2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing

PAM: a Packet Manipulation Mechanism for Mitigating Crosstalk Faults in NoCs

Zahra Shirmohammadi *shirmohammadi@kish.sharif.edu*
Mohsen Ansari *mansari@ce.sharif.edu*
Sanaz Kazemi Abharian *skazemi@ce.sharif.edu*
Sepideh Safari *ssafari@ce.sharif.edu*
Seyed Ghassem Miremadi *miremadi@sharif.edu*

Dependable Systems Laboratory, Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran

***Abstract*— This paper proposes an efficient mechanism that mitigates crosstalk faults in Network-on-Chips (NoCs). This is done by using a Packet Manipulating mechanism called PAM for reliable data transfer of NoCs. PAM investigates the transitions of a packet to minimize the forbidden transition patterns appearing during the flit traversal in NoCs. To do this, the content of a packet is manipulated using three different manipulating mechanisms. In other words, PAM manipulates the content of packet in three manipulating modes including: vertical, horizontal and diagonal modes. Then, comparing the transitions of these manipulating mechanisms, a packet with minimum numbers of transitions is selected to be injected into the channel. A tag field is added to the manipulated packet in the sender to be recovered in the receiver. The evaluation of PAM shows that the proposed mechanism can mitigate crosstalk faults with fewer overheads with respect to the other state-of-the art crosstalk mitigating mechanism.**

Keywords—component: Network-on-Chip; Reliability; Crosstalk Faults.

I. INTRODUCTION

Network-on-Chips (NoCs) have been proposed as a promising solution to account the ever increasing communication requirements of System-on-Chips (SoCs) [1]. In the architecture of NoCs, data is sent and received in the form of packets [2]. To exchange these packets between Processing Elements (PEs) in each cycle, packets are divided into the flow control units called flits [2]. Traversal of these flits is done using parallel and adjacent wires between PEs. With increasing the technology size, the thickness of these wires decreases faster than their width and height. This increases the ratio of the coupling capacitance to the total capacitance and increases the probability of crosstalk faults [3]. Crosstalk fault seriously threats the reliability of data transfer in the wires of NoCs. Timing violations such as delays in rising/falling transitions speed up in rising/falling transitions and also unwanted voltage glitches are the effects of crosstalk faults that threat the reliability of NoCs [4]. The intensity of crosstalk faults depends on the transition patterns appearing on the wires of NoCs. For example, transition patterns '-↑↑', '-↓↓', '↑↑-' and '↓↓-' in a 3-bit NoC communication channel, are the weakest patterns from the crosstalk fault point of view; while, transition patterns '↑↓↑' and '↓↑↓' or Triplet Opposite Direction (TOD) impose the strongest crosstalk effects. In more precisely 5-bit delay modeling, '↑↓↑↓↑' impose the strongest crosstalk effects [5]. In this content, symbols '↑' and '↓' are used to represent transitions (0→1) and (1→0) respectively also symbol '-' refers to no transitions. Several mechanisms in the different levels of design abstraction try to relieve the effects of crosstalk faults. In the lowest level of design abstraction, physical level mechanisms reduce the crosstalk faults using mechanisms such as shielding [6], repeater insertion [7] and wire spacing [8]. However, these mechanisms suffer from the area overheads that they impose to NoC-based systems. In transistor level, crosstalk faults are mitigated by intentional timing skewing the timings between a sender and receiver [9]. However, transistor level mechanisms are not favorable by designers. The first reason is that these mechanisms should be applied in the repeater inserted channels that they have the area overheads and the second reason is challenges in timing concerns between a sender and a receiver. In the next level of design abstraction, in Register Transfer Level (RTL) [10]-[15], coding mechanisms such as Crosstalk Avoidance Codes (CACs) prevent the crosstalk faults by omitting different transition classes. CACs are categorized into the One Lambda Codes (OLCs) [10], Forbidden Transition Codes (FTCs) [11] or Forbidden Pattern Free (FPFs) codes [12][24][30], and Forbidden Overlapped Codes (FOCs) [13] based on the patterns that they can remove from the code words. One of the CACs that reduce the crosstalk faults more efficiently are Forbidden Pattern Free (FPF) codes. In FPFs, crosstalk fault is reduced by omitting TODs in the form the codewords before injecting the data word into the channels. One of FPF coding mechanisms that have been proposed recently is Fibonacci coding mechanism. This coding mechanism uses Fibonacci sequence as bases to produce code words [12]. Producing these codewords requires the codec including the encoder in the sender to produce the codewords and the decoder in the receiver to recover the data word from the codewords. All of these coding mechanisms impose overheads to the switches of NoC-based systems. Although several mechanisms have been proposed to deal with the problem of crosstalk fault in NoCs, to the best of our knowledge, all of these mechanisms at different levels of design abstraction have deteriorations in important NoC parameters that are performance, power consumption, and area occupation. This paper proposes an efficient packet manipulating mechanism for mitigating the crosstalk faults in NoCs called PAM. In PAM, the content of a packet is manipulated using three different manipulating modes including vertical, horizontal and diagonal modes. Manipulating the content of a packet by applying these modes and then comparing the transitions of these manipulating mechanisms, a packet with minimum numbers of transitions is

978-1-5090-0154-5/15 $31.00 © 2015 IEEE
DOI 10.1109/CIT/IUCC/DASC/PICOM.2015.281

selected to be injected into the channel. A tag field is added to the manipulated packets to be recovered in the receiver. The evaluation of PAM shows that the proposed mechanism can mitigate the crosstalk faults with fewer overheads with respect to the state-of-the art Fibonacci coding mechanism. The rest of the paper is organized as following: An overview of NoC and crosstalk fault effects is presented in Section II. Section III reviews the related work and Section IV introduces the proposed crosstalk mitigation mechanism and evaluated in Section V, finally conclusion remarks are given in Section VI.

II. Background and Motivations

Due to the effects of the PAM in the architecture of NoCs and as the PAM targets to tackle crosstalk faults in NoCs; the architecture of NoC and the crosstalk fault model is discussed in this section in more details.

