

Low power system on chip bus encoding scheme with crosstalk noise reduction capability

Z. Khan, T. Arslan and A.T. Erdogan

Abstract: Inter-wire coupling is a major source of wire load and delay faults for on-chip buses implemented in ultra-deep submicron system on chip (SoC) systems. Elimination or minimisation of such faults is crucial to the performance and reliability of SoC designs. A novel on-chip bus encoding scheme targeting high-performance generic SoC systems is presented. In addition to its efficiency in terms of power, the scheme reduces delay faults by completely eliminating the most critical type of crosstalk coupled switched capacitance. The authors describe the technique and its implementation (using the widely adopted AMBA-AHB SoC bus standard) and provide experimental results indicating 22–36% energy saving for systems implemented in 0.18 μm CMOS technology.

1 Introduction

The scaling of complementary metal-oxide-semiconductor (CMOS) technology to ultra-deep submicron has increased the sensitivity of CMOS technology to various noise mechanisms such as crosstalk noise, power supply noise, leakage noise and so on. Of all these, the crosstalk noise because of 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 has several times larger magnitude 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 and so on, 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 centre wire becomes four times the coupled capacitance in case only one wire changes state, whereas all others remain silent. This increase in C_I causes four times increase in delay and energy consumption (because of four times increase in crosstalk noise) compared with single wire change [4].

Previous low-power coding schemes aimed at reducing the node switching activity for low power [5–7]. 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, the major source of energy consumption is the inter-wire coupled capacitance and therefore its minimisation 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 Kim *et al.* [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 in Victor and Keutzer [9] is well suited for reducing crosstalk only 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). As type-3 is a combination of a type-1 and a type-2 coupling, eliminating type-2 automatically reduces type-3. 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 minimising self and coupled switched capacitance. The work presented by Lyuh and Kim [10] is well suited for both power efficiency and elimination of all types of worst crosstalk (type-4, type-3 and type-2). However, as the method exploits the probabilistic information of the data stream, it cannot be applied to a data, the statistical properties of which cannot be known a priori, and therefore cannot be applied to generic SoC systems. The work by Tiehan *et al.* [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 that are neither local with regard to data nor temporally correlated. The work described in Bertozzi *et al.* [12] uses Hamming code for crosstalk error correction and/or detection. The work provides a comparison from power saving perspective between two strategies. The first is error correction because of crosstalk noise and the second is retransmission of the data, in case error is detected, at the cost of extra clock cycles. The author has proved that the retransmission is power efficient compared with error correction with the underlying assumption that the retransmission is error

© IEE, 2006

IEE Proceedings online no. 20050152

doi:10.1049/ip-cdt:20050152

Paper received 17th September 2005

The authors are with the School of Engineering and Electronics, University of Edinburgh, Alrick Building, King's Buildings, Mayfield Road, Edinburgh EH9 3JL, UK

E-mail: z.khan@ed.ac.uk

free. The work in this paper is an attempt to eliminate/reduce the root cause of the errors by manipulating the data and restricting those transitions that are the genesis of the crosstalk errors. The work is also targeted to high-throughput applications which will not tolerate wasting extra clock cycles. 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 as a result of reduced crosstalk noise). The method is well suited to generic high-speed data buses and does not require prior probabilistic information of the input data stream.

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 emphasise two aspects of the encoding scheme. The first is elimination/minimisation of worst crosstalk and second the energy efficiency.

For a 3-bit bus, a type-1 crosstalk occurs if either wire 1 or wire 3 changes state, for example, 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 the centre wire is in opposite state transition with one of its adjacent wires, whereas the other wire undergoes the same state transition as the centre wire, for example, a transition from 001 to 110 will cause a type-2 crosstalk. For this type of crosstalk, the coupled capacitance will be $2 \cdot C_I$. A type-2 crosstalk can also arise for the case in which the centre wire changes state, for example, the transition from 000 to 010. A type-3 crosstalk occurs if the centre wire undergoes opposite state transition with one of the two wires, whereas the other is quiet. A transition from 101 to 110 will cause a type-3 crosstalk and the coupled capacitance of the centre wire in this crosstalk will be $3 \cdot C_I$. For the case of type-4 crosstalk, all three wires move 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 centre wire rises to $4 \cdot C_I$. Type-4, type-3 and type-2 are the worst crosstalk [13].

