

Energy reduction through crosstalk avoidance coding in networks on chip

Partha Pratim Pande ^{a,*}, Amlan Ganguly ^a, Haibo Zhu ^a, Cristian Grecu ^b

^a School of Electrical Engineering and Computer Science, P.O. Box 642752, Washington State University, Pullman, WA, USA

^b SoC Research Lab, Department of Electrical and Computer Engineering, University of British Columbia, 2332 Main Mall Vancouver, BC, Canada V6T 1Z4

Received 22 December 2006; received in revised form 10 September 2007; accepted 17 September 2007

Available online 1 October 2007

Abstract

Commercial designs are currently integrating from 10 to 100 embedded processors in a single system on chip (SoC) and the number is likely to increase significantly in the near future. With this ever increasing degree of integration, design of communication architectures for large, multi-core SoCs is a challenge. Traditional bus-based systems will no longer be able to meet the clock cycle requirements of these big SoCs. Instead, the communication requirements of these large multi processor SoCs (MP-SoCs) are convened by the emerging network-on-chip (NoC) paradigm. Crosstalk between adjacent wires is an important signal integrity issue in NoC communication fabrics and it can cause timing violations and extra energy dissipation. Crosstalk avoidance codes (CACs) can be used to improve the signal integrity by reducing the effective coupling capacitance, lowering the energy dissipation of wire segments. As NoCs are built on packet-switching, it is advantageous to modify data packets by including coded bits to protect against the negative effects of crosstalk. By incorporating crosstalk avoidance coding in NoC data streams and organizing the CAC-encoded data packets in an efficient manner, so that total number of encoding/decoding operations can be reduced over the communication channel, we are able to achieve lower communication energy, which in turn will help to decrease the overall energy dissipation.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Networks on chip; Crosstalk; Crosstalk avoidance codes; Interconnect energy; Low power interconnects; Wormhole switching

1. Introduction and motivation

The network-on-chip (NoC) design paradigm is viewed as an enabling solution [1–3] for the integration of exceedingly high number of computational and storage blocks in a single chip. The common characteristic of NoC interconnect architectures is that the functional blocks communicate with each other with the help of a routing infrastructure consisting of switches and links. With technology scaling and deep sub-micron (DSM) effects, power consumption is the greatest challenge in today's multi-core SoC design. Subsequently, in NoC domain, a significant

amount of energy is dissipated by the communication links. With shrinking geometry, the inter-wire spacing decreases rapidly [4] while the height and width of the wires do not scale at the same rate. This in turn tends to increase the cross-sectional aspect ratio, increasing the effective coupling capacitance between intra-layer adjacent wires with negative effects on delay, power and signal integrity. The fact that the dielectric constant does not scale down at the same rate also contributes to the increase in coupling capacitance between adjacent wires in the same metal level. This phenomenon is commonly termed as crosstalk. Shielding can be effectively used to reduce crosstalk, and involves placing a grounded wire between every pair of signal lines. Although this is effective in preventing crosstalk within a bus, it has the effect of doubling the wire count. Layout-level solutions for mitigating crosstalk, such as wire spacing, have the disadvantage of being dependent on process

* Corresponding author. Tel.: +1 5093355223.

E-mail addresses: pande@eecs.wsu.edu (P.P. Pande), ganguly@eecs.wsu.edu (A. Ganguly), haibo@eecs.wsu.edu (H. Zhu), grecuc@ece.ubc.ca (C. Grecu).

variation, and their wire area overhead is comparable with shielding [5]. Therefore, doubling the inter-switch wire routing area following these conventional techniques is not the optimum solution. In the deep sub-micron (DSM) technologies crosstalk between adjacent wires is an issue in NoC communication fabrics as in any other VLSI system and it can cause timing violations and extra power dissipation. By incorporating crosstalk avoidance coding (CAC) in NoC data streams we can reduce the effective coupling capacitance of the inter-switch wire segments and hence the communication energy without incurring the non-optimal wire area overhead of shielding/spacing. CACs reduce the worst-case switching capacitance of a wire by ensuring that a transition from one codeword to another does not cause adjacent wires to switch in opposite directions [6].

In addition to the reduction of the effective switching capacitances of the inter-switch wire segments due to incorporation of CACs, two more energy dissipation sources need to be considered. The first one is the codec (coder/decoder) energy and the second one is the extra energy dissipated by the redundant wires introduced by the coding schemes. We propose a method to reduce the codec energy dissipation by modifying the packet structure in a manner that requires only coding/decoding for the first flit (header) of each packet. In addition to the energy dissipation, the timing characteristic of the codec blocks needs to be investigated. The switches together with the inter-switch links of a NoC establish a highly pipelined communication medium, with the delay of each stage within the limit of one clock cycle [7]. The CAC codecs will add extra delay. It is crucial that the codecs do not introduce additional timing overhead to the pipelined communication architecture. We show that through efficient design, the codec logic and the inter-switch link wires can be combined into a single pipeline stage that fits within the clock cycle limit generally accepted for high-performance chips. Thus we avoid adding extra stages of coding and decoding in the communication pipeline. Our aim in this paper is to study the performance of NoC architectures in presence of CAC in terms of energy efficiency, timing and silicon area overhead.

2. Related work

In recent years, there has been an evolving effort in developing on-chip networks to integrate increasingly large number of functional cores in a single die [1,2]. Even before the advent of the NoC paradigm, different research groups investigated various coding schemes to enhance the reliability of bus-based systems. In [8], the authors proposed to employ data encoding to eliminate crosstalk delay within a bus. They presented a detailed analysis of the self-shielding codes and established fundamental theoretical limits on the performance of codes with and without memory. In [9], the authors provided a comprehensive study of the capability of error correcting codes to reduce the crosstalk-

induced bus delay (CIBD), and proved that Dual Rail codes can handle crosstalk better than Hamming codes. The authors of [9] used single error correcting codes (SECs) to minimize crosstalk. These codes are not as efficient as CACs in handling crosstalk related effects. In addition, different low-power coding (LPC) techniques have been proposed to reduce power consumption of on-chip buses [10]. But these LPCs aim at reducing only the self-transitions of a wire. According to Kretzschmar et al. [11], the principal limitation of the applicability of the LPCs is that, due to higher power dissipation in the codec blocks, these codes are energy efficient only if the length of the wire segment exceeds a certain limit. In [12], the authors presented a unified framework for applying coding for systems on chips (SoCs), targeted principally for bus-based systems. In [13], performance of single error correcting and multiple error detecting Hamming codes and cyclic codes in an AMBA bus-based system has been discussed. Error resiliency in NoC fabrics and the trade-offs involved in various error recovery schemes are discussed in [14]. In this work, the authors investigated performances of simple error detection codes like parity or cyclic redundancy check codes and single error-correcting, multiple error-detecting Hamming codes in NoC fabrics. These codes do not have explicit crosstalk avoidance characteristics, which are absolutely necessary in the DSM technology.

