# A Dual Low Power and Crosstalk Immune Encoding Scheme for System-on-Chip Buses

Zahid Khan[1], Tughrul Arslan[1,2], and Ahmet T. Erdogan[1]

[1] University of Edinburgh, Scotland,UK
Z.Khan@ed.ac.uk
[2] Institute for System Level Integration livingston, Scotland

**Abstract.** Crosstalk causes logical errors due to data dependent delay degradation as well as energy consumption and is considered the biggest signal integrity challenge for long on-chip buses implemented in Ultra Deep Submicron CMOS technology. Elimination or minimization of crosstalk is crucial to the performance and reliability of SoC designs. This paper presents a novel on-chip bus encoding scheme targeting high performance generic SoC systems. In addition to its efficiency in terms of power, the scheme eliminates three types of crosstalk that cause miller-like transition on two or three adjacent wires simultaneously. The paper describes the technique, its implementation (using the widely adopted AMBA-AHB SoC bus standard) and provides experimental results indicating upto 38% energy saving for systems implemented in 0.18μm CMOS technology.

## 1 Introduction

The scaling of CMOS technology to ultra deep sub micron has increased the sensitivity of CMOS technology to various noise mechanisms such as crosstalk noise (due to capacitive and inductive coupling), 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 *($C_L$)* [1]. In addition to its dependence upon technology as well as structural factors such as wire spacing [2], 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 [1]. 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

E. Macii et al. (Eds.): PATMOS 2004, LNCS 3254, pp. 585–592, 2004.

energy consumption is the inter-wire coupled capacitance and 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 [7] 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 raw data. The work presented in [8] is well suited for both power efficiency and elimination of all types of worst. 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 [9] exploits the locality and temporal correlation that exist in address buses for both low power and reducing worst crosstalk coupling. However, the method cannot be applied to data buses which are neither local with regard to data nor temporally correlated. 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 worst crosstalk types (*type-4, type-3 and type-2 discussed in section 2).* By doing so, the errors associated with crosstalk induced delay and glitches can be eliminated. At the same time, the scheme provides power reduction by minimizing self and eliminating worst coupled switched capacitance. The method is well suited to generic data 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 Expression for Net Switching Activity

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 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 $C_r$. 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 $2 \cdot C_r$. 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 $3 \cdot C_r$. 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 $4 \cdot C_I$. *Type-4*, *type-3* and *type-2* are classified as the worst crosstalk coupling [4].

The authors in [10] have given approximate energy function for the self and coupled switching activity considering lumped model of the bus. The same lumped model is considered here to formulate energy expression in terms of supply voltage, node capacitance, coupling capacitance, self, type-4, type-3, type-2 and type-1 switching activity for a generic n-bit bus in the time interval *[0, N].* The expression is given in 1.

$$E=E_{0->1}\cdot \{N_x+ \lambda \cdot (4 \cdot N_4 + 3 \cdot N_3 +2 \cdot N_2 + 1 \cdot N_1)\} \tag{1}$$

Here $N_x$, $N_4$, $N_3$, $N_2$ *and* $N_1$ are the self, type-4, type-3, type-2 and type-1 switching activity in time interval [0, *N*]. $E_{0->1}=C_L V^2_{dd}$ *and* $\lambda=C_1/C_L$=ratio of coupling capacitance to wire-to-substrate capacitance. For 0.18μm and minimum distance between wires, $\lambda=3.2$[1]. The proof of equation 1 is not presented here. The net switching activity is given by 2*:*

$$E/E_{0->1}=N_x+ \lambda \cdot (4 \cdot N_4+3 \cdot N_3+2 \cdot N_2+1 \cdot N_1) \tag{2}$$

Equation 2 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 2. Equation 3 then gives the energy saving:

$$Energy\ Saving = (1-N_c/N_u) \cdot 100 \tag{3}$$

Where $N_u$ and $N_c$ are the net switching activity (as given by equation 2) in the unencoded and corresponding encoded data.

## 3 Proposed On-Chip Bus Encoding Scheme

Spatially redundant Limited Weight Code has hamming weight less than the data it represents and is used in off-chip communication for energy efficiency [2] [11]. The concept of the limited weight code can be exploited for the case of on-chip communication to achieve two performance goals. The first is worst crosstalk coupling elimination and the second is the energy efficiency.

