International Journal For Technological Research In Engineering
Volume 2, Issue 7, March-2015 *ISSN (Online): 2347 - 4718*

FAULT TOLERANCE IN NETWORK ON CHIP

Mohana. D[1], Saranya Rajathi[2]
[1]Associate Professor, [2]PG Scholar Applied Electronics
Department of ECE, T.J Institute of Technology, Chennai- 6000 97

Abstract: Creating Network on Chip is done here, we present a new network-on-chip (NoC) that handles accurate localizations of the faulty parts of the NoC. The implemented design NoC is based on new error detection mechanisms suitable for dynamic NoCs, where the number and position of processor elements or faulty blocks vary during runtime. Indeed, We designed a online detection of data packet and adaptive routing algorithm errors. Both presented mechanisms are able to distinguish permanent and transient errors and localize accurately the position of the faulty blocks (data bus, input port, output port) in the NoC routers, while preserving the throughput, the network load, and the data packet latency. We provide localization capacity analysis of the presented mechanisms, NoC performance evaluations, and field-programmable gate array synthesis.

I. INTRODUCTION

The continuing reduction of feature sizes into the nanoscale regime has led to dramatic increases in transistor densities. Computer architects are actively pursuing multi-core designs with billions of transistors on a single die. Integration at these levels has highlighted the criticality of the on-chip interconnects; global interconnect delays are dominating gate delays and affecting overall system performance. Packet based (NoC) architectures[5] are viewed as a possible solution to burgeoning global wiring delays in many-core chips, and have recently crystallized into a significant research domain. NoCs are steadily becoming the de facto interconnect solution in complex Systems-on-Chip (SoC), because of their scalability and optimized electrical properties. However, current research also indicates that the chip area and power budgets are increasingly being dominated by the interconnection network. To combat this escalating trend, attention should be paid to the optimization of the interconnect architecture. Unlike traditional multi-computer macro-networks, on-chip networks in still a new flavour to communication research due to their inherently resource-constrained nature. Scarcity in the area and power budgets devoted to the interconnection fabric necessitates a re-interpretation of the networking paradigm. Furthermore, despite the lightweight character demanded of the NoC components, modern designs require ultra-low communication latencies in order to cope with inflating data bandwidths. The work presented in this volume aims to address these issues through a comprehensive and holistic exploration of the design space. To truly appreciate the ordinances underlying the NoC realm, the design aspects of the on-chip network are viewed through a penta-faceted prism encompassing five major issues: (1) performance, (2) silicon area consumption, (3)power/energy efficiency, (4) reliability, and (5) variability. These five aspects serve as the fundamental design drivers and critical evaluation metrics in the quest for efficient NoC implementations.