Sotiriadis and Chandrakasan [14] have given approximate energy function for the self and coupled switching activity considering lumped model of the bus. The same lumped model (Fig. 1) is considered here for three adjacent wires. The model is used to provide expression for the energy consumption when each type of crosstalk is considered alone and then to derive energy expression when all types of crosstalk coupling occur together. The

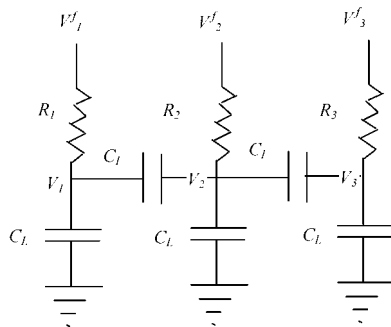


Fig. 1 Lumped model of the on-chip bus

energy expression for a 3-bit bus can be expressed as

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

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

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

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

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 wires 1, 2 and 3, respectively. For a 0.18 μm CMOS technology and minimum distance between wires, the ratio of coupled capacitance (C_I) to wire-to-substrate capacitance (C_L) is $\lambda = C_I/C_L = 3.2$ [15]. Equation (1d) gives the 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 by using (1d) and is given by

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

where $E_{0 \rightarrow 1}$ is the energy consumption because of 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 (2), we have $E/E_{0 \rightarrow 1} = 1 + 4 \cdot \lambda$, which gives the net switching activity on the 3-bit bus which in this case consists 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_{4x} and N_4 be total self and type-4 switching activity in the time interval $[0, N]$, the net switching activity will then be $(N_{4x} + 4 \cdot N_4 \cdot \lambda)$ and from (2), the total energy consumption is

$$E = E_{0 \rightarrow 1} \cdot (N_{4x} + 4 \cdot N_4 \cdot \lambda) \quad (3a)$$

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

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

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

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

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 (3a) to (3d)

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

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

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

The idea can be generalised 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, as the crosstalk on wire i because of wires $i + 1$ and $i - 1$ (for $i > 1$) is only

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 (5). Equation (6) then gives the energy saving

$$\text{Energy saving} = \left(\frac{1 - N_c}{N_u} \right) \cdot 100 \quad (6)$$

where N_u and N_c are, respectively, the net switching activity [as given by (5)] in the unencoded and corresponding encoded data.

3 Low power and crosstalk efficient bus encoding scheme

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

Consider a 4-bit vector 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 $Z_2(1010)$ (equivalent to 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 ability of the codes to eliminate type-4 couplings can be proved using the example given below.

Consider the case in which a 4-bit bus is in state 0101, and 1010 is the data to be transmitted on the bus. It can be shown that there is only one N_4 and one N_2 coupling in this bus transition. There exists a type-4 coupling between wires 1, 2 and 3 and a type-2 crosstalk between wires 3 and 4. This can be proved using (1). From (1b), we have

$$\begin{aligned} E_1 &= 0 \\ E_2 &= C_L \cdot \{-\lambda \cdot (0 - V_{dd}) + (1 + 2 \cdot \lambda) \cdot (V_{dd} - 0) \\ &\quad - \lambda \cdot (0 - V_{dd})\} \cdot V_{dd} \\ E_2 &= C_L \cdot \{\lambda \cdot V_{dd} + (1 + 2 \cdot \lambda) \cdot V_{dd} + \lambda \cdot V_{dd}\} \cdot V_{dd} \\ E_2 &= C_L \cdot (1 + 4 \cdot \lambda) \cdot V_{dd}^2 \\ E_3 &= 0 \\ E_4 &= C_L \cdot \{-\lambda \cdot (V_3^f - V_3^i) + (1 + \lambda) \cdot (V_4^f - V_4^i)\} \cdot V_4^f \\ E_4 &= C_L \cdot \{-\lambda \cdot (0 - V_{dd}) + (1 + \lambda) \cdot (V_{dd} - 0)\} \cdot V_{dd} \\ E_4 &= C_L \cdot (1 + 2 \cdot \lambda) \cdot V_{dd}^2 \end{aligned}$$

If the data 1010 is XORED with $Z_1(0101)$ and $Z_2(1010)$, one of the two resulting data 1111 or 0000 will definitely have no type-4 switching with data already present on the bus, which is 0101. In some cases, both resulting data will be free from the most critical type of crosstalk (type-4). For the case in which both are free from type-4 switching, the data sets are then searched for type-3 and type-2 crosstalk. The one that has less type-2 and/or type-3 crosstalk are transmitted on the bus.

This property of the 4-bit sequence can be extended to an arbitrary m -bit sequence (where $m = 2^x$ and x is any positive integer) for low power and worst crosstalk (N_4 , N_3 and N_2) minimisation. The extension of the method to an arbitrary sequence greater than four will not result in eliminating type-4 crosstalk, however, it will minimise the worst crosstalk couplings by reducing the relative switching activity. The underlying advantage is the use of less wire

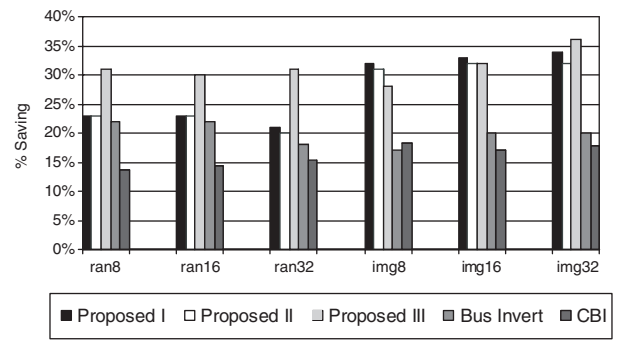


Fig. 2 Energy efficiency

overhead, but the disadvantage is that the probability of reducing the logical errors will be quite less compared with applying the method to the 4-bit sequence. This can be proved using the results as provided in Fig. 2.