For a codeword of width n, a *k-LWC* is a code whose code words have hamming *weight* $\leq k$. For a vector m of space $2^m$, *k-LWC* code must satisfy the following inequality [11].

$$\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+ \dots\dots\dots + \binom{n}{k} \geq 2^m \tag{4}$$

A semi perfect code uses all n-tuples with *weight <k* and only some of the *n-tuples* with *weight=k* as code words [11].

Based on the inequality a semi perfect *m/2-LWC* codebook (where m is the width of the raw input data) has been developed in such a manner that if two code words of hamming weight greater than zero are summed using a modulo-2 adder, the resulting pattern will have no worst crosstalk coupling. The resulting pattern will also reduce self switched capacitance to a considerable extent. A sample *4-to-6-bit* spatially redundant semi perfect *m/2-LWC* codebook is presented below in Table 1 for reference. The bit pattern in the codebook is chosen such that the code words have only *type-3* and *type-1* crosstalk couplings and when the code to be transmitted on the bus is modulo-2 added with the present bus state, the resulting pattern will have only *type-1* coupling. If at any instant, a sample is taken of the pattern present on the bus, the pattern will be completely different from the codeword. The codeword is absorbed in the bit stream and is unrecognizable as the stream contains information from the previous code words. The codeword is recovered at the receiving end by performing modulo-2 summation of the present and the previous bus states.

**Table 1.** Semi-Perfect *2-LWC* Codebook

| *4-bit data* | *6-bit code* | *4-bit data* | *6-bit code* | *4-bit data* | *6-bit code* | *4-bit data* | *6-bit code* |
|---|---|---|---|---|---|---|---|
| 0000b | 000000b | 0100b | 000101b | 1000b | 100100b | 1100b | 100000b |
| 0001b | 000001b | 0101b | 001000b | 1001b | 010001b | 1101b | 100001b |
| 0010b | 000010b | 0110b | 001001b | 1010b | 010010b | 1110b | 100010b |
| 0011b | 000100b | 0111b | 001010b | 1011b | 010100b | 1111b | 010000b |

The encoding algorithm consists of the following three steps:

*Step1:*

$$y^{m}(n) = x^{m}(n) \oplus x^{m}(n-1) \tag{5a}$$

The first step consists of modulo-2 adding the state (binary sequence) already transmitted $x^m(n-1)$ and the state to be transmitted $x^m(n)$.

*Step 2:*

$$L^{p}\ (n) = m/2 - LWCCodeboo\,k(y^{m}(n)) \tag{5b}$$

The *m-bit* binary pattern $y^m(n)$ is input to the semi perfect *m/2-LWC* codebook that is specifically developed to eliminate worst crosstalk coupling and reduce self switching, the output is the corresponding *p-bit* semi-perfect *m/2-LWC* codeword.

*Step 3:*

$$D^{p}(n) = L^{p}(n) \oplus D^{p}(n-1) \tag{5c}$$

The codeword $L^p(n)$ to be transmitted on the bus is modulo-2 added with the previous bus state $D^p(n-1)$. The resulting pattern is immune to worst crosstalk coupling. This step is the most crucial and important part of the Encoder architecture as it performs two functions. The first is to eliminate the worst crosstalk coupling and second to reduce self switching activity at the expense of a modest increase in type-1 coupling.

**Decoding** is also a three step process and every step is the exact reverse of its corresponding encoding step.

*Step 1:*
From 5c, modulo-2 adding both sides with $D^p(n-1)$

$$D^p(n) \oplus D^p(n-1) = L^p(n) \oplus D^p(n-1) \oplus D^p(n-1)$$

$$=> D^p(n) \oplus D^p(n-1) = L^p(n) \tag{6a}$$

In this step, present and previous bus states are modulo-2 added to recover the limited weight codeword.

*Step2:*

$$y^m(n) = m - LWCDecodebook(L^p(n)) \tag{6b}$$

In this step, the limited weight codeword recovered from the first step is used to regenerate the m-bit binary pattern.