A. NoC Architecture Review

The architecture of NoCs consists of *M*×*N* switches and Processing Elements (PEs) [14]. The structure of the switch of NoC is shown in Fig 1. PEs are connected in the form of specific topology using physical channels. Each of these physical channels consists of wires that their width varies from several ten to several hundred wires. Layout efficiency, appropriate electrical properties and simplicity are among the important factors for designers in selecting the type of topology in NoCs. According to these factors, mesh and hypercube are the most popular topologies in NoC designing [14]. In NoCs, communicating data between PEs on the wires is done using packets. These packets are divided into fixed-size units, called flits. The size of these flits is equal to physical channel width. In the architecture of NoCs, switches play an important role in connecting PEs as a processing part and channels as communicating parts of network. Each switch is connected to its north, south, east and west neighbor PEs using five physical channels. Each physical channel is timely multiplexed between some buffers, called virtual channels to increase the performance of the network. The way that the input channel is connected to the output channel of the switch is called switching method. Low buffering requirements and making average packet delivery time almost independent of the distance between source and destination makes wormhole switching as a widely used switching method. Each packet is consisting of multiple flits that header has the routing information. Arriving the header flit in the input of virtual channel either from the previous PE or the PE connected to the same switch, the destination of the packets (north, south, east, west or the processing element) is decided in the switch, and the switch is configured by sending appropriate signals to the crossbar. In the next step, the incoming flit is routed to the selected outgoing virtual channel. In the case of existence of a free virtual channel in the selected outgoing physical channel, the header flit will be transferred to the next PE and the other flits in the packet follow it in a pipelined fashion, otherwise the header flit has to wait until a virtual channel of the selected outgoing physical channel becomes free. During these flits traversal between PEs, appearing different transitions on a wire

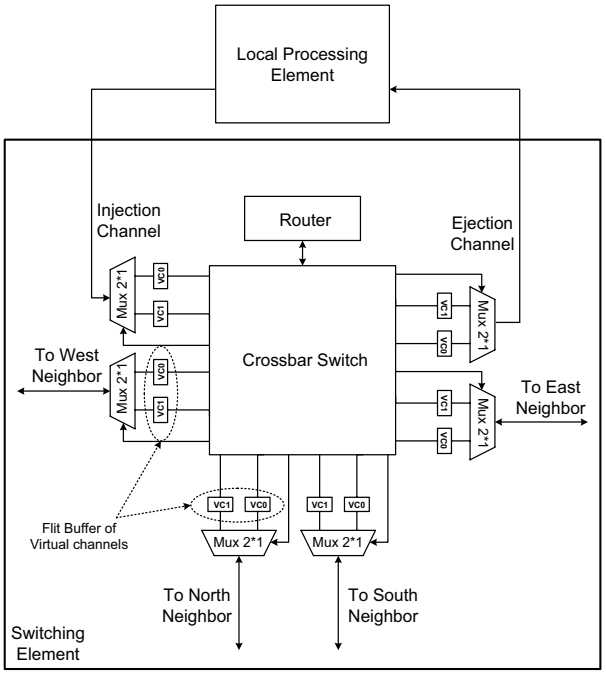

Fig.1. A mesh based NOC switch architecture

leads to charging and discharging the capacitance between wires and producing the crosstalk faults.

B. Crosstalk Faults

Although technology scaling has reached a miniaturization levels where multiple PEs can be integrated in a single die; but, with scaling down into the nanometer regime, a new evil rises: *unreliable NoC.* The reason behind increasing concerns for reliability in NoCs is that the effects of fault resources are becoming stronger with shrinking the transistor dimensions. Consequently, this will threat the NoC data reliability in the form of packet loss, misrouting and/or data corruption [14]. One of these reliability challenges in NoCs is crosstalk faults. According to ITRS, it is predicted that till 2020, the total length of wires will reach to 7000 $\frac{\text{m}}{\text{cm}^2}$ [15]. During sending and receiving flits using these wires, charging/discharging of coupling capacitances can inadvertently affect adjacent wires of the channel. This phenomenon referred to as crosstalk faults can seriously threat the reliability of flit traversal in NoCs. 1) Delay in rising/falling transitions 2) Speed up in rising/falling are the transitions appearing and 3) Unwanted voltage glitches including positive and negative glitches are the effects of crosstalk faults [4]. These effects can degrade the whole performance and reliability of NoC based systems. According to ITRS, wire delay is increasing in coming years. It is predicted that till 2015 the delay of global wires will reach to 1794×10^2 ps [16]. According to [17] the propagation delay of wire l of communication channel ($1 < l < n$) can be given by:

$$\mathrm{T} = \tau_0\left[(1+2\lambda)\Delta_\mathrm{l}^2 - \lambda\Delta_\mathrm{l}(\Delta_{\mathrm{l}-1} + \Delta_{\mathrm{l}+1})\right] \qquad \text{(Eq. 1)}$$

TABLE 1. Transition classes, patterns and delay of each class in 3- wire model

Transition Class	Patterns	Delay
0C	↑↑↑ ↓↓↓	π_0
1C	-↑↑ -↓↓ ↑↑- ↓↓-	$(1+\lambda)\pi_0$
2C	-↑- -↓-	$(1+2\lambda)\pi_0$
3C	-↑↓ ↑↓- ↓↑- -↑↓	$(1+3\lambda)\pi_0$
4C	↑↓↑ ↓↑↓	$(1+4\lambda)\pi_0$