The objective of this paper is to explore the energy savings capability of CACs in NoC domain. We characterize performance of multiple NoC architectures in presence of CACs.

3. Crosstalk avoidance in NoC links

A few NoC interconnect architectures have been proposed by different research groups [15]. The common characteristic of these NoC architectures is that the processor/storage cores communicate with each other through switches. Communication between constituent IP blocks in a NoC takes place through packet-switching. Generally wormhole switching is adopted for NoCs, which breaks down a packet into fixed length flow control units or *flits*. The first flit or the *header* contains routing information that helps establishing a path from the source to destination, which is subsequently followed by all the other *payload* flits. Some common NoC architectures are shown in Fig. 1.

3.1. Crosstalk avoidance codes

The effective coupling capacitance of an inter-switch wire segment in a NoC link depends on the transitions in the adjacent wires. As shown in [16] the worst case switching capacitance of a wire segment is given by $(1 + 4\lambda)C_L$, where λ is the ratio of the coupling capacitance to the bulk capacitance and C_L is the load capacitance, including the self capacitance of the wire. By incorporating CACs it is possible to reduce this switching capacitance to $(1 + p\lambda)C_L$,

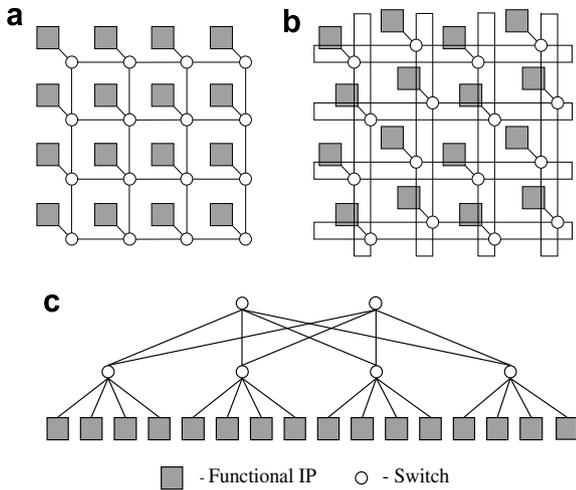


Fig. 1. NoC architectures: (a) mesh, (b) folded-torus and (c) butterfly fly tree (BFT).

where $p = 1, 2,$ or 3 and it is referred to as the maximum coupling. Thus, the worst case energy dissipation of a single wire segment in a NoC link is reduced from $(1 + 4\lambda)V_{dd}^2 C_L$ to $(1 + p\lambda)V_{dd}^2 C_L$, indicating a linear increase in energy savings in presence of CAC with the decrease in coupling capacitance. There are a number of crosstalk avoidance codes [6] proposed in the literature to achieve the above mentioned reduction in effective switching capacitance. Here we consider three representative CACs that achieve different degrees of coupling capacitance reduction.

3.1.1. Forbidden overlap condition (FOC) codes

As mentioned above, a wire has the worst-case switching capacitance of $(1 + 4\lambda)C_L$, when it executes a rising (falling) transition *and* its neighbors execute falling (rising) transitions. If these worst-case transitions are avoided, the maximum coupling can be reduced to $p = 3$. This condition can be satisfied if and only if a codeword having the bit pattern 010 does not make a transition to a codeword having the pattern 101 at the same bit positions. The codes that satisfy the above condition are referred to as forbidden overlap condition (FOC) codes. The simplest method of satisfying the forbidden overlap condition is half-shielding, in which a grounded wire is inserted after every two signal wires. Though simple, this method has the disadvantage of requiring a significant number of extra wires. Another solution is to encode the data links such that the codewords satisfy the forbidden overlap (FO) condition. However, encoding all the bits at once is not feasible for wide links due to prohibitive size and complexity of the codec hardware. In practice, partial coding is adopted, in which the links are divided into sub-channels which are encoded using FOCs. The sub-channels are then combined in such a way as to avoid crosstalk occurrence at their boundaries. Considering a 4-bit sub-channel, the FOC coding scheme is represented in Table 1. For brevity, the table shows a representative section of the entire truth table for a 4-bit wide channel.

Table 1
FOC₄₋₅ coding scheme

Data bits				Coded bits				
d_3	d_2	d_1	d_0	c_4	c_3	c_2	c_1	c_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	0	0	0	1
0	0	1	1	0	0	1	0	1
0	1	0	0	0	0	0	1	1
0	1	0	1	0	0	1	1	1
0	1	1	0	1	0	0	1	1
0	1	1	1	1	0	1	1	1

In this case, two sub-channels can be placed next to each other without any shielding, and not violating the FO condition as shown in Fig. 2. Following this method, a 32-bit uncoded flit becomes 40-bit long after encoding. Thus FOC is a (40, 32) code.

The Boolean expressions relating the original input (d_3 to d_0) and coded bits (c_4 to c_0) for the FOC scheme are expressed as follows:

$$\begin{aligned}
 c_0 &= d_1 + d_2\bar{d}_3 \\
 c_1 &= d_2\bar{d}_3 \\
 c_2 &= d_0 \\
 c_3 &= d_2d_3 \\
 c_4 &= d_1d_2 + d_3
 \end{aligned}$$

3.1.2. Forbidden transition condition (FTC) codes

The maximum capacitive coupling and, hence, the maximum delay, can be reduced even further by extending the list of non-permissible transitions. By ensuring that the transitions between two successive codes do not cause adjacent wires to switch in opposite directions (i.e., if a codeword has a 01 bit pattern, the subsequent codeword cannot have a 10 pattern at the same bit positions, and vice versa), the coupling factor can be reduced to $p = 2$. This condition is referred to as forbidden transition condition, and the CACs satisfying it are known as forbidden transition condition (FTC) codes. Inserting a shielding wire after each signal line can employ the simplest FTC, but causes unreasonable overhead in redundant wires. For wider inter-switch links, a hierarchical encoding is more suitable, where the inter-switch links are divided into sub-channels

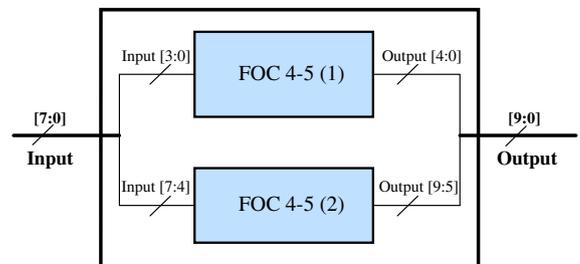


Fig. 2. Combining adjacent sub-channels in FOC coding.

