Volume : 3 | Issue : 5 | May 2014 • ISSN No 2277 - 8179

Research Paper

Architectures of Fault-Tolerant Network Interfaces and Router For Network-On-Chip

Engineering

KEYWORDS : Network-on-chip;Network Interface;ECC; fault-tolerance

Anusree S.

PG Scholar, Department of Electronics and Communication Engineering, Vedavyasa Institute of Technology, Malappuram, india.

ABSTRACT

This paper introduces a Fault-Tolerant Network Interfaces and Router for Network on Chip. Networks-on-Chip which constitute the interconnection architecture for the complex designs, may undergo malfuctions and failures due to the technology scaling and complexity. This can be overcome by adopting a functional fault model for the components of the NI and Router in the Network on Chip.The proposed Fault-Tolerant solution can be used to mitigate both temporary and permanent faults in the NI. This paper proposing a new architecture to the NI components,FIFOs and Look Up Tables, which are most sensitive to faults.This architectural solution can also provide a saving of up to 48% in the area overhead, as well as a significant energy reduction, with respect to an alternative standard Triple Modular Redundancy implementation of the NI, while maintaining a similar level of robustness to faults.

I. INTRODUCTION

Networks-on- Chip (NoCs) appeared as a communication system to connect and manage the communication between IP blocks in complex System-on-Chips (SoCs). It will provide reduced hardware overhead, better scalability, and higher data throughput compared to bus subsystems. Use of the NoCs instead of Soc buses will follow the same path of data communications and will reduces SoC manufacturing cost, SoC time to market, SoC time to volume, and SoC design risk or increases SoC performance. According to that, new types of malfunctions and failures in devices and interconnect of new complex designs composed of a high number of heterogeneous Intellectual Propriety (IP) cores, and connected by means of Networks-on-Chip (NoCs) will arise due to the scaling of the CMOS technology which will be hard to predict and to avoid with the current design methodologies. In order to deal with faults in such complex systems, new fault-tolerant approaches and architectural solutions are needed.

Typical NoC architecture consists of computational processing elements (PEs), Network interfaces (NIs) and routers.When packet sent from source PE to destination PE, packet forwarded on network depends on decision made by each router It indicates that fault-tolerance of Noc is mainly based on the architecures of NIs and routers. To deal with permanent and temporary faults in the links and in the router architecture several fault-tolerant solutions have been proposed.

Among these only few were dealing with the fault tolerance of Network Interfaces. NIs are mainly used to connect the IP cores to the overall system through the communication system.They represent a major role to create a fault tolerat NoC. Faults in these may cause the incorrect transmission of data and control information which cannot be detect easily and it may lead to the isolation of the working core from the rest of the system. So the design of the fault tolerant NI and router are very important for an efficient NoC. Usually fault-tolerant hardware implementation of sensitive components is based on triple module redundancy (TMR), where the three copies of the same component perform the same operation and the single output result is obtained by a voting system, which is expensive in terms of the amount of resources and energy needed. Especially in the case NIs and routers which often represent a significant part of the area of the overall communication subsystem, extensive use of hardware redundancy may not be economically viable.

In this paper proposed architecture is mainly based on the components of NIs and router which are very sensitive to faults.NI components that are most sensitive to faults are Look Up Table and FIFOs. Router design consists of FIFO buffer to provide temporary storage of packets that are in transit. The goal of this proposed work is to evaluate architectural solutions that can be make the NIs and router resistant to both permanent and temporary faults. The proposed architectural solutions for the NIs and router required only limited amount of redundancy to overcome the effects of both permanent and temporary faults. It mainly concentrating in the fault tolerance of the NIs and router by introducing a new architectural solution.

II. EXISTING SYSTEM

The current VLSI technology can support an extensive integration of transistors for the efficient implementation of the applications. So the number of computing resources in single-chip has enormously increased to meet the growing computation-intensive applications and the needs of low-power, high-performance systems. But as the addition of many computing resources such as CPU, DSP, specific IPs, etc to build a system in System-on-Chip increases, then its interconnection between each other becomes another challenging issue. because of its low-cost and simple control characteristics a shared bus interconnection which needs an arbitration logic to serialize several bus access requests, is adopted to communicate with each integrated processing unit in most System-on-Chip applications. However, limitation in its scalability is an drawback of the shared bus interconnection because the bus accesses should be serialized and only one master at a time can utilize the bus. So some other interconnection methods should arise in the environment where the number of bus requesters is large and their required bandwidth for interconnection is more than the current bus.