Where in Eq. 1, τ_0 is the delay of ideal channel without crosstalk effect, λ is the ratio of coupling capacitance to bulk capacitance on the wire l and Δ_l equals to '1' for '0' → '1' (or ↑), and '-' for '1' → '0' (or ↓), transitions respectively. flittransition patterns are classified into five different classes according to the relative delay of a wire with respect to its adjacent wires. TABLE 1 shows these five classes of transitions, their pattern and the delay of each class. Class 4C and 3C of the transitions impose the worst delay on the victim wire. 4C class includes Triplet Opposite Direction (TOD) patterns including '↑↓↑' and '↓↑↓'. This class can cause the upper delay limit of $(1+4\lambda)\pi_0$ to the victim wire. 3C transition patterns have patterns with a pair of opposite direction transitions. Patterns of '-↑↓','-↓↑ ','↑↓- ' and, '↓↑-' the worst case delay that these transition patterns impose $(1+3\lambda)\pi_0$ to the victim wire. Those patterns that have '↓', or '↑' transition whereas they are not member of 4C or 3C categories are *2C transition patterns*. 2C transition patterns are considered as '-↑-' and '-↓-' transition patterns. They impose the worst case delay of $(1+2\lambda)\pi_0$ of this class on the victim wire. 1C transition patterns is a class with transition patterns '-↓↓','-↑↑ ','↓↓- ' and' ↑↑-' and produce the worst case delay of $(1+\lambda)\pi_0$. However the classification of transition patterns based on this model has two drawbacks [5]:

- **First**, the model in [5] has limited accuracy because of its dependency on only three wires: the model overestimates the delays of patterns in 1C through 4C while it underestimates the delays of patterns in 0C.
- **Second**, the actual delay ranges in some classes overlap with others.

This implies that the delays of existing CACs are not tightly controlled. These drawbacks motivate the writers of [5] to include more wires and to classify the transition patterns without overlapping delay ranges. The new classification proposed in [5] is based on 5 wires. First, all transition patterns are partitioned with respect to the delays on the middle wire. In other words, the patterns with close delays are grouped in one class. By grouping these patterns according to their evaluated delays, a finer classification of patterns without overlapping delays between adjacent classes this classification is shown in TABLE 2.

III. Related Work

Crosstalk fault tackling mechanisms can be applied at different levels of design abstraction. These mechanisms can be applied in physical level, transistor level and Register Transfer Level (RTL). Mechanisms at the lowest level of

TABLE 2. Transition classes, patterns and delay of each class in 5- wire model

Transition Class	Pattern
0C	↑↑↑↑↑
	-↑↑↑↑, ↑↑↑↑-
	↑-↑↑↑, ↑↑↑-↑
1C	-↑↑↑-, ↓↑↑↑↑, ↑↑↑↑↓
	- -↑↑↑, ↑↑↑- -, -↑↑-↑, ↑-↑↑-
	↑-↑-↑, ↑↑↑↓↑, ↑↓↑↑↑
2C	-↑↑↑↓, ↓↑↑↑-
	- -↑↑-, -↑↑- -, ↓-↑↑↑, ↓↑↑-↑, ↑-↑↑↓, ↑↑↑-↓
	↓↑↑↑↓
3C	- -↑↑↓, ↓↑↑- -, -↑↑-↓, ↓-↑↑-
	↓-↑↑↓, ↓↑↑-↓
	- -↑-↑, ↑-↑- -, -↑↑↓↑, ↑↑↑↓-, -↓↑↑↑, ↑↓↑↑-
4C	- -↑- -, ↑-↑-↓, ↓-↑-↑, -↑↑↓-, ↑↑↑↓↓, ↓↑↑↓↑, -↓↑↑-, ↑↓↑↑↓, ↓↓↑↑↑
	- -↑-↓, ↓-↑- -, -↑↑↓↓, ↓↑↑↓-, -↓↑↑↓, ↓↓↑↑-
	↓-↑-↓, ↓↑↑↓↓, ↓↓↑↑↓
5C	↓↓↑-↓, ↓-↑↓↓
	- -↑↓↓, ↓↓↑- -, -↓↑-↓, ↓-↑↓-
	- -↑↓-, -↓↑- -, ↑-↑↓↓, ↑↓↑-↓, ↓-↑↓↑, ↓↓↑-↑
	- -↑↓↑, ↑↓↑- -, -↓↑-↑, ↑-↑↓-
	↑-↑↓↑, ↑↓↑-↑
6C	↓↓↑↓↓
	↓↓↑↓-, -↓↑↓↓
	-↓↑↓-, ↑↓↑↓↓, ↓↓↑↓↑
	↑↓↑↓-, -↓↑↓↑
	↑↓↑↓↑

design abstraction mitigate/omit the rate of crosstalk faults by the use of crosstalk aware fabrication process. Changing geometrical dimensions of wires [8], increasing the spacing between wires [8], shielding wires [6] and repeater insertion [7] are among the mechanism at physical level of design abstraction. Passive shielding [18] and active shielding [19] are two different kinds of shielding mechanisms. In passive shielding, the shield wires, which are statically connected to power or ground, are inserted in either side of wires. In passive shielding, the signal wires in either side of wires are isolated from its neighboring signal wires reducing the effects of capacitive coupling. In repeater insertion mechanisms, the crosstalk fault is reduced by separating the wire into several segments and driving each segmented part by an inverting or non-inverting buffer. Without repeater insertion, the delay of wires increases quadratically with the wire length. The main drawbacks of physical level mechanism are their high area overheads. At a higher level of design abstraction, transistor level mechanism can tackle crosstalk faults by timing skewing the transitions [20]. In skewed transitions, simultaneous opposite switching between adjacent bus wires is avoided by introducing relative delay ΔT. This relative delay can be generated statistically or dynamically. In skewed transition, with static ΔT, as its name implies, the relative delay is always applied between the wires of the channel regardless of their switching patterns. The skewing delay is done by inserting different time shifts in repeater inserted channel. Producing ΔT can be done by inserting delay elements (e.g., inverter chain) at the beginning of alternate channel. In the dynamic approach,