By looking at the energy equation as given in (1), it becomes evident that crosstalk is the major source of energy consumption in on-chip communication. Worst crosstalk also causes considerable and unpredictable delay in signal transitions and therefore can result in logical errors which can bring the system to complete failure. Therefore emphasis is given to eliminate or minimise the worst crosstalk coupling. As the method is primarily developed to eliminate/minimise the worst crosstalk, effect on N_1 is not considered. The scheme is briefly described below.

The data $d(n)$ on a 4-bit bus is modulo-2 added with two basis functions Z_1 and Z_2 (Fig. 3). The output $x_{z1}(n)$ ($x_{z1}(n) = d(n) \oplus Z_1$) is compared with the previous bus state $x(n-1)$ in a crosstalk check module that checks the level of the crosstalk. The crosstalk check module consists of N_4_count and N_2_count modules. The first counts N_4 , whereas the second counts N_2 couplings in $x_{z1}(n)$ with respect to the previous bus state $x(n-1)$. It can be proved from the basic energy equations that in a 4-bit bus, the maximum number of type-4 coupling is one, whereas that of type-2 is three. Therefore modules N_4_count and N_2_count provide 1-bit and 2-bit outputs, respectively. A similar crosstalk check module is implemented for $x_{z2}(n)$ ($x_{z2}(n) = d(n) \oplus Z_2$). The outputs from N_2_count modules of both $x_{z1}(n)$ and $x_{z2}(n)$ are compared in a 2-bit comparator module such that if the number of type-2 couplings in $x_{z1}(n)$ is greater than the number of type-2 couplings in $x_{z2}(n)$, the output of the comparator is 1 otherwise it is 0. The 1-bit output from N_4_count of $x_{z1}(n)$ 1-bit output from N_4_count of $x_{z2}(n)$, and the 1-bit comparator output are used to select either $x_{z1}(n)$ or $x_{z2}(n)$. The selection (multiplexing) is done such that if N_4_count is present in $x_{z1}(n)$, data XORED with $Z_2(x_{z2}(n))$ is selected

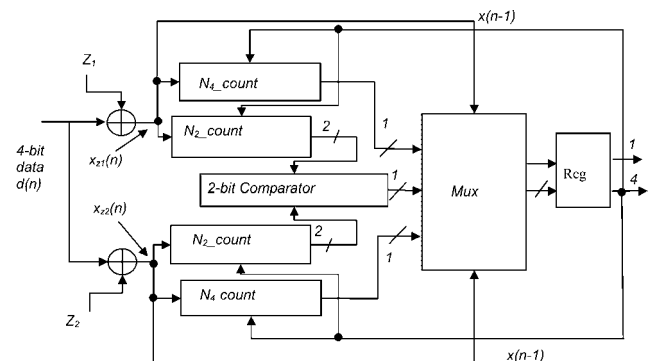


Fig. 3 Block diagram of the proposed encoding scheme

and the decode bit is set to logic 1. Because of the alternate bit complement, both cannot have N_4 count at the same time. If N_4 count is present in $x_{z2}(n)$, data XORED with $Z_1(x_{z1}(n))$ is selected and vice versa. For the case in which both N_4 counters are zero, the selection is based on the comparator output and if the comparator output is logic 1, $x_{z2}(n)$ is selected and the decode bit is set, whereas for logic 0, $x_{z1}(n)$ is chosen and decode bit is reset.

The output from the multiplexer (4-bit data and the decode bit) is registered at the next positive clock edge. The register outputs the encoded data on the bus together with decode bit as decode information for the decoder.

The number of N_4 coupling can be either 1 (max) or 0 (min) per 4-bit data transfer. The N_4 coupling can occur either among bits 0, 1 and 2 or among bits 1, 2 and 3. A very simple combinational logic as shown in Fig. 4 is used to implement N_4 counter. The idea behind the development of this counter is based on the fact that in type-4 coupling, all three bits switch opposite with respect to their previous state and two adjacent bits must not be at the same state in either their present or previous states. Consider the first three bits in the transition from 0101 (previous state) to 1010 (present state). This is an example of a type-4 switching. The three bits (010) are switching opposite to their previous state (101) and no two adjacent bits are at the same logic state in either their present or previous states. Therefore if the present and previous states of the individual bits are modulo-2 added and then ANDED, the output must be one. In the case of the transition from 000 to 111 (for example), the output from the AND gate will also be at logic 1. To differentiate between the two states, it is necessary to add the modulo-2 sum of any two adjacent bits of either the previous or the present state. The selection of these two adjacent bits should be made in order to implement the counter efficiently regarding reducing the number of gates. As mentioned earlier, type-4 switching can occur either between 0, 1 and 2 or between 1, 2 and 3. Therefore for efficient implementation, bits 1 and 2 of the present state ($x_1(n)$ and $x_2(n)$) are selected for modulo-2 addition. With this additional information, the inputs to the AND gate increase to four. N_4 check is implemented for 0, 1, 2 and 1, 2, 3 bits. The output from the two AND gates are then logical ORED to generate a single-bit output from N_4 counter.

