UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ARTHUR PEREIRA FRANTZ

Designing Fault Tolerant NoCs to Improve Reliability on SoCs

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dra. Fernanda Gusmão de Lima Kastensmidt
Advisor

Porto Alegre, February 2007

CIP – CATALOGING-IN-PUBLICATION

Frantz, Arthur Pereira

Designing Fault Tolerant NoCs to Improve Reliability on SoCs / Arthur Pereira Frantz. – Porto Alegre: PPGC da UFRGS, 2007.

80 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2007. Advisor: Fernanda Gusmão de Lima Kastensmidt.

1. Networks-on-chip. 2. Fault tolerance. 3. Soft errors. 4. Crosstalk. I. Kastensmidt, Fernanda Gusmão de Lima. II. Título.

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

BI	Bus Inverted Code
DVS	Dynamic Voltage Scaling
ECC	Error Corretion Code
EDC	Error Detection Code
FSM	Finite State Machine
IP	Intellectual Property
NoC	Network-on-Chip
SEU	Single-Event Upset
SoC	System-on-Chip
TMR	Triple Modular Redundancy
TR	Timing Redundancy

LIST OF FIGURES

ABSTRACT

As the technology scales down into deep sub-micron domain, more IP cores are integrated in the same die and new communication architectures are used to meet performance and power constraints. Networks-on-Chip have been proposed as an alternative communication platform capable of providing interconnections and communication among on-chip cores, handling performance, energy consumption and reusability issues for large integrated systems.

However, the same advances to nanometric technologies have significantly reduced reliability in mass-produced integrated circuits, increasing the sensitivity of devices and interconnects to new types of failures. Variations at the fabrication process or even the susceptibility of a design under a hostile environment might generate errors. In NoC communications the two major sources of errors are crosstalk faults and soft errors. In the past, it was assumed that connections cannot be affected by soft errors because there was no sequential circuit involved. However, when NoCs are used, buffers and sequential circuits are present in the routers, consequently, soft errors can occur between the communication source and destination provoking errors. Fault tolerant techniques that once have been applied in integrated circuits in general can be used to protect routers against bit-flips.

In this scenario, this work starts evaluating the effects of soft errors and crosstalk faults in a NoC architecture by performing fault injection simulations, where it has been accurate analyzed the impact of such faults over the switch service. The results show that the effect of those faults in the SoC communication can be disastrous, leading to loss of packets and system crash or unavailability. Then it proposes and evaluates a set of fault tolerant techniques applied at routers able to mitigate soft errors and crosstalk faults at the hardware level. Such proposed techniques were based on error correcting codes and hardware redundancy. Experimental results show that using the proposed techniques one can obtain zero errors with up to 50% of savings in the area overhead when compared to simple duplication. However some of these techniques are very power consuming because all the tolerance is based on adding redundant hardware. Considering that software-based mitigation techniques also impose a considerable communication overhead due to retransmission, we then propose the use of mixed hardware-software techniques, that can develop a suitable protection scheme driven by the analysis of the environment that the system will operate in (soft error rate), the design and fabrication factors (delay variations in interconnects, crosstalk enabling points), the probability of a fault generating an error in the router, the communication load and the allowed power or energy budget.

Keywords: Networks-on-chip, fault tolerance, soft errors, crosstalk.

Projeto de NoCs Tolerantes a Falhas para o Aumento da Confiabilidade em SoCs

RESUMO

Com a redução das dimensões dos dispositivos nas tecnologias sub-micrônicas foi possível um grande aumento no número de *IP core*s integrados em um mesmo chip e consequentemente novas arquiteturas de comunicação são usadas bucando atingir os requisitos de desempenho e potência. As redes intra-chip (*Networks-on-Chip*) foram propostas como uma plataforma alternativa de comunicação capaz de prover interconexões e comunicação entre os *cores* de um mesmo chip, tratando questões como desempenho, consumo de energia e reusabilidade para grandes sistemas integrados.

Por outro lado, a mesma evolução tecnológica dos processos nanométricos reduziu drasticamente a confiabilidade de circuitos integrados, tornando dispositivos e interconexões mais sensíveis a novos tipos de falhas. Erros podem ser gerados por variações no processo de fabricação ou mesmo pela susceptibilidade do projeto, quando este opera em um ambiente hostil. Na comunicação de NoCs as duas principais fontes de erros são falhas de *crosstalk* e *soft errors*. No passado, se assumia que interconexões não poderiam ser afetadas por *soft errors*, por não possuirem circuitos seqüenciais. Porém, quando NoCs são usadas, buffers e circuitos seqüenciais estão presentes nos roteadores e, consequentemente, podem ocorrer *soft errors* entre a fonte e o destino da comunicação, provocando erros. Técnicas de tolerância a falhas, que tem sido aplicadas em circuitos em geral, podem ser usadas para proteger roteadores contra bit-flips.

Neste cenário, este trabalho inicia com a avaliação dos efeitos de *soft errors* e falhas de *crosstalk* em uma arquitetura de NoC, através de simulação de injeção de falhas, analisando detalhadamente o impacto de tais falhas no roteador. Os resultados mostram que os efeitos dessas falhas na comunicação do SoC podem ser desastrosos, levando a perda de pacotes e travamento ou indisponibilidade do sistema. Então é proposta e avaliada a aplicação de um conjunto de técnicas de tolerância a falhas em roteadores, possibilitando diminuir os *soft errors* e falhas de crosstalk no nível de hardware. Estas técnicas propostas foram baseadas em códigos de correção de erros e redundância de hardware. Resultados experimentais mostram que estas técnicas podem obter zero erros com 50% a menos de *overhead* de área, quando comparadas com a duplicação simples. Entretanto, algumas dessas técnicas têm um grande consumo de potência, pois toda essas técnicas são baseadas na adição de hardware redundante. Considerando que as técnicas de proteção baseadas em software também impõe um considerável *overhead* na comunicação devido à retransmissão, é proposto o uso de técnicas mistas de hardware e software, que podem oferecer um nível de proteção satisfatório, baseado na análise do ambiente onde o sistema irá operar (*soft error rate*), fatores relativos ao projeto e fabricação (variações de atraso em interconexões, pontos susceptíveis a *crosstalk*), a probabilidade de uma falha gerar um erro em um roteador, a carga de comunicação e os limites de potência e energia suportados.

Palavras-chave: Networks-on-Chip, tolerância a falhas, soft errors, crosstalk.

1 INTRODUCTION

The integration of a complete system onto the same silicon die has become feasible as a consequence of the increasing integration densities made available by deep submicron lithography technologies and the computational requirements of the most aggressive computing applications in the multimidia, automotive and ambient intelligence domains (BERTOZZI; BENINI, 2004).

Such systems, so called Systems-on-Chip (SoCs), represent high-complexity semiconductor products that incorporate building blocks from multiple sources, in particular, general purpose processors, DSPs, dedicated hardware accelerators, memory blocks, I/O blocks, etc. Consequently, the performance of SoCs with hundreds of IP blocks is limited by the ability to efficiently interconnect different functional blocks and to accomodate their communication requirements (BERTOZZI; BENINI, 2004).

In order to overcome such new challenges, a new communication infrastructure, named Network-on-Chip (NoC), has been proposed. This communication approach takes advantage of well known communication architectures, offering scalability and reusability of multipoint connections and parallelism and small link lengths of point-to-point connections (BJERREGAARD; MAHADEVAN, 2006). An example of a NoC topology with its architectural components is shown in Figure 1.1.

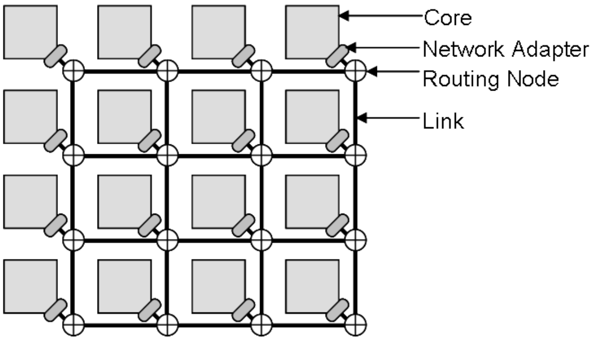


Figure 1.1: Illustration of a 4-by-4 NoC grid topology, indicating the architectural components

In the other hand, the same technological evolutions to nanometric processes have drastically reduced reliability in mass-produced integrated circuits, making devices and interconnects more sensitive to new types of malfunctions and failures. Errors might be generated at the fabrication process due to process variations, or even from the susceptibility of the design under a hostile environment.

The increasing speed and shrinking geometries of new integrated circuits has focused the attention toward nonconventional fault models, such as those due to inductive and capacitive coupling between metal wires inside the circuits. Such failures are commonly referred to as crosstalk faults and may result in incorrect signals inside the circuit when one or more coupled signals switch.

Another kind of error is the soft errors, that are transient faults provoked by the interaction of energetic particles with the silicon substrate, which produces an ionizing path that can charge or discharge the hit node generating a transient current pulse. The major effect is a bit-flip in a memory cell element also known as Single Event Upset (SEU). Figure 1.2 exemplifies the two phenomena: crosstalk due to coupling effects and a substrate ionization due to a energetic particle hit.

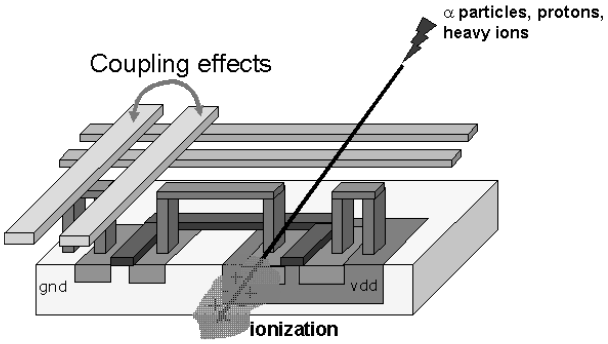


Figure 1.2: Soft error and crosstalk effects in integrated circuits

In the literature, several works have addressed the problem of ensuring reliability in ICs produced with new deep sub-micron technologies. Considering the intermitent faults such as crosstalk, a great variety of solutions has been proposed. Many of such solutions were designed to be used in on-chip buses, but also might be applied in NoC architectures.

The authors in (FAVALLI; METRA, 1999) evaluate four error correction codes (ECCs) (parity, dual-rail, m-out-of-n and Berger codes) for protection of on-chip buses against crosstalk. Dual-rail code presented better detection capabilities than the other codes, but such code efficiency is sensitive to the circuit layout (coding lines positions). In (ROSSI et al., 2005), the use of ECCs was also evaluated in combination with layout-level shieldings, in order to avoid crosstalk faults. Again, dual-rail codes resulted in better gains in terms of reduction of the delays induced by crosstalk. In (ROSSI et al., 2005), the authors propose a modification on the dual-rail code (adding 1 extra check bit) in order to improve the correction efficiency. Such method presented better power savings and delay reductions when compared to Hamming and the original dual-rail codes, with and without layout optimizations.

In (BERTOZZI; BENINI; MICHELI, 2002) and (BERTOZZI; BENINI; MICHELI, 2005), it is investigated the reliability impact of the use of DVS (Dynamic Voltage Scaling) in on-chip communication links in combination with the use of error detection/correction codes. In fact, they evaluate the ECC/EDCs efficiency when reducing the bus voltage operation, exploiting the trade-offs between reliability and power/energy consumption. In (NIEUWLAND et al., 2005), it is proposed a merged implementation of a Bus Inverted (BI) code to reduce the bus switching and error detection/corretion code in a single circuitry, achieving an increase in the signal integrity in on-chip buses. Bus inverted coding is based on inverting all the signals on the bus if more than half of the signals on the bus would switch. At the receiving end, the signals are re-inverted when indicated by the bus invert signal.

The authors in (LAJOLO, 2001) propose the use of distributed dedicated combinatorial hardware modules along the SoC buses in order to monitor the integrity of the trasmitted information and correct such information in case of error. The distributed *bus guardian* scheme is shown in Figure 1.3. It is based on online testing and diagnosis, followed by recovery.

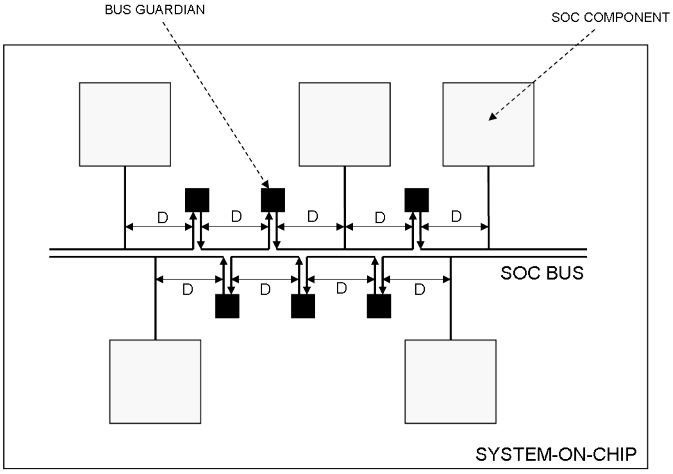


Figure 1.3: Distributed bus guardian scheme overview

A more detailed view of the bus guardian detection/recovery mechanism is depicted in Figure 1.4. Each *bus guardian* module acts constantly as controllers and regenerators of the correct information sent on the bus. Each module contains a decoding logic to separate data bits from check bits. A combinatorial error detection and correction logic is then exploited to check the correctness of the information. If possible, depending on the adopted bus enconding scheme, the correct information might be restored and sent again on the bus.

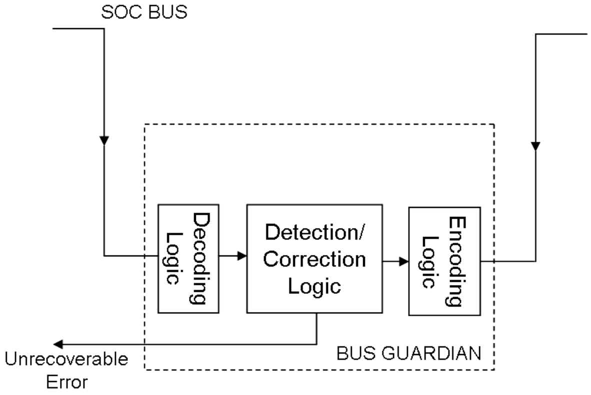


Figure 1.4: Bus guardian recovery mechanism

Moving to the NoC domain, the authors in (GRECU et al., 2006) propose the use of a code-disjoint approach to implement NoC switches, in order to detect faults at links or in the router. Code-disjoint ciruits are a special class of self-checking circuits, where both data and parity bits are computed. In case of mismatch of the input and output parity bits, an error has ocurred in the computation. The authors also estimate the energy, latency and throughput of end-to-end, switch-to-switch and code-disjoint approaches, where the latter presented better results.

In (MURALI et al., 2005), it is evaluated the energy, error protection and performance efficiency of end-to-end, switch-to-switch and hybrid recovery approaches for NoCs. In terms of packet latency, the hybrid approach presented better results, followed by the switch-to-switch approach. When considered the power consumption, the end-to-end approach presented better results only for very small error rates, being followed by the hybrid and switch-to-switch approaches.

Differently, (DUMITRAS; KERNER; MARCULESCU, 2003) and (MARCULESCU, 2003) propose a stochastic communication method where it is used a probabilistic broadcast algorithm in order to guarantee the messages delivery independently of the occurrence of faults.

In (TAMHANKAR; MURALI; MICHELI, 2005) it is proposed a timing error tolerant communication system called *Terror*. In order to tackle the delay faults in NoC links, the links can be pipelined by adding buffers. Basically, pipeline buffers divide the links into stages (see Figure 1.5), and the links might be seen as shift registers.

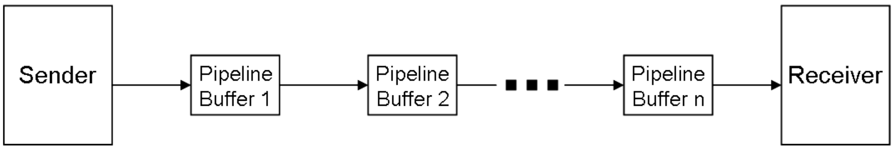


Figure 1.5: Terror pipelined link design with *n* stages

In the *Terror* system, each pipeline buffer (*main flip-flop*) is augmented with a second flip-flop (*delayed flip-flop*) as shown in Figure 1.6. Each pipeline buffer might operate in two different modes: *normal mode* or *delayed mode*. Initially, all the buffers are set to the normal mode and data tramission begins. In every cycle, at the regular clock edge *clk* (see Figure 1.6), the main flip-flop captures and transmits the incoming data. At the delayed clock edge *clkd*, the delayed flip-flop captures the incoming data and the error detection control circuit checks whether there is any diference between the main and the delayed flip-flops. If there is a difference, there is an error in the main flip-flop value. Then the *Terror* buffer enters the delayed mode, and the correct data from the delayed flip-flop is re-sent to the next pipeline buffer.

Regarding faults induced by radiation, such as soft errors, a number of solutions have been also reported in the literature. Such solutions range from only hardware solutions to combined hardware and software-based approaches. In (NICOLAIDIS, 2005), the author reports a great variety of hardware protection techniques, including well-known ones such as Triple Modular Redundancy (TMR). The TMR approach triplicates the sensitive parts of circuit (sequential logic for soft errors) and uses majority voters at the output of the triplicated parts in order to vote out the correct values. An overview of the TMR approach is shown in Figure 1.7.

Other soft error mitigation techniques are based on timing redundancy (TR), self-checking circuits, error detection/correction codes (EDC/ECC) for memories and even transistor level hardening approaches.

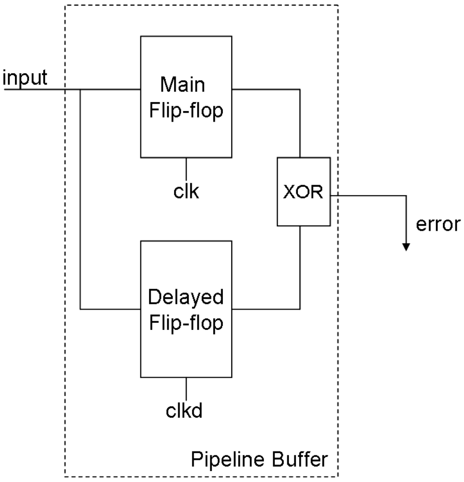


Figure 1.6: Terror pipeline buffer design

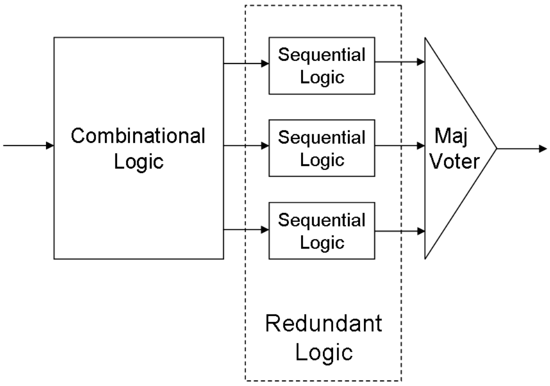


Figure 1.7: TMR logic overview

The techniques previously presented cope with crosstalk and soft errors independently, based on the assumption that connections cannot be affected by soft errors (no sequential circuit involved), whereas sequential circuits cannot be affected by crosstalk faults. However, when NoCs are used, buffers and sequential circuits are present in the routers, while the number of connections between any two routers can be very large.

Thus, although one can find in the literature a number of solutions to independently mitigate soft errors in sequential blocks and crosstalk faults in interconnects, as previously mentioned, new fault-tolerant approaches are required to cope with the simultaneously occurrence of those two faults.

In a recent work (PARK et al., 2006), the authors proposed a set of architectural and algorithmic techniques to tackle link and intra-router errors (crosstalk and soft errors, respectively). The proposed techniques are based on a hop-by-hop retransmission scheme to cope with link errors and a new virtual-channel implementation combined with other techniques to deal with deadlocks and intra-router errors. The potential problems associated with the above mentioned protection schemes are that if the fault has a non-transient nature (permanent or intermittent as crosstalk), the retransmission approach might not work. The retransmitted data cannot be guaranteed to be correct; on the contrary, it is very likely that it will be affected by the same fault. Another important observation is that deadlock protection technique was proposed only for virtual-channel-based NoC switches.

In order to develop efficient ways to avoid the simultaneously ocurrence of soft errors and crosstalk faults, this work was split in three main stages. First, a very detailed evaluation of the effects of soft errors and crosstalk faults over each NoC router part has been performed. Then, based on the effects evaluation, new hardware approaches that simultaneously tackle soft errors and crosstalk faults in NoC routers have been developed and evaluated. After the hardware techniques evaluation, it was observed that only-hardware approaches are very power consuming, and then mixed hardware-software approaches have been also designed and evaluated.

To perfom all these experiments the router architecture used is the RASoC router (ZEFERINO; KREUTZ; SUSIN, 2004). In order to represent crosstalk faults and soft errors in the NoC behaviour, modifications in the RASoC VHDL design have been done, as well as a VHDL fault injection environment has been developed. EDA tools such as Mentor Graphics Modelsim simulation tool and Mentor Graphics Leonardo Spectrum synthesis tool have been used.

As it can be seen, this work has been developed in a task-oriented way. After the conclusion of each stage, its results have been submitted for publication. It resulted in a set of publications ordered by the amount of results achieved. Taking it into consideration, this work was organized as a sequence of papers with comments about the development, challenges, results and taken choices.

In the second chapter are presented the experiments and results of fault injection simulations over a NoC router architeture, as well as, the router architecture description, fault models and sensitive parts.