the relative delay is applied only when adjacent channel wires are switching in the opposite direction. The timing concerns between a sender and a receiver is challenging, these mechanisms are not favorable by designers. At the next higher level, RTL, data coding is widely used to reduce the rate of crosstalk faults in on-chip channels. In coding mechanisms, the data word is manipulated before transmitting into the channel. The basic concept of the coding mechanism is to add redundancy to information in order to correct/or prevent crosstalk faults Error Detecting/Correcting Codes (EDC/ECCs) [21], Joint Crosstalk Avoidance and Error Correction Codes (CAC/SECs), Low Power Codes (LPC) [22] and Crosstalk Avoidance Codes (CACs) [23]-[25] are coding mechanisms used to tackle crosstalk faults. Duplicated-added-Parity (DAP) [26], Modified Dual Rail (MDR) [16] Boundary Shift Code (BSC) [27][28], and joint Crosstalk Avoidance and Multiple-Error-Correction Codes (CAC/MEC) are examples of CAC/SECs. The probability of crosstalk occurrence is reduced by duplicating each wire and adding a parity bit to the duplicated data. BSC is like DAP but, the only difference in BSC mechanism with respect to DAP is the location of the parity that is not fixed like the DAP, and bit places opposite side of the code word at each cycle. CAC/SECs have higher error detection and correction and can detect and/or correct the errors. However, as the crosstalk fault is not certain, these codes are not as efficient as other coding mechanisms. In CACs, crosstalk faults effects are prevented by omitting transition patterns of certain class. These codes can be categorize into Forbidden Overlap Condition (FOC) [13] codes, Forbidden Transition Free (FTF) [21] codes, Forbidden Pattern Free (FPF) [12][24][30] codes and One Lambda Code (OLC) [13]. FPFs reduce the crosstalk faults by omitting TODs. They restrict the worst case delay to 3C, 2C and 1C respectively. All of these coding mechanisms tackle the crosstalk faults by considering the 3-wire model and they do not consider the 5-wire model. The main challenge of CACs is to apply these codings with sufficient overheads. One way to reduce the overhead is to partition the channel and to use the separate coding in each partition separately. However channel partitioning faces with the forbidden transitions in the border between the partitions. The other mechanism is to use the numerical system. Numerical system is the notation for representing the codewords in a sufficient way. Fibonacci numerical system is among the systems that produces the TOD free code words. In Fibonacci numerical system, Fibonacci sequence is used as the base to produce FPF codes, e.g. for 6-bit space, the biases are 1 1 2 3 5 8. TABLE 3 shows 6-bit cookbooks generated by Fibonacci coding mechanism. The main drawback of Fibonacci coding mechanism is: 1. Complex encoder 2. The coding algorithm ambiguity; for example, this numerical system has two code word presentations for 12, which are '111001' and '111110'. This ambiguity is expiated by adding additional X_n wire in optimal version of coding in [12].

IV. Proposed Crosstalk Mitigation Mechanism

As discussed before, crosstalk faults can be tackled by preventing/omitting the specific transition patterns on the wires between PEs. In this section, considering the importance of crosstalk faults, the proposed packet management mechanism (PAM) is explained.

A. Packet Manipulating Mechanism

The main idea behind PAM is to reduce the crosstalk faults by reducing/omitting the numbers of transitions. As it was mentioned before, different transition classes impose delay to the victim wire. In this regards, PAM targets to reduce the crosstalk fault by reducing/omitting TODs. In PAM, flits with minimum numbers of transition patterns are selected according to the previous sent flit on the channel. This is done by manipulating the content of a packet using three different modes. These three modes are vertical, horizontal and diagonal manipulating mechanisms that are applied on the packet to select the proper mode between them. As using one of the vertical, horizontal and diagonal modes manipulate the content of flit, sum of these modes are called as a PAM mechanism in this content. Selecting the proper manipulation mechanism between these there modes is done by comparing between the numbers of transitions in configuration controller. After counting the number of transitions, PAM applies the mode with minimum numbers of transitions between tandem flits. To manipulate the content of the packet in PAM crosstalk mitigation mechanism, the content of packet is placed in the $N \times N$ matrix in the source, before injection the flits of packets into the network. Where the first N is the row and the second is the column of the matrix. If the content of the matrix cannot be filled by the main content of the packet to form a $N \times N$ matrix, ones or zeros are inserted in the content of the matrix in such a

TABLE 3. 6bit FPF code word generated by Fibo-CAC[26]

Data word	Fibo-CAC code word	Data word	Fibo-CAC code word
	8 5 3 2 1 1		8 5 3 2 1 1
0	0 0 0 0 0 0	13	1 1 0 0 0 0
1	0 0 0 0 0 1	14	1 1 0 0 0 1
2	0 0 0 0 1 1	15	1 1 0 0 1 1
3	0 0 0 1 1 0	16	1 1 1 0 0 0
4	0 0 0 1 1 1	17	1 1 1 0 0 1
5	0 0 1 1 0 0	18	1 1 1 1 0 0
6	0 0 1 1 1 0	19	1 1 1 1 1 0
7	0 0 1 1 1 1	20	1 1 1 1 1 1
8	0 1 1 0 0 0, 1 0 0 0 0 0	21	----------
9	0 1 1 0 0 1, 1 0 0 0 0 1	22	----------
10	1 0 0 0 1 1, 0 1 1 1 0 0	23	----------
11	0 1 1 1 1 0, 1 0 0 1 1 0	24	----------
12	0 1 1 1 1 1, 1 0 0 1 1 1	25	----------