N_2 counter, as shown in Fig. 5, counts N_2 couplings in parallel with its corresponding N_4 counter. N_2 coupling is between two adjacent bits. The maximum N_2 coupling in a particular 4-bit data transfer is three. In N_2 coupling, two adjacent bits switch opposite to each other. The first stage in N_2 counter is the identification of the

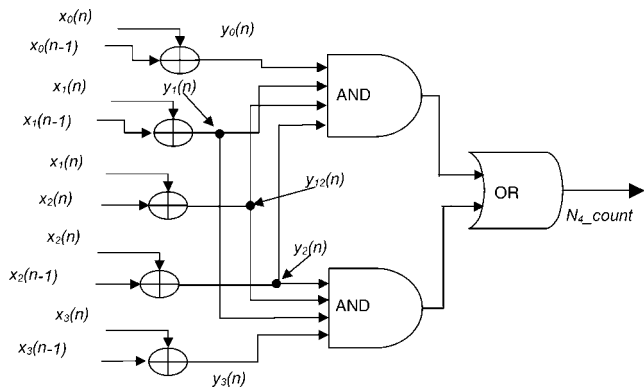


Fig. 4 N_4 coupling counter

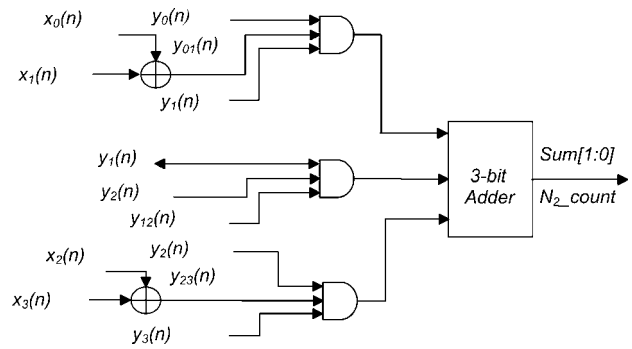


Fig. 5 N_2 coupling counter

coupling. This stage consists of XORING, the corresponding bits in the present and previous data. In order to differentiate this from the transition 00-to-11, the two adjacent bits are also XORED and the output is ANDED with the two other outputs as shown in Fig. 5. The three N_2 check outputs are then added in the second stage to obtain the total number of N_2 couplings. It can be shown that $N_3 = N_2 + N_1$, therefore, N_3 counting is redundant and not used here.

4 Implementation on AMBA-based SoC platform

The methodology can be implemented in different ways. Each type of implementation will have different capability of reducing crosstalk couplings and saving energy. The types are determined by the percentage of redundant wires used. Three schemes are introduced (proposed I, proposed II and proposed III) for cases corresponding to 25%, 62.5% and 12.5% wire redundancy as described below.

4.1 Case 1

The first implementation is based on partitioning the n -bit bus into clusters of 4-bits each. Each 4-bit cluster is separately encoded and decoded according to the proposed encoding scheme. An extra bit is added per 4-bit encoded data for unique decodability. The n -bit bus thus expands to $n + n/4$ bits. This is equivalent to a wire redundancy of 25%. The decoder (Fig. 6) for a 4-bit cluster consists of selecting one of the two basis functions based on the decode bit and then modulo-2 adding the incoming coded data with the correct basis function to regenerate the original data. The decoder modules required for the n -bit bus are $n/4$ which are also the number of the encoder modules.

4.2 Case 2

The encoding scheme in Case 1 guarantees to eliminate N_4 and reduce N_3 and N_2 worst crosstalk from the 4-bit clusters. But as the bus is partitioned into $n/4$ clusters and the bus is also expanded spatially to carry decode information, N_4 , N_3

