

A Novel Bus Encoding Scheme from Energy and Crosstalk Efficiency Perspective for AMBA based Generic SoC Systems

Zahid Khan¹, Tughrul Arslan^{1,2}, Ahmet T. Erdogan¹
z.khan@ed.ac.uk

Abstract

Inter-wire coupling is a major source of power consumption and delay faults for on-chip buses implemented in UDSM SoC Systems. Elimination or minimization of such faults is crucial to the performance and reliability of SoC designs. This paper presents a new on-chip bus encoding scheme targeting high performance generic SoC systems. In addition to its efficiency in terms of power, the scheme reduces delay faults by completely eliminating the most critical type of crosstalk coupling that causes three adjacent wires to undergo Miller-like transition simultaneously. The paper describes the technique, its implementation (using the widely adopted AMBA-AHB SoC bus standard) and provides results indicating between 24% to 38% energy saving for systems implemented in 0.18 μ m CMOS technology.

1. Introduction

¹The scaling of CMOS technology to ultra deep sub micron has increased the sensitivity of CMOS technology to various noise mechanisms such as crosstalk noise, power supply noise, leakage noise etc. Of all these, the crosstalk noise due to capacitive coupling is dominant as it causes delay faults, logical malfunctions and energy consumption on long on-chip buses. The coupled capacitance (C_I) between long parallel wires is of magnitude several times larger than the wire-to-substrate capacitance (C_L). In addition to its dependence upon technology as well as structural factors such as wire spacing [1], wire width, wire length [2], wire material, coupling length, driver strength [3], signal transition time etc, the coupled capacitance also depends upon the data dependent transitions and will increase or decrease depending upon the relative switching activity between adjacent bus wires [4]. For the case in which three adjacent wires undergo opposite state transition, the coupled capacitance on the center wire becomes 4 times the coupled capacitance in case only one wire changes state while all others remain silent. This increase in C_I causes 4 times increase in delay and energy consumption (due to four times increase in crosstalk noise) compared to single wire change [4].

Previous low power coding schemes aimed at reducing the node switching activity for low power [5] and [6]. This is efficient for off-chip buses where node capacitance is several times larger than the coupled capacitance and where impedances are properly adjusted to reduce crosstalk noise. However, for on-chip buses major source of energy consumption is the inter-wire coupled capacitance and therefore its minimization is necessary for saving energy consumption.

Reducing the inter-wire coupling capacitance without eliminating any type of worst case crosstalk (*type-4*, *type-3* and *type-2* discussed in section 2) will result in low power but will not reduce the maximum bound on delay penalty that limits the performance and reliability of high speed on-chip buses. The CBI Scheme in [8] reduces the net coupled switched capacitance but does not eliminate any type of worst case crosstalk. This implies that from delay perspective, the method is not much advantageous over the unencoded data. The scheme [9] is well suited as the coding eliminates all worst crosstalk types. However, there is no guarantee that the method will also be power efficient. The encoding scheme presented in this paper targets the crosstalk problem from both power and delay perspectives. It transforms the incoming data in such a way as to eliminate two worst crosstalk types (*type-4* and *type-2*). By doing so, the worst case delay in signal transition will be eliminated and the delay will now depend on the crosstalk which is less severe. At the same time, the scheme provides power reduction by minimizing self and coupled switched capacitance. The work presented in [10] is well suited for both power efficiency and elimination of all types of worst crosstalk (*type-4*, *type-3* and *type-2*). However, since the method exploits the probabilistic information of the data stream, it cannot be applied to a data the statistical properties of which can not be known a priori and, therefore, cannot be applied to generic SoC systems. The work in [11] exploits the locality and temporal correlation that exist in address buses for both low power and reducing interconnect coupling. However, the method cannot be applied to data buses which are neither local with regard to data nor temporally correlated. The method proposed in this paper is unique and novel in the sense that it targets reduction of self and worst crosstalk coupled switched capacitance for achieving two design goals (low power and improved error immunity). The method is well suited to generic data

¹ University of Edinburgh, Scotland, UK

² ISLI, Livingston, Scotland, UK

buses and does not require prior probabilistic information of the input data stream. The remaining sections of the paper are organized such that section 2 provides an expression for energy consumption as function of self and coupled switching activity, section 3 explains the method and its implementation using a generic SoC platform based on AMBA-AHB bus protocol, section 4 and 5 provide results and conclusion.