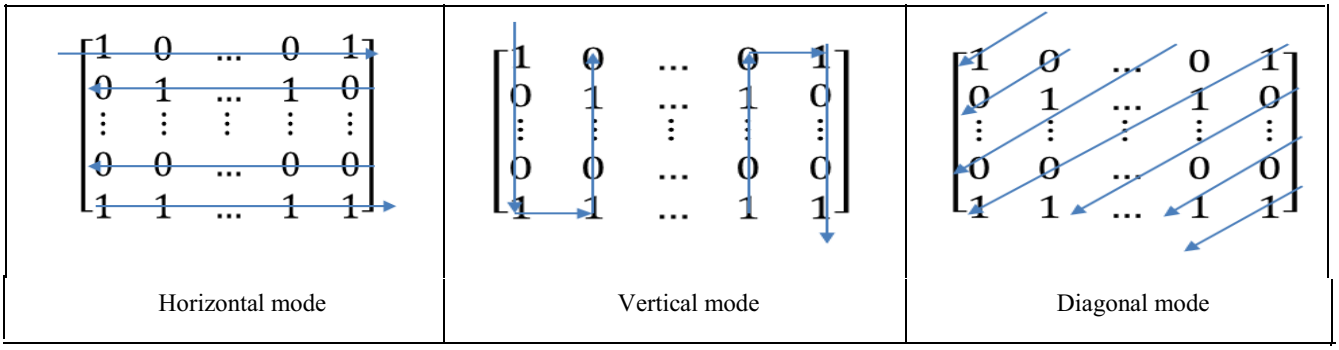

Fig. 2. PAM Modes in a packet

way that to reduce the transitions efficiently and prevention to be appear. This is especially efficient idea for manipulating diagonal mechanism. In order to fill the content of flits in the matrix, PAM mechanism is applied in the content of a packet as follows:

- ***Vertical mode:*** in ***vertical*** manipulation mode, the content of a packet is manipulated vertically and the content of flit is filled by reading each column of the matrix.
- ***Horizontal mode:*** in ***horizontal*** manipulation mode, the content of a packet is manipulated horizontally and the content of flit is filled by reading each row of the matrix
- ***Diagonal model***: in ***diagonal*** manipulation mode, the content of a packet is manipulated diagonally and the content of flit is filled by reading each row of the matrix diagonally from left to right.

Fig. 2 shows these three manipulating modes. Before injecting the flits to the network, the content of a packet is manipulated using these three modes. When each of these manipulating mechanisms are applied, the numbers of transitions between tandem flits in each of these manipulating mechanism are counted and the mechanism with minimum numbers of transitions is selected to be applied before injecting the flits to the network. One of the main advantages of the proposed mechanism is that PAM is the end-to-end mechanism and does not require to be applied in each switch. This is due to the pipeline nature of wormhole switching mechanism that causes wires to experience the same transition behavior during the flits traversal. In other words, PAM does not require manipulating the content of the packet in each switch during the flit traversal from a source to a destination and manipulating and ensuring decision is take part only in the source and destination. The architecture of the proposed PAM mechanism is shown in Fig. 3 that are controlled by Configuration Controller (CC). This leads to reducing the overheads of PAM with respect to other existing mechanisms in the literature and not manipulating the architecture of the switch. As header flit contains the information of routing, the content of header flit is not participated in PAM mechanism. Arriving header flit in the input virtual channel of a switch from either the processing element connected to the same switch or the previous PE, the packet's destination direction (north, south, east, west or the processing element) is decided, and an appropriate signal is sent to the crossbar and the crossbar switch is configured. In the next step, the flits incoming virtual channel is connected to the selected outgoing one. The header flit will be transferred to a next PE if there exists a free virtual channel in the selected outgoing physical channel. The other flits in the packet follow it in a pipelined fashion; otherwise the header flit must wait until a virtual channel of the selected outgoing physical channel becomes free. A tag field is added to the manipulated packets to make able the receiver PE to recover the content of the packet in the

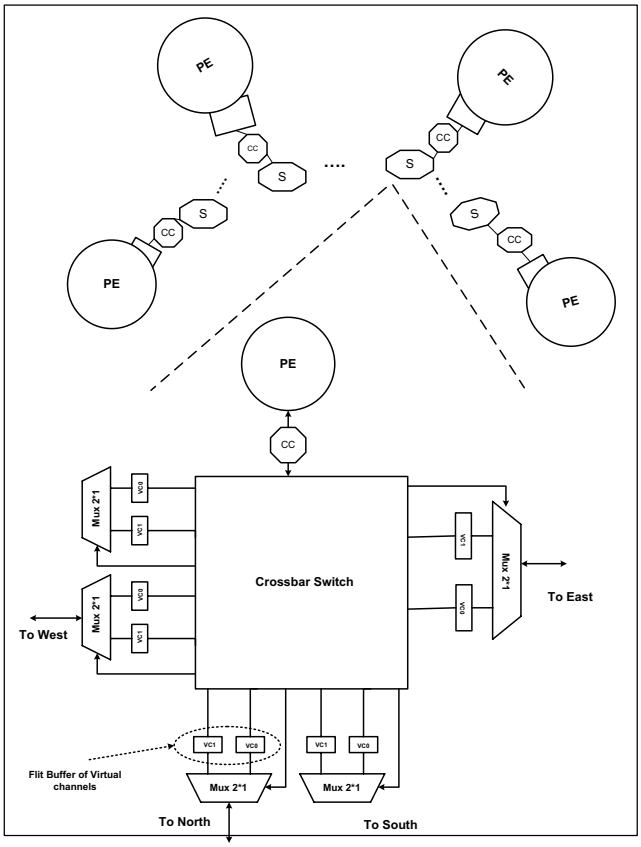

Fig. 3. Architecture of PAM in NoC

sender. This tag adds 2 bit redundancies to the packet. In other words, PAM has 2+flit width redundancy that can be applied in the two forms of policies: 1) Constant flits width with variant channel width and 2) Variant flits width give the designer good sense from the performance and the area overhead of the proposed method are the advantages of these two policies.

