACs, in fact a closer examinations of the effects of the type of codes on the area overheads, reveals that the higher the reduction in effective crosstalk capacitance the crosstalk codes achieve, the larger the saving that can be obtained from the partial coding compared to full coding, this is because the larger the reduction in coupling capacitance a CAC achieves, the higher its area cost will be, therefore partial coding method achieves the highest area saving when applied to OLC CACs. Finally, it can be noted that the proposed partial coding approach achieves, in all cases, significantly less area overheads (up to 70% reduction) compared to their respective equivalent shielding methods.

Second, the effect of applying partial coding algorithm on the energy cost is examined.

Tables (3 and 5) shows that significant savings in the energy dissipation can be obtained using the partial coding methods compared to a full coding approach, these savings are larger for wider buses and shorter links for the same reason discussed above.

Shielding equivalent method can in all cases under consideration achieve better results as they have no energy overheads for coding circuitry, this is however at the cost of increased area overheads as seen in Tables 2 &4.

The energy costs of the optimal partial coding solutions are also compared to those of optimal repeater insertion as shown in table 6, the comparison also include the bus delay of each technique normalised to un-coded bus. Results, from table 6, show that FOC and

FPC coding approaches have less energy costs than that of repeater insertion. The energy efficiency of OLC technique compared to repeater insertion is dependent on wire length. In general, energy costs of coding approach compared to un-coded bus decrease with wire lengths, as the longer the bus, the less significant the energy dissipation of the codec circuitry becomes compared to the energy consumed by data transition on the bus. On the other hand, the energy cost of repeater insertion compared to un-buffered bus independent of bus length. Therefore, the longer the bus the more energy efficient coding becomes compared to repeater insertion. An important metric that should be considered is the energy-delay product, which encompasses both energy cost and the achievable delay reduction achievable by each method. The results from table 6 show that coding archived lower energy-delay product in all cases under consideration compared to repeater insertion. Finally it should be noted that It should be noted here that energy results will be different for another value of (λ), in general, the higher λ is the more energy saving is expected by using crosstalk avoidance methods.

Table 6: Partial Crosstalk Avoidance Coding vs. Repeater Insertion

Design Technique	Network Type	Energy Cost	Normalised Delay	Energy-Delay Product
Optimal Partial Coding FOC	Mesh	1.48	0.78	1.1544
	Folded Torus	1.18		0.9204
	BFT	1.01		0.7878
Optimal Partial Coding FPC	Mesh	1.5	0.56	0.84
	Folded Torus	1.12		0.6272
	BFT	0.94		0.5264
Optimal Partial Coding OLC	Mesh	3.31	0.34	1.1254
	Folded Torus	2		0.68
	BFT	1.43		0.4862
Optimal Repeater Insertion	Mesh	1.93	0.9	1.737
	Folded Torus	1.93	0.49	0.9457
	BFT	1.88	0.28	0.5264

7. Conclusions

Partial coding algorithm has been proposed to allow area and energy efficient implementation of crosstalk avoidance codes. Quantitative analysis in 32nm technology has shown that the proposed approach can achieve in most cases the same bus speed up but with lower energy and area overheads compared to existing coding schemes, shielding and repeater insertion. The current technology scaling trend leads to more crosstalk noise; this indicates that crosstalk avoidance methods will become an essential tool for IC designers. In addition, the aggressive drive for high performance systems to satisfy growing consumer application requirements for high speed means that on-chip buses will need to run faster and carry more data, hence, become wider. Therefore it is anticipated that the proposed partial coding approach will play an increasingly important role in achieving the speed requirement at low area and energy costs.

8. Appendixes

8.1. Appendix 1

An example algorithm is presented here for area minimisation for crosstalk avoidance codes with (p=3) applied on a bus with effective width (M=8). The available crosstalk codes in this case are:

$$C_8 = \{C(N,K,p): 1 \le K \le 8,\ p = 3\}$$
$$C_8 = \{C(4,3,3), C(5,4,3), C(6,5,3), C(7,6,3), C(8,7,3), C(9,8,3)\}$$

The algorithm is shown below

```
Begin
   $M = 8$
   $AO_{min} = 100$
    **find valid solutions (S)
    j=0
    For $(i_3 = 1; i_3 \le \left(\frac{M}{i_3}\right); i_3 ++); (i_4 = 1; i_4 \le \left(\frac{M}{i_4}\right); i_4 ++)); (i_5 = 1; i_5 \le \left(\frac{M}{i_5}\right); i_5 ++));$
        $(i_6 = 1; i_6 \le \left(\frac{M}{i_6}\right); i_6 ++)) (i_7 = 1; i_7 \le \left(\frac{M}{i_7}\right); i_7 ++)); (i_8 = 1; i_8 \le \left(\frac{M}{i_8}\right); i_8 ++));$
    Begin
          if $(3i_3 + 4i_4 + 5i_5 + 6i_6 + 7i_7 + 8i_8 = M)$
          Begin
          $j++$
          $s[j] = \{i_3, i_4, i_5, i_6, i_7, i_8\}$
         End
         $E(S) = j$
    End
   **find the optimum partial coding set $(s_{OPT})$
   $AO_{max} = 100$
   $s_{opt} = s_1$
  For $(J = 1; J \le E(S); J++)$;
  Begin
         If $(AO(s_j) \le AO_{max})$ then:
         $s_{OPT} = s_j$
  End
```

End

8.2. Appendix 2: proof of lemma 2

Lemma 1 shows that E(S) represents the number of ways we can divide M bits into smaller parts be encoded separately, each of these parts may have different number of bits K_i .