Chapter 3 presents hardware approaches to cope with the simultaneous action of crosstalk and soft errors in a NoC router architecture. It is also presented the evaluation of the proposed approaches in terms of area and frequency overhead, and protection efficiency.

In chapter 4 are proposed mixed hardware-software solutions to also protect a NoC router against soft errors and crosstalk. Area, frequency and protection efficiency are also evaluated, as well as, energy consumption for the proposed approaches.

Finally, in chapter 5 are presented some conclusions and future works.

2 SOFT ERRORS AND CROSSTALK EFFECTS IN A NOC SWITCH

As previously presented, drastic device shrinking, high logic complexity, power supply reduction, and high operating speeds that accompany the technological evolution to nanometric technologies have reduced dramatically the reliability of deep sub-micron ICs. Thus, significant problems are related to soft errors induced by radiation and crosstalk faults induced by capacitive coupling. In order to represent these upsets in a NoC behaviour, a fault injection system was developed.

The fault injection experiments were performed over the RASoC router. RASoC router is the basic building block of the SoCIN network. RASoC is a VHDL soft-core, parameterized in three dimensions: communication channels width, input buffers depth and routing information width. More details about RASoC implementation are present in (ZEFERINO; KREUTZ; SUSIN, 2004).

Aiming at analyzing the effects of soft errors and crosstalk in the RASoC behaviour, the experiments have been divided in three diferent steps. First, only soft errors were injected in the sensitive parts of the NoC architecture (sequential logic elements). Such experiments and results have been presented in (FRANTZ; KASTENSMIDT, 2006) and proved that transient faults such as soft errors might lead to permanent effects, such as the crash of the router. Then, crosstalk faults have been injected in the local link of the router and, again, results proved that unreliable links might compromise drastically the communication in a NoC router. These results have been published in (FRANTZ et al., 2006). In order to evaluate the combination of these effects, new fault injection experiments were performed. At this time, the RASoC router has been exposed to the simultaneous action of soft errors and crosstalk. As expected, the effects of such combined faults were worse than only single faults. Such results were presented in (FRANTZ et al., 2006a). Attempting to clear the undestanding of these experiments and results, the refered publications are presented in the next sections.

Such results provided enough knowledge to develop suitable protection mechanisms in order to avoid faults in the NoC router. Such approaches are presented in the next chapters.

2.1 SEU Effects Evaluation on a NoC Router Architecture

Category: Full Paper

Conference: LATW2006: Proceedings of the 7th IEEE Latin American Test Workshop

Location: Buenos Aires - Argentina

Date: March, 2006

SEU Effects Evaluation on a NoC Router Architecture

Arthur Pereira Frantz
Universidade Federal do Rio Grande do Sul
afprantz@inf.ufrgs.br

Fernanda Lima Kastensmidt
Universidade Federal do Rio Grande do Sul
fglima@inf.ufrgs.br

Abstract

Future generations of systems-on-chip (SoCs) will consist of hundreds of pre-designed IPs assembled together. To solve the communication problems among these IP blocks, Networks-On-Chip (NoCs) has been proposed as the future communication architecture. Furthermore, as the complexity of designs increases and the technology scales down into the deep sub-micron domain, devices and interconnect are subject to new types of malfunctions and failures. This work intends to evaluate the effect of a Single Event Upset in a NOC router architecture by developing a fault injection mechanism, allowing an accurate analysis of the impact of SEUs over the router service. The results show that the SEUs may affect the proper router work, causing lost of packets, packet information errors or even compromising the router service, provoking permanent routing problems. These results can guide the future studies on SEU mitigation techniques for Networks-on-Chip.

1. Introduction

The development of deep sub-micron lithography processes has allowed integration of a huge amount of transistors in a single die. Consequently, complex system designs can be integrated into a single chip. Such kind of systems, named Systems-on-Chip (SoCs), are based on the reuse of pre-designed and pre-verified blocks, called cores or intellectual property (IP) blocks [1]. For communication among these IP blocks a dedicated architecture is a need. Nowadays, a shared bus is the most common alternative. However, communication performance and energy consumption increases considerably with the number of cores connected to it. To meet energy and performance requirements of future integrated systems, a new alternative of communication architecture has been proposed. Such architecture, named Network-on-Chip (NoC), is based on the same concepts adopted on the building of interconnection networks for parallel computers [1].

In the other hand, with the shrinking of the transistors dimensions and the reduction of the power supply voltage, the sensibility of the circuit to internal and external noises has increased, which can affect the correct work of integrated circuits. Such errors might come from the fabrication process (process variability), frequency increasing (crosstalk between metal wires) and from the environment that the circuit is working into (radiation effects). In the last case, a radiation effect in the space environment might be a Single Event Upset (SEU). SEUs can provoke bit-flips in sequential logic elements, such as registers or memories. It is a transient effect but the design must be aware of this transient effect and it may be protected against SEU in order to ensure a correct operation in the presence of a fault. NOCs are complex architectures composed of many registers, buffers, finite state machines (FSMs) and logic that can be susceptible to SEU.

In this context, this work intends to evaluate the effect of a SEU in a NOC architecture by developing a fault injection mechanism, allowing an accurate analysis of the impact of SEUs over the router work. The results show that the SEUs may affect the proper router work, causing lost of packets, packet information errors or even compromising the router service, provoking permanent routing problems. Such evaluation can provide resources to guide a suitable harware protection against SEUs.

This paper is organized as follows. In the next section is presented some related work. In the third section is shown an overview of the NoC router architecture used for evaluation of SEU effects. Afterwards are presented the fault injection system and the evaluation approach used to inject faults and analyze the results, respectively. Some experimental results are shown in section 5. Finally, are presented some conclusions and future works.

2. Related Work

Much work has been developed in order to improve reliability on NoC communication. Most of this work regards only protection against noise induced errors in interconnection links. Bertozi, Benini and De Micheli have explored the trade-offs between power/energy consumption and communication reliability mechanisms for on-chip communication links/buses [4][5]. Murali et. al. have presented a discussion of trade-offs (energy efficiency, error protection efficiency and performance impact) involved in various error recovery schemes [6]. Tamhankar has proposed a timing error tolerant communication system named Terror that uses pipelined links [7]. Zimmer has proposed a new fault model notation and an encoding scheme to cope with faults in switch-to-switch links [8]. In another way, Marculescu and Dumitras have proposed a randomized fault-tolerant routing algorithm to cope with switch and link errors [9][10]. It is important to notice that no work regards the evaluation or protection of NoC routers against SEU.

3. RASoC Architecture

In this work, the RASoC router is used as a case study for evaluation of SEU effects. RASoC router is the basic building block of the SoCIN network. SoCIN is a NoC architecture that has a direct topology and might be configured as a 2-D grid, a 2-D torus or a double torus [1].

RASoC is a VHDL soft-core, parameterized in three dimensions: communication channels width, input buffers depth and routing information width [1].

RASoC is built on a distributed way. Its architecture is based on the wormhole switching approach and it uses a deterministic source-based routing algorithm. Also, it applies the handshake protocol for link flow control, and uses round-robin arbitration and input buffering [2].

Externally, RASoC is a routing switch with up to five bi-directional ports (Local, North, East, South and West). Such ports include two unidirectional opposite channels, each one with its data, framing and flow control signals. Internally, RASoC is basically composed by instances of two kinds of modules: input channel and output channel [2]. The RASoC channels are represented in Figure 1.

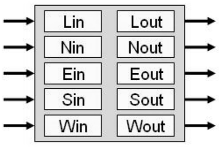


Figure 1. RASoC channels illustration

3.1. The Input Channel Module

The input channel is composed by four blocks named Input Flow Controller (IFC), Input Buffer (IB), Input Controller (IC) and Input Read Switch (IRS) [2]. The internal organization of the input channel module is show in Figure 2.

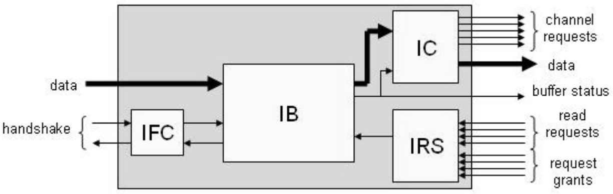


Figure 2. Input Channel module organization

The Input Flow Controller block performs the translation between the handshake and the input buffer flow control protocols. The Input Buffer block is responsible to store flits (the flow control units) of the incoming packets while they cannot be forwarded to an output channel. The Input Controller block performs the routing function, selecting an output channel, emitting a request to the selected output channel, and, finally, updating the routing information in the header. The Input Read Switch block basically controls the read requests from the output channels to the input buffer [2].

3.2. The Output Channel Module

The output channel is also composed by four blocks named Output Controller (OC), Output Data Switch (ODS), Output Read Switch (ORS) and Output Flow Controller (OFC). The internal organization of the output channel module is represented in Figure 3.

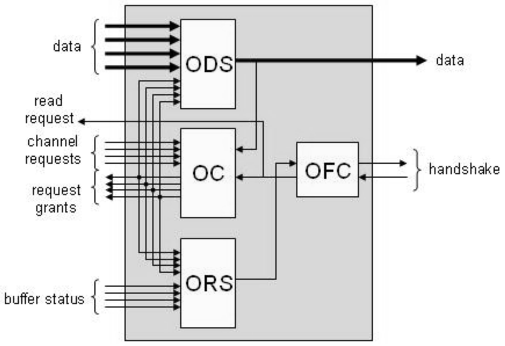


Figure 3. Output Channel module organization

The Output Controller block runs the arbitration task over the requests emitted by the input channels.

The Output Data Switch and Output Read Switch blocks connect the selected input channel to the external output channel interface. Similarly to the Input Flow Controller, the Output Flow Controller block performs the translation between the input buffer flow control and the handshake protocol.

This routing architecture was chosen basically because it implements all routing concepts (such as buffering, arbitration and flow control) in a simple way, allowing us to identify exactly the SEU effects over each router part. As the RASoC is a parameterized router architecture, in our experiments we used a RASoC with the follow parameters (Table 1).

Table 1. RASoC parameters in our experiment

Data Width	8 bits
Buffer Depth	4 Flits
Flit Size	10 bits

4. Fault Injection System

Integrated circuits operating in space environment and more recently at the sea level, can be upset by radiation particles present in such environment [3]. To represent these upsets in the NoC work a fault injection system was developed. This fault injection system was implemented based on a SEU Fault model.

4.1. SEU Fault Model

The space environment is composed of various particles generated by the sun activity. The particles can be classified as two major types: charged particles and electromagnetic radiaton. The charged particles can be electrons, protons or heavy ions. Electromagnetic radiation (photons) can be x-ray, gama-ray or ultraviolet light [3].

When an electrically charged particle hits and passes through an IC, it interacts with the IC silicon and creates a track of electron hole pairs. If a sensitive node, typically the drain of an off transistor, is in the proximity of the ionization track of an electrically charged particle, it collects a significant part of the generated charge carriers (holes or electrons), resulting in a transient current pulse on this node [11].

In a storage cell (e.g., a memory cell, a latch or a flip-flop), such a sufficiently strong transient current pulse can reverse the cell state, producing the inversion in the stored value. In other words, it provokes a bit flip in the storage cell. This phenomenon is called Single Event Upset (SEU) [3] and its effect in a data register is illustrated in Figure 4.

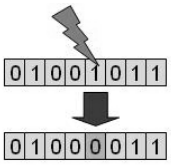


Figure 4. Illustration of the SEU effect in a data register

4.2. RASoC Sensitive Points

As illustrated earlier, SEUs can affect sequential logic components in a circuit. The sensitive points in RASoC architecture are all flip-flops implemented in the router design. Such flip-flops are basically placed in two different design blocks. The first one is the Input Buffer (IB) block (Figure 2). In this block there are a FIFO buffer and a 2-bit register (state register of a Finite State Machine). This FSM is responsible to the buffer control. The buffer is implemented as 4-deep 10-wide shift register. The internal organization of the Input Buffer block is shown in Figure 5.

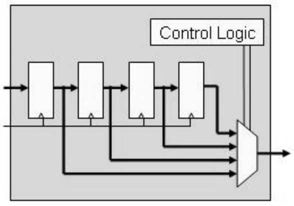


Figure 5. Input Buffer internal organization

The second sensitive block is the Output Controller (OC) block (Figure 3). In this block there are two 2-bit registers and one 4-bit register. The first 2-bit register is the state register of a FSM. The second one and the 4-bits register are responsible for arbitration priority control. In summary, each router port has 50 sensitive bits (42 in the Input Buffer block and 8 in the Output Controller block), totalizing 250 sensitive bits in the RASoC router design.

4.3. Fault Generation

The fault generation mechanism was implemented based on the SEU fault model shown earlier. This mechanism emulates the SEU effects in memory related components (single flip-flops or latches, registers and memories) provoking bit flips into them. The fault location and the time that the fault should occur are chosen pseudo-randomly. This implementation does not concern the mean time between failures; in other words, it just generates one bit flip per execution, without considering the fault occurrence frequency. The implementation of bit flip in a simple register is shown in Figure 6.

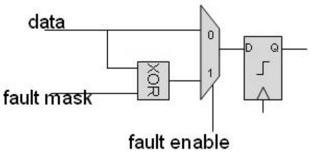


Figure 6. Bit flip emulation mechanism

The implemented fault generation mechanism is divided into 2 main design blocks. Random Generator (RG) generates pseudo-random numbers to be decoded by the Mask Generator. Mask Generator (MG) generates all the fault enable signals to all sensitive points (based on the given random numbers). It also chooses the time and position (fault masks) of the bit flip fault.

The fault generation system generates six fault enable signals, one to each sensitive point in the design (such points were shown earlier) including the FIFO position that the error should occur in. It also generates four different types of fault masks (each one according with the number of bits of the registers).

4.4. Traffic Generation

The traffic generation system used in our experiment is the same one used in [1] in order to validate the RASoC router. This traffic generation architecture (TG) is composed by a set of sub-modules implementing simple traffic generation (STG). Each STG module is responsible for just one generation pattern. Each TG might have more than one STG, according with the number of generation patterns needed.

The objective of using a traffic generation system instead of a real application is to exercise in a random manner the majority of the router circuits. Such approach allows the evaluation of SEU effects independently of the application that is running on the NoC. It is important to notice that the sensitivity of the NOC to SEU must be exhaustively evaluated in order to develop solutions to mitigate the problem.

4.5. Evaluation Approach

In order to evaluate the effects of SEUs on the RASoC work, we used the “golden chip” approach. In this approach, two circuits are simulated together. The first circuit is fault-free and the second one is faulty. During the simulation, the results are compared, and a fault is detected each time the outputs mismatch. The method allows routing problems and packet information errors detection. An overview of the evaluation scenario is illustrated in Figure 7.

The simulation scenario is basically composed by the fault generation mechanism, the traffic generation system and two instances of the RASoC router. Each instance of the RASoC router receives traffic generated by 5 different instances of the Traffic Generation module (one TG for each input channel). Each TG module generates 20 packets by execution, with packet lengths varying pseudo-randomly from 3 to 11 flits. The time interval between packets is also randomly chosen. The comparison and storage of the results is performed by the testbench entity. An overview of the whole simulation scenario is shown in Figure 8.

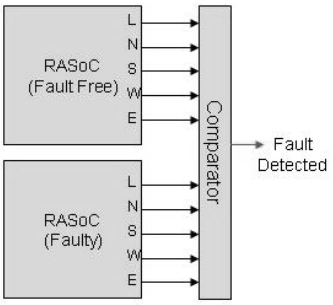


Figure 7. Evaluation scenario overview

The simulation was performed using the ModelSim tool. The total simulation time was 600ms, with 780 faults injected randomly in the 250 sensitive bits and 78,000 packets transmitted. The clock frequency used in our experiment was 100MHz. The RASoC maximum frequency found is 132Mhz (after synthesis for the Xilinx Virtex II FPGA Platform).

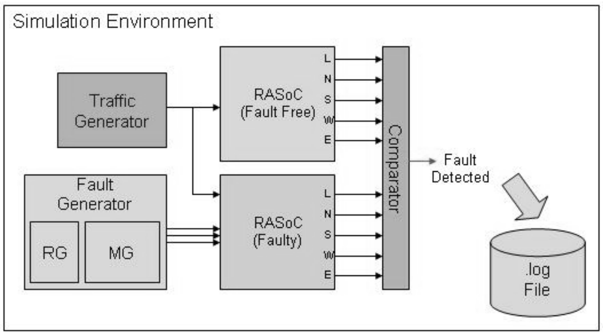


Figure 8. Simulation scenario overview

5. Experimental Results

After the fault injection simulation, an exhaustive analysis was performed. The fault effects that were identified are detailed below:

Packet Missing: This effect occurs when an incoming packet is not routed. Actually, the packet is lost inside the router, that is, the packet can suffer starvation or being overwritten in the buffer.

Single/Multiple Packet Routing Error: Such error occurs when one or more than one packet are routed unproperly. It is not a permanent effect.

Payload Error: This error occurs when the packet payload is changed. Such error might be provoked by a simple bit flip or by increasing/decreasing the number of payload flits.

Packet Formation Error: This error might produce packets without a begin- or end-of-packet marks. Such error can compromise the proper routing.

Router Crash: It is a permanent effect. Such error occurs when the router starts working unproperly. The packets are routed incorrectly or not routed. Such effect can be repaired only with the router reset.

The encountered errors were classified by the location where the faults have been injected into. The experimental results are shown in Table 2.

Table 2: Fault injection results

Fault Location	Faults	Fault Effects		
Arbitration Priority Registers	240	Packet Missing	01	0.4 %
		Router Crash	01	0.4 %
		No effect	238	99.2 %
Arbitration Control FSM State Register	120	Packet Missing	03	2.5 %
		Single/Multiple Packet Routing Error	46	38.3 %
		Router Crash	49	40.8 %
		No effect	22	18.4 %
FIFO Buffer	280	Payload Error	26	9.3 %
		Single/Multiple Packet Routing Error	04	1.4 %
		Router Crash	14	5 %
		No effect	236	84.3 %
FIFO FSM State Register	120	Packet Missing	01	0.8 %
		Packet Formation Error	17	14.2 %
		Router Crash	08	6.7 %
		No effect	94	78.3 %

When faults were injected into the arbitration priority registers, 99.2% of the faults have not provoked any error. Such situation has occurred because the faults have just changed the packet priority, causing only a small delay in the packet delivery. When the faults were injected into the state register of the arbitration control FSM, most of the faults have caused some kind of error. It is important to notice that almost 41% of the faults have provoked the router crash. Such effect could be also identified in 5% of the faults injected into the FIFO buffer. It might have occurred by the corruption of the header flit of a packet, leading the routing control FSM to an unpredictable state and locking the routing FSM work. To repair this kind of error, the unique alternative is to perform the router reset. 84.3% of the injected faults into the FIFO buffer have not caused any error. Such effect has occurred probably because the faults were injected into empty positions of the buffer. When faults were injected into the state register of the FIFO FSM, most of the faults have provoked no errors. However, 14.2% of the faults have provoked some kind of packet formation error.

6. Conclusions and Future Works

After all experiments, we could observe that the occurence of SEUs in a NoC router architecture can affect the proper NoC service. Such effects can vary from a simple lost of packet up to a permanent interruption of the router service, depending on *where* and *when* the fault has occured.

Once evaluated the effects of SEU on each router sensitive part, the results can provide enough resources to guide a suitable hardware protection against SEUs.

Future work includes the study and implementation of SEU mitigation techniques for NoC routers. Such protection can include just one kind of well-known SEU mitigation techniques (such as Triple Modular Redundancy, Time Redundancy, Error Detection/Correction Codes) or a combination of them (according with the trade-off between performance impact, protection efficiency, area overhead and power/energy consumption).

7. References

[1] Zeferino, C. A., Redes-em-Chip: Arquiteturas e Modelos para Avaliação de Área e Desempenho. Ph.D. Thesis – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, RS, 2003.

[2] Zeferino, C. A., Susin. A. A. SoCIN: A Parametric and Scalable Network-on-Chip. In: Proceedings of 17th Symposium on Integrated Circuits and Systems (SBCCI), IEEE CS Press, 2003. pp.169-174.

[3] Kastensmidt, F. G. L., Designing Single Event Upset Mitigation Techniques for Large SRAM-Based FPGA Components. Ph.D. Thesis – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, RS, 2003.

[4] Bertozzi, D.; Benini, L.; De Micheli, G.; “Error control schemes for on-chip communication links: the energy-reliability tradeoff”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 24, Issue 6, June 2005.

[5] Bertozzi, D.; Benini, L.; De Micheli, G.; “Low power error resilient encoding for on-chip data buses”, In: Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2002. 4-8 March 2002.

[6] Murali, S.; Theocharides, T.; Vijaykrishnan, N.; Irwin, M.J.; Benini, L.; De Micheli, G.; “Analysis of error recovery schemes for networks on chips”, IEEE Design & Test of Computers, Volume 22, Issue 5, Sept.-Oct. 2005.

[7] Tamhankar, R.R.; Murali, S.; De Micheli, G.; “Performance driven reliable link design for networks on chips”, In: Proceedings of the Asia and South Pacific Design Automation Conference, 2005. Volume 2, 18-21 Jan. 2005.

[8] Zimmer, H.; Jantsch, A.; “A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip”, First IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, 1-3 Oct. 2003.

[9] Marculescu, R.; “Networks-on-chip: the quest for on-chip fault-tolerant communication”, 2003. In: Proceedings of IEEE Computer Society Annual Symposium on VLSI, 20-21 Feb. 2003.

[10] Dumitras, T.; Kerner, S.; Marculescu, R.; “Towards on-chip fault-tolerant communication”, In: Proceedings of the Asia and South Pacific Design Automation Conference, 2003. 21-24 Jan. 2003.