B. Configuration Controller

In order to select between triple manipulation modes in the PAM mechanism, the Configuration Controller (CC) is embedded in each switch of the network. However, in each pair of paths between a sender and a receiver, the decision between the modes takes place only in the sender. Then, in the receiver, using tag, the packet's mode is recognized. In order to minimize the power consumption overhead of PAM mechanism, the hardware of this mechanism are clock gated to be disabled by CC. The modules of vertical, horizontal and diagonal manipulation mechanisms in PAM are enabled/ disabled by a control signal from the CC. The block diagram of PAM is represented in Fig. 4 as it is shown in Fig. 4, the CC determines the current mode in sending the packet in the sender. CC contains a comparator for counting the numbers of transitions between a tandem flits in a packet. As decision about the current mode of packet manipulation is occurred according to the numbers of transitions, Current Flit to Send (CFS) is compared with respect to the content of Previously Sent Flit (PSF) and determines the mode with minimum numbers of TODs in each of the modes. The output of this module selects among triple modes. The modules of these triple mechanisms are embedded before injection the packets to the network and are enabled/disabled by a control signal from the CC. As in PAM the currernt and privious flits should be compared in order to select between modes, a buffer is embeded in each CC to hold the PSF in each cycle. Comparator for counting the numbers of transitions between tandem flits in a packet. As decision about the current mode of packet manipulation is occurred according to the numbers of transitions, CFS is compared with respect to the content of PSF and this comparator determines the mode with minimum numbers of transitions in each of the modes. The output of this module selects among triple modes. The modules of these triple mechanisms are embedded before injection the packets to the network and are enabled/disabled by a control signal from the CC.

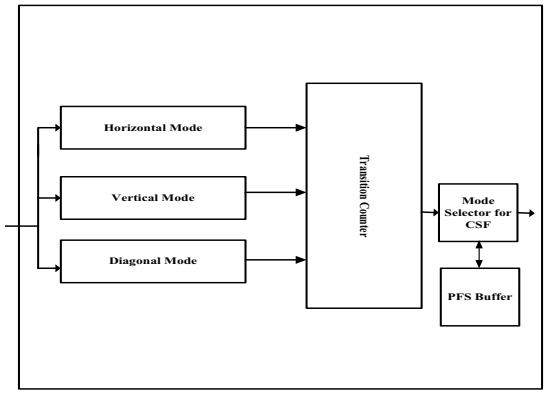

Fig. 4. Block diagram of the CC

V. Experimental Evaluations

In order to evaluate the proposed crosstalk mitigating mechanism, in this section PAM is evaluated using exhaustive simulations. In the first subsection, the effect of three modes on different video benchmarks is applied to determine the effectiveness of PAM on different classes of transitions. In the second subsection the imposed overheads of PAM on the architecture of NoC switches is evaluated.

A. The Efficiency of PAM on Real Traffic Bitstreams

To verify compliance with reality, H. 264 codec is used to produce a real multimedia application in each PE. To obtain realistic traffic patterns and also wire switching factors, JM 15. 1 [29] simulator is used. Standard video bit stream benchmarks including: Football and Mobile in the H.264 encoder is used to obtain realistic traffic patterns and wire switching factors. TABLE 4 shows parameters for these bit streams. Different applications with 128, 64 and 32 flit widths and the number of different crosstalk-induced transition patterns are counted to compare the effects of each PAM mode on the Football and Mobile. These results are shown in Fig. 5 and Fig 6 respectively. Based on these results, the PAM mechanism is

TABLE 4. Tested video bit streams parameters

Video benchmark	Format	Flit width
Football	CIF 328×228	128
Mobile	SIF 352×240	128

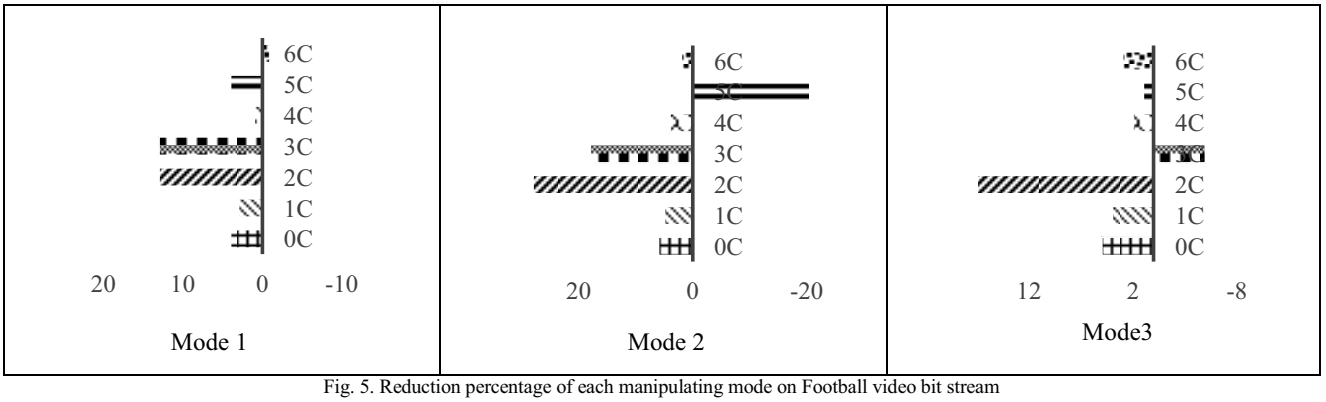

Fig. 5. Reduction percentage of each manipulating mode on Football video bit stream

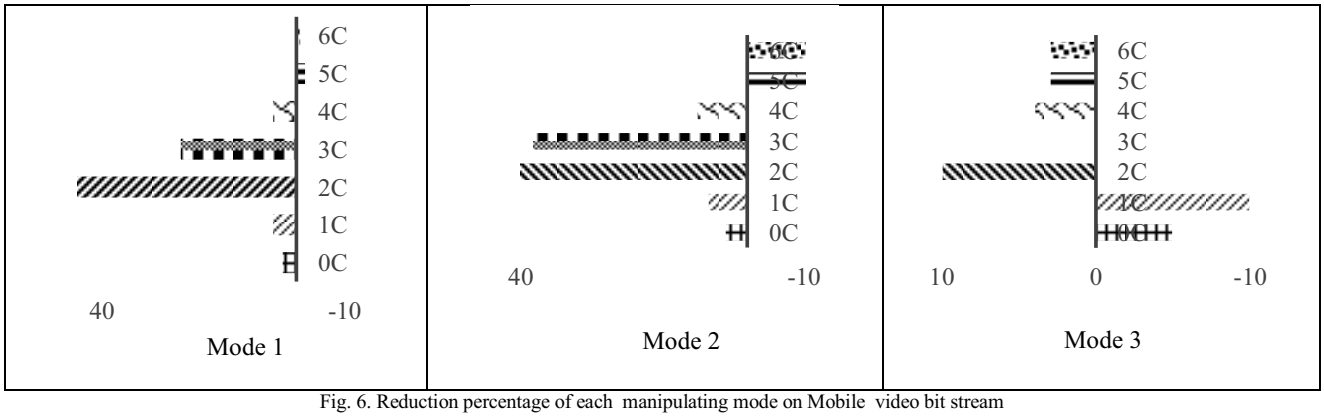

Fig. 6. Reduction percentage of each manipulating mode on Mobile video bit stream