Table 2
FTC₃₋₄ coding scheme

Data bits			Coded bits			
d_2	d_1	d_0	c_3	c_2	c_1	c_0
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	0	1
0	1	1	0	1	0	1
1	0	0	0	1	1	1
1	0	1	1	1	0	0
1	1	0	1	1	0	1
1	1	1	1	1	1	1

that are encoded individually. Considering a 3-bit sub-channel the coding scheme is expressed in Table 2.

For wider message words the entire flit can be subdivided into multiple sub-channels, each having a 3-bit width, and then the individual coded sub-words recombined following the scheme shown in Fig. 3. This scheme of recombination simply places a shielded wire between each adjacent pair of sub-channels. This ensures no forbidden transitions even at the boundaries of the sub-channels. This makes FTC a (53,32) code if the uncoded flit had 32 bits.

The Boolean expressions relating the original input and coded bits for the FTC scheme are expressed as follows:

$$c_0 = d_1 + d_2\bar{d}_0$$

$$c_1 = d_0d_1d_2 + \bar{d}_0\bar{d}_1d_2$$

$$c_2 = d_0 + d_2$$

$$c_3 = d_0d_2 + d_1d_2$$

3.1.3. Forbidden pattern condition (FPC) codes

The same reduction of the coupling factor as for FTCs ($p = 2$) can be achieved by avoiding 010 and 101 bit patterns for each of the code words. This condition is referred to as forbidden pattern condition, and the corresponding CAC is known as forbidden pattern condition (FPC) codes. Considering a 4-bit sub-channel, the coding scheme is expressed in Table 3. Again, for brevity, the table has only 8 entries of the entire truth table for all 4 bits. While combining the sub-channels we made sure that there is no forbidden pattern at the boundaries. Fig. 4 depicts the scheme of avoiding forbidden pattern at the boundaries, considering 4-bit sub-channels. The MSB of a sub-channel

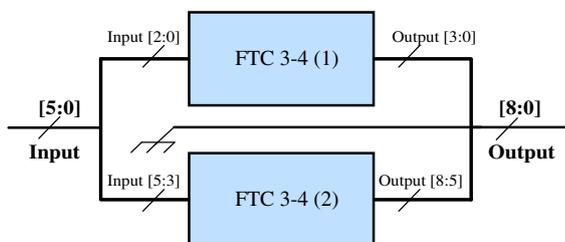


Fig. 3. Block diagram of combining adjacent sub-channels in FTC coding.

Table 3
FPC₄₋₅ coding scheme

Data bits				Coded bits				
d_3	d_2	d_1	d_0	c_4	c_3	c_2	c_1	c_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	1	1	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	1	1	0	0
0	1	0	1	0	0	1	1	1
0	1	1	0	0	1	1	1	0
0	1	1	1	0	1	1	1	1

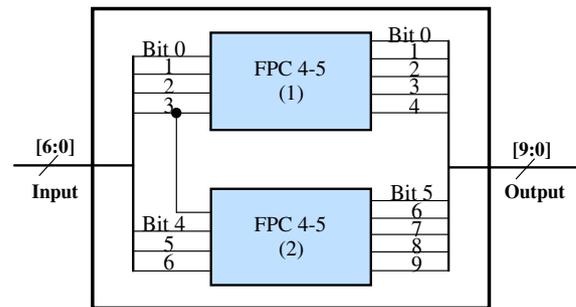


Fig. 4. Combining adjacent sub-channels in FPC coding.

is fed to the LSB of the adjacent one. This method is more efficient than simply placing shielding wires between the encoded sub-channels and consequently results in lesser redundancy in terms of wire overhead. This recombination technique makes the FPC a (52,32) code, if the original flit was 32 bits wide.

The Boolean expressions relating the original input (d_3 to d_0) and coded bits (c_4 to c_0) for the FPC scheme are expressed as follows:

$$c_0 = d_0$$

$$c_1 = d_0d_1 + d_2d_1 + d_1\bar{d}_3 + d_0d_2\bar{d}_3$$

$$c_2 = d_2\bar{d}_3 + d_1d_2 + \bar{d}_0d_2 + d_1\bar{d}_0\bar{d}_3$$

$$c_3 = d_2d_3 + \bar{d}_0d_2 + d_2d_1 + d_1d_3\bar{d}_0$$

$$c_4 = d_3$$

In this paper, we investigate the applicability of these three CAC coding schemes in the NoC domain. Our aim is to study the energy saving, and timing characteristics of these schemes at the cost of extra area overhead they introduce.

3.2. Communication pipelining in NoC in presence of CACs

The exchange of data among the constituent blocks in a SoC is becoming an increasingly difficult task because of growing system size and non-scalable global wire delay. To cope with these issues, designers must divide the end-to-end communication medium into multiple pipelined stages, with the delay in each stage comparable to the clock-cycle budget. In NoC architectures, the inter-switch

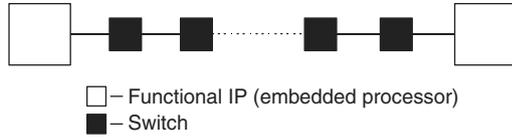


Fig. 5. Data transfer in NoC fabrics.

wire segments, along with the switch blocks, constitute a highly pipelined communication medium characterized by link pipelining, deeply pipelined switches, and latency-insensitive component design [17].

The generic communication medium of any NoC fabric is shown in Fig. 5. Between a source and destination pair there is a path consisting of multiple switch blocks [7] involving several inter-switch and intra-switch stages. The switches generally consist of multiple pipelined stages. The number of intra-switch pipelined stages can vary with the design style and the features incorporated within the switch blocks. However, through careful circuit-level design and analysis, designers can make each intra-switch stage's delay less than the target clock period in a particular technology node. In accordance with ITRS [18], a generally accepted rule of thumb is that the clock cycle of high performance SoCs will saturate at a value in the range of 10–15 FO4 (Fan-out of 4) delay units. We need to ensure that by adding the CAC codec blocks, the constraints on timing can still be met.