The use of on-chip packet-switched micro-network of interconnects, generally known as Network-on-Chip (NoC) architecture can be satisfied such scalable bandwidth requirement. The basic idea came from traditional large-scale multi-processors and distributed computing networks. Due to the scalable and modular nature of NoCs and their support for efficient on-chip communication, efficient NoC-based system implementations are emerged. For integrating intellectual property (IP) cores with diverse communication requirements, system on chip (SoC) based designs requires standardized interfaces to make the network on chip (NoC) as a communication backbone. These Network Interfaces have to be simple and generic for implementation with minimal overhead.

But their complicated configurations and implementation complexity make it hard to be adopted as an on-chip interconnection methodology even though the current network technologies are well developed and their supporting features are excellent. Faults in them lead to failure of entire system.In order to meet typical SoCs or multi-core processing environment, basic module of network interconnection like switching logic, routing algorithm and its packet definition should be fault tolerant to result in easily implemental solutions. Usually fault-tolerant hardware implementation of sensitive components is based on triple module redundancy (TMR), where the three copies of the same component perform the same operation and the single output result is obtained by a voting system, which is expensive in terms of the amount of resources and energy needed.

Volume : 3 | Issue : 5 | May 2014 • ISSN No 2277 - 8179

Fault-tolerant solutions are proposed to mitigate transmission errors due to cross-talk, electromagnetic radiations, or alpha particles where the fault-tolerance of NoC-based systems has been addressed by a significant amount of research effort. These solutions are mainly based on the use of error detecting and correcting codes and retransmission. By exploiting the intrinsic redundancy of NoC paths for providing alternatives to faulty links or faulty components in routers, architectural solutions have been studied for increasing fault-tolerance in routers and the NoC.The permanent and temporary errors in the NI has to detect to make it fault free. In the previous works about NI were mainly dealing with the permanent faults in the link connecting the core to the NI. In this paper we address not only permanent faults in the link connecting the core to the NI, but we propose a solution able to deal with both permanent and temporary faults in all the main architectural elements of the NI.

III. SECDED CODE

As it's name indicates Single Error Correction and Double Error Detection it is an Error Correcting Code (ECC). This means that the symbol alphabet consists of just two symbols (which we denote 0 and 1), that the receiver can correct a transmission error without asking the sender for more information or for a retransmission, and that the transmissions consist of a sequence of fixed length blocks, called *code words*. This is based on an extended Hamming Code. Hamming's development is a very direct construction of a code that permits correcting single-bit errors. He assumes that the data to be transmitted consists of a certain number of *information bits* ***u***, and he adds to these a number of *check bits* ***p*** such that if a block is received that has at most one bit in error, then ***p*** identifies the bit that is in error (which may be one of the check bits). Specifically, in Hamming's code ***p*** is interpreted as an integer which is 0 if no error occurred, and otherwise is the 1-origined index of the bit that is in error.

But for many applications a single error correcting code would be considered unsatisfactory and leads to the development of a SEC-DED code which seems safer by adding one check bit, which is parity bit on all the bits in the SEC, to the Hamming code where it is known as extended Hamming Code. Consider the table.1 , It is assumed *a* priority that either 0, 1, or 2 transmission errors occur. As indicated in the table, if there are no errors, the overall parity (the parity of the entire *n*-bit received code word) will be even, and the syndrome of the -bit SEC portion of the block will be 0. If there is one error, then the overall parity of the received block will be odd. If the error occurred in the overall parity bit, then the syndrome will be 0. If the error occurred in some other bit, then the syndrome will be nonzero and it will indicate which bit is in error. If there are two errors, then the overall parity of the received block will be even. If one of the two errors is in the overall parity bit, then the other is in the SEC portion of the block. In this case the syndrome will be nonzero (and will indicate the bit in the SEC portion that is in error). If the errors are both in the SEC portion of the block, then the syndrome will also be nonzero, although the reason is a bit hard to explain.

Table.1. Adding a parity bit to make a SEC-DED code

Syndro me	Overall Parity Type (P5)	Error Type	Notes
0	0	No error	
!=0	1	Single Error	Correctable
!=0	0	Double Error	Not Correctable
0	1	Parity Error	P5 is in error, correctable

Table.2. Extended Hamming code data and parity bits

BIT POSITION	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BIT NUMBER	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA/PARITY BIT	P5	D 15	D 14	D 13	D 12	D 11	P4	D 10	D9	D8	D7	D6	D5	D4	P3	D3	D2	D1	P2	D0	P1	P0

The parity bits P0-P4 are created for single error detection and correction and are created as follows.