2. Energy Expression Formulation

This section presents four types of crosstalk by taking three adjacent wires into consideration. The classification of the crosstalk into types is done to emphasize two aspects of the encoding scheme. The first is elimination/minimization of worst crosstalk and second the energy efficiency. For a 3-bit bus, a *type-1* crosstalk occurs if one of the three wires changes state e.g. a transition from 110 to 111 will cause a *type-1* crosstalk. For *type-1* crosstalk, the coupled capacitance is C_i . A *type-2* crosstalk occurs if center wire is in opposite state transition with one of its adjacent wires while the other wire undergoes the same state transition as the center wire, e.g. a transition from 001 to 110 will cause a *type-2* crosstalk. For this type of crosstalk, the coupled capacitance will be $2 C_i$. A *type-3* crosstalk occurs if the center wire undergoes opposite state transition with one of the two wires while the other is quiet. A transition from 101 to 110 will cause a *type-3* crosstalk and the coupled capacitance of the center wire in this crosstalk will be $3 C_i$. For the case of *type-4* crosstalk, all three wires transition to opposite state with respect to each other and their previous bus state. A transition, for example, from 101 to 010 will cause a *type-4* crosstalk and the coupled capacitance of the center wire rises to $4 C_i$. *Type-4*, *type-3* and *type-2* are the worst crosstalk [12].

The authors in [13] have derived an approximate energy expression for an n-bit bus as function of self and coupled switched capacitance. The same lumped model is considered here for a three bit bus. The model is used to provide an expression for the energy consumption when each type of crosstalk is considered alone and then to derive energy expression when all types of crosstalk occurs together. The idea is then generalized for an n-bit bus.

Consider the 3-bit bus shown in Figure 1

The total energy can be expressed as the algebraic sum of the energy consumed in self and coupled switching.

$$E = E_{self\ switching} + E_{coupled\ switching}$$

The operation of the three bit bus is expressed using the following energy equations.

$$E_1 = C_L \{ (1 + \lambda) (V_1^f - V_1^i) - \lambda (V_2^f - V_2^i) \} V_1^i \quad (1a)$$

$$E_2 = C_L \{ -\lambda \cdot (V_1^f - V_1^i) + (1 + 2\lambda) (V_2^f - V_2^i) - \lambda \cdot (V_3^f - V_3^i) \} \cdot V_2^i \quad (1b)$$

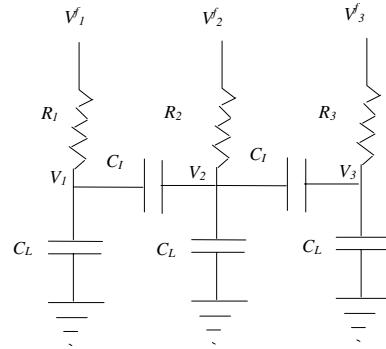


Figure 1: A lumped Model of the on-chip bus [11]

$$E_3 = C_L \{ -\lambda (V_2^f - V_2^i) + (1 + \lambda) (V_3^f - V_3^i) \} V_3^i \quad (1c)$$

$$E = E_1 + E_2 + E_3 \quad (1d)$$

Here V_1^f, V_2^f, V_3^f are final and V_1^i, V_2^i, V_3^i are the initial states of the three wires respectively. $V_1^f, V_2^f, V_3^f, V_1^i, V_2^i, V_3^i$ can be either V_{dd} or 0. E_1, E_2 , and E_3 represent energy for wire1, 2 and 3 respectively. For 0.18 μ m CMOS technology and minimum distance between wires, the ratio of coupled capacitance (C_i) to wire-to-substrate capacitance (C_L) is $\lambda = C_i/C_L = 3.2$ [14]. Equation 1d gives total energy consumption for a 3-bit bus. For a *type-4* crosstalk (101-to-010), the energy consumption on the 3-bit bus can be found using equation 1d and is given by:

$$E = E_{0->1} (1 + 4\lambda) \quad (2)$$

where $E_{0->1}$ is the energy consumption due to self transition and is equal to $C_L V_{dd}^2$. Equation 2 implies that for a *type-4* crosstalk, the energy consumption is increased by 4λ . From equation 2, we have $E/E_{0->1} = 1 + 4\lambda$ which gives the net switching activity on the 3-bit bus which in this case comprises of one *self* (0-to- V_{dd}) and one *type-4* coupled switching activity. The contribution of the *type-4* switching to net switching activity is 4λ . The energy consumption, therefore, depends upon the self and *type-4* switching activity. If we take N clock periods and let N_{sx} and N_t be total *self* and *type-4* switching activity in the time interval $[0, N]$, the net switching activity will then be $(N_{sx} + 4 N_t \lambda)$ and from equation 2, total energy consumption is:

$$E = E_{0->1} (N_{sx} + 4 N_t \lambda) \quad (3a)$$

Similarly, it can be shown that the total energy drawn from the power supply during the interval $[0, N]$ in the case of *type-3*, *type-2* and *type-1* couplings occurring

alone will be as below:

$$E = E_{0 \rightarrow 1} (N_{3x} + 3 N_3 \lambda) \quad (3b)$$

$$E = E_{0 \rightarrow 1} (N_{2x} + 2 N_2 \lambda) \quad (3c)$$

$$E = E_{0 \rightarrow 1} (N_{1x} + I N_1 \lambda) \quad (3d)$$

where N_3 , N_2 and N_1 are respectively type-3, type-2 and type-1 couplings and N_{3x} , N_{2x} , N_{1x} are respectively the associated self transitions in the corresponding crosstalk type. If a 3-bit bus has all types of crosstalk, the approximate energy consumption is then given by adding equations 3a, 3b, 3c and 3d:

$$E = E_{0 \rightarrow 1} \cdot \{N_x + \lambda \cdot (4 N_4 + 3 N_3 + 2 N_2 + I N_1)\} \quad (4)$$

Where $N_x = N_{4x} + N_{3x} + N_{2x} + N_{1x}$ is the total self switching activity in the time interval $[0, N]$. The net switching activity is given by 5:

$$E/E_{0 \rightarrow 1} = N_x + \lambda \cdot (4 N_4 + 3 N_3 + 2 N_2 + I N_1) \quad (5)$$

The idea can be generalized for a n-bit bus to compute the total energy consumption by considering N_x , N_4 , N_3 , N_2 and N_1 be respectively total self, type-4, type-3, type-2 and type-1 switching activity in the n-bit data set for the interval $[0, N]$.

Equation 5 provides only approximate results for the total energy consumption on the bus for the crosstalk on wire i due to wire $i+1$ and $i-1$ (for $i > 1$) is considered. The assumption is valid as coupling effect varies inversely with the spacing between wires. The results presented in this paper for energy estimation are based on calculating N_x , N_4 , N_3 , N_2 , N_1 and then finding out the net switching activity as given by equation 5. Equation 6 then gives the energy saving:

$$\text{Energy Saving} = (1 - N_c/N_u) 100 \quad (6)$$

where N_u and N_c are respectively the net switching activity (as given by equation 5) in the encoded and corresponding encoded data. In all the encoding schemes, the switching activity is measured in a careful manner such that one type does not include the switching activity of other types.

The Power consumption could be measured using the commercially available tools like Synopsys design compiler. However, these tools are based upon the self switched capacitance and do not take into account the coupled switched capacitance. Therefore, the energy consumption on the bus will be estimated using equation 6. The power consumption of the Codec Architecture

(Encoder and Decoder) will be calculated through the available power estimation tools like the Synopsys design power.

3. Low Power Bus Encoding Scheme

3.1 Methodology Definition

The proposed encoding scheme is based on an intrinsic property exhibited by 4-bit binary sequence (vector space V_4), which is explained below:

Consider a 4-bit bus that represents a maximum of sixteen 4-bit binary sequences (4-tuples). If any one of the 4-tuples is taken, modulo-2 summed with two basis functions Z_1 (0101) and Z_2 (1010) (alternate bit complement) and compared with the remaining 4-tuples, it is observed that one of the two xored data (either data xored with Z_1 or data xored with Z_2) will have no type-4 switching ($N_4=0$) with respect to the remaining fifteen 4-tuples. The proposed encoding scheme exploits this property of V_4 for low power, worst crosstalk (N_4 and N_2) elimination and N_3 crosstalk minimization. Since the method is primarily developed to eliminate/minimize the worst crosstalk, effect on N_1 is not considered. The scheme is briefly described:

The data on a 4-bit bus is modulo-2 added with two basis functions Z_1 and Z_2 (Figure 2). The output $x_{z1}(n)$ is compared with the previous bus state $x(n-1)$ in a crosstalk check module that checks the level of the crosstalk.