more efficiently in 2C either in Football and Mobile bit streams. Results of evaluations on these two bit streams confirm that the best reduction of PAM is in the 2C with an average of 42% reduction in Football and Foreman bit streams. As PAM has the best improvement on the 2C based on these results, 2C is selected as a metric to check the content of tandem flits and selecting the proper modes that produces the minimum numbers of 2C. In other words, the numbers of 2C in tandem flits of the packet, using each mode of the PAM mechanism the flit with minimum numbers of 2C is selected to be injected into the network in the source. As it was mentioned in the previous section a three bit tag is added to the packets to make able the receiver to inform about the mode used in the sender.

B. The Overheads of PAM Controller

In order to evaluate the imposed overhead of PAM controller to the switches of NoCs, the controller of PAM is implemented using VHDL-based simulations. In these simulations, packets with 128, 64 and 32 flits widths are considered in the simulations and the data of packet is arranged in matrixes to apply each mode in PAM mechanism. In order to estimate PAM controller overheads in the terms of critical path and power consumptions, the controller is synthesized using Design Compiler tool in TSMC 45 nm technology. In order to have fair comparisons with the other state-of-art crosstalk mitigation mechanism, the encoder and decoder of Fibo-CAC is implemented and compared with PAM mechanism. Fig. 7 shows the critical path of CC and also the critical path of Fibo-CAC codec in 45 nm technology. These results confirm that PAM outperforms Fibo-CAC. The results of power consumption of the CC and codec of Fibo-CAC are shown in Fig. 8 in 45 technology. These results confirm that we have an average of 32% improvement in amount of power consumption with respect to Fibo-CAC in different technologies. Also, the results of PDP is reported in Fig. 9 and states that PAM has better results with respect to Fibo-CAC. According to these results with incresing the flit width, PAM shows its efficency

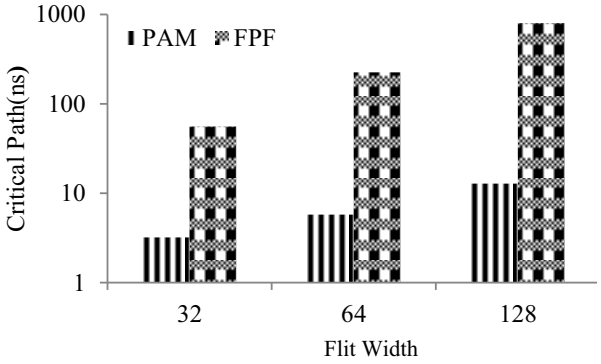

Fig. 7. Critical Path of CC with respect to Fibo-CAC codec in 45nm

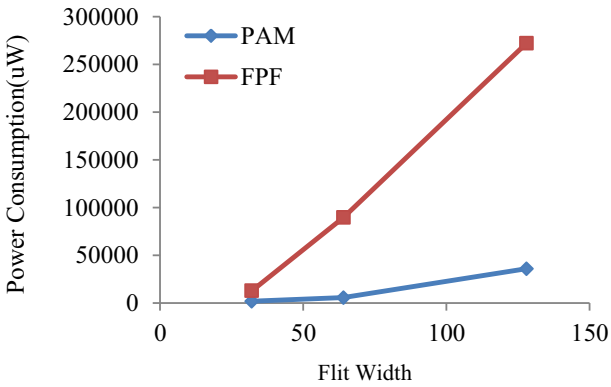

Fig. 8. Power Consumption of CC with respect to Fibo-CAC codec in 45nm

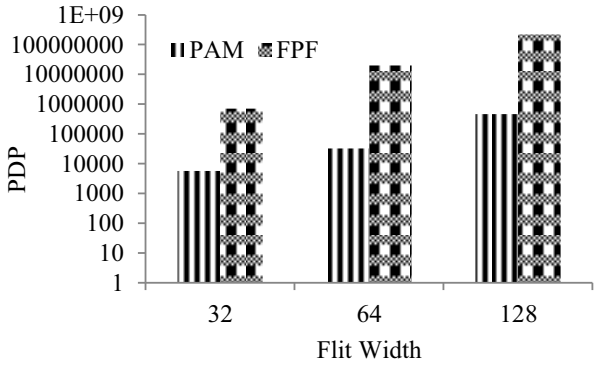

Fig. 9. PDP of CC with respect to Fibo-CAC codec in 45nm

VI. CONCLUSION

A Packet Manipulating mechanism called PAM for reliable data transfer of NoCs is proposed in this paper. PAM investigates the transitions of a packet to minimize the forbidden transition patterns appearing during the flit traversal in NoCs. To do this, the content of a packet is manipulated using three different manipulating mechanisms. In other words, PAM manipulates the content of packet in three manipulating mode including: vertical, horizontal and diagonal modes. Then, comparing the transitions of these manipulating mechanisms, a packet with minimum numbers of transitions is selected to be injected into the channel. A tag field is added to the manipulated packet in the sender to be recovered in the receiver. The evaluation of PAM confirms that the proposed mechanism has an average of 32% improvement in amount of power consumption with respect to state-of-the art Fibo-CAC coding mechanism.