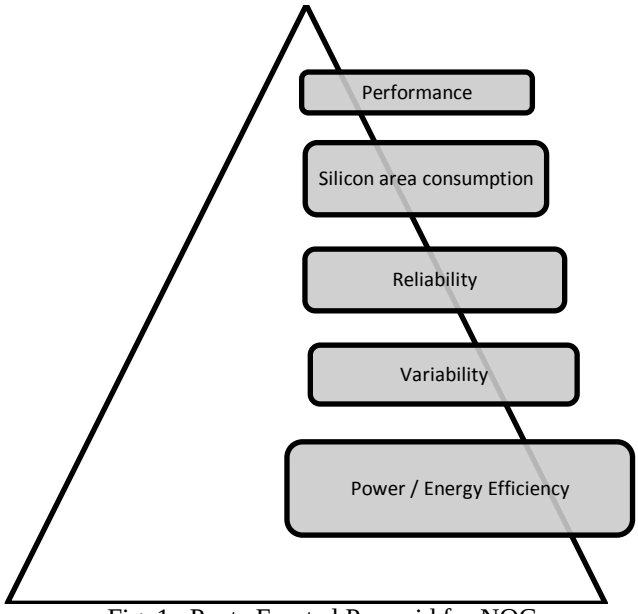

Fig. 1. Penta Faceted Pyramid for NOC

The research described in this volume explores the field by employing a two pronged approach: (a) MICRO-architectural innovations within the major NoC components, and (b) MACRO-architectural choices aiming to seamlessly merge the interconnection backbone with the remaining system modules. These two research threads, along with the aforementioned five key metrics mount a holistic and in-depth attack on most issues surrounding the design and integration of NoCs in modern multi-core architectures. Based on this premise of two complementary core themes, the volume is divided into two corresponding parts; the first part delves into the world of MICRO-architectural exploration of the NoC paradigm, while the second part shifts the focus to a MACRO-architectural abstraction level.
Ultimately, both parts work in unison in attacking several pressing issues concerning on-chip interconnects in the new multi/many-core reality. A new network-on-chip (NoC) that handles accurate localizations of the faulty parts of the NoC is presented. The implemented design NoC is based on new error detection mechanisms suitable for dynamic NoCs[1], where the number and position of processor elements or faulty blocks vary during runtime. Indeed, We designed a online detection of data packet and adaptive routing algorithm errors. Both presented mechanisms are able to

distinguish permanent and transient errors and localize accurately the position of the faulty blocks (data bus, input port, output port) in the NoC routers, while preserving the throughput, the network load, and the data packet latency. We provide localization capacity analysis of the presented mechanisms, NoC performance evaluations, and field-programmable gate array synthesis. The proposed NoC is based on new error detection mechanisms suitable for dynamic NoCs, where the number and position of processor elements or faulty blocks vary during runtime. Recently the trend of embedded systems has been moving toward multiprocessor systems-on-chip in order to meet the requirements of real-time applications. The complexity of these SoCs is increasing andthe communication medium is becoming a major issue of them MPSoC. Generally, integrating a network-on-chip into the SoC provides an effective means to interconnect several processor elements (PEs) or intellectual properties (IP) .The NoC medium features a high level of modularity, flexibility, and throughput.A NoC comprises routers and interconnections allowing communication between the PEs and/or IPs. The NoC relies on data packet exchange. The path for a data packet between a source and a destination through the routers is defined by the routing algorithm. Therefore, the path that a data packetis allowed to take in the network depends mainly on the addictiveness permitted by the routing algorithm, which is applied locally in each router being crossed and to each data packet . The increasing complexity and the reliability evolution of SoCs, MPSoCs are becoming more sensitive to phenomena that generate permanent, transient, or intermittent faults. These faults may generate data packet errors, or may affect router behavior leading to data packet losses or permanent routing errors. Indeed, a fault in a routing logic will often lead to packet routing errors and might even crash the router. To detect these errors, specific error detection blocks are required in the network to locate the faulty sources. Moreover, permanent errors must be distinguished from transient errors. Indeed, the precise location of permanent faulty parts of the NoC must be determined, in order for them to be bypassed effectively by the adaptive routing algorithm. To protect data packets against errors, error correcting codes are implemented inside the NoC components. Among the well known solutions, three are usually applied for the MPSoC communications based NoC. First, the end-to-end solution requires an ECC to be implemented in each input port of the IPs or PEs in the NoC . The main drawback of this solution is its incapacity to locate the faulty components in the NoC. Consequently, it is inadequate for dynamic NoCs, where the faulty and unavailable zones must be bypassed. Second, the switch-to switch detection is based on the implementation of an ECC in each input port of the NoC switches. For instance, in a router of four communication directions, four ECC blocks are implemented. Therefore, when a router receives a data packet from a neighbor, the ECC block analyzes its content to check the correctness of the data. This process detects and corrects data errors according to the effectiveness of the ECC being used. Third, another proposed solution is the code disjoint. In this approach, routers include one ECC in each input and output data port. This solution localizes the error sources, which can be either in the switches or on the data links between routers. However, if an error source is localized inside a router, this solution mechanism disables the totality of the switch. These online detection mechanisms cannot disconnect just the faulty parts of the NoC, and hence do not give an accurate localization of the source of errors. The result is that the network throughput decreases while the network load and data packet latency increase. Moreover, they are not able to distinguish between permanent and transient errors. For all these techniques, each ECC implemented in the routers of the network adds cost in terms of logic area, latency in data packet transmission, and power consumption.