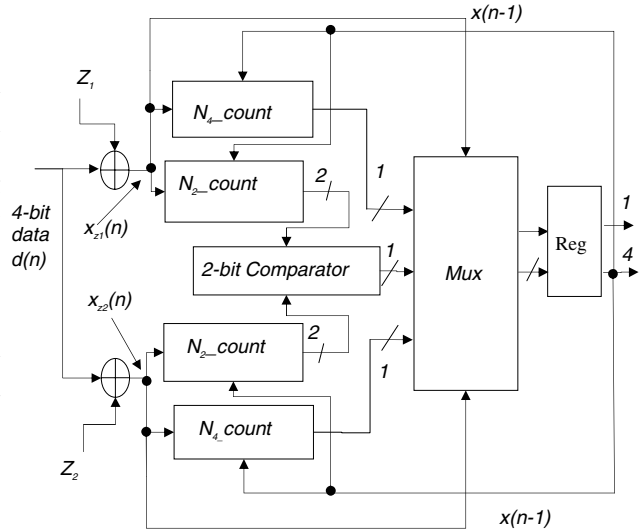


Figure 2: Encoder for a Generic 4-bit bus

The crosstalk check module consists of N_4_count and N_2_count modules. The first counts N_4 while the second counts N_2 couplings in x_{z1} with respect to the previous bus state $x(n-1)$. Similar crosstalk check module is implemented for x_{z2} . The output from the two N_2_count

modules from both x_{Z1} and x_{Z2} are compared in a 2-bit comparator module. The comparison is done in such a way that if 2-bit N_2_count in x_{Z1} is greater than 2-bit N_2_count in x_{Z2} , the output of the comparator is 1 otherwise it is 0. The 1-bit output from N_4_count of x_{Z1} , 1-bit output from N_4_count of x_{Z2} and the 1-bit comparator output are used to select either x_{Z1} or x_{Z2} . The multiplexing is well explained by the pseudo code given below:

If type-4 coupling is present in x_{Z1} , select x_{z2} and set decode bit **Else if** type-4 coupling is present in x_{z2} , select x_{z1} and reset decode bit **Else if** Comparator Output is one, select x_{z2} and set decode bit **Else** select x_{z1} and reset decode.

The output from the Mux (4-bit data and the decode bit) are registered at the next positive clock edge. The entire operation occurs within one clock cycle. The register outputs the encoded data on the bus together with decode bit as decode information for the decoder. A shield wire is placed between the 4-bit encoded data and the decode bit to avoid worst crosstalk at the expense of type-1 crosstalk as some energy loss occurs due to coupling of the shield wire with its two adjacent wires. Thus a 4-bit bus is expanded to 6 bits for low power and crosstalk noise reduction.

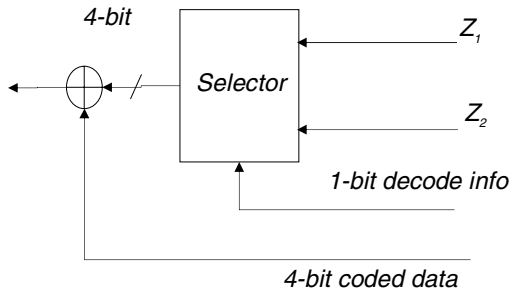


Figure 3: 4-bit Decoder

The decoder (Figure 3), based on the decode information, exclusive-ors the incoming data with either Z_1 or Z_2 and generates the original data.

It can be proved from the basic energy equations as given in 1d that the maximum number of N_4 coupling can be either 1 (max) or 0 (min) per 4-bit data transfer. A very simple combinational logic as shown in Figure 4 is used to implement N_4 counter. The logic detects N_4 coupling either between wires 1, 2 and 3 or between wires 2, 3 and 4. N_2 counter counts N_2 coupling. The maximum N_2 coupling in a particular data transfer is 3 and the N_2 counter consists of the slim combinational logic as shown in figure 5. The 3-bit adder is used to add or in other words count the N_2 couplings. It can be shown that $N_3 = N_2 + N_1$, therefore, N_3 counting is redundant and not used here.