[11] Nicolaidis, M.; “Design for soft error mitigation”, IEEE Transactions on Device and Materials Reliability, Volume 5, Issue 3, Sept. 2005.

2.2 Evaluating SEU and Crosstalk Effects in Network-on-Chip Routers

Category: Short Paper

Conference: IOLTS2006: Proceedings of the 12th IEEE International On-Line Testing Symposium

Location: Como - Italy

Date: July, 2006

Evaluating SEU and Crosstalk Effects in Network-on-Chip Routers

Arthur Pereira Frantz, Luigi Carro, Érika Cota, Fernanda Lima Kastensmidt
UFRGS, Instituto de Informática – DELET, PPGC, Porto Alegre, RS, Brazil
{apfrantz, carro, erika, fglima}@inf.ufrgs.br

Abstract

This work intends to evaluate the effect of a Single Event Upsets (SEUs) and crosstalk faults in a NoC router architecture by developing a fault injection mechanism, allowing an accurate analysis of the impact of SEU and crosstalk over the router service. Results show that such faults may affect the router behavior, causing loss of packets, errors in packet information or even compromising the router service, provoking permanent routing problems.

1. Introduction

The shrinking of the transistors dimensions and the reduction of the power supply voltage of the new ICs fabrication technologies have increased the sensitivity of circuits to internal and external noise, which can affect the correct operation of the system [1,2]. As an example, the radiation present in the space environment can induce faults with transient effects in the circuit. Such faults are known as soft errors and the Single Event Upset (SEU) is the most typical fault in this category [3,4]. Furthermore, delay variations and crosstalk noise have become an issue with the continuously shrinking geometry of semiconductor devices and the increasing switching speed [5].

To deal with the aforementioned problems, some work has been developed to protect on-chip interconnects against crosstalk [2,5,6] and sequential circuits against SEU faults [3,4]. The techniques proposed in the literature cope with crosstalk and SEUs independently, based on the assumption that connections cannot be affected by SEUs (no sequential circuit involved), whereas sequential circuits cannot be affected by crosstalk faults. However, when Networks-on-Chip (NoCs) are used, buffers and sequential circuits are present within the interconnections (NoC routers).

In this context, this paper intends to evaluate the effects of SEU and crosstalk faults in a NoC router architecture by performing fault injection simulations, allowing an accurate analysis of the impact of such faults over the router service. Results show that the effects of those faults in the system operation can be disastrous, ranging from a simple loss of packets up to the permanent interruption of the router service.

2. Fault Injection System

In order to represent SEU and crosstalk faults in the router behavior, a fault injection system was developed. The case study NoC architecture used in the experiments was the RASoC router [7]. The router was configured to have 5 channels, buffer depth of 4 positions and 8-bits data width, resulting in 3,352 gates and 250 flip-flops.

In the fault injection system SEU faults were represented as bit-flips in sequential logic components, with fault location and fault instant pseudo-randomly chosen. The sensitive points to SEU in the router design are the input buffers (buffer and control logic) and registers of the arbitration control.

Crosstalk faults have been represented as glitches and delays in the local link lines. Such representation was based on the MAF model [8] that just considers a fault as affecting only one link line, termed victim, at a time. The remaining lines are designated aggressors, and act collectively to generate an error condition on the victim.

The fault injection system uses a pseudo-random traffic generator instead of a real application, in order to exercise in a random manner the majority of the router circuits, independently of the application that is running on the NoC.

3. Experimental Results

A total of 858 faults were injected in all sensitive points of the router design, with 11.42% of the faults representing crosstalk faults and 88.58% representing SEU faults. During the simulations a total of 78,000 packets have been transmitted.

Table 1: Fault injection results for a pseudo-random traffic (total simulation time = 700ms)

Fault Type	Fault Location	# of faults	Fault Effects					
			Packet Missing	Packet Routing Error	Payload Error	Packet Formation Error	Router Crash	No effect
SEU	Arbitration Priority Registers	240	0.4%	0%	0%	0%	0.4%	99.2%
	Arbitration FSM State Register	120	2.5%	38.3%	0%	0%	40.8%	18.4%
	FIFO Buffer	280	0%	1.4%	9.3%	0%	5%	84.3%
	FIFO FSM State Register	120	0.8%	0%	0%	14.2%	6.7%	78.3%
	SEU Effects Average (in 700ms simulation)		0.66%	6.58%	3.42%	2.24%	9.47%	77.1%
Crosstalk	NoC Links	98	0%	6.1%	83.7%	0%	10.2%	0%

After the fault injection simulations, a number of fault effects was identified. The packet missing error occurs when an incoming packet suffers starvation or is overwritten inside the buffer. A payload error occurs when the packet content is changed during the transmission. A single/multiple packet routing error occurs when one or more packets are routed incorrectly, leading to the loss of such packets. The packet formation error might produce packet without the correct framing signals, compromising the routing task. The router crash is a permanent effect, causing the interruption of the routing service. Such effect can be repaired only with the router reset. The encountered errors were classified by the location where the faults have been injected into (Table 1).

One can observe in Table 1 that 100% of the crosstalk faults have provoked some kind of error, while 77.1% of the SEU faults (average) have caused no effect. Most of the crosstalk faults (83.67%) and 3.42% of SEU faults (average) have caused payload errors. This effect might be corrected by software-implemented techniques and higher-level approaches based on packet retransmission. However, more than 10% of the crosstalk and 9.47% of SEU faults (average) caused the router crash. It is important to notice that this effect cannot be corrected by packet retransmission or software techniques, and needs the hardware reset.

By analyzing the experimental results, one can notice that the combination of SEU and crosstalk only makes the problem worse, because in that case the system can be dealing with multiple faults at same time.

4. Conclusions and Current Work

After all experiments, one can observe that the occurence of SEU and crosstalk faults in a NoC router architecture can affect its proper function. Such effects can vary from a simple loss of packet up to a permanent interruption of the router service, depending on *where* and *when* the fault has occured.

Once evaluated the effects of SEU and crosstalk fault over the router sensitive parts, the results can provide enough resources to guide a suitable hardware protection against such faults.

Current techniques used to protect links against crosstalk based on link encoding or retransmission can deal only with the crosstalk problem, but not necessarily with SEU faults and the combination of both upsets. On going research aims at finding protection alternatives to NoC routers in order to simultaneously mitigate multiple upsets, such as SEU, Single Event Transient (SET) and crosstalk faults.

References

[1] Murali, S.; Theocharides, T.; Vijaykrishnan, N.; Irwin, M.J.; Benini, L.; De Micheli, G.; “Analysis of error recovery schemes for networks on chips”, IEEE Design & Test of Computers, Volume 22, Issue 5, Sept.-Oct. 2005.

[2] Acquaviva, A.; Bogliolo, A.; “A Bottom-Up Approach to On-Chip Signal Integrity”, Lecture Notes in Computer Science, Volume 2799, Jan 2003.

[3] Nicolaidis, M.; “Design for soft error mitigation**”,** *IEEE Transactions on Device and Materials Reliability*, Volume 5, Issue 3, Sept. 2005, pp. 405-418.

[4] Kastensmidt, F., Carro, L., Reis, R.; *Fault-Tolerance Techniques for SRAM-based FPGAs*, Series: Frontiers in Electronic Testing, Springer, Vol. 32, 2006. 180 p.

[5] Rossi, D.; Metra, C.; Nieuwland, A.K.; Katoch, A.; “Exploiting ECC redundancy to minimize crosstalk impact”, IEEE Design & Test of Computers, Volume 22, Issue 1, Jan 2005.

[6] Nieuwland, A.K.; Katoch, A.; Rossi, D.; Metra, C.; “Coding techniques for low switching noise in fault tolerant busses”, *In*: 11th IEEE International On-Line Testing Symposium, 2005. *Proceedings...* pp. 183-189, 6-8 July 2005.

[7] Zeferino, C. A., Susin, A. A., “SoCIN: A Parametric and Scalable Network-on-Chip”. *In*: 17th Symposium on Integrated Circuits and Systems (SBCCI), 2003. *Proceedings…* pp. 169-174, 2003.

[8] Cuviello, M.; Dey, S.; Bai, X.; Zhao, Y.; “Fault modeling and simulation for crosstalk in system-on-chip interconnects”, *In*: 1999 IEEE/ACM International conference on Computer-Aided Design. *Digest of Technical Papers*, pp. 297-303, 7-11 Nov. 1999.

2.3 Evaluation of SEU and Crosstalk Effects in Network-on-Chip Switches

Category: Full Paper

Conference: SBCCI2006: Proceedings of the 19th Annual Symposium on Integrated Circuits and Systems Design

Location: Ouro Preto - Brazil

Date: September, 2006

Evaluation of SEU and Crosstalk Effects in Network-on-Chip Switches

Arthur Pereira Frantz, Fernanda Lima Kastensmidt, Luigi Carro, Érika Cota
UFRGS, Instituto de Informática, PPGC
Av. Bento Gonçalves, 9500 Bloco IV, Porto Alegre, RS, Brazil
{apfrantz, fglima, carro, erika}@inf.ufrgs.br

ABSTRACT

As the complexity of designs increases and the technology scales down into the deep sub-micron domain, devices and interconnections are subject to new types of malfunctions and failures. This work intends to evaluate the effect of Single Event Upsets (SEUs) and crosstalk faults in a Network-on-Chip switch by performing fault injection simulations, allowing an accurate analysis of the impact of these faults over the switch service. The results show that such faults might affect the switch behavior, with errors ranging from simple loss of packets up to the permanent interruption of the switch service.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance.

General Terms

Design, Reliability.

Keywords

Network-on-Chip, Single-Event Upset, Crosstalk.

1. INTRODUCTION

The advances in deep sub-micron technologies have allowed integration of a huge amount of transistors in a single die. Consequently, complex system designs could be integrated into a single chip (SoC – System-on-Chip). Such systems are based on the reuse of pre-designed and pre-verified IP cores.

With the growing of the number of cores embedded in the system, interconnection among them has become a major bottleneck to system design. Recently, NoCs (Networks-on-chip) have been proposed as an alternative communication platform capable of handling performance, energy consumption, and reusability issues of large integrated systems [1,2]. NoCs are based on the same concepts adopted by interconnection networks used in parallel computers [3], and provide high performance communication for systems with intensive communication requirements.

In the other hand, with the shrinking of the transistors dimensions and the reduction of the power supply voltage, the sensitivity of the circuit to internal and external noises has increased, being able to affect the proper work of the system [4,5]. Errors might come from the fabrication process (process variability), frequency increase, or from the operation environment. The radiation present in the space environment can induce faults with transient effects in the circuit. Such faults are known as soft errors and the Single Event Upset (SEU) [6] is the most typical fault in this category. SEU faults are bit-flips in sequential logic elements, such as registers and memories [7]. Furthermore, delay variations and crosstalk noise have become an issue with the continuously shrinking geometry of semiconductor devices and the increasing switching speed [8].

When NoCs are used, buffers and sequential circuits are present in the routers, while the number of connections between any two switches can be very large. Therefore, since each fault can be originated by statistically independent events, one must consider the possibility of crosstalk and SEU faults affecting the same communication channel at the same time.

Based on this assumption, this paper intends to evaluate the effects of SEU and crosstalk faults in a NoC architecture by performing fault injection simulations, allowing an accurate analysis of the impact of such faults over the switch service. The results show that the effect of those faults in the system operation can be disastrous, leading to loss of packets and system crash or unavailability.

This paper is organized as follows. In the second section, it is presented some related work. In the third section, it is presented the fault injection system. In section 4 it is shown an overview of the NoC switch architecture used for evaluation of SEU and crosstalk fault effects. Section 5 presents the evaluation approach used to inject faults and analyze the results. Some experimental results are shown in section 6. Finally, in section 7 are presented some conclusions and future works.

2. RELATED WORK

Several works have addressed the problem of ensuring reliability in ICs produced with new deep-sub-micron technologies. A number of solutions to cope with soft errors are reported in [6] ranging from well-known techniques, such as triple modular redundancy (TMR) and time redundancy (TR), to combined hardware and software-based approaches. In [9], another solution to protect ICs against radiation-induced errors is proposed, based on the reuse of design-for-testability and debug resources.

Mitigation of crosstalk-induced errors has also been considerably addressed in the literature. Rossi et. al. use error correction codes

SBCCI'06, August 28–September 1, 2006, Minas Gerais, Brazil.

(ECC) to minimize the crosstalk impact in interconnect busses [8]. In the same direction, Nieuwland et. al. propose the use of coding techniques to reduce switching noise in fault tolerant busses [9]. In [10], Bertozzi et. al. investigate energy and power efficient ways of encoding data buses to deal with transient and noise induced faults. Lajolo proposes an online solution for detection and correction of faults in SoC buses [11], consisting in small detection/recovery modules distributed along the bus interconnection architecture.

Moving to the NoC domain, Tamhankar proposes a timing error tolerant communication system, named Terror, which uses pipelined links [12]. Marculescu and Dumitras propose a randomized fault-tolerant routing algorithm to cope with switch and link errors [13,14]. More recently, a discussion of the trade-offs between energy efficiency, error protection efficiency, and performance impact involved in various error recovery schemes is presented in [4] and [15].

Thus, although one can find in the literature a number of solutions to independently mitigate SEU faults in sequential blocks and crosstalk faults in interconnects, as previously mentioned, very few work has addressed an analysis of the effects of such faults on Networks-on-Chip. In [16], the effects of SEU have been analyzed in a CAN-based network. However, the presented results do not show details about the correlation between the location of the fault and its effects in the entire system. This paper investigates in more details such correlation and also analyzes the combined effects of crosstalk and SEU faults in NoCs.

3. FAULT INJECTION SYSTEM

Drastic device shrinking, high logic complexity, power supply reduction, and high operating speeds that accompany the technological evolution to nanometric technologies have reduced dramatically the reliability of deep sub-micron ICs [12]. Significant problems are related to soft errors induced by radiation and crosstalk faults. In order to represent these upsets in the NoC behavior, a fault injection system was developed. This fault injection system was implemented based on the SEU and crosstalk fault models presented below.

3.1 SEU Fault Model

The space environment is composed of various particles generated by the sun activity. Such particles can be classified as two major types: charged particles and electromagnetic radiation [7].

When an electrically charged particle hits and passes through an IC, it interacts with the IC silicon and creates a track of electron hole pairs. If a sensitive node, typically the drain of an off transistor, is in the proximity of the ionization track of an electrically charged particle, it collects a significant part of the generated charge carriers (holes or electrons), resulting in a transient current pulse on this node [17].

In a storage cell (e.g., a memory cell, a latch or a flip-flop), such a sufficiently strong transient current pulse can reverse the cell state, producing the inversion in the stored value. In other words, it provokes a bit flip in the storage cell. This phenomenon is called Single Event Upset (SEU) [7] and its effect in a memory cell is illustrated in Figure 1.

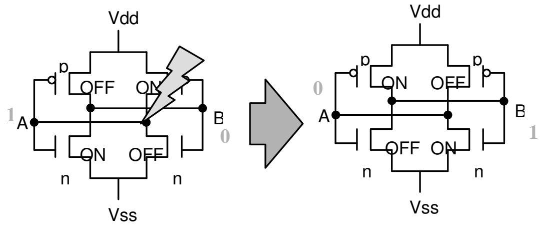


Figure 1. Illustration of the SEU effect in a memory cell

3.2 Crosstalk Fault Model

Due to the continuous shrinking geometry of semiconductor devices, the widths of metallic interconnects are being reduced, while their resistances are being augmented. As a consequence, the delay introduced by these interconnections is increasing. One way to deal with these problems is to increase the thickness of the metal lines in order to augment their cross section and, hence, reduce their resistance. However, such solution increases the crosstalk problems because, in this way, metal lines on different layers have less space between them [14].

An widely adopted simulation model for crosstalk faults is the MAF model (Maximal Aggressor Fault) [18]. In order to reduce the fault set and speed-up the fault simulation, the MAF model just considers a fault as affecting only one wire, termed victim, at a time. The remaining wires are designated aggressors, and act collectively to generate an error condition on the victim. In the MAF model the fault set is also reduced considering only the worst-case combinations of coupling impedance among all possible aggressors. Thus, the MAF model considers all N-1 aggressors on a bus to be transitioning in the same direction as a fault [18]. An example of the possible effects in the MAF model is shown in Figure 2.

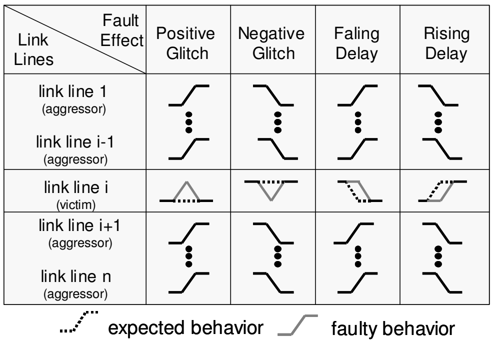


Figure 2. Possible crosstalk effects in the MAF model

In the MAF model just four types of errors might occur:

- **Positive Glitch Error**: occurs when all aggressor lines are transitioning simultaneously from ‘0’ to ‘1’ and the victim line is ‘0’;
- **Negative Glitch Error**: occurs when all aggressor lines are transitioning simultaneously from ‘1’ to ‘0’ and the victim line is ‘1’;

- **Rising Delay Error**: occurs when all aggressor lines are transitioning simultaneously from '1' to '0' and the victim line is transitioning from '0' to '1';

- **Falling Delay Error**: occurs when all aggressor lines are transitioning simultaneously from '0' to '1' and the victim line is transitioning from '1' to '0'.

3.3 Fault Generation

The fault generation mechanisms were implemented based on the SEU and crosstalk fault models previously presented.

The SEU fault injection mechanism simulates the SEU effects in memory related components (single flip-flops or latches, registers and memories) provoking bit flips into them. The implementation of the SEU bit flip simulation mechanism in a simple register is shown in Figure 3.

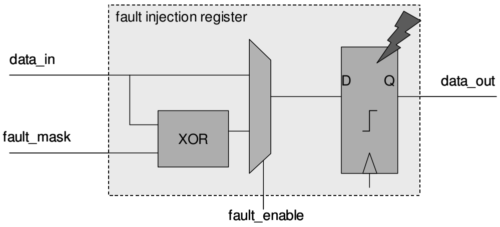


Figure 3. SEU bit flip simulation mechanism

In the SEU fault injection mechanism, the fault location and the time that the fault should occur are chosen pseudo-randomly. However, this implementation does not concern the mean time between failures; in other words, it just generates one fault per execution, without considering the fault occurrence frequency. The implemented fault generation mechanism is divided into 2 main design blocks. Random Generator (RG) generates pseudo-random patterns that are decoded by the Mask Generator. Mask Generator (MG) generates all the fault enable signals that control the fault injection logic added to all sensitive points in the design (registers and buffers).

The crosstalk injection mechanism simulates crosstalk faults on the switch local link, introducing undesired glitches and delays based on the MAF model. The implementation of the crosstalk injection mechanism in a link line is presented in Figure 4.

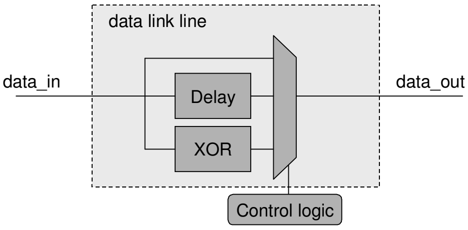


Figure 4. Crosstalk simulation mechanism

In the crosstalk fault injection mechanism, each line of the local link was adapted, including the fault injection block presented in Figure 4. As the traffic used in the experiments is pseudo-randomly generated (low probability of all lines transitioning at the same time), only the fault effects (glitches and delays) and the characteristics of one victim at a time have been implemented. Thus, similarly to the SEU fault injection mechanism, the fault time and fault location are chosen in a pseudo-random way.

The fault generation system is able to pseudo-randomly inject single SEU faults (see Figure 5.a), single crosstalk faults (see Figure 5.b) and a combination of a single SEU and a single crosstalk faults (see Figure 5.c). It is important to observe that the execution time (time of a complete traffic simulation) and the time that the faults should occur are pseudo-randomly chosen.

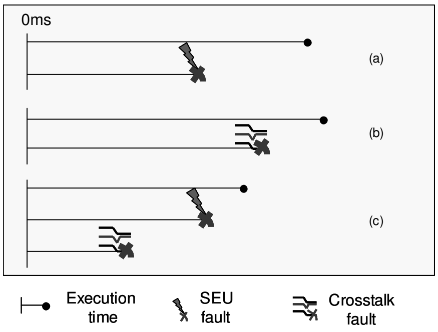


Figure 5. Fault time distribution illustration

3.4 Traffic Generation

Aiming at generating traffic to simulate the switch design, each router port has been connected to one traffic generation bloc (TG). The traffic generation block is composed by a set of sub-modules, each one implementing one traffic pattern (STG – Simple Traffic Generation). Each TG block might have more than one STG, according with the number of generation patterns needed.

The objective of using a traffic generation system instead of a real application is to exercise in a random manner the majority of the router circuits. Such approach allows the evaluation of SEU and crosstalk effects independently of the application that is running on the NoC. It is important to notice that the sensitivity of the NOC to SEU and crosstalk must be exhaustively evaluated in order to develop solutions to mitigate the problem.

4. CASE STUDY ARCHITECTURE

We based our analysis on a packet-switching network model proposed in [19]. It is implemented in a 2-D mesh topology. In such a model the communication channels between two adjacent routers are defined to be 10-bit wide. This network is based on the RASoC router, which is composed of five input and five output ports. One pair of input/output ports is dedicated to the connection between the router and the core, while the remaining four pairs connect the router with the four adjacent routers, as depicted in Figure 6. Such ports include two unidirectional opposite channels, each one with its data, framing and flow control signals.