P0 = D15⊕D13 ⊕D11 ⊕ D10 ⊕D8 ⊕D6 ⊕D4⊕D3⊕D1⊕D0
P1 = D13 ⊕ D12 ⊕ D10 ⊕ D9 ⊕ D6⊕D5⊕D3⊕D2⊕D0
P2= D15 ⊕ D14 ⊕ D10 ⊕ D9 ⊕ D8 ⊕ D7⊕D3⊕D2⊕D1
P3= D10 ⊕ D9 ⊕ D8 ⊕ D7 ⊕ D6 ⊕ D5⊕D4
P4 = D15⊕ D14 ⊕ D13 ⊕ D12 ⊕ D11

The block diagram of the SECDED which is used in the architecturec of LUT and FIFO is shown in the below figure 1.For the 22 bit input applied to the block diagram, output is a16 bit data and the 2 bit to indicate type of error in the data. If the error in the data is single then it will detect and correct by using the syndrome generated. But for the double error it can detect the error occured but can't correct them.

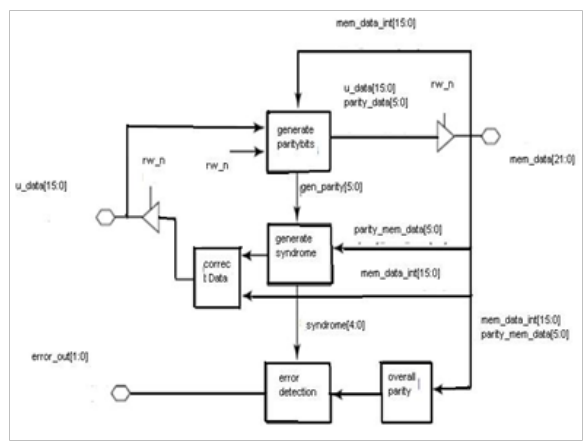


Fig.1. SECDED Block Diagram

IV. PROPOSED SYSTEM

As a solution to the interconnect problem, Networks-On-Chip (NoC) have been proposed for highly complex chips. NoCs help designing chips in several ways. That are

- in deep submicron technologies replace the wires
- complexity decreases by sharing
- limitation in scalability of wires can overcome
- can be energy efficient and reliable
- computation from communication can be decouple through welldefined interfaces, enabling IP modules and interconnect to be designed in isolation, and to be integrated and reused more easily

Networks consist of routers, which transport the data from one place to another, and network interfaces (NIs), which implement the interface to the IP modules.As the size of IP modules attached to the NoC is relatively small, on-chip NIs must provide a low area overhead. The important features of our NI are it's low area requirement ., we must provide a smooth transition from buses to NoCs to enable the reuse of existing IP modules.

The goal of the proposed work is to deal with both the permanent and temporary fault by providing a new architecture to NI components FIFOs and Look Up Table which are sensitive to faults. The fault free NI which assumes a particular relevance in the design of a reliable MPSoC, includes a front-end and a back-end sub-modules. The communication protocol adopted by the core will be implemented by the front end. The back-end module is in charge of implementing basic communication services,

such as data packetization, and routing and control flow related functions. Moreover, additional services, such as link error detection and error recovering strategies, transaction ordering, support for cache coherence and security, can also be implemented. In this paper, we are dealing with an NI providing basic communication services.

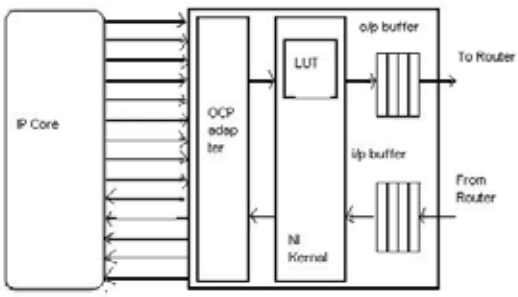


Fig. 2. Overview of the reference NI architecture considered in the experiments.

The NI considered here is an independent hardware block located between the core and the communication infrastructure. The main NI components are:

- An Open Core Protocol (OCP) adapter implements the OCP protocol which is acting as initiators at cores where the adapter implements a Slave interface and it implements a Master interface at target cores.
- The NI kernel receives and transmits data and control information from/to the adapter, packetizes and de-packetizes messages, schedules and inserts packets in the output FIFO buffer, retrieves them from the input FIFO buffer, and implements the control flow mechanism;
- The function of the output FIFO buffer is to store packets ready to be inserted into the NoC, and the input FIFO buffer stores incoming packets.