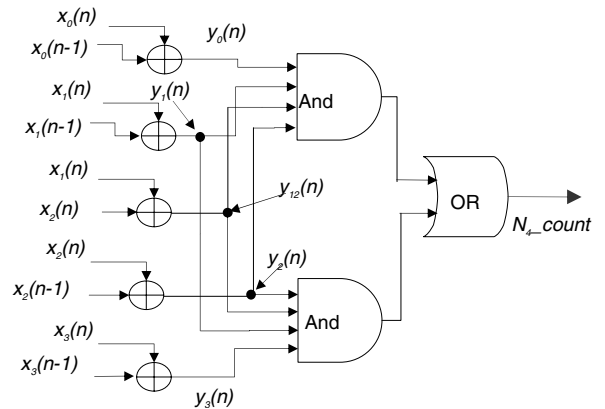


Figure 4: N_4 Coupling Counter (1-bit)

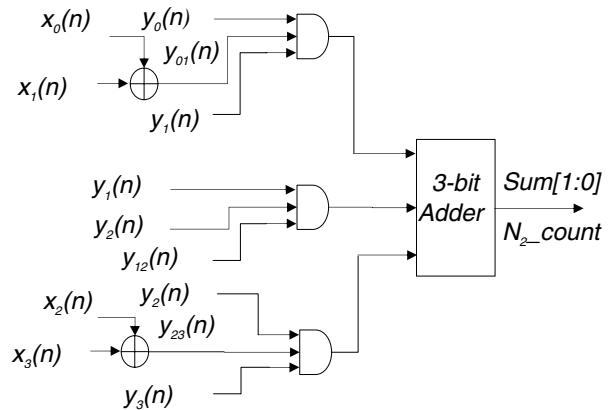


Figure 5: N_2 Coupling Counter (2-bit)

For 8-bit transfer, the 8-bit bus is first partitioned into two 4-bit clusters and coding is applied to each cluster. The inter-cluster worst crosstalk is eliminated by placing a shield wire between the two clusters. The decode information for the two clustered decoders could be transmitted using only two bits. However, 3 bits are used so that no worst crosstalk occurs between decode bits. A shield wire is also placed between adjacent 4-bit coded data and the 3-bit decode information for the same purpose. The decode information for two 4-bit clusters is given in Table 1.

Operation on 1 st 4-bit data	Operation on 2 nd 4-bit data	3-bit Decode Info
Z_1	Z_1	000b
Z_1	Z_2	001b
Z_2	Z_1	011b
Z_2	Z_2	111b

Table 1 (Decode Info for 8-bit decoding)

The encoded data on the bus is organized as 4-bit coded data+shield wire+4-bit coded data+shield wire+3-bit

decode information. Therefore, in this scheme an 8-bit bus is expanded to $4+1+4+1+3=13$ -bit bus. The same procedure is adopted in the case of 16-bit transfer and the bus expands to $13+1+13=27$ bits. The encoding scheme is implemented for 8 and 16-bit transaction as a sample example. Any transfer size can be used to prove the dual purpose of the encoding scheme. The scheme can also be implemented without inserting shield wires and using only 25% spatial redundancy (one extra wire per 4-bit cluster) to carry decode information along with the coded clusters. This scheme will not guarantee elimination of type-4. However, type-4 is minimized to such an extent that its presence is almost negligible. The scheme will not reduce the maximum delay bound, however, it will reduce the bit error probability due to crosstalk noise. Energy efficiency will be slightly more than the case with shield wires due to less increased type-1 coupling.

3. Encoder and Decoder Implementation

The Encoder takes 16-bit data and 4-bit byte lane enable information from the AHB Bus. The AHB data bus is divided into 4 byte lanes so as to enable 8, 16 and 32-bit transfer. The Encoder is designed for the entire 16-bit data bus. However, the encoding is performed based on the byte lane information for either 8 or 16-bit data whichever is available on the bus. If 8-bit transfer is taking place on the AHB bus, the encoder configures itself for operating on lower 8-bit data and the upper 8-bit lane is grounded. The encoded data received together with the decode information are applied to the Decoder to get the desired data. Like Encoder, the Decoder is also designed for 16-bit bus. However, based on the appropriate byte lane, decoding is done for the desired byte lane(s) (either 8-bit or 16-bit) of the data bus to generate the original data. The Codec Architecture is implemented on AHB data bus with area overhead of 2.51% for 8-bit and 5.01% for 16-bit transactions.