RASoC router is a VHDL soft-core, parameterized in three dimensions: communication channels width, input buffers depth and routing information width. It is implemented using from 3,000 to 6,000 gates, depending on the bitwidth of the network channel and depth of the input buffers [19]. Its architecture is based on the wormhole switching approach and it uses a deterministic source-based routing algorithm. Also, it applies the handshake protocol for link flow control, uses round-robin arbitration and input buffering [19].

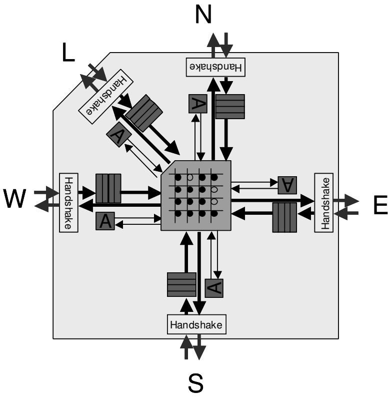


Figure 6. Basic structure of the RASoC router

As the RASoC is a parameterized architecture, table 1 shows the RASoC configuration used in the experiments presented in this paper.

Table 1. RASoC configuration used in the experiments

Data Width	8 bits
Buffer Depth	4 Flits
Flit Size	10 bits

4.1 RASoC Switch Sensitive Points

As previously illustrated, SEU faults can affect sequential logic components in a circuit. The first sensitive block is the input buffer block. In this block there are a FIFO buffer and a 2-bit control register (state register of a Finite State Machine). FIFO buffer is implemented as a 4-deep 10-wide shift register.

The second sensitive point in the RASoC design is the output controller. In this block, there are two 2-bit registers and one 4-bit register. The first 2-bit register is the state register of a FSM responsible to control the arbitration mechanisms. The second one and the 4-bits register are responsible for arbitration priority control.

Regarding crosstalk faults, they can affect the links in RASoC design. As the RASoC router uses input buffering, the first router task is to store the incoming data. Consequently, crosstalk faults can indirectly affect the data stored in the input buffers.

5. EVALUATION APPROACH

Aiming at evaluating the effects of SEU and crosstalk faults on the RASoC router behavior, the "golden chip" approach is used. In this approach, two circuits are simulated *in tandem* (the first one is fault-free and the second one is faulty). During the simulation, the router outputs are observed and compared, and a fault is detected each time the outputs mismatch. This method allows the detection of routing problems and errors in the packet payload.

The simulation scenario is basically composed by the fault generation mechanisms (SEU and crosstalk injection), the traffic generation system and two instances of the RASoC router. Each instance of the RASoC router receives traffic generated by 5 different instances of the Traffic Generation module (one TG for each input channel). The comparison and storage of the results is performed by the testbench entity. An overview of the whole simulation scenario is shown in Figure 7.

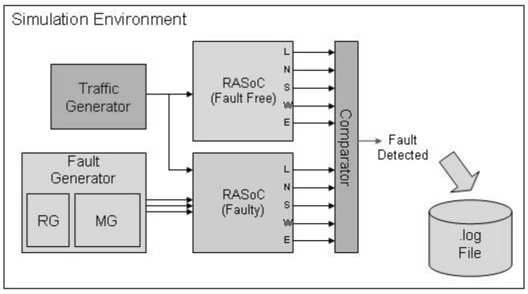


Figure 7. Simulation scenario overview

The simulation was performed using the ModelSim tool. The total simulation time was 15.2s, with 18,909 SEU faults injected randomly in the 250 sensitive bits, 17,108 crosstalk faults injected on the local link (10 lines).

6. EXPERIMENTAL RESULTS

After the SEU and crosstalk fault injection simulation, the generated results have been analyzed. The fault effects that were identified are detailed below:

- **Single/Multiple Packet Routing Error**: Such error occurs when one or more packets are routed improperly. A packet can also suffer starvation or being overwritten inside the buffer. It is not a permanent effect.
- **Payload Error**: This error occurs when the packet payload is changed. Such error might be provoked by a simple bit flip or by increasing/decreasing the payload length. It might also produce packets without the correct framing signals (marks for begin- and end-of-packet).
- **Router Crash**: It is a permanent effect. Such error occurs when the router starts working improperly. The packets are routed incorrectly or not routed. Such effect can be repaired only with the router reset.

Considering the SEU injected faults, the encountered errors were classified by the location where the faults have been injected into. The experimental results are shown in Table 2.

When SEU faults were injected into the arbitration priority registers, 99.17% of the faults have not provoked any error. Such situation has occurred because the faults have just changed the packet priority, causing only a small delay in the packet delivery. When faults were injected into the state register of the arbitration control FSM, most of the faults have caused some kind of error. It is important to notice that almost 41% of such faults have provoked the router crash. It is also important to observe that this effect cannot be corrected by packet retransmission or software techniques, and it needs the hardware reset.

Table 2. Fault injection results

Fault Type	Fault Location	Fault Effects							
		Packet Routing Error		Payload Error		Router Crash		No Effect	
		#	%	#	%	#	%	#	%
SEU	Arbitration Priority Registers	0	0.00	1	0.42	1	0.42	238	99.17
	Arbitration FSM State Register	46	38.33	3	2.50	49	40.83	22	18.33
	FIFO Buffer	80	5.54	168	11.63	24	1.66	1,173	81.18
	FIFO FSM State Register	0	0.00	18	15.00	8	6.67	94	78.33
Crosstalk	NoC Local Link	6	4.84	82	66.13	10	8.06	26	20.97
SEU + Crosstalk	Arbitration Priority Registers + link	48	4.03	313	26.30	64	5.38	765	64.29
	Arbitration FSM State Register + link	330	55.46	38	6.39	144	24.20	83	13.95
	FIFO Buffer + link	1,264	8.66	4,447	30.45	791	5.42	8,102	55.48
	FIFO FSM State Register + link	56	9.41	129	21.68	93	15.63	317	53.28

Router crash errors could be also identified when faults were injected into the input buffer, but in a very small percentage. 81.18% of the injected faults into the input buffer have not caused any error. Such effect has occurred because such faults were injected into empty positions of the buffer. Regarding the state register of the input buffer FSM, most of the injected faults have provoked no errors. However, almost 7% of such faults have provoked the router crash.

Considering crosstalk faults, one can observe that almost 80% of the injected faults have provoked some kind of error. An expressive amount of such faults (8.06%) generated router crash errors, while 66.13% of them have caused payload errors.

Since the physical effects that cause the occurrence of SEU and crosstalk faults are different, both events have a probability of occurring simultaneously. This is possible because the faults are independent and there is no correlation between the occurrence of crosstalk and SEU faults. It is also important to notice that different faults can compensate the effect of each other, but such probability is very low because, such fact depends on the time and location of the faults.

Analyzing the experimental results, one can notice that the combination of SEU and crosstalk only makes the problem worse, because in that case the system can be dealing with multiple faults at same time, increasing the probability of router crash errors.

When crosstalk and SEU faults were injected simultaneously in the local link and arbitration priority registers, respectively, the occurrence of errors has increased in 4.96%. When such faults have been injected in the local link and input buffer, such percentage has increased in 3.76%. Considering the state register of the input buffer FSM, such growth is about 8.96%.

7. CONCLUSIONS AND FUTURE WORKS

After all experiments, one can observe that the occurrence of SEU and crosstalk faults in a NoC switch architecture can affect its proper function. Such effects can range from a simple loss of packet up to a permanent interruption of the switch service, depending on where and when the faults have occurred.

Once evaluated the effects of SEU and crosstalk faults over the router sensitive parts, the results can provide enough resources to guide a suitable hardware protection against such faults. Due to the high probability of router crash errors, it is important to investigate fault-tolerant solutions implemented in hardware in order to avoid the necessity of reset operations.

Future work includes the study of SEU and crosstalk mitigation techniques for NoC routers. Such protection can include just one kind of mitigation techniques (such as Triple Modular Redundancy, Time Redundancy, Error Detection/Correction Codes) or a combination of them (according with the trade-off between performance impact, protection efficiency, area overhead and power/energy consumption).

8. REFERENCES

[1] Benini L.; Micheli. G. D.; “Networks on Chips: A New SoC Paradigm”. *IEEE Computer*, Vol. 35, January, 2002, pp. 70-78.

[2] Dally, W. J.; Towles, B.; “Route Packets, Not Wires: On-Chip Interconnection Networks”. *In*: Design Automation Conference, 2001, *Proceedings...* pp. 684-689, 2001.

[3] Duato, J.; Yalamanchili, S.; Ni, L.; Interconnection Networks: An Engineering Approach”. IEEE Computer Society, Los Alamitos, CA 1997.

[4] Murali, S.; Theocharides, T.; Vijaykrishnan, N.; Irwin, M.J.; Benini, L.; De Micheli, G.; “Analysis of error recovery schemes for networks on chips”, *IEEE Design & Test of Computers*, Volume 22, Issue 5, Sept.-Oct. 2005, pp. 434-442.

[5] Acquaviva, A.; Bogliolo, A.; “A Bottom-Up Approach to On-Chip Signal Integrity”, Lecture Notes in Computer Science, Volume 2799, pp. 540-549, Jan 2003.

[6] Nicolaidis, M.; “Design for soft error mitigation**”,** *IEEE Transactions on Device and Materials Reliability*, Volume 5, Issue 3, Sept. 2005, pp. 405-418.

[7] Kastensmidt, F., Carro, L., Reis, R.; *Fault-Tolerance Techniques for SRAM-based FPGAs*, Series: Frontiers in Electronic Testing, Springer, Vol. 32, 2006. 180 p.

[8] Rossi, D.; Metra, C.; Nieuwland, A.K.; Katoch, A.; “Exploiting ECC redundancy to minimize crosstalk impact”, *IEEE Design & Test of Computers*, Volume 22, Issue 1, Jan 2005, pp. 59-70.

[9] Nieuwland, A.K.; Katoch, A.; Rossi, D.; Metra, C.; “Coding techniques for low switching noise in fault tolerant busses”,

In: 11th IEEE International On-Line Testing Symposium, 2005. *Proceedings…* pp. 183-189, 6-8 July 2005.

[10] Bertozzi, D.; Benini, L.; De Micheli, G.; "Low power error resilient encoding for on-chip data buses", *In*: Design, Automation and Test in Europe Conference and Exhibition, 2002. *Proceedings…* pp. 102-109 , 4-8 March 2002.

[11] Lajolo, M.; "Bus guardians: an effective solution for online detection and correction of faults affecting system-on-chip buses", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Volume 9, Issue 6, Dec. 2001, pp. 974–982.

[12] Tamhankar, R.R.; Murali, S.; De Micheli, G.; "Performance driven reliable link design for networks on chips", *In*: Asia and South Pacific Design Automation Conference, 2005. *Proceedings…* pp. 749–754, Volume 2, 18-21 Jan. 2005.

[13] Marculescu, R.; "Networks-on-chip: the quest for on-chip fault-tolerant communication", 2003. *In*: IEEE Computer Society Annual Symposium on VLSI, 2003. *Proceedings…* pp. 8–12, 20-21 Feb. 2003.

[14] Dumitras, T.; Kerner, S.; Marculescu, R.; "Towards on-chip fault-tolerant communication", *In*: Asia and South Pacific Design Automation Conference, 2003. *Proceedings…* pp. 225-232, 21-24 Jan. 2003.

[15] Bertozzi, D.; Benini, L.; De Micheli, G.; "Error control schemes for on-chip communication links: the energy-reliability tradeoff", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Volume 24, Issue 6, June 2005, pp. 818–831.

[16] Perez, J.; Reorda, M.S.; Violante, M.; "Early, Accurate Dependability Analysis of CAN-Based Networked Systems", *IEEE Design & Test of Computers*, Volume 23, Issue 1, Jan. 2006.

[17] Nicolaidis, M.; "Design for soft error mitigation", IEEE Transactions on Device and Materials Reliability, Volume 5, Issue 3, Sept. 2005.

[18] Cuviello, M.; Dey, S.; Bai, X.; Zhao, Y.; "Fault modeling and simulation for crosstalk in system-on-chip interconnects", *In*: 1999 IEEE/ACM International conference on Computer-Aided Design. *Digest of Technical Papers*, pp. 297-303, 7-11 Nov. 1999.

[19] Zeferino, C. A., Susin, A. A., "SoCIN: A Parametric and Scalable Network-on-Chip". *In*: 17th Symposium on Integrated Circuits and Systems (SBCCI), 2003. *Proceedings…* pp. 169-174, 2003.

3 HARDWARE-BASED MITIGATION TECHNIQUES

As it can be observed in the previos chapter, soft errors and crosstalk faults can compromise the proper service of the RASoC router. To improve the reliability of the RASoC design and mitigate the crosstalk and soft error effects over the links and input buffer parts, we have implemented and evaluated five different solutions. These proposed implementations are presented in (FRANTZ et al., 2006b).

Many works in the literature refered to the use of Error Correction Codes (ECCs) to recover information in case of occurence of crosstalk faults in communication links. It is also refered the use of ECCs and TMR for protection against soft errors. Taking this information in consideration, the developed hardware solutions include the use of an ECC (hamming code) and the combination of the ECC and modifications in the design of the input buffers and communication channels.

The first developed approach uses only hamming code in the input buffer. Such solution presented a high dependability against soft errors, but a high vulnerability to crosstalk. It occured because crosstalk faults act on the communication links, where no protection has been implemented. In the second hardware solution, the vulnerability to crosstalk was reduced by keeping the input buffer encoded and also encoding the communication links. With that it was achieved a high dependability to single soft errors or crosstalk faults. However, in case of double faults (a crosstalk plus a soft error) such protection scheme cannot correct the faults and the router might fail.

By inserting an extra correction stage, as presented in the third solution, the vulnerability to soft error and crosstalk is reduced to zero. This result is very good but this solution can only deal with one crosstalk per link at a time (modeled like this only for simulation).

In the fourth solution, an adaptation of the protection scheme Terror (TAMHANKAR; MURALI; MICHELI, 2005) was implemented in the communication links. In order to protect the input buffer against soft errors, it was kept enconded by hamming code. This solution presented a good protection against crosstalk faults, but a high vulnerability to soft errors due to the unprotected Terror block sample registers.

The final solution implements a timing redundancy approach, where each input information is sampled three times, instead of twice in the Terror approach, and being followed by majority voter. Such an aditional sampling improves reliability due to the voting logic, which is also able to cope with soft errors.

However, analyzing more deeply the RASoC router design and the fault injection results, it has been possible to identify some reasons why the RASoC router crashed in the presence of some faults. The major two problems in the design are:

- In case of a fault in the links or input buffers that corrupted the framing data (packets without header, without trailer, with two headers or with two trailers), it provokes an obstruction of the input buffers (flits that cannot be routed are not discarded);

- In case of a fault in the arbitration finite state machine (FSM), it can make the FSM go to an invalid state, interrupting the arbitration service and consequently obstructing the communication.

Based on these results, we implemented small modifications in the router design and re-performed the fault injection simulations. Analyzing the results, such small changes in the router design avoided completely the occurence of crash faults, with an area overhead of less than 5% and no performance penalty. These results are presented in (FRANTZ et al., 2007a).

3.1 Dependable Network-on-Chip Router Able to Simultaneously Tolerate Soft Errors and Crosstalk

Category: Full Paper

Conference: ITC2006: Proceedings of the IEEE International Test Conference

Location: Santa Clara - USA

Date: October, 2006

Dependable Network-on-Chip Router Able to Simultaneously Tolerate Soft Errors and Crosstalk

Arthur Pereira Frantz, Fernanda Lima Kastensmidt, Luigi Carro, Érika Cota
PPGC - Instituto de Informática
Universidade Federal do Rio Grande do Sul
Po Box 15064, ZIP 91501-970, Porto Alegre, RS, Brazil
{apfrantz, fglima, carro, erika}@inf.ufrgs.br

Abstract

As the technology scales down into deep sub-micron domain, more IP cores are integrated in the same die and new communication architectures are used to meet performance and power constraints. However, the same technologic advance makes devices and interconnects more sensitive to new types of malfunctions and failures, such as crosstalk and transient faults. This paper proposes fault tolerant techniques to protect NoC routers against the occurrence of soft errors and crosstalk at the same time, with minimum area and performance overhead. Experimental results show that a cost-effective protection alternative can be achieved by the combination of error correction codes and time redundancy techniques.

1. Introduction

The development of deep sub-micron lithography processes has allowed the integration of a huge amount of transistors in a single die. Consequently, complex system designs can be integrated into a single chip (SoC – System-on-Chip) through the reuse of pre-designed IP cores.

With more cores embedded in the system, interconnection among them has become a major bottleneck to system design. Recently, NoCs (Network-on-chip) have been proposed as an alternative communication platform capable of handling performance, energy consumption, and reusability issues of large integrated systems [1,2]. NoCs are based on the same concepts adopted by interconnection networks used in parallel computers [3], and provide high performance communication for systems with intensive communication requirements.

Another consequence of the sub-micron technologies is the increased sensitivity of the circuit to internal and external noise, which can affect the correct operation of the system [4,5]. In communication architectures for instance, delay variations and crosstalk noise have become an important issue [8]. This sensitivity is caused by the shrinking of the transistors dimensions and the reduction of the power supply voltage. Errors might come from the fabrication process (process variability), frequency increase, or from the operation environment. Radiation present in the space environment can induce faults with transient effects in circuits. Such faults are known as soft errors and the Single Event Upset (SEU) [6] is the most typical fault in this category. SEU faults are bit-flips in sequential logic elements, such as registers and memories [7].

To deal with the aforementioned problems, some work has been developed to protect on-chip interconnects against crosstalk [5,8,9,10] and sequential circuits against SEU faults [6,7,11]. The techniques proposed in the literature cope with crosstalk and SEUs independently, based on the assumption that connections cannot be affected by SEUs (no sequential circuit involved), whereas sequential circuits cannot be affected by crosstalk faults. However, when NoCs are used, buffers and sequential circuits are present in the routers, while the number of connections between any two routers can be very large. Therefore, since each fault can be originated by statistically independent events, one must consider the possibility of crosstalk and SEU faults affecting the same communication channel at the same time. As we will show in the next sections, the effect of those faults in the system operation can be disastrous, leading to loss of packets and system crash or unavailability.

This paper proposes a new technique that can simultaneously deal with SEU and crosstalk effects in NoC routers. First, a set of fault-tolerance techniques is evaluated with respect to the trade-offs between fault protection efficiency, area overhead, and performance impact. We then propose a combination of error correction codes (ECC), and hardware and time redundancy to efficiently avoid errors caused by the occurrence of SEU and crosstalk at the same time in NoC routers. To the best of our knowledge, this is the first approach that tackles the occurrence of those two faults in the NoC. Related works, which will be refereed along the paper, can only deal with the problem of crosstalk and SEU separately. Experimental results show that one can obtain zero errors under double faults, with up to 50% of savings in the area overhead, when compared to simple duplication.

The paper is organized as follows. Section 2 presents the effects of SEU and crosstalk faults on sub-micron

1-4244-0292-1/06/$20.00 © 2006 IEEE

circuits. Section 3 presents the architecture of a NoC router used for evaluation of SEU and crosstalk mitigation techniques. Section 4 presents the evaluation of four fault tolerance techniques that can be used in the NOC domain. Section 5 proposes a new scheme for the simultaneous mitigation of SEU and crosstalk faults, while Section 6 presents some experimental results. In Section 7, conclusions and future works are discussed.

2. SEU and Crosstalk Concerns

NoCs typically use the message-passing communication model. Cores attached to the network communicate by sending request and receiving response messages. To be routed by the network, a message is typically composed by a header, a payload and a trailer. The header and the trailer frame the packet and the payload carries the data being transferred. The header also carries the information needed to establish the path between the sender and the receiver. Depending on the network implementation, messages can be split into smaller structures, so called packets, which can be individually routed. Packet-based networks present a better resource utilization, since packets are shorter and reserve a smaller number of channels during transportation compared to a whole piece of message.

It is well-known that new generations of integrated systems can be susceptible to faults of transient effect like crosstalk and SEU, which are bit-flips in the memory elements. NoC systems are composed of both, memory and communication vias. NoC routers include many internal memory elements that are responsible to implement buffers, control logics and other parts of the system.

Fault injection simulations of SEU faults over a NoC router architecture have been presented in [12]. More recently, crosstalk fault injections have been also performed over the links of the same router, using the MAF (Maximal Aggressor Fault) model [13] as crosstalk model. Results of both experiments show that SEU and crosstalk faults may affect the proper router service. The effects vary from a simple loss of packet up to a permanent interruption of the router service, depending on where and when the fault has occurred. The architecture of the router used in these experiments is detailed in Section 3. For the fault injection, the router was configured to have 5 channels, buffer depth of 4 positions and 8-bits data width, resulting in 3,352 gates and 250 flip-flops.

Faults were injected separately in all router sensitive points, including links (crosstalk faults), input buffers, registers and finite state machines (SEU faults). Table 1 resumes some of the fault injection results, regarding only crosstalk faults injected on the router links and SEU faults in the input buffers (the major SEU sensitive area in the router design). These two sensitive parts were initially chosen, because in such points SEU and crosstalk faults can have their effects overlapped, leading the system to deal with multiple faults. A total of 1569 faults were randomly injected in these two sensitive parts, with 7.91% of faults representing crosstalk and 92.09% representing SEU faults.

Table 1 Fault Injection Results

Fault Effects	% of Injected Faults	
	SEU	Crosstalk
Payload Error	11.63 %	66.13 %
Single/Multiple Packet Routing Error	5.54 %	4.84 %
Router Crash	1.66 %	8.06 %
No effect	81.18 %	20.97 %

After the fault injection simulations, a number of fault effects was identified. A payload error occurs when the packet content is changed during the transmission. A single/multiple packet routing error occurs when one or more packets are routed incorrectly, leading to the loss of such packets. The router crash is a permanent effect, causing the interruption of the routing service. Such effect can be repaired only with the router reset.