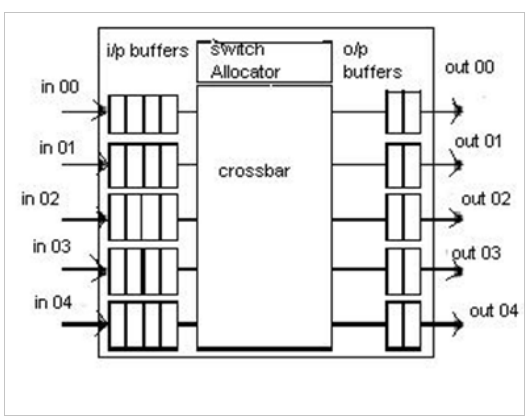


Fig. 3. Overview of the reference router architecture considered in the experiments.

Router is a main element of NoC to connect between the IP cores .It consists of FIFO buffer to provide temporary storage of packets which are in transit. Crossbar switch provide full connectivity between all available links and it routes data from input channel to output channel depends on routing decision make by control logic. High speed is achieved by allowing routing function to each input port, which gives high level of parallelism.

The NoC working is based on a wormhole flow-control, and a source based routing policy.The elements on the NoC are memory mapped and the implements a transaction-based communication on a shared-memory abstraction.At the time of new transaction which is requested by the processing element, the NI looks up the memory-mapped address of the OCP transaction by employing a programmable lookup table (LUT), located in the NI kernel. Then the LUT returns a sequence of bits which codes the path used by the packets to reach the destination node in the NoC. This routing information is inserted into the packet header, and its length depends on the dimension of the NoC and on its topology. At each router encountered along the path, a few bits of the sequence are employed for requesting the desired output port. When the output port is granted, such bits are discarded, and the header of the packet is updated. The LUT which programmable can store the information in the LUT and also can rewritten to support run-time modifications of the routing paths.

V. SYSTEM DESIGN AND IMPLEMENTATION

The main function of the Network Interface module in NoC is to convert the data to and from the format required by core Infrastructure. In this fault model, errors are mainly due to faults concerning the Lookup Table (LUT) and FIFOs which are considering as very sensitive to faults. Because of this we are mainly concentrating in these components. These NI components are mainly composed of memory cells (SRAM or Flip-Flops) and they are particularly affected by both permanent and temporary faults. The architectures which we proposed for fault tolerance are based on the use of Error Correcting and Detecting codes, in combination with limited redundancy, and a limited use of Triple Modular Redundancy (TMR). the term 'by using TMR' repeating twice. Please delete the last term and make the sentence as 'By using TMR the rest of the logic and components of the NI are implemented.'

A. Look Up Table

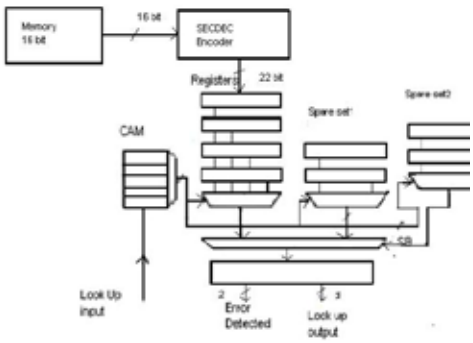
Figure 4 shows the architecture of the LUT which is proposed for increasing the fault-tolerance. As baseline architecture, we consider a LUT implemented as a combination of a non programmable Content-Addressable Memory (CAM) and either a RAM or a set of registers . Without loss of generality, in this work we refer to a register-based implementation. The CAM contains hard-coded the address boundaries of the memorymapped IP cores of the NoC. When initiating a new transaction, the most significant bits (MSBs) of the operation address are compared with the values coded into the lines of the CAM. The position of the CAM line matching the input address is used to select the register in which is stored the output of the lookup operation, i.e., the routing path to reach the destination node mapped to the input address.

In this we implemented a two level approach which employs Error Correcting and Detecting Codes and a limited amount of architectural redundancy which will allow us to deal with both temporary and permanent faults in the LUT. A Single Error Correcting and Double Error Detecting (SECDED) Hsiao code that is able to correct up to one error and detect up to two errors are used to store path information in each LUT register. when writing the register the Hsiao encoder encodes the information and a decoder decodes it after lookup as shown in the figure 4.