4. Simulation Results

The proposed method was implemented using 0.18μm CMOS technology and energy consumption was measured based on self and coupled transitions. Applied data streams consist of zero mean uniformly distributed random data (ran) and highly correlated application specific data such as those produced in image processing (img) and biomedical (bio) applications with transfer sizes of 8 and 16 bits. The total number of self (N_s) and coupled (N_4, N_3, N_2, N_1) transitions are calculated for the three types of data using two transfer protocols (8 and 16-bit) of the AMBA-AHB data bus for the case of unencoded, bus invert, CBI and the proposed low power method. The energy saving in each case is calculated using the relation given in equation 6. Figure 6 provides percentage energy

saving for bus invert, CBI and the proposed encoding scheme. The bus invert and CBI are used for comparison as both do not require in advance the probabilistic information of the data.

The power consumption of the internal combinational and sequential elements of the system with and without Codec Architecture has been measured using synopsys power compiler for a sample 8-bit biomedical data set example. The power consumption of the Codec is only 0.0228mW which is equivalent to a 2.25% increase. For 8-bit random and 8-bit image data, the percent increase in power due to Codec is respectively 2.45% and 1.98%. The increase in power is negligibly small as compared to saving on the on-chip bus. The results are analyzed by considering first energy saving and then crosstalk reduction.

The alternate bit complement (*xoring with either 0101 or 1010*) in the proposed encoding scheme favors elimination of N_4 , N_2 and much reduction of N_3 crosstalk and self switching activity (N_s) for highly correlated data such as those produced in image and biomedical applications. Therefore, the proposed encoding scheme results in appreciable energy saving for these data types. The power efficiency of the scheme increases as the worst crosstalk coupling in the generic data gets increased.

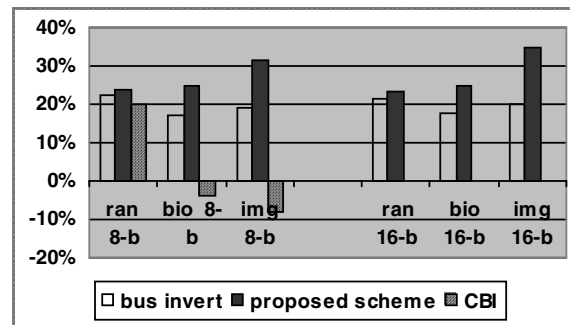


Figure 6: Percentage Energy saving

Therefore, for image data, increasing the transfer size from 8 to 16-bit results in increase in worst crosstalk coupling and minimizing more crosstalk coupling will result in more energy saving. The energy saving, therefore, increases from 32% to 35%. For the biomedical data set, the worst crosstalk increases by less than 1% with increase in transfer size, the energy saving remains almost constant at 24%. In the case of random data, the increase in transfer size from 8 to 16-bit results in slight increase in N_1 coupling which decreases the energy efficiency of the proposed encoding scheme from 24% to 23%.

The CBI proved to be more power consuming on biomedical and image data sets. The reason for this increase is that CBI depends on the total coupled switched

capacitance in a particular data transfer and if an 8-bit data sample has less than 8 total coupled switched transitions in a particular transfer, the CBI coder will not invert the data. The biomedical and image data sets are characterized by a high degree of correlation with minimum crosstalk coupling probability per data transfer. Out of the 14644 samples for the biomedical data, none of the data samples got inverted by the CBI coder. Similarly out of the 65535 samples of the 8-bit image data, only 2318 samples got inverted. The de-correlation at the output of the CBI encoder caused increase in total switched capacitance for these data sets. The CBI scheme is well suited to the case in which the data has more crosstalk coupling per data sample so that inversion of the data occurs quite frequently. The CBI results are available for only 8-bit data due to time constraints.

The proposed encoding scheme eliminates N_4 and N_2 crosstalk coupling completely from all data types with much more reduction in N_3 coupling. Figure 7 provides a sample graph for 8-bit biomedical data set that proves the claims made regarding coupling reduction. This also shows comparison with bus invert and coupling driven bus invert schemes. The proposed scheme proved to be more efficient than the bus invert and the CBI regarding power and worst crosstalk coupling reduction.

5. Conclusion

The authors have presented a technique that addresses energy loss and delay problems (due to crosstalk noise) faced by today's tightly coupled on-chip buses implemented in ultra deep sub micron SoC systems. The technique provides energy saving, for 0.18 μ m CMOS technology, ranging from 24% for highly de-correlated uniformly distributed random data to 35% for highly correlated application specific data such as those produced in image processing applications. From delay perspective, the technique eliminates N_4 and N_2 crosstalk coupling completely and minimizes N_3 , thereby reducing the bit error probability and improving the reliability and robustness of the on-chip communication to a considerable extent at the expense of some increase in bandwidth.