One can observe in Table 1 that 79.03% of the crosstalk faults have provoked some kind of error, while 84.3% of the SEU faults have caused no effect. Most of the crosstalk faults (66.13%) and 11.63% of SEU faults have caused payload errors. This effect might be corrected by software-implemented techniques and higher-level approaches based on packet retransmission. However, more than 8% of the crosstalk and 1.66% of SEU faults caused the router crash. It is important to notice that this effect cannot be corrected by packet retransmission or software techniques, and needs the hardware reset.

Since the physical effects that cause the occurrence of SEU and crosstalk are different, both events have a probability of occurring simultaneously. In this case, the combination of SEU and crosstalk faults only makes the problem worse, because the system could be dealing with multiple faults at same time, increasing the probability of router crash errors. This is possible because the faults are independent and there is no correlation between the occurrence of crosstalk and SEU faults. It is also important to notice that different faults can compensate the effect of each other, but such probability is very low because, such fact depends on the time and location of the faults.

Due to the high probability of router crash errors, it is important to investigate fault-tolerant solutions implemented in hardware to avoid the necessity of reset operations.

Several works have addressed the problem of ensuring reliability in ICs produced with new deep sub-micron technologies. A number of solutions to cope with soft errors are reported in [6] ranging from well-known techniques, such as triple modular redundancy (TMR) and time redundancy (TR), to combined hardware and software-based approaches. In [9], another solution to protect ICs against radiation-induced errors is proposed, based on the reuse of design-for-testability and debug resources.

Mitigation of crosstalk-induced errors has also been considerably addressed in the literature. Rossi et. al. use error correction codes (ECC) to minimize the crosstalk impact in interconnect busses [8]. In the same direction, Nieuwland et. al. propose the use of coding techniques to reduce switching noise in fault tolerant busses [9]. In [10], Bertozzi et. al. investigate energy and power efficient ways of encoding data buses to deal with transient and noise induced faults. Lajolo proposes an online solution for detection and correction of faults in SoC buses [14], consisting in small detection/recovery modules distributed along the bus interconnection architecture.

Moving to the NoC domain, Tamhankar proposes a timing error tolerant communication system, named Terror, that uses pipelined links [15]. Marculescu and Dumitras propose a randomized fault-tolerant routing algorithm to cope with switch and link errors [16,17].

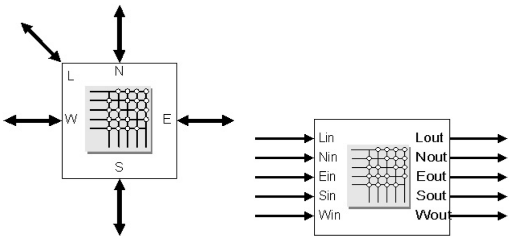
More recently, a discussion of the trade-offs between energy efficiency, error protection efficiency, and performance impact involved in various error recovery schemes is presented in [4] and [18].

Thus, although one can find in the literature a number of solutions to independently mitigate SEU faults in sequential blocks and crosstalk faults in interconnects, as previously mentioned, new fault-tolerant approaches are required to cope with the simultaneously occurrence of those two faults. Such an approach is proposed and evaluated in the sequel of this paper.

3. RASoC Architecture

We base our analysis on a packet-switching network model proposed in [19]. It is implemented in a 2-D mesh topology. The communication channels between two adjacent routers are defined to be 10-bit wide. The network is based on the RASoC (Router Architecture for Systems-on-Chip) router, which is composed of five input and five output ports, as shown in Figure 1(a). One pair of input/output ports is dedicated to the connection between the router and the core, while the remaining four pairs connect the router with the four adjacent routers, as depicted in Figure 1(b). Such ports include two unidirectional opposite channels, each one with its data, framing and flow control signals.

RASoC router is a VHDL soft-core, parameterized in three dimensions: communication channels width, input buffers depth and routing information width. It is implemented using from 3,000 to 6,000 gates, depending on the bitwidth of the network channel and depth of the input buffers [19]. Its architecture is based on the wormhole switching approach and it uses a deterministic source-based routing algorithm. Also, it applies the handshake protocol for link flow control, uses round-robin arbitration and input buffering [19].



a) external interface b) detailed interface

Figure 1 Basic structure of the RASoC router

Internally, RASoC is basically composed by instances of two types of modules: input channel and output channel [19]. The input channel is composed by four blocks named Input Flow Controller (IFC), Input Buffer (IB), Input Controller (IC) and Input Read Switch (IRS). The internal organization of the input channel module is show in Figure 2.

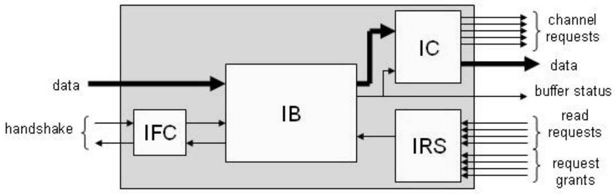


Figure 2 Input Channel module organization

The Input Flow Controller block performs the translation between the handshake and the input buffer flow control protocols. The Input Buffer block is responsible for storing flits (the flow control units) of the incoming packets while they cannot be forwarded to an output channel. The Input Controller block performs the routing function, selecting an output channel, emitting a request to the selected output channel, and, finally, updating the routing information in the header. The Input Read Switch block basically controls the read requests from the output channels to the input buffer [19].

The output channel is also composed by four blocks named Output Controller (OC), Output Data Switch (ODS), Output Read Switch (ORS) and Output Flow Controller (OFC). The internal organization of the output channel module is represented in Figure 3.

The Output Controller block runs the arbitration task over the requests emitted by the input channels. The Output Data Switch and Output Read Switch blocks connect the selected input channel to the external output channel interface. Similarly to the Input Flow Controller, the Output Flow Controller block performs the translation between the input buffer flow control and the handshake protocol.

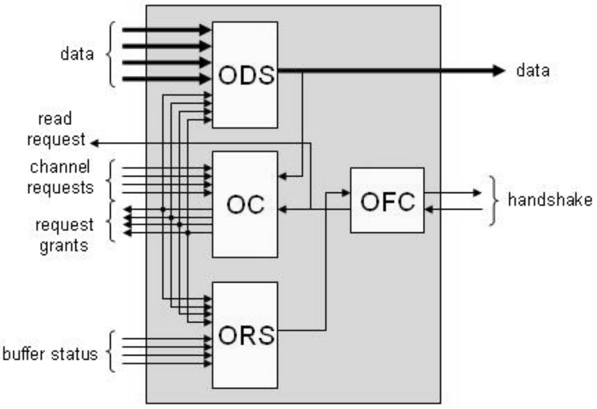


Figure 3 Output Channel module organization

Table 2 shows the RASoC configuration used in the experiments presented in this paper.

Table 2 RASoC configuration used in the experiments

Data Width	8 bits
Buffer Depth	4 Flits
Flit Size	10 bits

3.1 RASoC Sensitive Points

As previously said, SEU faults can affect sequential logic components in a circuit. The points sensitive to SEU in the RASoC architecture are located in two different design blocks: the Input Buffer block (IB) (Figure 2) and Output Controller block (OC) (Figure 3). IB block contains a FIFO buffer and a 2-bit register that stores the state of a finite state machine (FSM) that controls the buffer. The buffer is implemented as 4-deep 10-wide shift register. The internal organization of the Input Buffer block is shown in Figure 4.

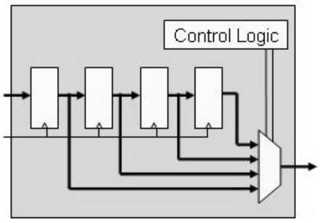


Figure 4 Input Buffer internal organization

Other blocks in the router design are composed only by combinatorial logic (not susceptible to SEUs).

As for crosstalk faults, they can affect only the links in RASoC design. However, as RASoC router uses input buffering, the first router task is to store the incoming data. Consequently, crosstalk faults can indirectly affect the data stored in the input buffer.

4. Mitigation Techniques Evaluation

As shown in Table 1, SEU and crosstalk faults can compromise the proper service of the RASoC router. To improve the reliability of the RASoC design and mitigate the SEU and crosstalk effects over the links and input buffer parts, we have implemented and evaluated four different solutions. Such solutions include the use of a single mitigation technique such as ECC and the combination of ECC and modifications in the design of the communication channels.

A brief explanation about each solution is presented below.

4.1 1st Solution – Hamming Code 1 (HC1)

The first protection alternative implements Hamming Code on the input buffers of the RASoC router. The incoming data is encoded (E) before being stored in the input buffer (B). When the data leaves the buffer, it is decoded and (possibly) corrected (D), and then follows to the routing, arbitration and switching circuitry (R), as shown in Figure 5.

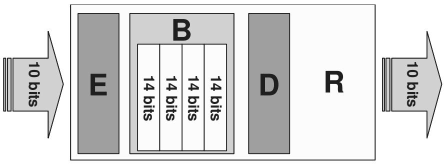


Figure 5 First solution (HC1) overview: E=encoding block, B=buffer block, D=decoding block, R= routing, arbitration and switching circuitry

To implement this solution, four bits are added to each word of the 4-deep FIFO making them 14-bit wide. In addition, the encoding/decoding logic is included in the router architecture.

Notice that this solution protects only the input buffer against SEU faults, and does not take into account the crosstalk faults in the links.

4.2 2nd Solution – Hamming Code 2 (HC2)

The second solution attempts to mitigate both the SEU and crosstalk effects using Hamming Code on links and on input buffers. An overview of the dataflow in the second solution is shown in Figure 6.

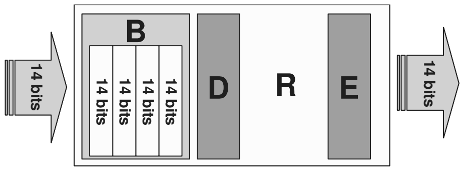


Figure 6 Second solution (HC2) overview: E=encoding block, B=buffer block, D=decoding block, R= routing, arbitration and switching circuitry

In this approach, data is encoded before being sent to the router output channel, i.e., before being transmitted to the next router in the network. Thus, single faults affecting the communication channel can be caught by the decoding block (D) when the data leaves the buffer. This means that each link has now 14 lines instead of 10 in the original design. Encoded incoming data is directly stored in the input buffer (B), whose words have also 14 bits. Before being processed by the router, the incoming data is decoded and (possibly) corrected (D), and then follows to the routing, arbitration and switching circuitry (R). Finally, if data must be re-transmitted (destination has not been reached) data is encoded (E) before it follows to an encoded link.

Although this solution can cope with single SEU and crosstalk faults in the NoC, it cannot solve the problem of multiple faults occurring to the same encoded word. Since larger parts of the circuit are now sensitive to transient faults, one must consider the possibility of simultaneous occurrence of a crosstalk in the link and a SEU in the input buffer before the data is checked by a decoding/correcting logic. Thus, another solution is proposed next.

4.3 3rd Solution – Hamming Code 3 (HC3)

Similarly to the second alternative, the third solution attempts to mitigate SEU and crosstalk effects using only Hamming Code. In this solution links and input buffers are kept encoded, but a decoding-correcting-encoding stage (D+E) is added between the link and the input buffer, as shown in Figure 7. This correcting stage detects and corrects any transient single fault present in the link before storing the corrected encoded data into the buffer. Thus, the second decoding/correcting logic in the buffer output detects and corrects any single transient fault in the buffer. Finally, the last encoding block in the router output encodes the correct data before transmitting it to the next router in the network.

Although this solution can effectively protect the NoC against both SEU and crosstalk faults, because the links are kept encoded, not only the router area is increased but also the wiring area compared to the unprotected solution. Moreover, this solution can only cope with MAF modeled faults, which cannot represent the entire set of crosstalk fault combinations. The next solution attempts to deal with the aforementioned problems while still keeping the fault tolerance efficiency.

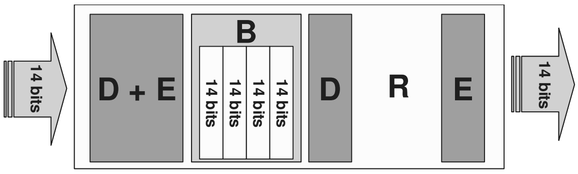


Figure 7 Third solution (HC3) overview: D+E=decoding, correcting and encoding block, B=buffer block, D=decoding block, R= routing, arbitration and switching circuitry, E=encoding block

4.4 4th Solution – Hamming Code 1 (HC1) + Terror

The fourth protection alternative uses a combination of Hamming Code and the Terror approach for the link design. The Terror approach is a timing error tolerant communication system, for aggressively designing links of NoCs. Terror proposes the use of pipelined links, where each pipeline stage has a timing error recovery mechanism [15]. The logic level implementation of Terror is presented in Figure 8.

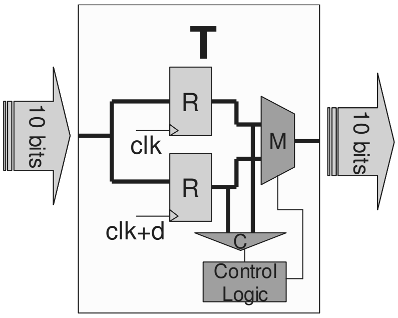


Figure 8 Terror block scheme [15]

The Terror recovery mechanism is based on the delayed sampling principle. Basically, the incoming data is sampled twice. The main pipeline register samples the incoming data at the main clock edge (clk), and a second register samples the incoming data at a delayed clock edge (clk+d). When a timing error occurs (such as a crosstalk fault) the main pipeline register has an incorrect value but the delayed pipeline register must have the right data. In case of register's mismatch, the link starts working in the delayed mode, and thus just the right data (stored in the delayed register) is passed to the next pipeline stage.

In this protection alternative each link implements only one Terror block and the input buffers are encoded using Hamming Code, as shown in Figure 9.

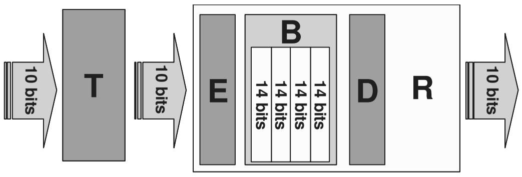
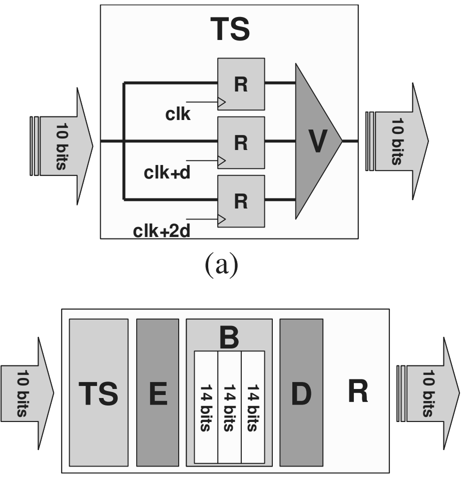


Figure 9 Fourth solution dataflow (HC1+Terror): T=Terror block, E=encoding block, B=buffer block, D=decoding block, R= routing, arbitration and switching circuitry

In this solution, as the link has only one pipeline stage (Terror block), the latency for each packet being transmitted is increased in one clock cycle. However, the frequency of the NoC is maintained. Although this solution can efficiently protect the links against crosstalk faults, the Terror block adds sensitive points to SEU - the two sample registers, which reduces the dependability of the entire solution to SEU. In the next section, a new solution is proposed to correct this problem.

5. The Proposed Solution – Hamming Code 1 + Triple Sampling (HC1+TS)

Aiming at increasing the fault tolerance efficiency, we have implemented a new protection scheme based on some principles presented above. Basically, this proposed solution uses the Hamming Code to protect the input buffers, combined with a variation of the delayed sampling principle of the Terror approach. A basic overview of the implemented sample mechanism is shown in Figure 10(a). In the proposed approach, the delayed sampling principle (TS) was implemented by using 3 sample registers. One of these registers is considered as the first buffer position, as shown in figure 10(b). The other two registers are added to the design, as defined by the Terror approach. However, since those additional registers are located within the router (and not in the channel as in Terror), they can also be protected against SEUs, as will be explained next, which maintains the system dependability. The incoming data is sampled three times (instead of twice as in the Terror approach) and a voter defines the correct data that will proceed in the router, as shown in Figure 10(a). By using the voting logic, it is possible to mitigate SEU faults in the sampling registers. After voting, the data is encoded (E) and follows to the N-1 buffer positions, for N the buffer depth. When data leaves the input buffer, it is decoded and corrected (D), and then follows to routing, arbitration and switching circuitry (R), similarly to the other protection alternatives.



(b)

Figure 10 Fifth solution (HC1+TS) overview: TS=triple sampling block, E=encoding block, B=buffer block, D=decoding block, R= routing, arbitration and switching circuitry

The HC1+TS solution can mitigate SEUs, which occur in the input buffers and in the sample registers, with simultaneously crosstalk effects in the router links. Moreover, since the TMR scheme is generic and it is implemented in each bit individually, it is possible to tolerate more than one crosstalk fault in the router link, not only one per link, as described in MAF model.

Considering the occurrence of SEU and crosstalk at the same time, which in this case means one SEU plus one crosstalk at the same clock cycle, there is a high probability of these faults being tolerated. The only possibility of these faults do not be tolerated is if these two faults occur at the same bit placed in two distinct sample registers at the same clock cycle. If the clock cycles are different, a SEU and crosstalk at the same bit can be tolerated by the method.

As one will observe in the experimental results, the proposed solution results in a better compromise between area overhead and system protection when compared to the HC1+Terror scheme. With the reuse of the first position of the input buffer as sample register, one has potentially the same area overhead of Terror approach, but a more reliable system. However, the number of flip-flops is actually reduced because there is now N-1 encoded buffer stages (against N of HC1+Terror). In addition, the number of logic gates is also reduced because the voting logic is simpler than the control logic of Terror.

6. Experimental Results and Discussion

The aforementioned solutions were implemented in the RASoC router and validated through fault injection simulations. In order to represent SEU and crosstalk faults in the router behavior, a fault injection system was implemented for each solution.

In the fault injection system SEU faults are represented as bit-flips in sequential logic components, with fault location and fault instant pseudo-randomly chosen. Such implementation does not concern the mean time between failures; in other words, it just generates one bit flip per execution, without considering the fault occurrence frequency. An example of the SEU effect caused by the hitting of a charged particle in a memory cell is illustrated in Figure 11.

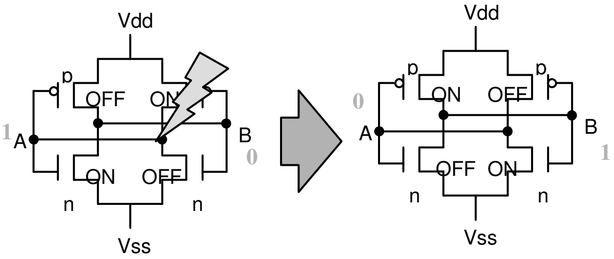


Figure 11 Illustration of the SEU effect in a memory cell

Crosstalk faults are represented as glitches and delays in the local link lines. Such representation is based on the MAF model [8] that just considers a fault as affecting only one link line, termed victim, at a time. The remaining lines are designated aggressors, and act collectively to generate an error condition on the victim. An example of the possible effects in the MAF model is shown in Figure 12.

The fault injection system also uses a pseudo-random traffic generator instead of a real application, in order to exercise in a random manner the majority of the router circuits, independently of the application that is running on the NoC. In order to evaluate the effects of the injected faults two circuits are simulated *in tandem* (the first one is fault-free and the second one is faulty). During the simulation, the results are compared, and a fault is detected each time the outputs mismatch.

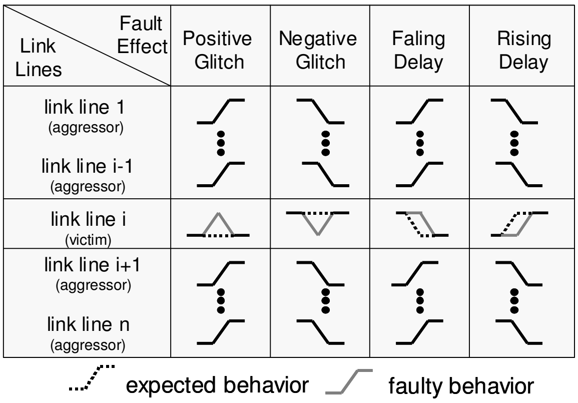


Figure 12 Possible crosstalk effects in the MAF model

After fault injection simulation, all implemented solutions have presented 100% of fault correction. It occurred due to the random characteristics of the injected faults and the traffic. Then, in order to evaluate more precisely the fault tolerance efficiency of each implemented solution, the Architectural Vulnerability Factor (AVF) metric [20] was used.

AVF measures the susceptibility of a logic structure to faults of transient effects. In practice, AVF considers the ratio between the number of sensitive bits that must tolerate upsets to ensure the proper execution, and the total number of bits in the structure. A dependable architecture should have an AVF close to 0%, which means that the probability of a fault to cause an error in the logic architecture is very low. AVF results for SEU and crosstalk faults of the solutions discussed in Sections 4 and 5 are shown in Table 3.