Because of the uniform distribution of the XORs in the implementation of the encoder and the decoder in the Hsiao code, the number of levels of logic ports and the overall delay of the modules can be reduce. The Error Correcting Code (ECC) corrects single-bit errors, due to transient or permanent faults. However, an error caused by a permanent fault will recur every time the bad cell of the register is used, and, as faults accumulate, the device eventually becomes unusable. The design of a certain number of spare registers that are meant to substitute LUT registers provide architectural redundancy to the LUT. These spare registers are of critical importance because a defective LUT register will cause an entire core not to be reachable from that NI. A bit in a status register specifies whether a specific register of the LUT is working or faulty, by selecting either the regular register or the spare register. Spare registers are simply addressed through the least significant bits (LSBs) of the addressing signals. We implemented the status register, as well as all the control logic, by using TMR.

Volume : 3 | Issue : 5 | May 2014 • ISSN No 2277 - 8179

Fig.4. Overview of the proposed LUT architecture.



At the time of applying LookUp input to the CAM the output will be generated from the CAM which can fetch those data from the stored register by comparing the MSBs of the operation address with the value coded into the lines of the CAM. These data will be multiplexed to the SECDED decoder and performs the operation to find the fault and to correct the data. If the error is single error,then the corrected data will be the output. If it is double error, it will indicate as error detected but will not correct the error.

By detecting and correcting the degradation of the routing information both in the case of permanent and temporary faults the presented LUT architecture allows the NI to be protected from Routing Path Errors. Routing Path Errors can also avoided by protecting the control registers of the LUT by implementing them in TMR which will reduce the probability of selecting the wrong routing path register associated to an input destination address.

B. FIFOs

Decoupling of the computation in IP cores from the communication operations is performed by the FIFOs in NIs. This proposed architecture is a register-based synchronous FIFO circular buffer, whose datapath is equal to the flit dimension (data and control signals). From the figure 5 it is clear that in addition to the storage elements, logic is needed for managing the pointer to the element to be extracted (read pointer), the pointer to the first available position in the FIFO (write pointer), and for implementing control signals notifying whether the FIFO is full (Full) and whether at least one element is present in the FIFO (Exists). The read and write pointers are implemented as counters which are updated depending on the write or read operation performed on the FIFOs.

Similarly to the solution discussed for the LUT, the presented architecture employs a two-level approach. Information in the FIFO is encoded, one flit at the time, by using a SECDED Hsiao code.The SECDED encoder encodes the 16 bit data to 22 bit by generating the parity bits and store it in the registers.

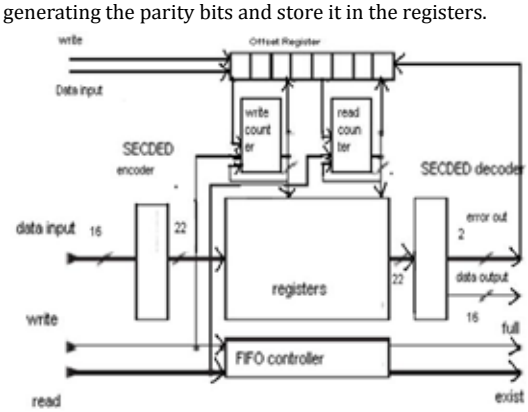


Fig.5. Overview of the proposed FIFO architecture.

While reading the data the decoder decodes it to 16 bit data and finds the error if any. While decoding if any error occured, it will provide the corrected output if it is single error and occurence of error will be indicated if the error is double error. The proposed architecture protects the NI from the corruption of the information stored into the FIFO. Functional run-time errors such as Corrupt Data Errors, Routing Path Errors, and Control Flow Errors, due to faults in the FIFO slots storing, respectively, body flits, header flits, and control flow information, are avoided or significantly reduced.

VI. SIMULATION RESULTS

The main building block of the architectures of FIFOs and LUT are SECDED decoder and encoder which can perform the detection and correction of the single errors and detection of the double errors in the NI components. The FIFO and LUT architecture is fault tolerant due to the use of these. The simulated output of these architectures from the Xilinx ISE 14.4 are shown in the below figures.