4. Energy savings profile in presence of CAC

When flits travel on the interconnection network, both the inter-switch wires and the logic gates in the switches toggle, resulting in energy dissipation. The flits from the source nodes need to traverse multiple hops consisting of switches and wires to reach destinations.

The motivation behind incorporating CAC in the NoC fabric is to reduce switching capacitance of the inter-switch wires and hence make communication among different blocks more energy efficient. So, the metric of interest is the average savings in energy per flit with coding compared to the uncoded case. All the schemes have different number of bits in the encoded flit. A fair comparison in terms of energy savings demands that the redundant wires be also taken into account while comparing the energy dissipation profiles. The metric used in this work for comparison thus takes into account the savings in energy due to the reduced crosstalk, additional energy dissipated in the extra redundant wires and the codecs. The savings in energy per flit per hop is given by

$$E_{\text{savings},j} = E_{\text{link,uncoded}} - (E_{\text{link,coded}} + E_{\text{codec}}) \quad (1)$$

where $E_{\text{link,uncoded}}$ and $E_{\text{link,coded}}$ are the energies dissipated by the uncoded flit and the coded flit in each inter-switch link respectively taking into consideration the redundant wires in the encoded flits due to CAC schemes. E_{codec} is the energy dissipated by each codec. The energy savings

in transporting a single flit, say the i th flit, through h_i hops can be calculated as

$$E_{\text{savings},i} = \sum_{j=1}^{h_i} E_{\text{savings},j} \quad (2)$$

The average energy savings per flit in transporting a packet consisting of P such flits through h_i hops for each flit will be given as

$$\bar{E}_{\text{savings}} = \frac{\sum_{i=1}^P \sum_{j=1}^{h_i} (E_{\text{savings},j})}{P} \quad (3)$$

The metric \bar{E}_{savings} is independent of the specific switch implementation, which may vary based on the design.

In order to quantify the energy savings profile for a NoC interconnect architecture, we determine the energy dissipated in each codec, E_{codec} by running Synopsys™ Prime Power on the gate-level netlist of the codec blocks. To determine the inter-switch link energy in presence and absence of coding, that is, $E_{\text{link,coded}}$ and $E_{\text{link,uncoded}}$, respectively, the capacitance of each interconnect stage, $C_{\text{interconnect}}$ is calculated taking into account the specific layout of each topology and it can be estimated according to the following generic expression:

$$C_{\text{interconnect}} = C_{\text{wire}} \cdot w_{a+1,a} + n \cdot m \cdot (C_G + C_J) \quad (4)$$

where C_{wire} is the wire capacitance per unit length. $w_{a+1,a}$ is the wire length between two consecutive switches. C_G and C_J are the gate and junction capacitance of a minimum size inverter, respectively, n denotes the number of inverters (when buffer insertion is needed) in a particular inter-switch wire segment and m is their corresponding size with respect to a minimum size inverter. However, the inter-switch wire lengths are not long enough to necessitate buffer insertion in the architectures used in this paper and hence Eq. (4) reduces to the expression below.

$$C_{\text{interconnect}} = C_{\text{wire}} \cdot w_{a+1,a} \quad (5)$$

While calculating C_{wire} without any coding we have considered the worst case switching scenario, where the two adjacent wires switch in the opposite direction of the signal line simultaneously. The typical value of C_{wire} at the 90 nm technology used here is 0.22 pF/mm [21]. The parameter $w_{a+1,a}$ can be calculated depending on the network architecture used. For mesh architecture the inter-switch wire length is given by

$$w_{a+1,a} = \frac{\sqrt{\text{Area}}}{\sqrt{N} - 1}, \quad (6)$$

where Area is the area of the silicon die used and N is the number of individual IP blocks in the SoC. The inter-switch wire length for folded-torus architecture is twice that of the mesh as it connects every alternate IP block in the network. The same inter-switch wire length for the BFT architecture between levels $a+1$ and a is given by Eq. (7), where levels is the total number of levels needed for implementing the BFT architecture given by $\log_4 N$.

$$w_{a+1,a} = \frac{\sqrt{\text{Area}}}{2^{\text{levels}-a}} \quad (7)$$

In the presence of CACs the effective wire capacitance will be reduced according to the coding scheme and this will help in reducing the link energy. On the other hand the additional energy dissipated by the codecs and redundant wires added by the coding schemes need to be considered as well. Our aim is to study the effects of all these factors on the overall energy savings of NoC communication infrastructures.

5. Increasing energy savings by modifying the flit structure

As shown in Fig. 5, when data routing are performed, the flits need to be coded and decoded at each intermediate switch node. These operations will have a significant effect on overall energy dissipation. If the packet structure can be modified in such a way that coding/decoding is needed only at the source and destination nodes, then there will be no extra power dissipation arising out of the codec blocks in the intermediate nodes. The original packet structure is shown in Fig. 6a [15]. If the flit structure is modified so that only the header flits contain the control information, then the payload flits need not be coded or decoded at each intermediate switch node. Eventually, this will help in reducing the overall communication energy dissipation. The original and modified flit structures are shown in Figs. 6a and b, respectively.

A new field called *flit count* is incorporated in the header flit, which keeps track of the number of payload flits in a packet. The *type* field in the payload flits becomes unnecessary as the switch knows the number of payload flits that follow a header from the *flit count* field in the header. The *pktid* field in the payload flits of Fig. 6 links each flit to a particular packet. After decoding the header flit the switch knows the number of payload flits to expect from the *flit count* and then systematically routes all the payload flits bearing the same *pktid* along the path set by the header flit. This can be done without decoding the payload flits as CAC encoding of the *pktid* produces unique identifiers which can be directly used to link the payload flits with a particular header. The switches only need a negligible modification to use the CAC encoded *pktid* bits instead of the original *pktid* for mapping each payload flit to its corre-

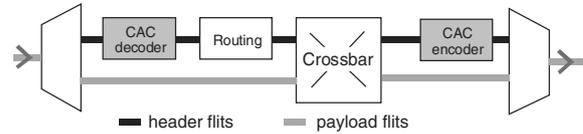


Fig. 7. Data path for header and body flits in the switch.

sponding packet. The payload flits need not be decoded and encoded at each intermediate switch, thus reducing the codec overhead and making the incorporation of CAC schemes in a NoC communication fabric more energy efficient. The path of the header and payload flits through the switch blocks in presence of modified flit structure is shown in Fig. 7.