On the other p(M) represent the number of ways we can write M as the sum of positive integers [22]. The two definitions can be identical if we assume that the number of bits in each sub-link K_i can take any value between 1 and M. However; the values choses for each K_i should be based on the available crosstalk avoidance codes. For example for CAC (P=2), there are codes for all possible integers (e.g. k=1, use shielding technique). However for CAC(p=1), it is not possible to find codes for k=1

This means there are restrictions on the number of bits (K_i) in each sub-link, these restriction will vary based on the type of CAC under consideration.

Based on the above discussion, It can be concluded that $E(S)$ has an upper limit bound to $par(M)$, which can only be reached if there are CAC able to encode any number of information bits $K_i \leq M$.

9. References

1. International Technology Roadmap for Semiconductors (www.itrs.net).
2. Victor, B. and K. Keutzer, Bus encoding to prevent crosstalk delay. IEEE/ACM International Conference on Computer Aided Design, 2001: p. 57-63.
3. Sridhara, S.R. and N.R. Shanbhag, Coding for Reliable On-Chip Buses: A Class of Fundamental Bounds and Practical Codes. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 2007. 26(5): p. 977-982.
4. Zhang, Q., J. Wang, and Y. Ye, Delay and Energy Efficient Design of On-Chip Encoded Bus with Repeaters. International Conference on VLSI Design, 2008: p. 377-382.
5. Sridhara, S.R. and N.R. Shanbhag, Coding for system-on-chip networks: a unified framework. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2005: p. 655-667.
6. Shah, H., et al., Repeater insertion and wire sizing optimization for throughput-centric VLSI global interconnects. IEEE/ACM International Conference on Computer Aided Design, 2002: p. 280-284.
7. Vaidyanathan, B. and Y. Xie, Crosstalk-Aware Energy Efficient Encoding for Instruction Bus through Code Compression. IEEE International SOC Conference, 2006: p. 193-196.
8. Sridhara, S.R., A. Ahmed, and N.R. Shanbhag, Area and energy-efficient crosstalk avoidance codes for on-chip buses. IEEE International Conference on Computer Design, 2004: p. 12-17.
9. Halak, B. and A. Yakovlev, Fault-Tolerant Techniques to Minimize the Impact of Crosstalk on Phase Encoded Communication Channels. Computers, IEEE Transactions on, 2008. 57(4): p. 505-519.
10. Pamunuwa, D., L.-R. Zheng, and H. Tenhunen, Maximizing throughput over parallel wire structures in the deep submicrometer regime. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2003. 11(2): p. 224-243.
11. Goossens, K., J. Dielissen, and A. Radulescu, AEthereal network on chip: concepts, architectures, and implementations. Design & Test of Computers, IEEE, 2005. 22(5): p. 414-421.

12. Halak, B. and A. Yakovlev, Throghput Optimisation for Area-Constrained Links with Crosstalk Avoidance Methods. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, (Forth Coming Issue).
13. Halak, B., The Analysis and Optimization of Performance and Reliability Metrics of Capacitive Links in Deep Submicron Semiconductor Technologies, in Electronic , Electrical and Computer School. 2009, Newcastle University: Newcastle upon Tyne.
14. Zhang, Q., J. Wang, and Y. Ye, Low-Power Crosstalk Avoidance Encoding for On-Chip Data Buses. IEEE Asia Pacific Conference on Circuits and Systems, 2006: p. 1611-1614.
15. Halak, B. and A. Yakovlev, Bandwidth-Centric Optimisation for Area-Constrained Links with Crosstalk Avoidance Methods. Design, Automation and Test Conference in Europe, 2008: p. 438-443.
16. Ying, Z., et al. Codeword Selection for Crosstalk Avoidance and Error Correction on Interconnects. in VLSI Test Symposium, 2008. VTS 2008. 26th IEEE. 2008.
17. Duan, C., V.H.C. Calle, and S.P. Khatri, Efficient On-Chip Crosstalk Avoidance CODEC Design. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2009. 17(4): p. 551-560.
18. Xuebin, W. and Y. Zhiyuan. CAC CODEC designs based on numeral systems. in Signal Processing Systems, 2009. SiPS 2009. IEEE Workshop on. 2009.
19. Shafaei, M., A. Patooghy, and S.G. Miremadi. Numeral-Based Crosstalk Avoidance Coding to Reliable NoC Design. in Digital System Design (DSD), 2011 14th Euromicro Conference on. 2011.
20. Xuebin, W. and Y. Zhiyuan, Efficient CODEC Designs for Crosstalk Avoidance Codes Based on Numeral Systems. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2011. 19(4): p. 548-558.
21. Duan, C., A. Tirumala, and S.P. Khatri, Analysis and avoidance of cross-talk in on-chip buses. IEEE Conference on Hot Interconnects, 2001: p. 133-138.
22. Andrews, G.E. and K. Eriksson, Integer Partitions. 2nd ed. 2004: Cambridge University Press.
23. Sotiriadis, P.P. and A.P. Chandrakasan, A bus energy model for deep submicron technology. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2002. 10(3): p. 341-350.
24. Partha Pratim, P., et al., Performance evaluation and design trade-offs for network-on-chip interconnect architectures. Computers, IEEE Transactions on, 2005. 54(8).
25. Mutyam, M., Fibonacci Codes for Crosstalk Avoidance. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2012. 20(10): p. 1899-1903.