*Step 3:*
The third step recovers the original binary sequence.

$$x^m(n) = y^m(n) \oplus x^m(n-1) \tag{6c}$$

Initially the first step of modulo-2 adding the present and previous states of the raw input data (5a) was not included in the encoding algorithm. The method proved to be more power consuming on the data that has repeated same non-zero sequences. The inclusion of the first step is proved below:

Let $x^m(n)$, $x^m(n+1)$ and $x^m(n+2)$ are three consecutive raw input data values to be transmitted on the bus. The corresponding code words are $L^p(n)$, $L^p(n+1)$ and $L^p(n+2)$ and the bit pattern on the channel is $D^p(n)$, $D^p(n+1)$ and $D^p(n+2)$ where

$D^p(n) = D^p(n-1) \oplus L^p(n) => D^p(n) = L^p(n)$ as $D^p(n-1) = 0$
(initial condition)

$D^p(n+1) = D^p(n) \oplus L^p(n+1) = L^p(n) \oplus L^p(n+1)$

$D^p(n+2) = D^p(n+1) \oplus L^p(n+2) = L^p(n) \oplus L^p(n+1) \oplus L^p(n+2)$ Therefore, if
$x^m(n)=x^m(n+1)=x^m(n+2)$ then $L^p(n)=L^p(n+1)=L^p(n+2)$ and
$D^p(n)=L^p(n)$, $D^p(n+1)=0$ and $D^p(n+2)=L^p(n)$

The data pattern at the bus should remain constant as the input data pattern remains the same. However, transition increases due to inversion on every second non-zero same data input values. This can be avoided if step (5a) is inserted in encoding the data on the bus.

## 4 Encoder and Decoder Implementation

The proposed low power and crosstalk immune encoder and decoder architecture has been implemented on AMBA-AHB data bus. AMBA-AHB is the abbreviation for Advanced Microcontroller Bus Architecture- Advanced High Performance Bus. AHB is the standard of ARM and is widely used in today's high performance SoC systems. AHB data bus has been divided into different byte lanes so as to enable *8, 16, 32-bi* or higher order transactions. The proposed Codec Architecture (encoder and decoder) has been implemented to comply with all transfer protocols of the AMBA AHB bus. However, the research work presented in this paper is limited to *8, 16* and *32 bit* transfer sizes.

For *8-bit* transfer of data on AHB bus, the *8-bit* lane is partitioned into two *4-bit* clusters. The partitioning is done for area efficiency of the Codec Architecture. For *4-bit* cluster the memory required for the codebook of table 1 is only *16*x*6=96* bits. For two clusters, that would be *96*x*2=192* bits. For 8-bit lane that has *256* binary sequences and to encode all $2^8$ binary sequences will need a codebook that will occupy $2^8$ x *12=3072* bits of memory. This implies that partitioning provides a saving of *93.75%* memory at the expense of one extra shield wire between the two partitions. Therefore, *m=4* and *p=6* are chosen for the Codec Architecture in figures 1 and 2. The *2-LWC* codebook presented in Table 1 is used to encode the clustered data. A shield wire is inserted between the code words of the two clusters. This shield wire will eliminate any possibility of worst crosstalk coupling between code words of the two clusters. The insertion of the shield wire will cause some loss of energy due to *type-1* coupling. An *8-bit* bus is therefore extended to *6-bit codeword+ shield wire + 6-bit codeword= 13-bits.* The Encoder and Decoder cause some delay in signal propagation. However, the delay is well within tolerable limits of the pipelined architecture and does not degrade the performance of the AHB data bus. The area overhead of 32-bit Codec Architecture is calculated to be only 7.012% of the overall architecture of the AMBA-AHB bus.

## 5 Simulation Results

The proposed method was implemented using *0.18μm* CMOS technology and energy consumption was measured based on self and coupled switching activity. Applied data streams consist of zero mean uniformly distributed random data (r), and application specific biomedical (b) data with transfer sizes of *8, 16* and *32* bits. The total number of self $(N_x)$ and coupled $(N_4, N_3, N_2, N_1)$ switching transitions are calculated (all switching activity discussed in the result are calculated at the out data bus from the encoder) for the three types of data using three transfer protocols *(8, 16 and 32-bit)* of the AMBA-AHB data bus for the case of unencoded, Bus Invert (BI), coupling driven bus invert (CBI) and the proposed low power method. The energy saving is calculated using the relation given in equation 3. Figure 1 provides percentage energy saving for BI, CBI and the proposed encoding scheme while Figure 2 provides percentage presence of the switching activity for a sample 8-bit random (r-8) data set. The BI and CBI are used for comparison as both do not require in advance the probabilistic information of the data. The CBI scheme proved to be more power consuming. 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 data is characterized by a high degree of correlation with minimum crosstalk coupling probability per data transfer. Out of the 14642 samples, none of the data samples got inverted by the CBI coder. The de-correlation at the output of the CBI encoder caused increase in total switched capacitance for the biomedical data set. The CBI Scheme is power efficient in the case in which the data has more crosstalk