The energy dissipated by the codec will not be included for the body flits in Eq. (1) when the modified flit structure of Fig. 6b is used. Thus, for payload flits (1) will be modified as follows:

$$E_{\text{savings},j} = E_{\text{link,uncoded}} - E_{\text{link,coded}} \quad (8)$$

Thus the modification of the flit structure increases the savings in energy at each hop that the flit traverses. Hence the savings in energy for the whole network will increase by adopting the modified flit structure.

6. Experimental results and analysis

To study the effects of the CAC schemes on the performance of different NoC infrastructures, we considered a system consisting of 64 IP blocks and mapped them onto the interconnect architectures, as shown in Fig. 1. This system size was selected to reflect the state of the art emerging SoCs. In [3], Intel has demonstrated an 80-core processor arranged in an 8 × 10 regular grid built on fundamental NoC concepts. We characterize the NoCs in terms of three principal metrics: energy savings, area overhead and timing. Messages were injected with a uniform traffic pattern (in each cycle, all IP cores can generate messages with the same probability). The routing mechanism used for the mesh and folded-torus architectures was the *e-cube* (dimension order) routing and for BFT was the least common ancestor (LCA) determination [19]. Simulations were performed using 90 nm technology node parameters. The codec blocks were synthesized with the CMP [20] standard

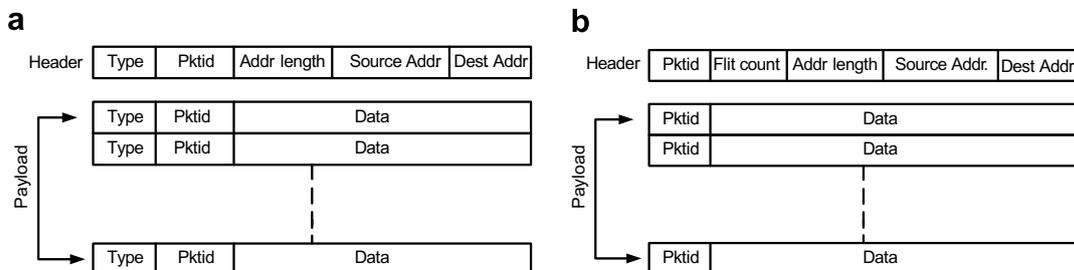


Fig. 6. Flit structure: (a) original and (b) modified.

Table 4
Simulation parameters

Architecture	Message length (Flits)	Buffer depth (Flits)	Number of ports
Mesh	16	2	5
Folded Torus	16	2	5
BFT	16	2	6

cell libraries. The parameters used for the purpose of simulations are listed in Table 4.

6.1. Energy savings profile

The average energy dissipation profile for any NoC follows a saturating trend with injection load [15]. Consequently, the energy savings profile will maintain the same trend. The energy dissipation and hence savings in energy of each inter-switch wire segment is a function of λ , the ratio of the coupling capacitance to the bulk capacitance. For a given interconnect geometry, the value of λ depends on the metal coverage in upper and lower metal layers [12]. We investigated the energy savings profiles for comparison at the two representative values of $\lambda = 1$ and 4. The trend in the energy savings profile is found to be identical for both cases and hence the profiles at only $\lambda = 1$ are shown in the paper.

Figs. 8–10 show the variation in energy savings per flit for mesh, folded torus and BFT-based NoC architectures, respectively, at $\lambda = 1$.

As seen in Figs. 8–10, maximum energy savings are obtained for the folded-torus architecture. The energy savings profile for the NoC depends on the two factors: the inter-switch wire lengths and the average path length. This occurs because the savings in energy is only due to reduction of effective capacitance on the wires, which in turn depends on the length of the wire and how frequently that wire is traversed i.e., the average path length of a flit. The inter-switch wire lengths and the average path-lengths (in terms of number of hops) for each of the architectures are listed in Table 5. In accordance with Eq. (7), the wire lengths in BFT architecture vary with the levels of the

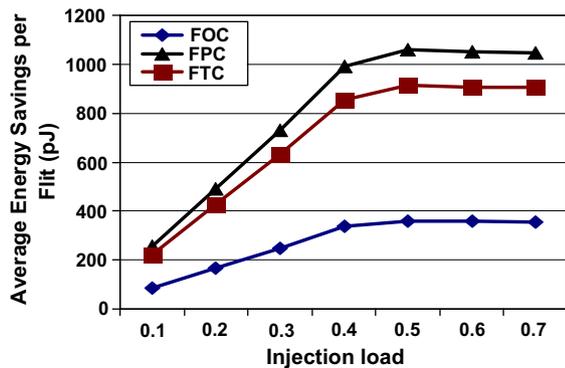


Fig. 8. Energy savings profile for a mesh based NoC.

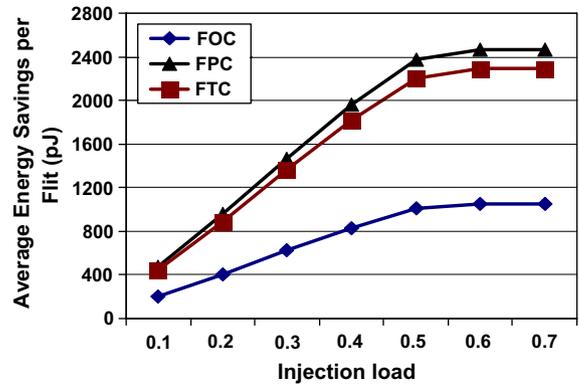


Fig. 9. Energy savings profile for a folded-torus based NoC.

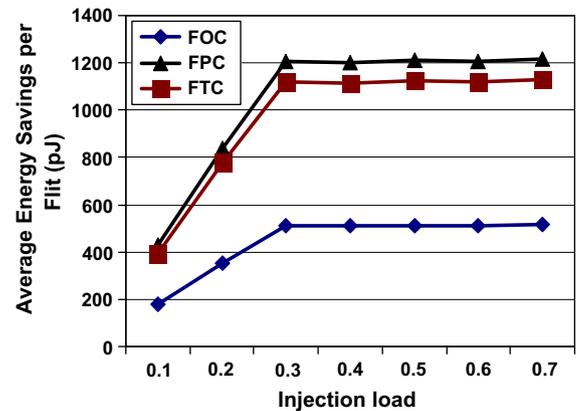


Fig. 10. Energy savings profile for a butterfly fat tree-based NoC.