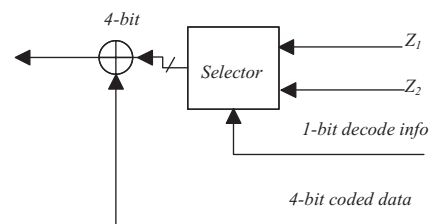


Fig. 6 Generic decoder for a 4-bit clustered coded data

and N_2 crosstalk coupling may occur in the inter-cluster region or between the decode bits and the clusters. The probability of such coupling can be eliminated by inserting shield wires in three different locations, namely the inter-cluster region, the cluster adjacent to the decode bits and between the decode bits. This will eliminate any possibility of the worst crosstalk occurrence; however, the bus bandwidth will further increase. In addition to increase in bus bandwidth, N_1 crosstalk coupling will also increase as a result of increased bandwidth. With the insertion of the shield wires, the bus bandwidth increases from 8–13, 16–27 and 32–55 which is equivalent to wire overhead of 62.5%. This is the maximum spatial redundancy. If the purpose of the encoding scheme is limited to only reducing logical faults by reducing crosstalk coupled switched capacitance, the shield wires can be avoided and the algorithm can be implemented with a small increase in bus bandwidth. If the purpose is to eliminate both worst case delay as well as to minimise logical faults, the algorithm has to be implemented with maximum spatial redundancy. In most cases, the goal is only to reduce logical faults much more by trading tolerable wire overheads. In such situations, the method requires only 25% wire overheads and guarantees much reduction in worst-case crosstalk couplings.

4.3 Case 3

The CODEC architecture is also implemented with one wire redundancy per 8-bit transaction. The wire redundancy is kept the same as in the case of BI and CBI. The method is implemented for 8, 16 and 32-bit transactions. For this implementation, basis functions Z_1 and Z_2 are extended so as to cover the 8-bit transaction. For example, Z_1 is now 01010101b, whereas Z_2 is 10101010b. The type-4 and

type-2 coupling counters (refer to Figs. 5 and 6) are also extended to count corresponding couplings in 8-bit transaction. As the maximum number of type-4 couplings in an 8-bit transaction is 3, the output of type-4 counter will now be 2-bit instead of the previous 1-bit. Similarly, the maximum number of type-2 couplings per 8-bit transaction is 7, therefore the output of type-2 counter is extended to 3-bit instead of the previous 2-bit output. The one having less type-4 couplings is transmitted. In case if both codes do not have type-4 couplings, the decision is made based on type-2 couplings and the one having less number of type-2 couplings is then transmitted along with the corresponding decoding bit.

5 Simulation results

A sample SoC architecture in which different IPs are connected through AMBA bus is shown in Fig. 7, whereas Fig. 8 provides a simulation environment used to simulate both encoder and decoder. IP-1 is transmitting data through the on-chip AMBA AHB bus to be stored on the external memory. The external memory is controlled through external memory controller. AMBA AHB bus controller and codec have been synthesised down to 0.18 μm CMOS technology. The BI and CBI have also been synthesised together with AMBA AHB bus controller. Table 1 gives the area overheads of the three encoding schemes. The area overheads of the proposed encoding scheme are approximately the same in all three implementations and therefore area overhead of only case 1 is presented in Table 1. As the proposed encoding scheme consists of a few cascaded gates, the area overhead of the proposed scheme is 11 and 18% less than the area overheads of BI and CBI, respectively. The delay introduced by the codec overhead is insignificantly small