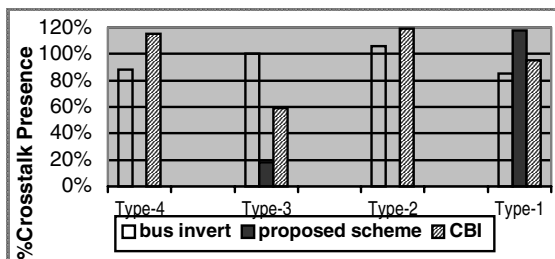


Figure 7: Graph representing percent crosstalk presence in a sample 8-bit biomedical data.

$$\%Crosstalk\ Presence = (Crosstalk\ in\ encoded\ data / Crosstalk\ in\ unencoded\ data) \cdot 100$$

5. References

- [1] L. Macchiarulo, E. Macii, M. Poncino, "Wire placement for crosstalk energy minimization in address buses", Design, Automation and Test in Europe Conference and Exhibition, 2002 Proceedings, 4-8 March 2002, Page(s): 158 -162
- [2] C. Guardiani, C. Forzan, B. Franzini, D. Pandini, "Modeling the Effect of Wire Resistance in Deep Submicron Coupled Interconnects for Accurate Crosstalk Based Net Sorting", xputers.informatik.uni-kl.de/conferences/patmos/patmos98/guardiani.pdf
- [3] T. Xiao, M. Marek-Sadowska, "Crosstalk reduction by transistor sizing", Design Automation Conference, 1999. Proceedings of the ASP-DAC '99. Asia and South Pacific, 18 21 Jan. 1999. Page(s): 137 -140 vol.1
- [4] K. Hirose, H. Yasuura, "A bus delay reduction technique considering crosstalk", Design Automation and Test in Europe Conference and Exhibition 2000. Page(s): 441-445
- [5] M.R. Stan, W.P. Burleson, "Bus-invert coding for low- power I/O", VLSI Systems, IEEE Transactions, Volume:3 Issue: 1, Mar 1995, Page(s): 49-58
- [6] L. Benini, G.D. Micheli, E. Macii, D. Sciuto, C. Silviano, "Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems", 7th Great Lakes Symposium on VLSI, 1997, Page(s): 7-82
- [7] S. Osborne, A.T. Erdogan, T. Arslan, D. Robinson, "Bus encoding architecture for low-power implementation of an AMBA-based SoC platform", Computers and Digital Techniques, IEE Proceedings, Volume: 149 Issue: 4, July 2002. Page(s): 152 -156
- [8] Ki-Wook Kim, Kwang-Hyun-Baek, N. Shanbhag, C.L.Liu, Sung-Mo Kang, "Coupling-driven signal encoding scheme for low-power interface design", Computer Aided Design, 2000. ICCAD-2000. IEEE/ACM International Conference on, 2000 Page(s): 318 -321
- [9] B. Victor, K. Keutzer, "Bus encoding to prevent crosstalk delay", Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conference on, 4-8 Nov. 2001 Page(s): 57-63
- [10] C.G. Lyuh, Taewhan Kim, "Low power bus encoding with crosstalk delay elimination", ASIC/SOC Conference, 2002. 15th Annual IEEE International Conference, 2002 Page(s): 389 -393
- [11] L. Tiehan, J. Henkel, H. Lekatsas, H. W. Wolf, "Enhancing signal integrity through a low-overhead encoding scheme on address buses", Design, Automation and Test in Europe Conference and Exhibition, 2003, March 3-7,2003, Page(s): 542-547
- [12] Chunjie Duan, Anup Tirumala, S.P. Khatri, "Analysis and avoidance of cross-talk in on-chip buses", Hot Interconnects 9, 2001, 2001 Page(s): 133 -138
- [13] P.P. Sotiriadis, A. Chandrakasan, "Low power bus coding techniques considering inter-wire capacitances", Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000, Page(s): 507 -510
- [14] P.P. Sotirsadis, A. Chandrakasan, "Bus energy minimization by transition pattern coding (TPC) in deep sub-micron technologies", Computer Aided Design, 2000. ICCAD-2000 IEEE/ACM International Conference, 2000, Page(s): 322-327