Table 5
Average path length distribution

Architecture	Length of interconnect (mm)	Average path lengths (hops)
Mesh	2.86	7.31626
Folded-Torus	5.72	6.04331
BFT		
Level 1	10	1.52242
Level 2	5	1.90302
Level 3	2.5	1.99958

nodes. The inter-switch wire length in the folded-torus architecture is twice that of the mesh topology. Although the upper level inter-switch links in BFT are longer than those of folded-torus, the number of links spanning the lower levels is higher and those links are much shorter [7]. So, the energy savings in folded-torus architecture are most pronounced as the inter-switch wires are longer than those in mesh and both have comparable average path lengths. Though the top level wires in the BFT architecture are longer than those of folded-torus, because there are very few hops in that level, the savings are less than that of the folded-torus.

However, the energy savings can be considerably improved by modifying the flit structure as shown in Fig. 6. As all the routing and control information is contained only in the header flit and the payload flits follow the already established path, there is no need to code/decode the payload flits on the fly at every switch. Instead, it is sufficient to code the payload flits at the source and decode at the destination switch. Thus, the coding and decoding overhead is greatly reduced as this process is now done only at the source and destination switches once for all the payload flits. The header flit however, still undergoes coding and decoding at all the switches. With this modified flit structure, the energy savings for $\lambda = 1$ are plotted for a mesh architecture in Fig. 11, a folded-torus architecture in Fig. 12 and for a BFT architecture in Fig. 13.

It is observed that the energy savings can be made more significant by adopting the modified flit structure with negligible increment in the complexity of the switch blocks. Another important point to note here is that the FOC scheme is the least energy efficient one. FTC and FPC have very similar energy savings profile and they are better than FOC. Table 6 quantifies the additional gain in energy sav-

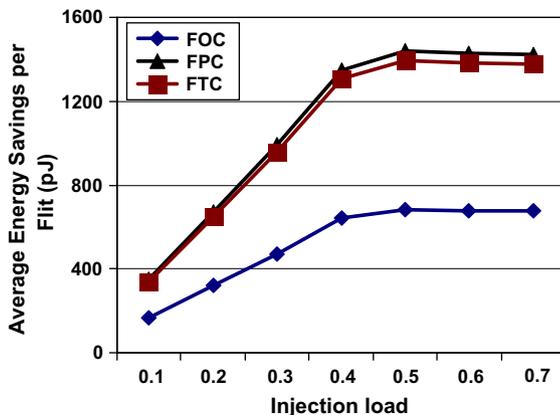


Fig. 11. Energy savings profile for a mesh based NoC at $\lambda = 1$ with modified flit structure.

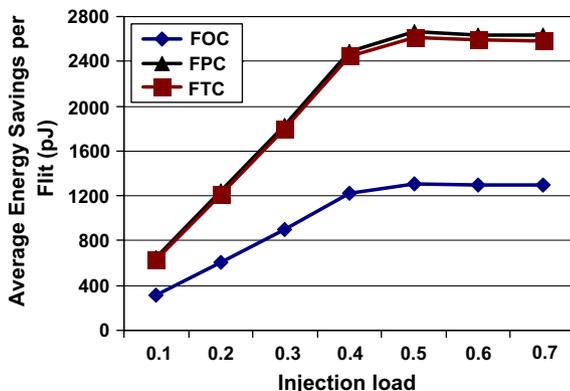


Fig. 12. Energy savings profile for a folded-torus based NoC at $\lambda = 1$ with modified flit structure.

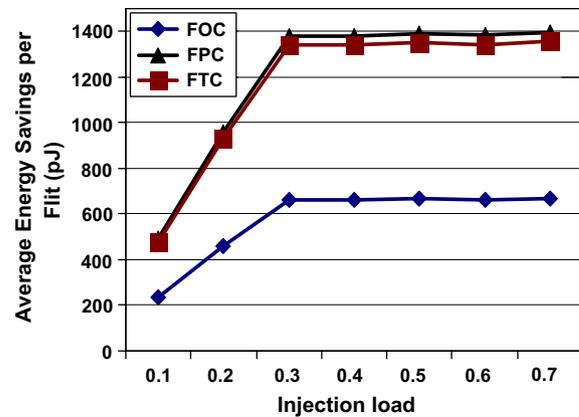


Fig. 13. Energy savings profile for a butterfly fat tree-based NoC at $\lambda = 1$ with modified flit structure.

Table 6
Gain in energy savings with modified flit structure

Architecture	CAC scheme	Energy savings with original flit structure (pJ)	Energy savings with modified flit structure (pJ)	Gain in energy savings (%)
Mesh	FOC	982	1307	33.0
	FTC	2134	2614	22.5
	FPC	2282	2664	16.7
Folded-Torus	FOC	2264	2570	13.5
	FTC	4675	5128	9.6
	FPC	4858	5218	7.4
Butterfly Fat Tree	FOC	1108	1261	13.8
	FTC	2291	2517	9.8
	FPC	2382	2562	7.5

ings at network saturation by adopting the modified flit structure for $\lambda = 4$.

6.2. Area overhead

While evaluating the performance of CAC schemes we need to consider the extra silicon area the codec blocks add to the NoC switch blocks as well as the additional wiring area caused in the routing channels by the redundant wires. Through RTL level design and synthesis in 90 nm technology node, we found that the switches, without any coding scheme consist of approximately 30 K gates. Here, we consider a two-input minimum-sized NAND structure as a reference gate. In comparison, the codecs for FOC, FPC and FTC have around 650, 1000 and 770 gates respectively. Consequently the extra area overhead added by the CAC codecs is relatively insignificant.