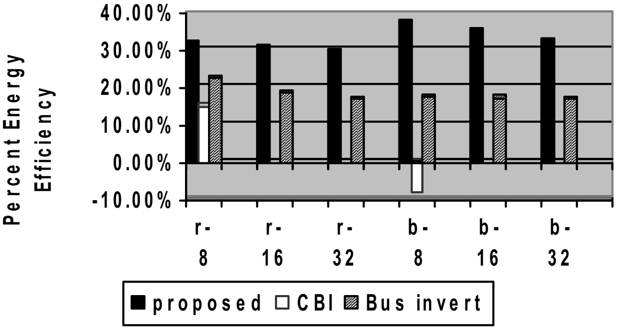

**Fig. 1.** Percent Energy Efficiency

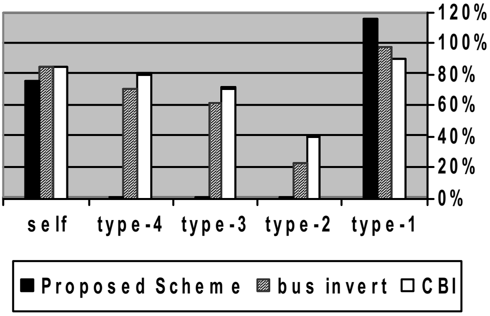

**Fig. 2.** % switching activity in 8-bit random data set (Percent switching activity presence= switching activity in the encoded data / switching activity in the unencoded data)

coupling so that inversion of the data occurs more frequently. The proposed scheme can also be applied to high capacitance off-chip buses as it reduces the self switching activity from *25%* for uniformly distributed random data to *49%* for highly correlated data such as those produced in biomedical applications. The power consumption of the internal combinational and sequential elements of the Codec Architecture has been measured using synopsis design power for a sample 8-bit random data set example. The power consumption is found to be only 2.01% of the overall power consumption of the SoC system on which the Codec Architecture has been implemented. This implies that the internal power consumption of the codec is negligible as compared to saving on the on-chip data bus. The method proposed proved to be more power as well as crosstalk noise efficient than BI, and CBI.

## 6 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 submicron SoC systems. The technique provides energy saving, for 0.18µm CMOS technology, ranging from 32% for highly decorrelated uniformly distributed random data to 38% for highly correlated application specific data such as those produced in biomedical applications. From delay perspective, the technique eliminates $N_4$, $N_3$ and $N_2$ crosstalk completely thereby reducing the bit error probability and improving the reliability and robustness of the on-chip communication to a considerable extent.

## References

[1] 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

[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. Patmos International Workshop on Power and Timing Modeling, Optimization and Simulation. Oct 7-9, 1998

[3] Payam Heydari, Massoud Pedram. Analysis and Reduction of Capacitive Coupling Noise in High Speed VLSI Circuits IEEE International Conference on Computer Design (ICCD), Austin, Texas, Sept. 2001. pp.104-109

[4] Chunjie Duan, Anup Tirumala, S.P. Khatri. Analysis and avoidance of cross-talk in on-chip buses. Hot Interconnects 9, 2001. 22-24 Aug. 2001.Page(s):133-138

[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. VLSI, 1997. Proceedings. Seventh Great Lakes Symposium on, 13-15 Mar. 1997 Page(s): 77 - 82

[7] 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 , 5-9 Nov. 2000 Page(s): 318 -321

[8] C.G. Lyuh, Taewhan Kim. Low power bus encoding with crosstalk delay elimination. ASIC/SOC Conference, 2002. 15th Annual IEEE International, 25-28 Sept.2002 Page(s): 389 -393

[9] Lu. Tiehan, J.Henkel, H. Lekatsas, W. Wolf. Enhancing signal integrity through a low-overhead encoding scheme on address buses. Design, Automation and Test in Europe Conference and Exhibition, 2003, Mar. 3-7, 2003

[10] 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, 21-24 May 2000 Page(s): 507 –510

[11] M.R. Stan, W.P. Burleson. Coding a terminated bus for low power. VLSI, 1995. Proceedings., Fifth Great Lakes Symposium on , 16-18 Mar. 1995, Page(s): 70 -73