VII. REFERENCES

[1] L. Benini, and G. De Micheli, "Networks on chips: a new SoC paradigm," Computer, vol. 35, no.1, pp. 70-78, January 2002.

[2] W. J. Dally and B. Towles, "Principles and practices of interconnection networks," San Mateo, CA: Morgan Kaufmann, 2004.

[3] M. Radetzki, C. Feng, X. Zhao and A. Jantsch, "Mechanisms for fault tolerance in network on chip," ACM Computing Survey, vol. 44, pp. 1-36, January 2013.

[4] A. P. Frantz, F. L. Kastensmidt, L. Carro, and E. Cota, "Dependable network-on-chip router able to simultaneously tolerate soft errors and crosstalk", in Proceedings of the International Test Conference (ITC), pp.1-9, October 2006

[5] F. Shi, X. Wu and Z. Yan, "New crosstalk avoidance codes based on a novel pattern classification," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 10, pp. 1892-1902, October 2012.

[6] J. Zhang and E. G. Friedman, "Effect of shield insertion on reducing crosstalk noise between coupled interconnects," Proceedings of the International Symposium on Circuits and Systems (ISCAS), pp. 23-26, May 2004.

[7] C. J. Akl and M. A.. Bayoumi, "Reducing Interconnect Delay Uncertainty via Hybrid Polarity Repeater Insertion," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, no. 9, pp. 1230-1239, September 2008.

[8] K. Agarwal, D. Sylvester and D. Blaauw, "Modeling and analysis of crosstalk noise in coupled RLC interconnects," IEEE Transactions on Computer-Aided Design of Integrated Circuits and System, vol. 25, no. 5, pp.892-901, May 2005.

[9] K. Nose, and T. Sakurai, "Two schemes to reduce interconnect delay in bi-directional and uni-directional buses," in Proceedings of Symposium on VLSI Circuits Digest (VLSIC'01), pp. 193-194, 2001.

[10] S. R. Sridhara, and N. R. Shanbhag, "Coding for reliable on-chip buses: a class of fundamental bounds and practical codes," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 5, pp. 977 – 982, May 2007.

[11] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in Proceedings of IEEE/ACM International Conference on Computer Aided Design," (ICCAD'01), pp . 57-63, 2001.

[12] C. Duan, V. H. C. Calle and S.P. Khatri, "Efficient on-chip crosstalk avoidance codec design," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 4, pp. 551-560, April 2009.

[13] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag, "Area and energy-efficient crosstalk avoidance codes for on-chip buses," in Proceedings of IEEE International Conference on Computer Design (ICCD'04), pp. 12-17, October 2004

[14] W. J. Dally and B. P. Towles, "Principles and practices of interconnection networks," Morgan Kaufmann, 2004.

[15] International Technology Roadmap for Semiconductors (ITRS), 2005 edition, Technical Report, 2005, [Onlline]. Available: http://public.itrs.net

[16] International Technology Roadmap for Semiconductors (ITRS), 2007 edition, Technical Report, 2007, [Onlline].Available: http://public.itrs.net.

[17] P. P. Sotiriadis and A. Chandrakasan, "Reducing bus delay in submicron technology using coding," in Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC'01), pp. 109-114, 2001.

[18] X. Huang, Y. Cao, D. Sylvester, S. Lin, T. J. King, and C. Hu, "RLC signal integrity analysis of high-speed global interconnects," in Proceedings of International Electron Devices Meeting (IEDM'00), pp. 731-734, 2000.

[19] H. Kaul, D. Sylvester, and D. Blaauw, "Performance optimization of critical nets through active shielding," IEEE Transactions on Circuits Systems I, vol. 51, no. 12, pp. 2417-2435, 2004.

[20] K. Hirose and H. Yasuura, "A bus delay reduction technique considering crosstalk," in Proceedings of the Conference on Design, Automation and Test in Europe (DATE'09), pp. 441-445, 2000.

[21] A. Ganguly, P. P. Pande, and B. Belzer, "Crosstalk-aware channel coding schemes for energy efficient and reliable noc interconnects", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no.11, pp. 1626–1639, Nov 2009.

[22] Y. Shin, S. I. K. Chae and K. Choi, "Partial bus-invert coding for power optimization of system level bus," in Proceedings of International Symposium on Low power Electronics and Design, (ISLPED'98), pp. 127-129, 1998.

[23] C. Duan, A. Tirumala and S. Khatri, "Analysis and avoidance of cross-talk in on-chip buses," Hot Interconnects, pp. 133-138, August 2000.

[24] Z. Shirmohammadi and S. G. Miremadi, "Crosstalk avoidance coding for reliable data transmission of network on chips," in Proceedings of the International Symposium on System-on-Chip 2013 (SoC'13), pp. 1-4, Tampere, Finland, October 2013

[25] C. Duan, V. H. C Calle and S.P. Khatri, "Efficient on-chip crosstalk avoidance codec design," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 4, pp. 551-560, April 2009.

[26] S. R. Sridhara and N. R. Shanbhag, "Coding for system-on-chip networks: a unified framework," IEEE Transactions on Very Large Scale Integration (VLSI'05) Systems, vol. 13, no. 6, pp. 655-667, 2005.

[27] D. Rossi, C. Metra, A. K. Nieuwland and A. Katoch, "New ECC for crosstalk impact minimization," IEEE Design & Test of Computers, vol. 22, no. 4, pp. 340-348, 2005.

[28] K. N. Patel and I. L. Markov, "Error-correction and crosstalk avoidance in DSM busses," in Proceedings of International Workshop on System-level Interconnect Prediction (SLIP'03), pp. 9-14, 2003.

[29] H264/AVC JM Refrence [Online] http://iphome.hhi.de/suehing/tml

[30] Z. Shirmohammadi and S. G. Miremadi, "S2AP: An efficient numerical-based crosstalk avoidance code for reliable data transfer of NoCs," in Proceedings of the the IEEE International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC '15) June -July 2015.