Table 3 AVF results for SEU and crosstalk faults

Solution	AVF %		
	SEU	Crosstalk	SEU + Crosstalk
HC1	0 %	100 %	100 %
HC2[1]	0 %	0 %	23.2 %
HC3[1]	0 %	0 %	0 %
HC1+Terror[2]	13.15 %	0 %	13.15 %
HC1+TS[2]	0 %	0 %	2.7 %

[1] able to correct only MAF crosstalk model

[2] able to correct all crosstalk models

Results from Table 3 show that solution HC1, which is based only on Hamming Code in the input buffer, presents a high dependability against SEU (AVF=0%), but a high vulnerability to crosstalk (AVF=100%). Figure 13(a) presents this AVF evaluation. Note that if there is a crosstalk at any link line, this input data is not correctly encoded and it may generate one or more wrong bits in the input data register. This is true for 100% of the crosstalk occurrence, and it does not matter if the buffer is protected or not against SEU.

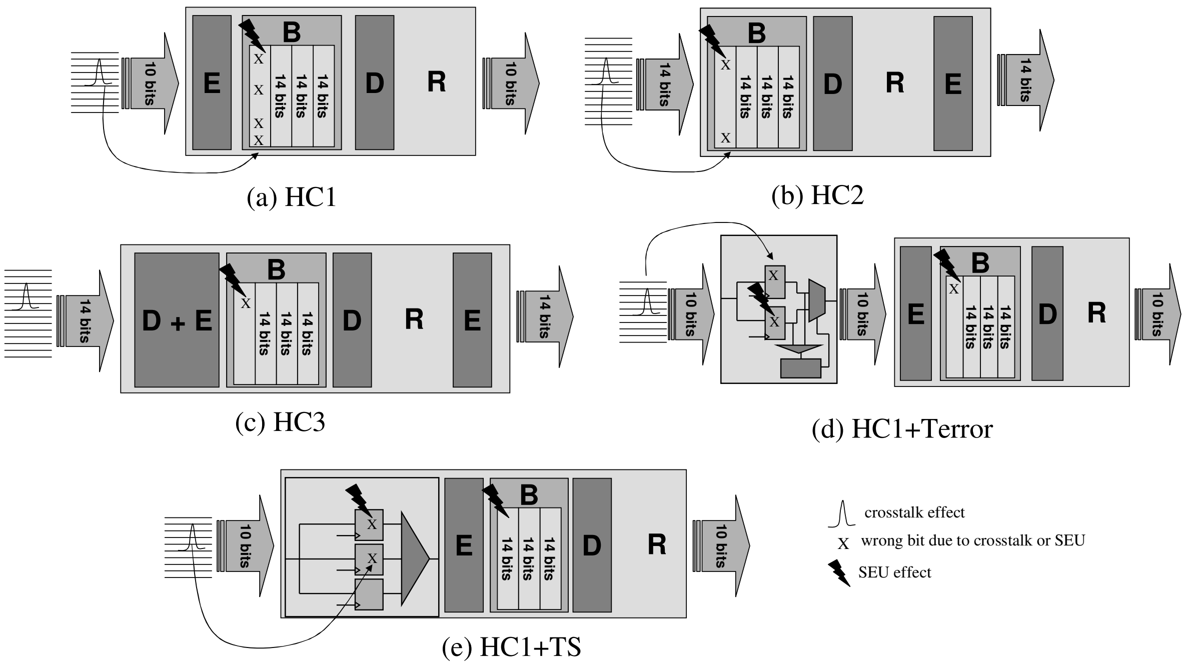
The vulnerability to crosstalk can be reduced by keeping the links encoded, which is shown in HC2 solution (AVF=23.2%). Figure 13(b) represents the AVF evaluation. When a crosstalk occurs at any link line, this transient error may be stored in one bit of the input register. If there is a SEU in one of the bits of the input register, there are 13 bit positions out of 56 that can produce an error in the fault-tolerant technique, because in this case, there are multiple faults, which overcome the Hamming Code protection.

By inserting an extra correction stage, as presented in HC3 solution, the vulnerability factor to SEU and crosstalk is reduced to 0%. This result is very good but this solution can only deal with one crosstalk per link at a time (MAF model). Note the AVF evaluation in Figure 13(c). If there is a SEU in one bit position, a crosstalk can occur at any link line and the input data will continue with the correct value because of the correcting block that avoids the propagation of the crosstalk fault. However, this correcting block cannot avoid the propagation of multiple crosstalk faults.

HC1+Terror solution is efficient to crosstalk, as previously mentioned, but it has an AVF of 13.15% to SEU due to the sample registers that are not protected. In the Terror block, if the data in the sample registers mismatch, the second register is always considered as the correct one.

Consequently, there will be an error in the method only if there is a SEU in the second sample register, which represents 10 bit positions out of 76 bit positions, as presented in Figure 13(d).

The final solution (HC1+TS) presents full reliability to SEU (AVF=0%) and a small vulnerability factor when crosstalk and SEU occurs at the same time (AVF=2.7%). This percentage comes from the ability of the TMR to vote the correct value when there are two out of three correct inputs. When a SEU and a crosstalk happen in the same bit, at the same time, and at different sample registers of the TMR, the voted output is not correct. This case is represented in Figure 13(e). Although this solution does not present an AVF for crosstalk equal to zero as the HC3 solution, it is very attractive since it can tolerate crosstalk in two or more link lines, which can occur in very deep sub-micron technologies.



(a) HC1

(b) HC2

(c) HC3

(d) HC1+Terror

(e) HC1+TS

Figure 13 AVF analysis in the studied techniques for crosstalk combined with SEU

Table 4 Area for each solution

Solution	Comb. Circuit (# gates)	#Flip-Flops (#gates)	Voter (# gates)	Hamming Enc./Dec. (# gates)	Total (# gates)	Overhead	Maximum clock frequency
No Protection	3,352	250 (1,500)	-	-	4,852	-	72.4 MHz
HC1	4,216	330 (1,980)	-	550	6,746	39.03%	57.6 MHz
HC2	4,217	330 (1,980)	-	550	6,747	39.05%	57.6 MHz
HC3	4,223	330 (1,980)	-	1,100	7,303	50.52%	57.6 MHz
HC1+Terror	4,466	430 (2,580)	-	550	7,596	56.55%	57.6 MHz
HC1+TS (proposed)	4,299	410 (2,460)	150	550	7,459	53.73%	48.7 MHz

To evaluate area and performance impact of each implemented design, each solution was synthesized and the results are presented in Table 4. Synthesis results were obtained with Leonardo Spectrum synthesis tool, using the AMI 0.35mm library. As can be seen in Table 4, solutions HC1, HC2, HC3 and HC1+Terror present an area overhead not larger than 56%, which is much less than full hardware redundancy solutions, such as TMR (minimum of 200% of area overhead). The performance penalty of these solutions is around 20%. The HC1+TS solution implies in 53% of area overhead and 32% of performance penalty.

7. Conclusions and Future Works

This paper has shown a set of fault tolerant techniques able to protect SEU and crosstalk in NoC routers. Two of the proposed solutions are able to tolerate SEU and crosstalk at the same time. In the first one (HC3), the NoC is totally dependable with respect to SEUs and single crosstalk faults at the router link (AVF=0% for both). However, to tolerate multiple crosstalk faults in addition to SEUs, which is an usual requirement, technique HC1+TS can be used (AFV=0% for SEU and AVF as low as 2.7% for crosstalk). Both techniques present up to 56% of area overhead and a low performance penalty (up to 32%) in the router. We note that the router represents a small fraction of the whole system area, thus the area overhead for the whole system is even smaller. It is also important to notice that all evaluated solutions can be used in other NoC architectures, since they use input buffering.

Future work includes the study of SEU mitigation techniques for the other sensitive parts of the RASoC router. It also includes the evaluation of the AVF for SEU and crosstalk in different NoC router architectures protected by the proposed mitigation techniques.

8. References

[1] L. Benini & G. D. Micheli, “Networks on Chips: A New SoC Paradigm”. *IEEE Computer*, Vol. 35, January, 2002, pp. 70-78.

[2] W. J. Dally & B. Towles, “Route Packets, Not Wires: On-Chip Interconnection Networks”. *Proceedings Design Automation Conference,* 2001, pp. 684-689.

[3] J. Duato, S. Yalamanchili and L. Ni, “Interconnection Networks: An Engineering Approach”. *IEEE Computer Society*, Los Alamitos, CA 1997.

[4] S. Murali, et al, “Analysis of error recovery schemes for networks on chips”, *IEEE Design &*

Test of Computers, Volume 22, Issue 5, Sept.-Oct. 2005, pp. 434-442.

[5] A. Acquaviva & A. Bogliolo, "A Bottom-Up Approach to On-Chip Signal Integrity", *Lecture Notes in Computer Science*, Volume 2799, pp. 540-549, Jan 2003.

[6] M. Nicolaidis, "Design for soft error mitigation**",** *IEEE Transactions on Device and Materials Reliability*, Volume 5, Issue 3, Sept. 2005, pp. 405-418.

[7] F. Kastensmidt, L. Carro and R. Reis, *Fault-Tolerance Techniques for SRAM-based FPGAs*, Series: Frontiers in Electronic Testing, Springer, Vol. 32, 2006. 180 p.

[8] D. Rossi, et al, "Exploiting ECC redundancy to minimize crosstalk impact", *IEEE Design & Test of Computers*, Volume 22, Issue 1, Jan 2005, pp. 59-70.

[9] A. K. Nieuwland, et al, "Coding techniques for low switching noise in fault tolerant busses", *Proceedings IEEE International On-Line Testing Symposium*, 2005, pp. 183-189.

[10] D. Bertozzi, L. Benini and G. De Micheli, "Low power error resilient encoding for on-chip data buses", *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, 2002, pp. 102-109.

[11] S. Mitra, et al, "Robust system design with built-in soft-error resilience", *IEEE Computer*, Volume 38, Issue 2, Feb. 2005, pp. 43-52.

[12] A. P. Frantz & F. G. L. Kastensmidt, "SEU Effects Evaluation on a NoC Router Architecture", *Proceedings Latin-American Test Workshop*, 2006, pp. 117-122.

[13] M. Cuviello, et al, "Fault modeling and simulation for crosstalk in system-on-chip interconnects", *Digest of Technical Papers IEEE/ACM International conference on Computer-Aided Design*, 1999, pp. 297-303.

[14] M. Lajolo, "Bus guardians: an effective solution for online detection and correction of faults affecting system-on-chip buses", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Volume 9, Issue 6, Dec. 2001, pp. 974–982.

[15] R. R. Tamhankar, S. Murali and G. De Micheli, "Performance driven reliable link design for networks on chips", *Proceedings Asia and South Pacific Design Automation Conference*, 2005, pp. 749–754, Volume 2.

[16] R. Marculescu, "Networks-on-chip: the quest for on-chip fault-tolerant communication", *Proceedings IEEE Computer Society Annual Symposium on VLSI*, 2003, pp. 8–12.

[17] T. Dumitras, S. Kerner and R. Marculescu, "Towards on-chip fault-tolerant communication", *Proceedings Asia and South Pacific Design Automation Conference*, 2003, pp. 225-232.

[18] D. Bertozzi, L. Benini and 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*, Volume 24, Issue 6, June 2005, pp. 818–831.

[19] C. A. Zeferino, A. A. Susin, "SoCIN: A Parametric and Scalable Network-on-Chip". *Proceedings 17th Symposium on Integrated Circuits and Systems (SBCCI)*, 2003, pp. 169-174.

[20] G. A. Reis, et al, "Design and Evaluation of Hybrid Fault-Detection Systems". *Proceedings 32nd International Symposium on Computer Architecture*, 2005, pp. 148–159.

3.2 Avoiding Router Crash Faults in NoCs at Design Level

Category: Extended Abstract

Conference: DATE2007: Workshop on Diagnostic Services in Network-on-Chips - Test, Debug, and On-Line Monitoring - Design, Automation and Test in Europe

Location: Nice - France

Date: April , 2007 (to Appear)

Avoiding Router Crash Faults in NoCs at Design Level

Arthur Pereira Frantz, Maico Cassel, Fernanda Lima Kastensmidt, Luigi Carro, Érika Cota

PPGC - Instituto de Informática
Universidade Federal do Rio Grande do Sul
Po Box 15064, ZIP 91501-970, Porto Alegre, RS, Brazil

{apfrantz, mcassel, fglima, carro, erika}@inf.ufrgs.br

Drastic device shrinking, high logic complexity, power supply reduction, and high operating speeds that accompany the technological evolution to nanometric technologies have reduced dramatically the reliability of deep sub-micron ICs [1]. Significant problems are related to faults of transient effect, such as soft errors induced by radiation and crosstalk faults.

Many works in the literature considers crosstalk and soft errors independently, based on the assumption that connections cannot be affected by soft errors (no sequential circuit involved), whereas sequential circuits cannot be affected by crosstalk faults. However, when NoCs are used, buffers and sequential circuits are present in the routers, while the number of connections between any two routers can be very large. Therefore, since each fault can be originated by statistically independent events, one must consider the possibility of crosstalk faults and soft errors affecting the same communication channel at the same time.

We have based our analysis on a packet-switching network model proposed in [2]. It is implemented in a 2-D mesh topology. This network is based on the RASoC router, which is composed of five input and five output ports. One pair of input/output ports is dedicated to the connection between the router and the core, while the remaining four pairs connect the router with the four adjacent routers. RASoC router is a VHDL soft-core, parameterized in three dimensions: communication channels width, input buffers depth and routing information width. It is implemented using from 3,000 to 6,000 gates, depending on the bit width of the network channel and depth of the input buffers [2]. Its architecture is based on the wormhole switching approach and it uses a deterministic source-based routing algorithm. Also, it applies the handshake protocol for link flow control, uses round-robin arbitration and input buffering [2].

As the RASoC is a parameterized architecture, in the experiments it was configured to have 5 channels, buffer depth of 4 positions and 8-bits data width, resulting in 3,352 gates and 250 flip-flops.

As previously mentioned, soft errors can affect sequential logic components in a circuit. The sensitive points in RASoC architecture are distributed over the design. The first sensitive block is the input buffer. In this block there are a FIFO buffer and a 2-bit control register (state register of a Finite State Machine).

The second sensitive point in the RASoC design is the output controller. In this block, there are two 2-bit registers and one 4-bit register. The first 2-bit register is the state register of a FSM responsible to control the arbitration mechanisms. The second one and the 4-bits register are responsible for arbitration priority control.

Regarding crosstalk faults, they can affect the links in RASoC design. As the RASoC router uses input buffering, the first router task is to store the incoming data. Consequently, crosstalk faults can indirectly affect the data stored in the input buffers.

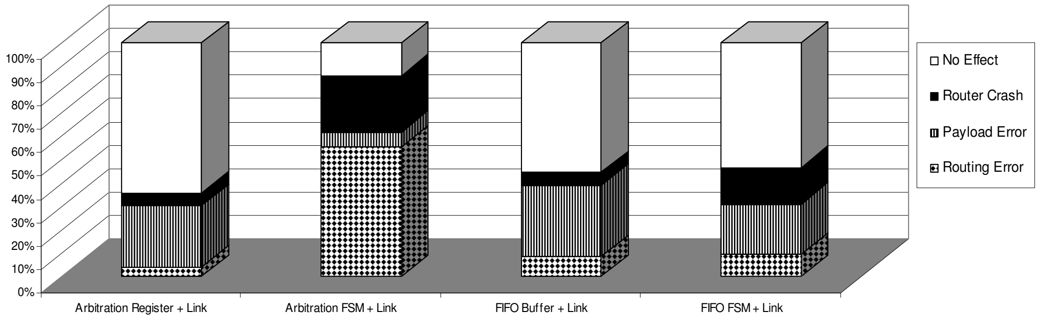


Figure 1 Fault Injection Results for the Non-Protected Version of the RASoC router

Fault injection simulations of soft errors and crosstalk faults over a NoC router architecture have been performed. Faults were injected in all router sensitive points, including links (crosstalk faults), input buffers, registers and finite state machines (soft errors). Fault location and fault time of both faults have been randomly chosen, in order to represent the independent behavior of such faults. We considered the injection of single faults (one soft error per transmission) and double faults (one soft error and one crosstalk fault per transmission).

Experimental results show that soft errors and crosstalk faults may affect the proper router service. After fault injection simulation, an exhaustive analysis was performed. The fault effects that were identified are detailed below:

- *Packet Routing Error*: Such error occurs when one or more packets are routed improperly. A packet can also suffer starvation or being overwritten inside the buffer. It is not a permanent effect.

- *Payload Error*: This error occurs when the packet payload is changed. Such error might be provoked by a simple bit flip or by increasing/decreasing the payload length. It might also produce packets without the correct framing signals (marks for begin- and end-of-packet).

- *Router Crash*: It is a permanent effect. Such error occurs when the router starts working improperly. The packets are routed incorrectly or not routed. Such effect can be repaired only with the router reset.

Considering the injected soft errors, the encountered errors were classified by the location where the faults have been injected into. Experimental results are shown in Figure 1.

Faults classified as payload error and routing error can be corrected by system level approaches. A common methodology uses parity checking and packet re-transmission. This approach does not need modification at the design level and it presents performance and power dissipation overhead in the re-transmission that varies according to the error rate due to crosstalk and soft errors. However, router crash faults cannot be corrected by this approach because this fault has a permanent effect in the router switching behavior requiring the router reset. When analyzing in more detail, a router crash occurs when the FIFO buffer is completely filled and the fault is obstructing the routing operation. Consequently, the packet is permanently stuck at FIFO, which stops the network operation in that path.

In this work, we present a technique that modifies the FIFO FSM logic to avoid router crashes. This implementation presents a small area overhead (less than 5%) and no performance penalty. It is based on a small set of logic gates, which enables the dropping of erroneous packets in the FIFO. Therefore, all router crashes can be seen as routing errors and system level techniques are able to deal with them. In order to evaluate how efficient is such technique, soft errors and crosstalk faults were re-injected in all sensitive points (sequential logic and links) with the protection logic implemented.

Figure 2 presents the new fault injection results with this implemented technique.

One can observe that with a small change in the way the packets are processed inside the router, 0% crash faults can be achieved with less than 5% area overhead, compared to traditional protection approaches, such as TMR, that have up to 200% area overhead.

References

[1] Tamhankar, R.R.; Murali, S.; De Micheli, G.; "Performance driven reliable link design for networks on chips", *In*: Asia and South Pacific Design Automation Conference, 2005. *Proceedings…* pp. 749–754, Volume 2, 18-21 Jan. 2005.

[2] Zeferino, C. A., Susin, A. A., "SoCIN: A Parametric and Scalable Network-on-Chip". *In*: 17th Symposium on Integrated Circuits and Systems (SBCCI), 2003. *Proceedings…* pp. 169-174, 2003.

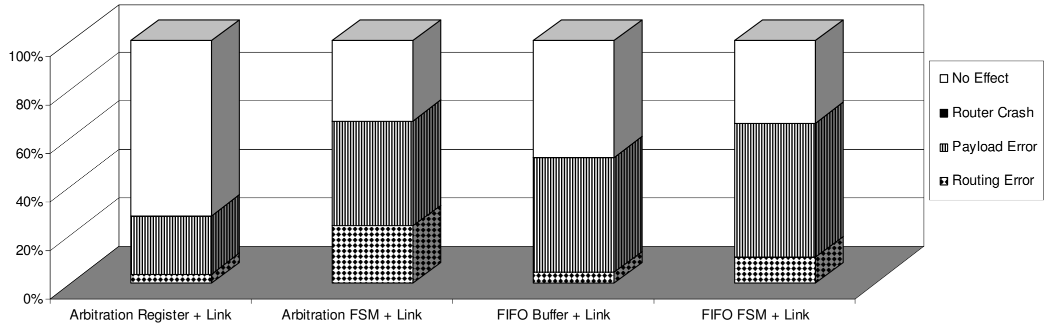


Figure 2 Fault Injection Results for the Protected Version of the RASoC router

4 MIXED HARDWARE-SOFTWARE MITIGATION SCHEMES

The previous chapter has shown a set of fault tolerant techniques applied to NoC routers able to mitigate crosstalk and soft errors at the hardware level. Such proposed techniques were based on error correcting codes and hardware redundancy. However, some of these techniques are very power consuming because all the tolerance is based on adding redundant hardware. Then, one could consider a software-based mitigation technique, whose principle of operation would be the retransmission of a message that has been detected as erroneous. But, this imposes a considerable communication overhead due to retransmission. Consequently, in these both cases, hardware and software protection, there is a penalty in terms of energy consumption.

However, the previous chapter has also shown that small modifications in the router design might avoid the occurrence of crash faults, the most critical error in a NoC router. Such modifications increase the router power consumption in less than 1%. Based on that, once the critical faults might be avoided by a small aditional hardware, the remain faults might be corrected by a software based approach. In this scenario, three mixed hardware-software solutions have been developed. Such approaches are different combinations of hardware- and software-based protection and error recovery techniques. All proposed schemes have been evaluated in terms of protection efficiency, area overhead and perfomance penalty. An energy comsumption analisys has also been done, making possible to point out which solution presents the best consumption for a given environment (error rate). These approaches and analisys are presented in (FRANTZ et al., 2007b).

4.1 Energy Efficient Mixed Hardware-Software Techniques for Routers Dependability in Presence of Crosstalk Faults in High Soft Error Rate Environments

Category: Full Paper

Magazine: Design & Test of Computer - Special Issue on Design and Test for Building Ultra High-Speed Networks

Date: July-August, 2007 (waiting for Acceptance)

Energy Efficient Mixed Hardware-Software Techniques for Routers Dependability in Presence of Crosstalk Faults in High Soft Error Rate Environments

Arthur Pereira Frantz, Maico Cassel, Fernanda Lima Kastensmidt,
Luigi Carro, Érika Cota

UFRGS, Instituto de Informática, PPGC
Av. Bento Gonçalves, 9500 Bloco IV, Porto Alegre, RS, Brazil

{apfrantz, mcassel, fglima, carro, erika}@inf.ufrgs.br

ABSTRACT

As the complexity of designs increases and the technology scales down into the deep sub-micron domain, devices and interconnections are subject to new types of malfunctions and failures. NoC routers are responsible to ensure the proper communication of on-chip cores, however, crosstalk faults and soft errors can affect the NoC service compromising the communication integrity, the performance and the energy of the entire integrated system. Fault tolerant techniques able to mitigate crosstalk faults and soft errors with a low area, performance and power consumption overhead are mandatory in future SoCs. This work presents mixed hardware-software solutions that can cope with crosstalk faults in high soft error rate environments and it evaluates the advantages and disadvantages of each one of them considering performance and energy efficiency.