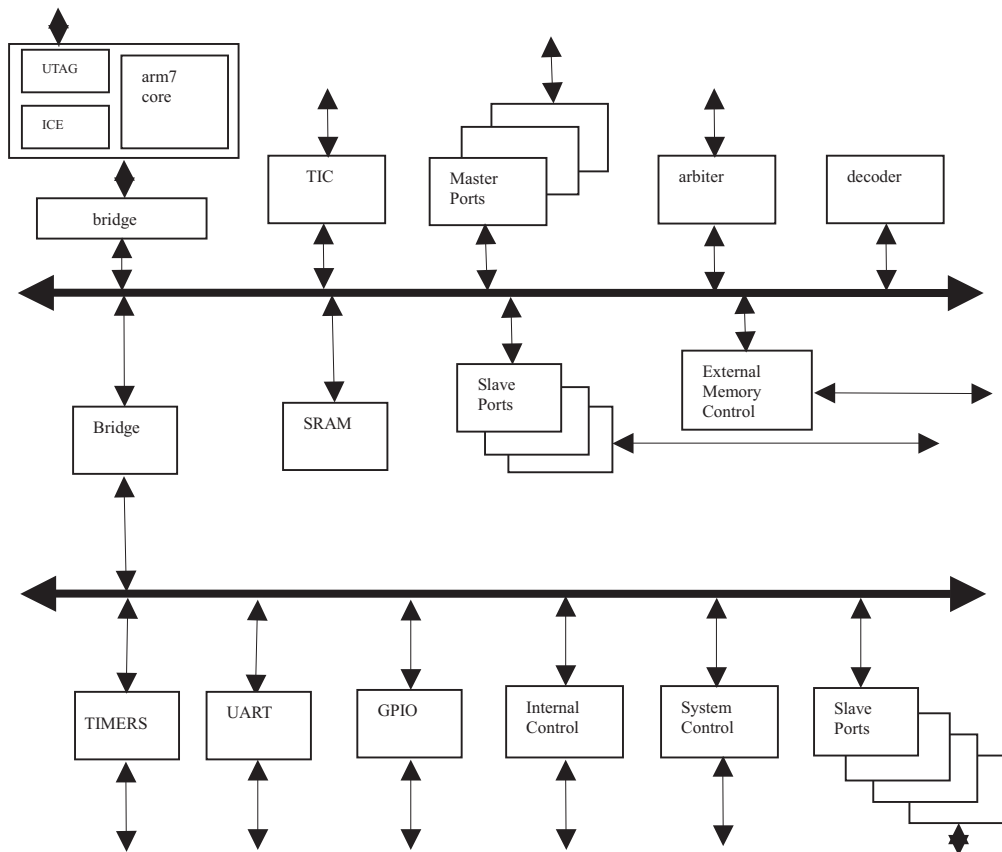


Fig. 7 AMBA AHB-based generic SoC system

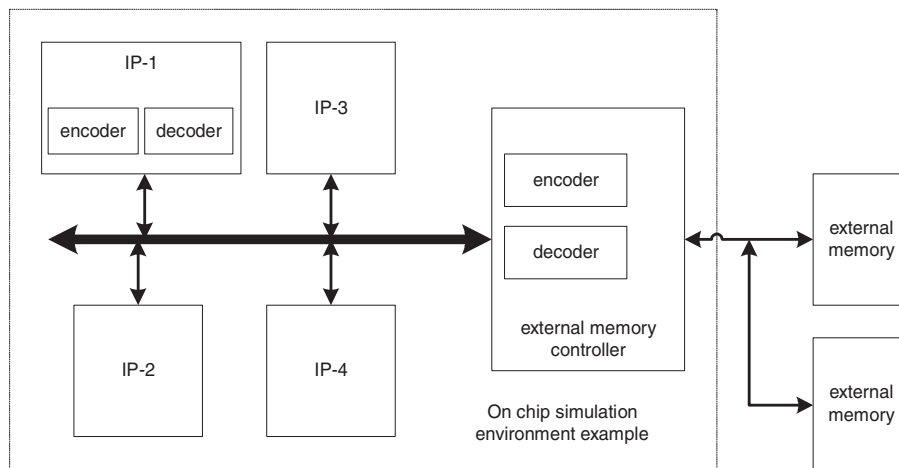


Fig. 8 Simulation environment

compared with the clock period and hence there is no danger of losing extra clock cycles. The Netlist is then simulated at a clock frequency of 120 MHz. The proposed encoding scheme is simulated for 25, 62.5 and 12.5% wire redundancy corresponding to cases 1, 2 and 3 outlined earlier. In all cases of implementations, the extended coded data bus is considered while monitoring all types of switching activity, for example, in the case of 8-bit transaction and using 25% redundancy scheme (case 1 in Section 4), switching activity on the extended 10-bit bus is monitored and the results are compared with the results of BI and CBI.

It is worth mentioning that the research work done previously on reducing energy consumption through minimising the crosstalk from on-chip buses, has used bus models to estimate the energy consumption. The energy estimation in this research is also based on the bus model presented in Section 2 (5) which is derived from the bus model described in Sotiriadis and Chandrakasan [14]. This bus model, like the previous bus models, provides approximate result for the energy consumed because of the combined effect of self and crosstalk couplings. This approximation is due to the use of lumped values for inter-connect capacitance, wire resistance and capacitance and wire-to-substrate capacitance.

Equation (5) in Section 2 requires count of type-4, type-3, type-2, type-1 and self transitions which can simply be obtained with RTL. However, in order to account for the area and power overhead of the codec, the external memory controller in Fig. 8 was synthesised and simulated at gate level. The power consumption of the internal combinational and sequential elements of the codec architecture has been measured using Synopsys power compiler for a sample 8-bit image data set. The total power consumption is 0.0228 mW and is equivalent to a 2.25% power overhead. Similar results can be obtained for higher transfer sizes. Clearly, this power overhead is negligibly small when compared with 22–36% power savings achieved on the on-chip bus.

Table 1: Area comparisons (in μm^2)

Encoding scheme	8-bit	16-bit	32-bit
Proposed encoding	2849	5982	11 367
BI	3203	6726	12 779
CBI	3467	7280	13 833

5.1 Case 1: 25% wire redundancy

Fig. 2 provides percentage energy saving for the corresponding encoded data (proposed I), BI and CBI schemes for three transfer protocols namely 8, 16 and 32-bit of the AHB data bus. For random data (ran8, ran16, ran32), all three encoding schemes are power efficient with the proposed method slightly more power efficient when compared with BI and CBI. The power efficiency of BI is from 18 to 21%. It is 14–15% for CBI and 21–23% for the proposed encoding scheme. However, in the case of application specific data such as image data set (img8, img16 and img32) the proposed scheme outperforms both BI and CBI. The power efficiency of BI is from 17 to 20%. It is 17–18% for CBI and 31–33% for the proposed encoding scheme. The CBI scheme proved to be less power saving compared with BI for both random and image data sets. The reason is that many of the data samples have couplings not enough to invert the data and most of the time the data passes unencoded. The CBI scheme is well suited to the case in which the data has more crosstalk coupling probability per data sample, so that inversion of the data occurs quite frequently.