The different CAC schemes add different number of redundant wires. The (40, 32) FOC code adds eight extra bits, the (53, 32) FTC code adds 21 and the (52, 32) FPC code adds 20 additional bits to guard against crosstalk. Traditional crosstalk avoidance schemes like shielding or spacing double the routing channel area overhead. This is because, when using shielding, a ground wire is placed between each signal

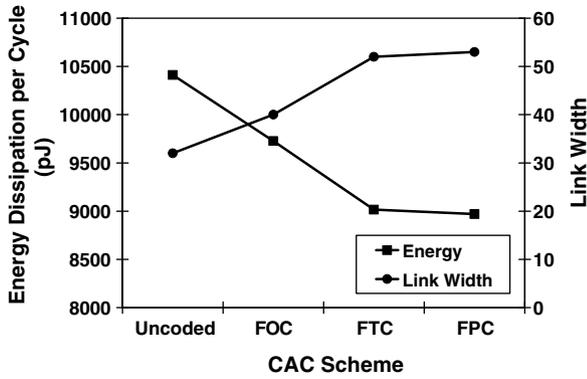


Fig. 14. Energy dissipation in a mesh based NoC along with the link width for each CAC scheme.

wire requiring 63 bits for an original 32-bit interconnect. Spacing out the wires to reduce coupling capacitance also has the effect of doubling the wiring area. Thus, the wiring area overhead for implementing these CAC coding schemes is less than that caused by traditional layout based crosstalk avoidance schemes. Fig. 14 shows the actual energy expenditure at network saturation for a mesh based NoC along with the increase in link width for each CAC scheme considering the modified flit structure. The silicon area overhead in the switches is summarized in Table 7.

As can be seen from Fig. 14 and Table 6, the savings in energy is considerable compared to the actual expenditure and hence the silicon and wiring area overheads are justified.

6.3. Timing

As discussed in Section 3.2 the switches consist of multiple pipelined stages. As shown in [7] one of the possible implementations the switches may consist of three stages: (1) input arbitration, (2) routing and (3) output arbitration. It is already shown in [7] that each intra-switch stage’s delay can be made less than this target clock period in a particular technology node. In presence of CAC there can be additional pipelined stages corresponding to encoder and decoder blocks, as shown in Fig. 15.

Through RTL design and synthesis using a 90 nm standard cell library and Synopsys synthesis tools, we obtain the delays along the critical paths of each encoder and decoder for all the coding schemes. The delay values corresponding to all the coding schemes are shown in Table 8. It is evident that all the coding schemes achieve the target

Table 7
Percentage increase in silicon area of the NoC switches for each CAC scheme

CAC scheme	Switch area with CAC codec	Percentage increase over uncoded case (%)
Uncoded	30,000	–
FOC	30,650	2.17
FTC	30,770	2.57
FPC	31,000	3.33

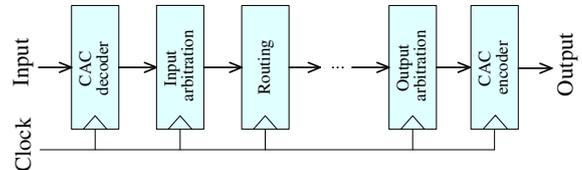


Fig. 15. Pipelined intra-switch stages in presence of coding.

delay values within the limit of one clock cycle. Consequently they will not affect the data introduction rate.

However, the delays of the codec blocks are very small. Consequently the sum of inter-switch wire delay and the delays of the encoder and decoder is less than the 15 FO4 limit. As a result the CAC encoder, decoder and the inter-switch wire segments can be merged into a single pipelined stage. Table 9 shows the delay on the inter-switch link for all the architectures studied in this paper.

The CAC schemes reduce the effective crosstalk in the inter-switch wire segments and hence reduce the corresponding crosstalk induced bus delay (CIBD) [9]. The worst case CIBD for an interconnect without any crosstalk prevention is $(1 + 4\lambda)\tau_0$, where τ_0 is the delay of an isolated wire. Depending on the particular crosstalk pattern that each scheme avoids, the worst case delay on the links is reduced to $(1 + 3\lambda)\tau_0$ for the FOC scheme and $(1 + 2\lambda)\tau_0$ for the FTC and FPC schemes. As a result the inter-switch wire delay in FOC is more than that of FPC and FTC. But as shown in Table 6 the delays of the FOC encoder and

Table 8
Critical path delay of codec blocks

Scheme	Block	Delay (FO4)
FOC	Encoder	0.50
	Decoder	0.25
FPC	Encoder	4.25
	Decoder	3.75
FTC	Encoder	2.75
	Decoder	2.50

Table 9
Inter-switch wire delays

Coding Scheme	Architecture	Delay (FO4)
Uncoded	Mesh	0.9
	Folded-Torus	3.6
	BFT level 1	11.0
	BFT level 2	2.75
	BFT level 3	0.7
FOC	Mesh	0.7
	Folded Torus	2.9
	BFT level 1	8.8
	BFT level 2	2.2
	BFT level 3	0.6
FTC/FPC	Mesh	0.5
	Folded Torus	2.2
	BFT level 1	6.6
	BFT level 2	1.7
	BFT level 3	0.4

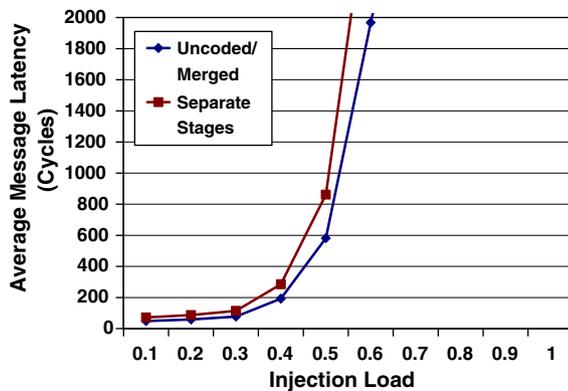


Fig. 16. Average message latency in presence of coding.

decoder are extremely small. Consequently the combined delay of the inter-switch wire segments and the codec in FOC is still well within the one clock cycle limit of 15 FO4. Though the critical path delays of the FTC and FPC codecs are longer than that of FOC, but they are able to reduce the delay on the inter-switch wires more and hence they can also be similarly combined with the interconnects into a single pipelined stage. This implementation is more efficient in terms of system latency as no additional pipelined stages are added to the switch anymore. Fig. 16 plots the latency, measured in clock cycles, in two cases. In the first case, the encoder and decoder are considered as individual pipeline stages. In the second case, the encoder, decoder and the inter-switch wire segments are merged into a single pipeline stage, for a mesh based NoC architecture. Merging the CAC codecs with existing inter-switch stage is a better implementation, as it results in no latency penalty compared to the uncoded case.

In case of the modified flit structure the payload flits need not be decoded and encoded at each switch. Therefore, even if the encoder and decoder are implemented as separate stages the latency does not increase significantly.