Categories and Subject Descriptors

B.7.3 [**Integrated Circuits**]: Reliability and Testing
B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance.

General Terms

Design, Reliability.

Keywords

Network-on-Chip, Single-Event Upset, Crosstalk, Fault Tolerance.

1. Introduction

Networks-on-Chip (NoC) have been proposed as an alternative communication platform capable of providing interconnections and communication among on-chip cores, handling performance, energy consumption and reusability issues for large integrated systems [1, 2]. The design of large integrated systems handle drastic device shrinking, high logic complexity, power supply reduction, and high operating speeds that accompany the technological evolution to nanometric technologies. This evolution has dramatically reduced reliability [3]. Errors might be generated at the fabrication process due to process variations, or from the susceptibility of the design under a hostile environment. This paper considers two major sources of errors in NoC communication: crosstalk faults and soft errors. Delay variations and crosstalk faults have become an issue with the continuously geometry shrinking of semiconductor devices and the increasing switching speed [4]. Crosstalk affects the interconnection links and they are controlled at the layout level. Soft errors origin from the interaction of neutrons presented at the atmosphere with the material producing energized particles that can ionize the silicon substrate inducing faults with transient effects.

Single Event Upset (SEU) [5] is the most typical fault in this category. SEU faults are bit-flips in sequential logic elements, such as registers and memories.

Crosstalk faults and soft error can affect the NoC service compromising the communication integrity, the performance and the energy of the entire integrated system. As it is analyzed in this paper, soft errors and crosstalk are source-independent and they may occur in the system at the same time. The influence of these faults can be on payload, packet routing errors and even routing crashes. Related works have been investigated solutions to tolerate crosstalk faults at the hardware level in NoC interconnects. They are based on time redundancy (multiple input sampling) for detection and correction [3] and codes to reduce the data switching activity [6]. However, the concern of soft errors in the communication circuitry is very recent. In the past, it was assumed that connections cannot be affected by soft errors because there was no sequential circuit involved. However, when NoCs are used, buffers and sequential circuits are present in the routers, consequently, soft errors can occur between the communication source and destiny provoking errors. Fault tolerant techniques that once have been applied in integrated circuits in general can be used to protect routers against bit-flips. In [7], the authors have shown a set of fault tolerant techniques applied at routers able to mitigate soft errors at the hardware level. Such proposed techniques were based on error correcting codes and hardware redundancy. However some of these techniques are very power consuming because all the tolerance is based on adding redundant hardware. One could consider a software-based mitigation technique, whose principle of operation would be the retransmission of a message that has been detected as erroneous. This imposes a considerable communication overhead due to retransmission. Consequently, in these both cases, hardware and software protection, there is a penalty in terms of power consumption. Based on this scenario, the use of a mixed hardware-software technique can develop a suitable protection scheme driven by the analysis of the environment that the system will operate in (soft error rate), the design and fabrication factors (delay variations in interconnects, crosstalk enabling points), the probability of a fault generating an error in the router and communication load, and the allowed power or energy budget.

This paper proposes three approaches that combine the use of hardware and software mitigation techniques in order to simultaneously deal with soft errors and crosstalk effects in NoC routers considering minimize retransmission in present of faults and consequently ensuring performance and energy efficiency. All mixed hardware-software fault-tolerant techniques are compared to the full hardware based approach, where they are evaluated with respect to the trade-offs between fault protection efficiency, area overhead, performance impact, power consumption and energy efficiency.

2. NoC Router Susceptibility to Crosstalk and Soft Errors

Drastic device shrinking, high logic complexity, power supply reduction, and high operating speeds that accompany the technological evolution to nanometric technologies have reduced dramatically the reliability of deep sub-micron ICs [3]. Significant problems are related to soft errors induced by radiation and crosstalk faults. Crosstalk faults are due to coupling effects at the layout level [4]. Soft errors are provoked by the interaction of energetic particles with the silicon substrate, which produces an ionizing path that can charge or discharge the hit node generating a transient current pulse. The major effect is a bit-flip in a memory cell element known as Single Event Upset (SEU). Figure 1 exemplifies the two phenomena: crosstalk due to coupling effects and a substrate ionization due to a energetic particle hit.

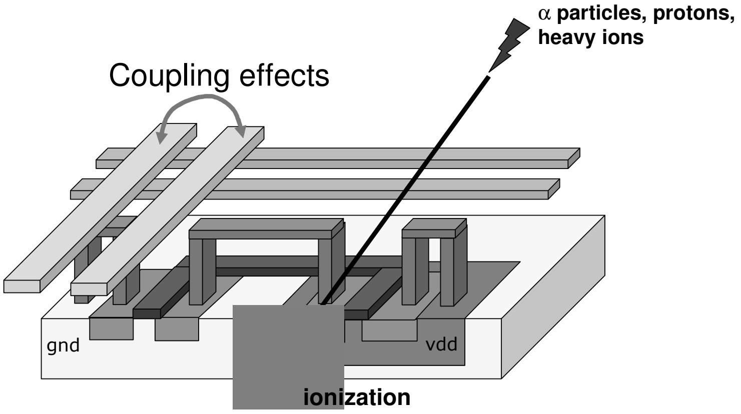


Figure 1. Soft error and crosstalk effects in integrated circuits

A NoC router behavior under crosstalk faults and soft errors has been analyzed by a fault injection system. We have based our analysis on a packet-switching network model proposed in [8]. It is implemented in a 2-D mesh topology. This network is based on the RASoC router, which is composed of five input and five output ports. One pair of input/output ports is dedicated to the connection between the router and the core, while the remaining four pairs connect the router with the four adjacent routers, as depicted in Figure 2. RASoC router is a VHDL soft-core, parameterized in three dimensions: communication channels width, input buffers depth and routing information width. Its architecture is based on the wormhole switching approach and it uses a deterministic source-based routing algorithm. Also, it applies the handshake protocol for link flow control, uses round-robin arbitration and input buffering [8]. As the RASoC is a parameterized architecture, in the experiments it was configured to have 5 channels, buffer depth of 4 positions and 8-bits data width, resulting in 3,352 gates and 250 flip-flops.

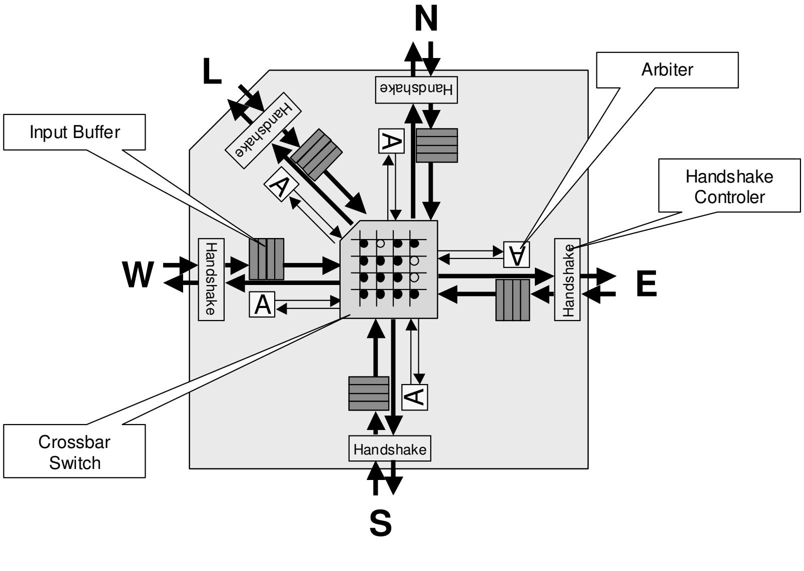


Figure 2. Basic structure of the RASoC router

A random traffic generation system is used instead of a real application aiming to exercise the router circuitry. Such approach allows the evaluation of soft errors and crosstalk effects independently of the application that is running on the NoC router giving a good average approximation. Soft errors are modeled as bit-flips in all memory cells and crosstalk faults are simulated based on the MAF model (Maximal Aggressor Fault) [9]. The MAF model considers a fault as affecting only one wire, termed victim, at a time. The remaining wires are designated aggressors, and act collectively to generate an error condition on the victim. In the MAF model, two effects may be represented: glitches, which occurs when all aggressor lines are transitioning simultaneously and the victim line stable; or delays, which occurs when all aggressor lines are also transitioning simultaneously and the victim line is transitioning in the opposite direction.

The fault generation system is able to pseudo-randomly inject single SEU faults, single crosstalk faults and a combination of a single SEU and a single crosstalk faults. It is important to observe that the execution time (time of a complete traffic simulation), the time that the faults should occur and the fault location (soft error or crosstalk) are pseudo-randomly chosen. A total fault injection simulation time of 15.2s has evaluated 18,909 soft error faults, which were injected randomly in the 250 sensitive bits, and 17,108 crosstalk faults injected on the local link (10 lines), totalizing single and double faults. Three major effects have been analyzed, as presented in table 1:

- **Single/Multiple Packet Routing Error**: Such error occurs when one or more packets are routed improperly. A packet can also suffer starvation or being overwritten inside the buffer. It is not a permanent effect.
- **Payload Error**: This error occurs when the packet payload is changed. Such error might be provoked by a simple bit flip or by increasing/decreasing the payload length. It might also produce packets without the correct framing signals (marks for begin- and end-of-packet).
- **Router Crash**: It is a permanent effect. Such error occurs when the router starts working improperly. The packets are routed incorrectly or not routed. Such effect can be repaired only with the router reset.

Table 1. Fault injection results

Fault Type	Fault Location	Fault Effects							
		Packet Routing Error		Payload Error		Router Crash		No Effect	
		#	%	#	%	#	%	#	%
SOFT ERROR	Arbitration Priority Registers	0	0.00	1	0.42	1	0.42	238	99.17
	Arbitration FSM State Register	46	38.33	3	2.50	49	40.83	22	18.33
	FIFO Buffer	80	5.54	168	11.63	24	1.66	1,173	81.18
	FIFO FSM State Register	0	0.00	18	15.00	8	6.67	94	78.33
Crosstalk	NoC Local Link	6	4.84	82	66.13	10	8.06	26	20.97
SOFT ERROR + Crosstalk	Arbitration Priority Registers + link	48	4.03	313	26.30	64	5.38	765	64.29
	Arbitration FSM State Register + link	330	55.46	38	6.39	144	24.20	83	13.95
	FIFO Buffer + link	1,264	8.66	4,447	30.45	791	5.42	8,102	55.48
	FIFO FSM State Register + link	56	9.41	129	21.68	93	15.63	317	53.28

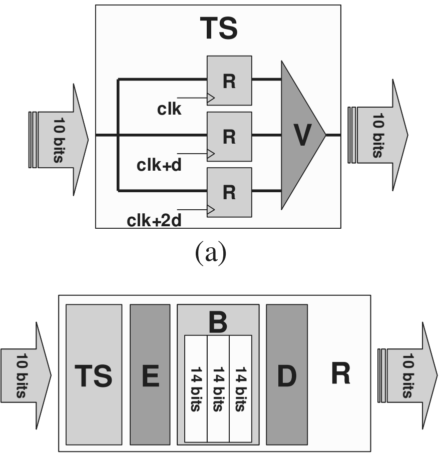
Since the physical effects that cause the occurrence of soft error and crosstalk faults are different, both events have a probability of occurring simultaneously. It is important to notice that different faults can compensate the effect of each other, but such probability is very low because, such fact depends on the time and location of the faults. Analyzing the experimental results, one can notice that the combination of soft error and crosstalk only makes the problem worse, because in that case the system can be dealing with multiple faults at same time, increasing the probability of router crash errors.

3. Fault Tolerant Router Approaches

Aiming at improving the reliability of the design and mitigate the crosstalk and soft errors effects over the links and sequential logic parts, we propose three protection approaches. Such approaches are different combinations of hardware- and software-based protection and error recovery techniques. The objective is to analyze each approach in terms of area overhead, performance penalty, power consumption and most important: the energy efficiency compared to the full hardware technique. While the full hardware based approach can mitigate all faults based on design changes but it presents a large area and power overhead, the mixed hardware-software approaches prioritize the protection of the most significant faults in order to achieve a good compromise in area, performance and power overhead, which is mandatory.

1) Full Hardware-based Approach

A full hardware protection scheme (HW-full) has been presented in [7]. Basically, this solution uses the Hamming Code to protect the input buffers against soft errors, combined with delayed sampling registers, in order to mitigate crosstalk faults. In this approach, the delayed sampling mechanism (TS) was implemented by using 3 sample registers. The incoming data is sampled three times and a voter defines the correct data that will proceed in the router, as shown in Figure 3(a). By using the voting logic, it is possible to mitigate soft errors in the sampling registers. After voting, the data is encoded (E) and follows to the N-1 buffer positions, for N the buffer depth. When data leaves the input buffer, it is decoded and corrected (D), and then follows to routing, arbitration and switching circuitry (R), as illustrated in figure 3(b). Other sensitive points in the router design (priority and FSM state registers) are protected with TMR. This technique protects the router against the occurrence of crosstalk faults and soft errors at the same time. The advantages and disadvantages of using this technique compared to mixed hardware-software solutions are evaluated in this work.



(a)

(b)

Figure 3. TS=triple sampling block, E=encoding block, B=buffer block, D=decoding block, R= routing, arbitration and switching circuitry

2) Mixed Hardware-Software based Approaches

The mixed hardware-software aims to protect the most critical parts of the router considering the application and the environment characteristics. The first mixed solution (HW-SW-basic) uses parity checking and packet retransmission in order to correct the effects caused by soft errors and crosstalk faults. In a preliminary analysis, this approach does not need modification at the design level; however, router crash faults (as detailed in the previous section) cannot be corrected by re-transmission, because this fault has a permanent effect in the router switching behavior, requiring a reset of the router. When analyzed in more detail, a router crash occurs due to two situations:

1st) When the FIFO buffer is completely full, and the fault obstructs the routing operation. Consequently, the packet is permanently stuck in FIFO, which stops the network operation in that path; or

2nd) when the fault leads any FSM to an invalid state, also it stops the correct network operation.

Therefore, to cope with router crashes minor modifications in the router architecture have been done. Such modifications enable the dropping of erroneous packets in the FIFO, and ensure safe states for all FSMs. Hence, all router crashes can be seen as routing errors, and the software-based approach is consequently able to deal with them.

The combination of this router crash avoidance at the hardware level with the re-transmission controlled by software can protect the NoC routers against soft errors. An example of the recovery mechanism used in this approach is depicted in Figure 4. Figure 4(a) shows a NoC communication, where one packet is being transmitted from node S (source) to node D (destiny), when a fault occurs (for instance a soft error), and the packet arrives to node D with a wrong payload. Using a parity checking mechanism, the error is detected and a NACK (no acknowledge) packet is sent back to node S. Then node S receives de NACK packet and retransmits the original one to node D. In case of no more errors, node D receives the right packet and then sends back an ACK (acknowledge) packet to node S, and the communication is finished. Other errors might occur during the transmission, such as the packet loss (it is incorrectly routed and never arrives to node D) or the ACK/NACK packets are erroneous or lost. In these cases, a timeout mechanism is used to detect the correct packet delivery. Figure 4 (b) shows the packet scheme using CRC and Figure 4(c) presents a communication scheme with packet retransmission every time a fault occurs.

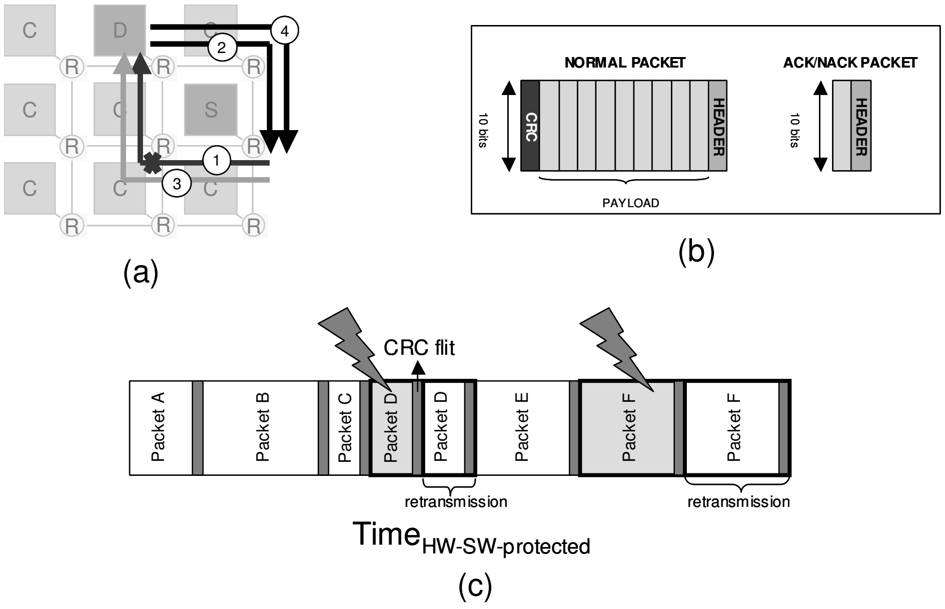


Figure 4. Example of error correction by packet retransmission

However, there are some types of faults that can be more efficiently protected in hardware and others in software. For example, crosstalk faults can not be protected by software techniques, because once there is a defect in one or more of the interconnection vias, this defect is going to behave like a permanent fault for the transmitted set of bits, affecting the communication regardless the number of performed retransmissions.

The second proposed mixed hardware-software solution (HW-SW-TS) deals with soft errors and crosstalk faults considering more then one victim wire at each time. In order to cope with more then one victim, which means more then one bit error at each flit transmission, one can use the triple sampling circuit at the input buffers of the routers. The triple sampling associated with design modifications that avoid router crashes at the hardware side and parity checking with retransmission mechanisms at the software side allows full protection of crosstalk and soft errors in NoCs. In this way, crosstalk faults and router crashes can be tolerated at the hardware level, while payload and routing errors are tolerated by retransmission. Note that the parity check helps to identify the payload errors, and the timeout mechanism assists the routing errors recognition.

However, triple sampling can be expensive in performance because three different clocks are needed and the delay of each clock must be evaluated to ensure the reliability. Solutions based on error correcting codes can reduce the performance penalty because they use only one clock domain. The third proposed mixed hardware-software solution (HW-SW-ECC) uses an encoded channel with a decoder block placed at the buffer input and an encoder block at the router output. The solution presented here uses hamming code, which corrects crosstalk faults based on the MAF model. But other correcting codes can be used able to correct more then one victim wire at each time. The HW-SW-basic, HW-SW-TS and HW-SW-ECC approaches presents communication overhead due the retransmission that varies according to the soft error rate and performance penalties according to the hardware modification at the router.

4. Experimental Results

The trade-offs in terms of area, performance and power consumption of the presented approaches were evaluated for the TSMC 0.18µm technology using the Mentor tools platform. Table 2 shows the results for the non-protected router (No Prot.), the full hardware version (HW-full), and the three mixed hardware-software solutions, the version where only router crashes are avoided in hardware (HW-SW-basic), and the ones that can deal with crosstalk by triple sampling (HW-SW-TS) and by error correcting codes (HW-SW-ECC).

The HW-full version presents a 41.41% area overhead due to the all extra logic used to protect against soft error and crosstalk, and 25.9% in performance penalty, where the delay applied in the clock are 1 ns (clk0, clk0+1ns and clk0+2ns). The HW-SW-basic version presents a very small overhead in area, less than 1%, while it can tolerate all the router crashes. However, it does not tolerate crosstalk faults. The HW-SW-basic approach also presents an improved in performance, which can be explained by the modifications of the VHDL code and the optimizations occurred during the re-synthesis flow.

Table 2. Area, frequency and power evaluation

Scheme	Protection against		Area		Frequency		Power Consumption	
	Soft error	Crosstalk fault	# gates	Overhead	MHz	Penalty	µW	Overhead
No Prot.	none	none	4935	-	226.8	-	7099.87	-
HW-full	Yes, at hardware level	Yes, at hardware level	6982	41.47%	168.1	25.9%	11629.63	63.8%
HW-SW-basic	Router crash errors protected at hardware level and payload and packet routing errors protected by CRC and time out with retransmission	none	4966	0.62%	235.0	-3.6%	7149.42	0.69%
HW-SW-TS		Yes, at hardware level	5243	6.24%	166.9	26.4%	9137.41	28.69%
HW-SW-ECC			5536	12.18%	256.1	12.91%	7537.82	3.63%

The HW-SW-TS solution has presented an equivalent performance of the full hardware version. This result was expected, because it is well-known that the triple sampling is one of the most expensive solutions in terms of performance, because it uses three flip-flops that must latch at three different instants of time, which reduces the performance of the system. The HW-SW-ECC presents the best compromise in terms of area, performance and power overhead. The error correcting logic is more efficient in terms of performance compared to the triple sampling; consequently, it presents good results compared to the HW-full solution. The use of HW-SW-TS or HW-

SW-ECC techniques is mandatory for protection against crosstalk. The choice between them depends on the NoC testing to identifies the crosstalk faults and map them in such a way that the error correcting code is able to deal with them. If the design is ensured to be crosstalk free, then the HW-SW-basic solution can be successfully used. In order to compare the fault tolerant techniques in terms of energy efficiency, it is necessary to evaluate the energy in the communication activity, which considers performance penalty, the number of flits used in the CRC protocol, the soft error rate, type of faults and number of retransmissions. The calculations of the average energy for the proposed mixed hardware-software techniques are presented in equations (1), (2), (3) and (4).