Another aspect of the encoding scheme (case I) is the reduction of worst crosstalk couplings from data flowing on the bus. This aspect is depicted in Figs. 9 and 10. These figures provide information about saving in switching activity in 8-bit transaction for the two types of data used in this experiment. In the case of random data, although the method is slightly more power efficient than BI and CBI, however, it is the reduction in worst crosstalk couplings (Fig. 9) that gives an edge to the proposed encoding scheme. For image data set, the proposed scheme

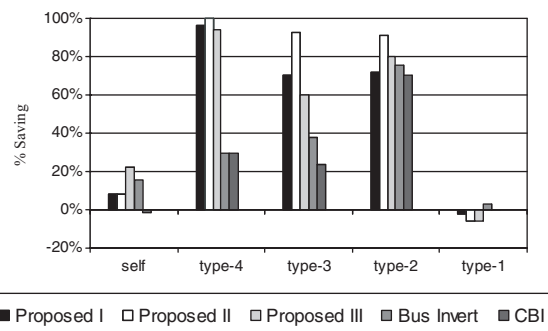


Fig. 9 Saving in switching activity for 8-bit random data

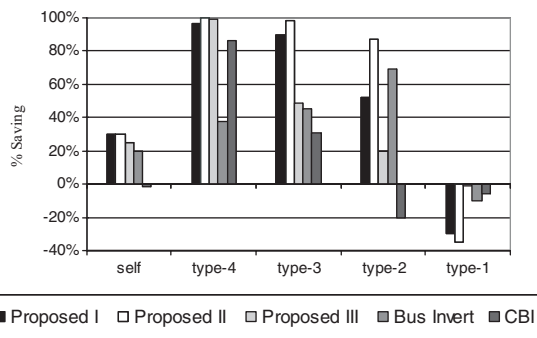


Fig. 10 Saving in switching activity for 8-bit image data

outperforms BI and CBI as well in reducing worst crosstalk couplings as evident from Fig. 10. The corresponding graphs for 16 and 32-bit transaction follow the same pattern and need not be plotted. Figs. 9 and 10 indicate that in the case of the proposed encoding scheme, significant (70–95%) reduction in the worst crosstalk coupling occurs with a 5% increase in type-1 coupling. The reduction in self-switching activity is 8% for random data and 30% for image data sets, respectively.

5.2 Case 2: 62.5% wire redundancy

The simulation results are obtained after placing shield wires to avoid inter-cluster crosstalk as well as between decode bits. The results for this scheme (case 2) are presented in Figs. 2, 9 and 10 using 8, 16 and 32-bit transaction of both random and image data sets. In the case of 8-bit transaction, decoding bits are transmitted using three wires according to Table 2. In the case of 16-bit transaction, the bus is partitioned into two 8 bits and decoding bits in each are transmitted using three wires according to Table 2. The same approach is followed with 32-bit transaction as well. Therefore with the insertion of the shield wires and expanding the decode bits on three wires, the bus bandwidth expands to 13 bits (for 8-bit transaction). The shield wires convert type-4 couplings into type-3 and type-2 into type-1 couplings. Therefore type-4 is completely eliminated, whereas type-3 and type-2 couplings are reduced significantly. Owing to expansion in the bus, type-1 coupling increases and hence energy saving in the proposed encoding scheme slightly reduces by 0.5–1.5% when compared with energy saving with no shield wires. However, it must be noted that shield wires are ground wires and do not change the self-switched capacitance.

The results for energy savings are presented in Fig. 2. With slight reduction, case 2 is still more energy efficient when compared with BI and CBI. By examining Figs. 9 and 10, it is clear that case 2 eliminates type-4 completely and reduces type-3 and type-2 to a considerable extent. This elimination/reduction enhances the error immunity of the scheme to a greater extent.

Table 2: Decode info for 8-bit decoding

Operation on first 4-bit data	Operation on second 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

5.3 Case 3: 12.5% wire redundancy

This corresponds to case 3 in Section 4 and is implemented with one extra wire per 8-bit transaction. The simulation results are plotted in Figs. 2, 9 and 10. In this implementation, the wire overhead is kept the same with BI and CBI. From Fig. 2, it is clear that even with equal wire redundancy, the proposed encoding scheme is more power efficient than BI and CBI. If case 3 is compared with cases 1 and 2, it becomes clear that in most cases case 3 is even more power efficient than its previous implementations.