7. Conclusions

Network on chip is emerging as a revolutionary method to integrate numerous cores in a single SoC. Widespread adoption of NoC paradigm will be possible if it addresses the signal integrity issues in addition to easing the design process. By incorporating crosstalk avoidance codes (CACs) in NoC data stream it is possible to reduce the worst-case coupling capacitance of inter-switch wire segments and consequently the energy dissipation in communication. The energy savings arising out of incorporating CACs depend on the distribution of inter-switch wires of different lengths and the packet structure. We proposed a method of reducing the energy dissipation by eliminating the need for CAC coding/decoding of payload flits at intermediate switches between communicating NoC cores. It is observed that maximum energy savings are achieved for folded-torus architecture as it consists of uniformly distributed long inter-switch

wire segments. We show how our method of modifying the packet structure and reducing the coding/decoding overhead makes it possible to achieve higher savings in energy in conjunction with crosstalk protection.

References

- [1] L. Benini, G. De Micheli, Networks on chips: a new SoC paradigm, *IEEE Computer* (January) (2002) 70–78.
- [2] P. Magarshack, P.G. Paulin, System-on-chip beyond the nanometer wall, *Proceedings of 40th Design Automation Conference (DAC 03)*, ACM Press, 2003, pp. 419–424.
- [3] S. Vangal, et al., An 80-Tile 1.28 TFLOPS network-on-chip in 65 nm CMOS, in: *Proceedings of the IEEE International Solid-State Circuits Conference*, 11–15 February, 2007, pp. 98–589.
- [4] R. Ho, K.W. Mai, M.A. Horowitz, The Future of Wires, *Proceedings of the IEEE* 89 (4) (2001) 490–504.
- [5] H. Tseng, L. Scheffer, C. Sechen, Timing-and crosstalk-driven area routing, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20 (4) (2001) 528–544.
- [6] S.R. Sridhara, N.R. Shanbhag, Coding for reliable on-chip buses: fundamental limits and practical codes, in: *Proceedings of the IEEE International Conference on VLSI Design*, 3–7 January, 2005, pp. 417–422.
- [7] C. Grecu, P.P. Pande, A. Ivanov, R. Saleh, Timing analysis of network on chip architectures for MP-SoC platforms, *Microelectronics Journal* 36 (9) (2005) 833–845.
- [8] B. Victor, K. Keutzer, Bus encoding to prevent crosstalk delay, in: *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD)*, 4–8 November, 2001, pp. 57–63.
- [9] D. Rossi, C. Metra, A.K. Nieuwland, A. Katoch, Exploiting ECC redundancy to minimize crosstalk impact, *IEEE Design & Test of Computers* 22 (1) (2005) 59–70.
- [10] M.R. Stan, W.P. Bursleson, Low-power encodings for global communication in CMOS VLSI, *IEEE Transactions on Very Large Scale Integration (TVLSI) Systems* 5 (4) (1997) 444–455.
- [11] C. Kretzschmar, A.K. Nieuwland, D. Muller, Why Transition Coding for Power Minimization of on-Chip Buses does not work, in: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, (DATE)*, 16–20 February, 2004, pp. 512–517.
- [12] S.R. Sridhara, N.R. Shanbhag, Coding for system-on-chip networks: a unified framework, *IEEE Transactions on Very Large Scale Integration (TVLSI) Systems* 13 (6) (2005) 655–667.
- [13] D. Bertozzi, L. Benini, G. De Micheli, Error control schemes for on-chip communication links: the energy-reliability tradeoff, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24 (6) (2005) 818–831.
- [14] S. Murali, G. De Micheli, L. Benini, T. Theocharides, N. Vijaykrishnan, M. Irwin, Analysis of error recovery schemes for networks on chips, *IEEE Design & Test of Computers* 22 (5) (2005) 434–442.
- [15] P.P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh, Performance evaluation and design trade-offs for network on chip interconnect architectures, *IEEE Transactions on Computers* 54 (8) (2005) 1025–1040.
- [16] P.P. Sotiriadis, A.P. Chandrakasan, A bus energy model for deep submicron technology, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 10 (3) (2002) 341–350.
- [17] L. Benini, D. Bertozzi, Xpipes: a network-on-chip architecture for gigascale systems-on-chip, *IEEE Circuits and Systems Magazine* 4 (2) (2004) 18–31.
- [18] ITRS 2005 Documents, <http://www.itrs.net/Links/2005ITRS/SysDrivers2005.pdf>.
- [19] J. Duato, S. Yalamanchili, L. Ni, *Interconnection Networks – An Engineering Approach*, Morgan Kaufmann, 2002.
- [20] CMP 90 nm technology library, <http://cmp.imag.fr/products/ic/?p=STCMOS090>.

- [21] P. Kapur, G. Chandra, J.P. McVittie, K.C. Saraswat, Technology and reliability constrained future copper interconnects – part II: performance implications, *IEEE Transactions on Electron Devices* 49 (4) (2002) 598–604.



Partha Pratim Pande is an Assistant Professor at the school of Electrical Engineering & Computer Science, Washington State University, Pullman, Washington, USA. Partha received his PhD in electrical and Computer Engineering from the University of British Columbia in 2005 and an MS in Computer Science from the National University of Singapore in 2002. His primary research interests lie in the area of design and Test of Networks on Chip, Fault Tolerance and reliability of Multi-processor SoC (MP-SoC) plat-

forms. Partha is serving in the program committees of different international conferences, like IOLTS, ATS, MWSACS and DELTA. He is a member of IEEE.



Amlan Ganguly is a doctoral candidate at the School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington, USA. He received his B.Tech (Hons.) degree in Electronics and Electrical Communication Engineering from the Indian Institute of Technology, Kharagpur, India. After a brief stint in Intel India Development Centre, Bangalore, India he Joined the graduate program at WSU. His research interests include design of fault tolerant interconnection infrastructures for

Multi-Processor SoC platforms and novel architectures for On-Chip networks. Amlan is a student member of the IEEE.



Haibo Zhu was born in Changzhou, China, in 1978. He received B.S. and M.S degrees in electrical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2001 and 2004 respectively. He also received M.S. degree in electrical engineering from Washington State University, Pullman, Washington, USA in 2007. Since then he joined Link-a-media Devices, in Santa Clara, CA.



Cristian Grecu received his B.S. and M.Eng. degrees in Electrical Engineering from the Technical University of Iasi, Romania, and the M.A.Sc degree from the University of British Columbia, Canada.

He is currently a doctoral candidate in the Department of Electrical and Computer Engineering, University of British Columbia, Canada. His research interests focus on design, test and fault tolerance of large SoCs, with particular emphasis on their data communication infrastructures.