Equation (1) presents the average power consumption calculation of the each protected design. The $\%_{hardware\_overhead}$ depends on the extra hardware used by each technique (*see Table 1*). Equation (2) presents the time spent in the transmission of a packet when a CRC fault tolerant technique is applied ($Time_{CRCprotected\_packet}$), where the original time ($Time_{original\_packet}$) is increased by the number of flits used by the retransmission protocol (CRC and ACK/NACK packets), according to the packet length ($\%_{CRC\_overhead}$). The CRC parity checker is composed of a parity flit inserted at the end of the payload and packet of two flits is used at the ACK/NACK protocol. For example for a packet measuring 20 flits, with a CRC code using 1 flit and 2 flits in the ACK/NACK packet the $\%_{CRC\_overhead}$ is 15%. If the packet increases to 100 flits, the $\%_{CRC\_overhead}$ reduces to 3%. Equation (3) presents the final packet transmission time ($Time_{HW\text{-}SW\text{-}protected\_packet}$), which is also increased by the performance penalty ($\%_{freq\_penalty}$).

Equation (4) presents the average energy consumption per communication, considering that the communication time is the sum of the communication of each packet. Parameters such as the router susceptibility to soft errors and crosstalk, the soft error rate and the retransmission cost in case of fault are considered. The $\%_{faulty\_comm}$ represents the percentage of communications affected by faults, $\%_{fault\_effects}$ is the percentage of faulty communications that generate errors, which depends on the router architecture and application, and $n_{transmissions}$ is the number of retransmissions in case of a fault. The $\%_{fault\_effects}$ must be evaluated for each NoC, by fault injection for instance, and the $\%_{faulty\_comm}$ must be evaluated for each application environment and design layout. In this paper, the data presented in Table 1 is used for the calculation ($\%_{fault\_effects}$) and for the referred retransmission technique; $n_{transmissions}$ is considered to be two.

$$P_{HW\text{-}SW\_protected\_average} = P_{original\_average} + P_{hardware\_overhead} \qquad (1)$$

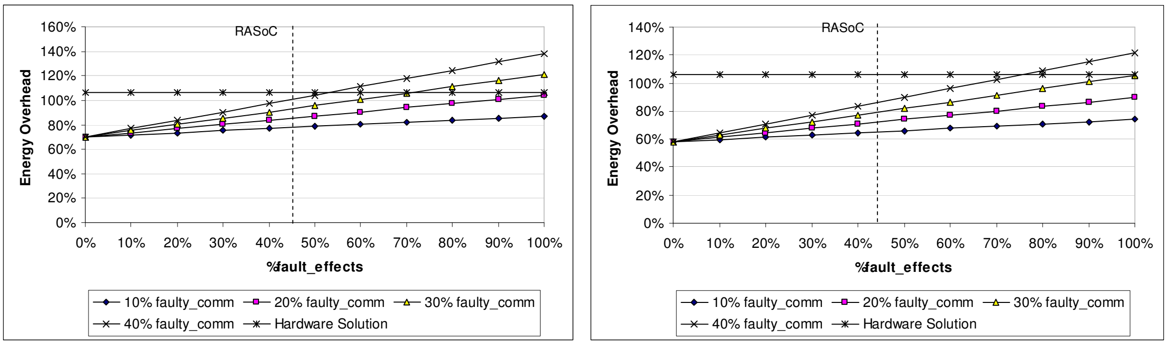
$$Time_{CRCprotected\_packet} = (Time_{original\_packet} + Time_{orignal\_packet} \times \%_{CRC\_overhead}) \qquad (2)$$

$$Time_{HW\text{-}SW\text{-}protected\_packet} = Time_{CRCprotected\_packet} + Time_{CRCprotected\_packet} \times \%_{freq\_penalty} \qquad (3)$$

$$E_{HW\text{-}SW\_protected\_\_average} = P_{protected\_average} \times \{[\Sigma(Time_{HW\text{-}SW\text{-}protected\_packet}) \times \%_{faulty\_comm} \times \%_{fault\_effects} \times n_{transmissions}] + [\Sigma(Time_{HW\text{-}SW\text{-}protected\_packet}) \times 1\text{-}(\%_{faulty\_comm} \times \%_{fault\_effects})]\} \qquad (4)$$

According to the application (packet lengths), the environment, and the router architecture, there is a design space for mitigation solutions that can simultaneously deal with soft errors and crosstalk with a minimal overhead in energy. Let one consider the first example for the HW-SW-TS approach plotted in Figure 5(a), where the dashed vertical line represents the percentage of faults that generated error in the communications ($\%_{fault\_effects}$) for the RASoC router. For the proposed techniques applied in the RASoC architecture, considering the percentage of faults that affect the normal operation ($\%_{fault\_effect}$) and a 15% for the $\%_{CRC\_overhead}$ parameter, the HW-full version can provide energy savings compared to the mixed solution only if the $\%_{faulty\_comm}$ is higher than 40%. In figure 5(b), the $\%_{CRC\_overhead}$ is changed for 3%, thanks to a larger packet size, for example, while the other parameters remain the same. In this case, the HW-SW-TS solution for the RASoC architecture presents better energy efficiency than the full hardware solution in all cases for $\%_{\text{faulty\_comm}}$ up to 40%.

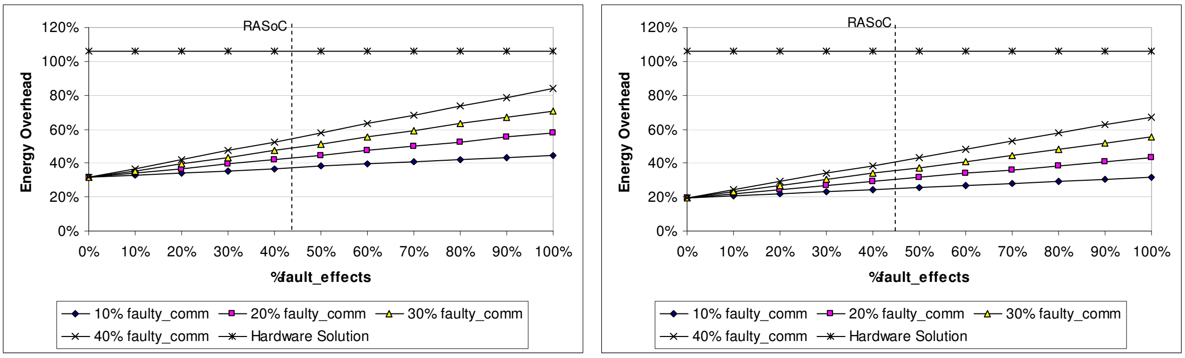
However, when considering a second example for the HW-SW-ECC approach plotted in Figure 6, where the dashed vertical line represents the percentage of faults that generated error in the communications ($\%_{\text{fault\_effects}}$) for the RASoC router as well, for both $\%_{\text{CRC\_overhead}}$ parameters of 15% or 3%, the HW-SW-ECC mixed hardware-software solution presents better energy savings in all cases. These results shows that full hardware solutions are too expensive in terms of energy and can be applied in routers with a high $\%_{\text{fault\_effects}}$ or for environments with very hostile conditions (a high $\%_{\text{faulty\_comm}}$).



(a) packets of 20 flits

(b) packets of 100 flits

Figure 5. HW-SW-TS and HW-full solutions energy overhead versus % of fault effects analysis



(a) packets of 20 flits

(b) packets of 100 flits

Figure 6. HW-SW-ECC and HW-full solutions energy overhead versus % of fault effects analysis

6. Conclusions

Mixed hardware-software techniques can be very efficient to protect NoC routers against crosstalk faults and soft errors. Solutions based on error correcting codes can present less overhead in area, performance and energy and they may make full hardware solutions such as the HW-full presented here too costly even in very high soft error rate environments. Results show that there is a design space to be explored regarding the robustness of a NoC against simultaneous occurrence of soft error and crosstalk, and that power can be saved, depending on the fault rates, router architecture susceptibility to soft errors and crosstalk, and the re-transmission cost in case of fault. So far, the susceptibility of the routers to soft errors does not justify in terms of energy efficiency the use of full hardware protection schemes because of its overhead in energy consumption and performance.

References

[1] L. Benini & G. D. Micheli, "Networks on Chips: A New SoC Paradigm". *IEEE Computer*, Vol. 35, January, 2002, pp. 70-78.

[2] W. J. Dally & B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks". *Proceedings Design Automation Conference,* 2001, pp. 684-689.

[3] R. R. Tamhankar, S. Murali and G. De Micheli, "Performance driven reliable link design for networks on chips", *Proceedings Asia and South Pacific Design Automation Conference*, 2005, pp. 749–754, Volume 2.

[4] D. Rossi, et al, "Exploiting ECC redundancy to minimize crosstalk impact", *IEEE Design & Test of Computers*, Volume 22, Issue 1, Jan 2005, pp. 59-70.

[5] M. Nicolaidis, "Design for soft error mitigation**",** *IEEE Transactions on Device and Materials Reliability*, Volume 5, Issue 3, Sept. 2005, pp. 405-418.

[6] A. K. Nieuwland, et al, "Coding techniques for low switching noise in fault tolerant busses", *Proceedings IEEE International On-Line Testing Symposium*, 2005, pp. 183-189.

[7] A. P Frantz, F. L. Kastensmidt, E. Cota, Luigi Carro, "Dependable Network-on-Chip Router Able to Simultaneously Tolerate Soft Errors and Crosstalk", *Proceedings International Test Conference (ITC)*, 2006, San Jose. IEEE, 2006. v. 1.

[8] C. A. Zeferino, A. A. Susin, "SoCIN: A Parametric and Scalable Network-on-Chip". *Proceedings 17th Symposium on Integrated Circuits and Systems (SBCCI)*, 2003, pp. 169-174.

[9] M. Cuviello, et al, "Fault modeling and simulation for crosstalk in system-on-chip interconnects", *Digest of Technical Papers IEEE/ACM International conference on Computer-Aided Design*, 1999, pp. 297-303.

5 CONCLUSIONS AND FUTURE WORKS

This work proposed the study and development of protection techniques against soft errors and crosstalk faults for application in NoC routers. Fault injection simulations proved that the effects of soft errors and crosstalk faults in a NoC router can be disastrous, ranging from a simple loss of packets to the permanent interruption of the communication service in the whole system. Another important consideration is that the combined action of soft errors and crosstalk faults increase significantly the probability of critical faults.

In this way, the problem of how to improve reliability in NoC routers at design level has been addressed in this work. A set of fault tolerance techniques for NoC routers has been proposed and evaluated. Experiments demonstrated that hardware-based schemes present a very good protection efficiency despite of the power and energy consumption overhead.

Regarding this major drawback of hardware-based techniques, the use of combined hardware-software approaches has been proposed and evaluated. Intermitent faults such as crosstalk have been addressed by the use of hardware protection techniques. In addition, transient faults, such as soft errors, have been addressed by a retransmission sofware-based approach. The proposed mixed hardware-software schemes presented excellent results in terms of power and energy consumption while mantaining a very good performance and protection efficiency. The use of full-hardware approaches is only justified in case of a very high error-prone environment.

The interest in the combined effects of crosstalk and soft errors in NoC routers is very recent and there is still a lot of work to be done in this area. Future works include the application and evaluation of the proposed mitigation techniques in other router architectures, as well as the evaluation of the fault effects in such architectures. It is also included the investigation of the impact of Single-Event Transients (SETs) in NoC routers.

REFERENCES

BERTOZZI, D.; BENINI, L. Xpipes: a network-on-chip architecture for gigascale systems-on-chip. **IEEE Circuits and Systems Magazine**, Washington, DC, USA, v.4, n.2, p.18–31, 2004.

BERTOZZI, D.; BENINI, L.; MICHELI, G. D. Low power error resilient encoding for on-chip data buses. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE AND EXHIBITION, DATE, 2002. **Proceedings...** Los Alamitos, IEEE Computer Society, 2002. p.102–109.

BERTOZZI, D.; BENINI, L.; MICHELI, G. D. Error control schemes for on-chip communication links: the energy-reliability tradeoff. **IEEE Transactions on Computer- Aided Design of Integrated Circuits and Systems**, Washington, DC, USA, v.24, n.6, p.818–831, June 2005.

BJERREGAARD, T.; MAHADEVAN, S. A survey of research and practices of Networkon- chip. **ACM Comput. Surv.**, New York, NY, USA, v.38, n.1, p.1, 2006.

DUMITRAS, T.; KERNER, S.; MARCULESCU, R. Towards on-chip fault-tolerant communication. In: ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE, ASP-DAC, 2003. **Proceedings…** [S.l.: s.n.], 2003. p.225–232.

FAVALLI, M.; METRA, C. Bus crosstalk fault-detection capabilities of error-detecting codes for on-line testing. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, Washington, DC, USA, v.7, n.3, p.392–396, Sept. 1999.

FRANTZ, A. P.; CARRO, L.; COTA, E.; KASTENSMIDT, F. L. Evaluating SEU and crosstalk effects in network-on-chip routers. In: IEEE INTERNATIONAL ON-LINE TESTING SYMPOSIUM, IOLTS, 2006. **Proceedings…** [S.l.: s.n.], 2006. 2p.

FRANTZ, A. P.; CASSEL, M.; KASTENSMIDT, F. L.; CARRO, L.; COTA, E. Avoiding Router Crash Faults in NoCs at Design Level. In: Submited for the Workshop on Diagnostic Services in Networks-on-Chip - Test, Debug and On-line Monitoring – Design, Automation and Test in Europe, 2007. 2p.

FRANTZ, A. P.; CASSEL, M.; KASTENSMIDT, F. L.; CARRO, L.; COTA, E. Energy Efficient Mixed Hardware-Software Techniques for Routers Dependability in Presence of Crosstalk Faults in High Soft Error Rate Environments. Waiting for acceptance to

IEEE Design & Test of Computers – Special Issue on Design and Test for Building Ultra High-Speed Networks, Washington, DC, USA, 2007. 10p.

FRANTZ, A. P.; KASTENSMIDT, F. L. SEU effects evaluation on a NoC router architecture. In: IEEE LATIN AMERICAN TEST WORKSHOP, LATW, 7., 2006. **Proceedings...** Porto Alegre: Evangraf, 2006. p.117–122.

FRANTZ, A. P.; KASTENSMIDT, F. L.; CARRO, L.; COTA, E. Evaluation of SEU and crosstalk effects in network-on-chip switches. In: ANNUAL SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEMS DESIGN, SBCCI, 2006. **Proceedings...** New York: ACM Press, 2006. p.202–207.

FRANTZ, A. P.; KASTENSMIDT, F. L.; CARRO, L.; COTA, E. Dependable Network-on-
Chip Router Able to Simultaneously Tolerate Soft Errors and Crosstalk. In: IEEE INTERNATIONAL TEST CONFERENCE, ITC, 37., 2006. **Proceedings...** New York: IEEE, 2006. 9p.

GRECU, C.; et al. On-line fault detection and location for NoC interconnects. In: IEEE INTERNATIONAL ON-LINE TESTING SYMPOSIUM, IOLTS, 12., 2006. **Proceedings...** [S.l.: s.n.], 2006. 6p.

LAJOLO, M. Bus guardians: an effective solution for online detection and correction of faults affecting system-on-chip buses. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, Washington, DC, USA, v.9, n.6, p.974–982, Dec. 2001.

MARCULESCU, R. Networks-on-chip: the quest for on-chip fault-tolerant communication.
In: IEEE COMPUTER SOCIETY ANNUAL SYMPOSIUM ON VLSI, ISVLSI, 2003. Tampa, Florida, USA. **New Trends and Technologies for VLSI Systems Design**: proceedings. Los Alamitos: IEEE Computer Society, 2003. p.8–12.

MURALI, S.; et al. Analysis of error recovery schemes for networks on chips. **IEEE Design & Test of Computers**, Washington, DC, USA, v.22, n.5, p.434–442, Sept./Oct. 2005.

NICOLAIDIS, M. Design for soft error mitigation. **IEEE Transactions on Device and Materials Reliability**, Washington, DC, USA, v.5, n.3, p.405– 418, Sept. 2005.

NIEUWLAND, A. K.; et al. Coding techniques for low switching noise in fault tolerant busses. In: IEEE INTERNATIONAL ON-LINE TESTING SYMPOSIUM, IOLTS, 11., 2005. **Proceedings...** [S.l.: s.n.], 2005. p.183–189.

PARK, D.; et al. Exploring Fault-Tolerant Network-on-Chip Architectures. In: INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS, DSN, 2006, Washington, DC, USA. **Proceedings...** Los Alamitos: IEEE Computer Society, 2006. p.93– 104.

ROSSI, D.; et al. New ECC for crosstalk impact minimization. **IEEE Design & Test of Computers**, Washington, DC, USA, v.22, n.4, p.340–348, July/Aug. 2005.

ROSSI, D.; et al. Exploiting ECC redundancy to minimize crosstalk impact. **IEEE Design & Test of Computers**, Washington, DC, USA, v.22, n.1, p.59–70, Jan./Feb. 2005.

TAMHANKAR, R. R.; MURALI, S.; MICHELI, G. D. Performance driven reliable link design for networks on chips. In: ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE ASP-DAC, 10., 2005. **Proceedings…** Piscataway, NJ, 2005. p.749–754.

ZEFERINO, C. A.; KREUTZ, M. E.; SUSIN, A. A. RASoC: a router soft-core for networks-on-chip. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE, DATE, 2004,Washington, DC, USA. **Proceedings…** Los Alamitos: IEEE Computer Society, 2004.

APPENDIX A PROJETO DE NOCS TOLERANTES A FALHAS PARA O AUMENTO DA CONFIABILIDADE EM SOCS

A integração de um sistema completo no mesmo *chip* tornou-se possível como uma conseqüência do aumento das densidades de integração disponibilizadas pelas tecnologias sub-micrônicas de litografia e pelos requisitos computacionais das aplicações mais agressivas nos domínios multimídia, automotivo e de ambientes inteligentes (BERTOZZI; BENINI, 2004).

Tais sistemas, chamados *Systems-on-Chip* (SoCs), representam produtos semicondutores de alta complexidade que incorporam blocos de múltiplas fontes, em particular, processadores de propósito geral, DSPs, aceleradores de hardware dedicados, memórias, blocos de E/S, etc. Consequentemente, o desempenho dos SoCs com centenas de blocos é limitado pela capacidade de se interconectar eficientemente diferentes blocos funcionais e acomodar seus requisitos de comunicação (BERTOZZI; BENINI, 2004).

Com o objetivo de ultrapassar tais desafios, uma nova infra-estrutura de comunicação, chamada *Network-on-Chip* (NoC), foi proposta. Esta abordagem de comunicação tira vantagem de arquiteturas de comunicação bem conhecidas, oferecendo escalabilidade e reusabilidade das conexões multiponto e o paralelismo e curtas conexões das conexões ponto-a-ponto (BJERREGAARD; MAHADEVAN, 2006).

Por outro lado, as mesmas evoluções tecnológicas para os processos nanométricos reduziram drasticamente a confiabilidade dos circuitos integrados produzidos em massa, tornando dispositivos e interconexões mais sensíveis a novos tipos de falhas. Tais erros podem ser gerados durante o processo de fabricação, devido às variações do processo, ou mesmo pela susceptibilidade de um projeto sob um ambiente hostil.

A crescente velocidade e diminuição dos tamanhos dos novos circuitos integrados tem trazido atenção para modelos de falhas não-convencionais, tais como aqueles devido a acoplamento capacitivo e indutivo entre fios de metal dentro dos circuitos. Tais falhas são comumente referenciadas como falhas de *crosstalk* e podem resultar em sinais incorretos dentro do circuito quando um ou mais sinais acoplados mudam.

Outro tipo de erro são os *soft errors*, que são falhas transientes provocadas pela interação de partículas energéticas com o silício, produzindo um caminho ionizado que pode carregar ou descarregar o nodo afetado gerando um pulso transiente de corrente. O maior efeito é um *bit-flip* em um elemento de memória também conhecido como *Single Event Upset* (SEU).

Na literatura, muitos trabalhos atacaram o problema da confiabilidade em CIs produzidos com nas novas tecnologias sub-micrônicas. Considerando as falhas intermitentes como o *crosstalk*, uma grande variedade de soluções foi proposta. Muitas dessas soluções foram projetadas para serem usadas em barramentos intra-chip, porém também podem ser aplicadas em arquiteturas de NoCs.

Entretanto, embora possa ser encontrado na literatura uma grande quantidade de soluções para tratar independentemente *soft errors* em blocos sequenciais e *crosstalk* em interconexões, novas abordagens de tolerância a falhas são necessárias para lidas com a ocorrência simultânea de tais falhas.

Com o objetivo de desenvolver formas eficientes de evitar a ocorrência simultânea de *soft errors* and falhas de *crosstalk*, este trabalho foi dividido em três estágios principais. Primeiramente, uma avaliação muito detalhada dos efeitos de *soft errors* e falhas de *crosstalk* sobre cada parte de um roteador de uma NoC foi realizada. Então, baseado na avaliação dos efeitos, novas abordagens de hardware que simultaneamente tratam dos *soft errors* e falhas de *crosstalk* em roteadores de NoCs foram desenvolvidos e avaliados. Após a avaliação das técnicas de hardware, foi observado que abordagens baseadas somente em hardware consomem muita potência, e então abordagens mistas de hardware e software foram também projetadas e avaliadas.