The second performance parameter is switching activity reduction which is presented in Figs. 9 and 10. The proposed encoding scheme is much more efficient than CBI in reducing self and worst crosstalk coupled switched capacitance. In comparison with BI, the proposed encoding scheme is quite efficient in reducing type-4 and type-3 coupling, however, reduction in type-2 coupling is comparable in case of random data and less than BI in the case of image data. It is also evident from Figs. 9 and 10 that the scheme reduces self transitions more than BI and CBI for both random and image data sets. Owing to better area, energy and crosstalk noise efficiency and equal wire redundancy, the proposed Scheme III is much better and more suitable for high-performance generic SoC implementations.

6 Conclusion

The authors have presented a technique that addresses energy loss and delay problems (because of crosstalk noise) faced by today's tightly coupled on-chip buses implemented in ultra-deep submicron SoC systems. The technique provides energy saving, for 0.18 μm CMOS technology, ranging from 22% for highly decorrelated uniformly distributed random data to 36% for highly correlated application specific data such as those produced in image processing applications. The encoding scheme has been implemented for minimum and maximum wire redundancies. In the case of maximum redundancy, the scheme eliminates N_4 couplings completely and minimises N_3 and N_2 couplings, whereas for the case of minimum redundancy, the scheme minimises to a larger extent all types of worst crosstalk couplings, 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.

7 References

- Macchiarulo, L., Macii, E., and Poncino, M.: 'Wire placement for crosstalk energy minimization in address buses'. Proc. Design, Automation and Test in Europe Conf. and Exhibition, Paris, France, 4–8 March 2002, pp. 158–162
- Guardiani, C., Forzan, C., Franzini, B., and Pandini, D.: '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
- Xiao, T., and Marek-Sadowska, M.: 'Crosstalk reduction by transistor sizing'. Design Automation Conf. 1999. Proc. ASP-DAC '99, Asia and South Pacific, 18–21 January 1999, vol. 1, pp. 137–140
- Hirose, K., and Yasuura, H.: 'A bus delay reduction technique considering crosstalk'. Design Automation and Test in Europe Conf. and Exhibition, Paris, France, 2000, pp. 441–445
- Stan, M.R., and Burleson, W.P.: 'Bus-invert coding for low-power I/O', *IEEE Trans. VLSI Syst.*, 1995, 3, (1), pp. 49–58
- Benini, L., Micheli, G.D., Macii, E., Sciuto, D., and Silvano, C.: 'Asymptotic zero-transition activity encoding for address buses in low-power microprocessor-based systems'. 7th Great Lakes Symp. on VLSI, Urbana, IL, USA, 1997, pp. 7–82

- 7 Osborne, S., Erdogan, A.T., Arslan, T., and Robinson, D.: 'Bus encoding architecture for low-power implementation of an AMBA-based SoC platform', *IEE Proc., Comput. Digit. Tech.*, 2002, **149**, (4), pp. 152–156
- 8 Kim, K.-W., Baek, K.-H., Shanbhag, N., Liu, C.L., and Kang, S.-M.: 'Coupling-driven signal encoding scheme for low-power interface design'. Computer Aided Design, ICCAD-2000. IEEE/ACM Int. Conf., San Jose, CA, USA, 2000, pp. 318–321
- 9 Victor, B., and Keutzer, K.: 'Bus encoding to prevent crosstalk delay'. Computer Aided Design, ICCAD 2001, IEEE/ACM Int. Conf., San Jose, CA, USA, 4–8 November 2001, pp. 57–63
- 10 Lyuh, C.G., and Kim, T.: 'Low power bus encoding with crosstalk delay elimination'. ASIC/SOC Conf., 15th Annual IEEE Int. Conf., Rochester, NY, USA, 2002, pp. 389–393
- 11 Tiehnan, L., Henkel, J., Lekatsas, H., and Wolf, H.W.: 'Enhancing signal integrity through a low-overhead encoding scheme on address buses'. Design, Automation and Test in Europe Conf. and Exhibition, Munich, Germany, 3–7 March 2003, pp. 542–547
- 12 Bertozzi, D., Benini, L., and De Micheli, G.: 'Low power error resilient encoding for on-chip data buses'. Proc. Design, Automation and Test in Europe Conf. and Exhibition, Paris, France, 4–8 March 2002, pp. 102–109. Digital Object Identifier 10.1109/Date.2002.998256
- 13 Duan, C., Tirumala, A., and Khatri, S.P.: 'Analysis and avoidance of cross-talk in on-chip buses'. Proc. 9th Hot Interconnects Conf., Stanford, USA, 2001, pp. 133–138
- 14 Sotiriadis, P.P., and Chandrakasan, A.: 'Low power bus coding techniques considering inter-wire capacitances'. Proc. IEEE Custom Integrated Circuits Conf., CICC 2000, Orlando, FL, USA, 2000, pp. 507–510
- 15 Sotiriadis, P.P., and Chandrakasan, A.: 'Bus energy minimization by transition pattern coding (TPC) in deep sub-micron technologies'. Computer Aided Design, ICCAD-2000, IEEE/ACM Int. Conf., San Jose, CA, USA, 2000, pp. 322–327