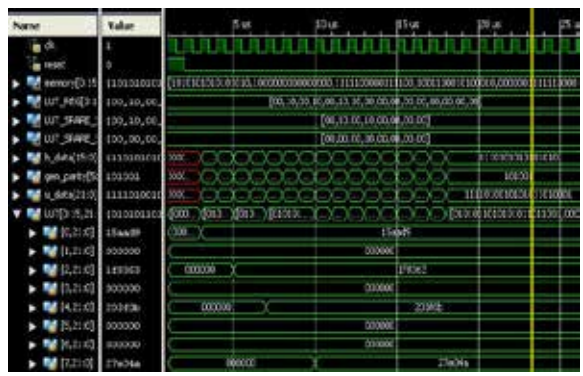


Fig.6. Simulated output of LUT

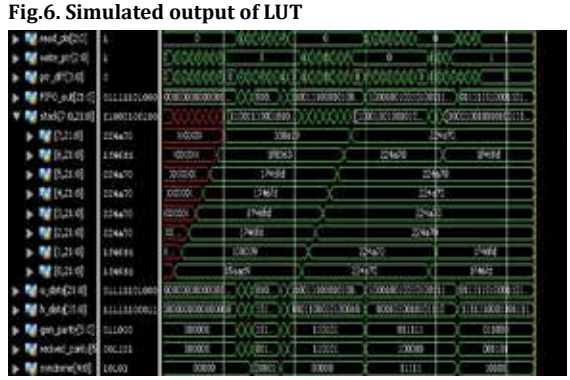


Fig.7. Simulated output of FIFO

VII. CONCLUSION AND FUTURE WORK

Research on fault-tolerant NoCs, which are a special class of interconnection network used to constitute the interconnection architecture of future, massively parallel multiprocessors that assemble hundreds to thousands of processing cores on a single chip, can therefore benefit from findings in this more general field. The NI and router could be the main source of errors in the NoC when the number of nodes in the network increases. The occurrence of permanent and temporary faults in the Network Interface will cause an unwanted behavior that may create unrecoverable situations in the NoC, such as deadlock or livelock conditions. By proposing a functional fault model for Network Interface and router based on the behavior on its main components, i.e., the lookup table and FIFOs, we are trying to develop an efficient NoC. New architectural solutions based on the use of error correcting and detecting codes and a limited amount of redundancy are proposing through this paper. Future work will be the Implementation of a fault-tolerant router along with the Network Interface which will lead to the generation of an efficient NoC where the NoC approach has a clear advantage over traditional busses and most notably system throughput.

VIII. ACKNOWLEDGMENT

The author gratefully acknowledges the support and facilities provided by Department of ECE, Vedavyasa Institute of Tech-

nology under University of Calicut. Author also extends her acknowledgement to Ms Anupama Varghese T(Asst. Professor, Dept of ECE) for her immense help during the course of project.

REFERENCE

1. Mariagiovanna Sami, Leandro Fiorin, "Fault-tolerant Network Interfaces for Networks-on-Chip," IEEE Transactions On Dependable And Secure Computing. | | 2. J. Hu, U. Ogras, and R. Marculescu, "System-level buffer allocation for application-specific networks-on-chip router design," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 25, no. 12, pp. 2919–2933, 2006. | | 3. R. Marculescu, "Networks-on-Chip: The Quest for on-Chip Fault-Tolerant Communication," in VLSI, 2003. Proceedings. IEEE Computer Society Annual Symposium on, feb. 2003, pp. 8 – 12. | | 4. S. Murali, T. Theocharides, N. Vijaykrishnan, M. | | Irwin, L. Benini,and G. De Micheli, "Analysis of Error Recovery Schemes for Networks on Chips," | | Design Test of Computers, IEEE, vol. 22, no. 5, pp. 434 – 442, sept.-oct. 2005. | | 5. T. Lehtonen, P. Liljeberg, and J. Plosila, "Online Reconfigurable Self-Timed Links for Fault Tolerant NoC," VLSI Design, vol. 2007, p. 13, 2007. | | 6. C. Grecu and M. Jones. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. IEEE Trans. Comput., 54(8):1025– 1040, 2005. | | 7. V. Rantala, T. Lehtonen, P. Liljeberg, and J. Plosila, | | "Multi Network Interface Architectures for Fault | | Tolerant Network-on-Chip," in Signals, Circuits and Systems, 2009. ISSCS 2009. International Symposium on, july 2009, pp. 1 –4. | 8. Radulescu, J.], S. Pestana, O. Gangwal, E. Rijpkema, | | P.Wielage, and K. Goossens, "An Efficient On-Chip NI Offering Guaranteed Services, Shared-Memory | | Abstraction, and Flexible Network Configuration," | | Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 24, no. 1, pp. 4 | | – 17, jan. 2005. |