II. NETWORK ON CHIP

Nowadays, the trend for Embedded Systems is moving toward Multiprocessor Systems on Chip (MPSoC) in order to meet the requirements of real-time applications. The complexity of these Systems on Chip (SoC) is increasing and the communication medium is becoming a major issue in MPSoC. Generally, integrating a Network-on-Chip (NoC) in the SoC provides an effective way to interconnect several Processor Elements (PEs) or Intellectual Properties (IPs) (processors, memory controllers, etc.) [1]. The NoC medium features a high level of modularity, flexibility, and throughput. A NoC is constituted of routers and interconnections allowing the communications between the PEs and/or IPs. Communication on NoC relies on data packet exchanges. The paths for the data packets between a source and a destination through the routers are defined by the routing algorithm. Therefore, the paths that data packets are allowed to take in the network depend mainly on the adaptiveness permitted by the routing algorithm (partially or fully adaptive routing algorithm) which is locally applied in each router being crossed and for each data packet [2, 3]. Dynamically reconfigurable 2D Mesh NoCs (DyNoC, CuNoC, QNoC, ConoChi, etc.) are suitable for FPGA-based systems [1, 4–7]. To achieve a reconfigurable NoC, an efficient dynamic routing algorithm of data packets is required. The goal is to preserve flexibility and reliability while providing high NoC performances in term of throughput.A dynamic reliable NoC presents the communications between several IPs depict a dynamic placement of an IP or the occurrence of a faulty node where bypasses determined by dynamic routing algorithm are required. Furthermore, faulty nodes or even faulty regions (areas in the network having a size larger than a tile) also make the communications in the networks harder, and even impossible for some routing algorithms and required fault-tolerant algorithms, as shown in Figure 1(c). Therefore, dynamic component placements and faulty nodes or regions are the main reasons why fault-tolerant or adaptive algorithms have been introduced and used in runtime dynamic NoCs [4]. Regarding adaptive or fault tolerant routing algorithms, several solutions have been proposed [9, 10]. Generally, these algorithms correspond to a modified

XY routing algorithm allowing to bypass faulty or unavailable regions. In the case of adaptive routing algorithms based on the turn model [11], zones corresponding to already detected faulty nodes or unavailable regions in the NoC are defined. The neighbor routers of these zones must not send data packets toward these faulty routers or unavailable regions. To achieve that, chains or rings around the adjacent faulty nodes or regions are formed in order to delimit rectangular parts in the NoC covering all the faulty nodes or unavailable regions. In these chains or rings of switches, the routing tables are modified and differ from the standard tables related to the XY routing algorithm. These specific switches integrate in their tables additional routing rules allowing to bypass the faulty zones or regions dedicated to dynamic IP/PE instantiations, while avoiding starvation, deadlock, and livelock situations [11]. Regarding increasing complexity and the reliability evolution of SoCs, MPSoCs are becoming more sensitive to phenomena generating permanent, transient, or intermittent faults [12]. These faults can affect the data packet contents or generate routing errors. To detect these errors, specific error detection blocks are required in the network to locate the faulty sources. Moreover, permanent errors must be distinguished from transient errors. Indeed, permanent faulty parts of the NoC must be located precisely in order to be bypassed thanks to the adaptive routing algorithm. Consequently, the considered fault models used are Stuck-at faults for localization of permanent errors, and Bit-flip faults for transient errors [8]. To protect the data packets against errors, Error Correcting Codes (ECC) are implemented inside the NoC components [6,7]. Among the well-known solutions, three are usually applied for the communications in MPSoC through a NoC. First, the end-to-end solution requires to implement an ECC in each input port of the IPs or PEs of NoC [2]. The main drawback of this solution is its incapacity to locate the faulty components (PE, IP, router, connection, etc.) in the NoC. Consequently, it is inadequate for dynamic NoCs where the faulty or unavailable zonesmust be bypassed. Second, the hop-by-hop detection is based on the implementation of ECC in each input ports of the NoC switches. For instance, in a router of four communication directions (North, South, East, West), four ECC blocks are implemented. Therefore, when a router receives a data packet from a neighbor, the ECC block analyzes the content of the packet to verify the correctness of the data. This process allows to detect and correct data errors according to the efficiency of the ECC being used. Third, another proposed solution is the code disjoint [2]. In this approach, routers include one ECC in each input and output data ports. This solution localizes the error sources which can be either in the switches or on the data links between routers. For all these techniques, each ECC implemented in the routers of the network requires an additional cost in terms of logic area, latency of the data packets and power consumption. Concerning the routing error detections and among the existing techniques able to detect faulty routing decisions, the analysis of the source and destination addresses presented in [6,7]. When a router receives a data packet, it compares its own address with the destination and source addresses. Then, the router checks its position in the deterministic XY path in the NoC of the considered data packet. The router performing this control is able to conclude if the switch from which it received the packet made or not a routing error according to the correct XY path. The drawback of this technique is the impossibility to handle the bypass of faulty nodes or regions. It results that this solution cannot be applied in adaptive or fault-tolerant routing algorithms. Indeed, as specified in a turn model algorithm [11], the structure of the reconfigurable NoC may contain bypass areas in which the switches must be able to take different routing decisions than the XY routing algorithm. To handle message routing errors in dynamic networks, a new faulty switch detection mechanism is required for adaptive or fault-tolerant routing algorithms. The aim is defining a new mechanism allowing the bypasses of the faulty nodes or regions (statically or dynamically PEs/IPs placed). In several application domains, such as multi-media processing, the bandwidth requirement between the cores in SoCs is increasing. The aggregate communication bandwidth between the cores is in the GBytes/s range for many video applications. In the future, with the integration of many applications onto a single device and with increased processing speed of cores, the bandwidth demands will scale up to much larger values. Each block corresponds to a core and the edges connecting the cores are labeled with bandwidth demands of the. communication between them. The bandwidth demands are in the order of hundreds of MBytes/s.Traditionally, bus-based architectures have been used to interconnect the various cores of the MPSoCs. To meet the increasing communication demands, the bus-basedarchitectures have evolved over time from a single shared bus to multiple bridgedbuses and to crossbar-based designs .Current state-of-the art bus architectures, such as the AMBA multi-layer , STBus and SonicsMX enable the instantiation of multiple buses operating in parallel, thereby providing a crossbar architecture. However, as all the cores in the design need to connect to the crossbar, such architecture is inherently non-scalable for large number of cores in the design. To effectively tackle the interconnect complexity of current and future MPSoCs, a micro-networks based interconnect architecture is needed to connect the cores. A communication-centric design approach, Networks on Chips (NoCs), has recently emerged as the design paradigm for designing such scalable micro-networks for MPSoCs.A typical NoC consists of switches, links and Network Interfaces (NIs). A NI connects a core to the network and co-ordinates the transmission and reception of packets from/to the core. A packet is usually segmented into multiple flow control unITS (°its). The switches and links are used to connect the various cores and NIs together. To tackle the delay of long NoC links, a latency insensitive design approach in which the links are pipelined can be utilized . Link pipelining increases the link throughput and decouples the cycle time of the communication system from the link length. The use of a NoC to replace bus-based wiring has several key advantages:

- Better scalability at the architectural and physical levels. NoCs can add bandwidth as needed and segment wires as required.
- Better performance under high loads. NoCs can run at or beyond 1 GHz (on
- a 1300nm technology process), cope with large bandwidth demands, and parallelize track streams.
- Better decoupling of protocol-level and transport-level issues in the communication protocol stack. Any standard interface can be deployed at the NoC boundary, then several degrees of freedom can be exploited within the fabric.
- Quicker design closure. NoC are more predictable: they intrinsically provide wire segmentation, which helps ensuring that design re-spins will not be needed in thelast phases of the design °ow, when they are more costly.
- More freedom in the design. NoCs are decentralized; therefore, features such as power management, clock domain crossing, frequency and voltage scaling can be independently added to the NoC sub-domains, reducing the issues presented by global infrastructures,more customizability.
- NoC topologies can be arbitrary and NoC architectures can be tuned according to a large range of settings. This extensive customization allows for perfect tailoring of the NoC to the target application(s).
- More streamlined design °ows. NoCs provide a solution to designing and verifying the whole architecture in a single automated pass, while bus-based architectures are struggling to keep up with application demands by means of
- very complicated and hand-crafted assemblies of buses, crossbars, bridges and converters. Moreover, these bus-based designs typically require several timeconsuming feedback loops in the design and veri¯cation phases due to the intensive manual intervention of designers, not needed with NoCs. NoCs facilitate modularity by orthoganalizing the design of the communication architecture design from the computation architecture, thereby leading to reduced design efforts.

Another effect of the shrinking feature size is that the power supply voltage and device Vt decreases and the wires become unreliable, as they are increasingly susceptible to various noise sources such as cross-talk, coupling noise, soft errors and process variations. The use of aggressive voltage scaling techniques to reduce the power consumption of the system further increases the susceptibility of the system to various noise sources. Moreover, wires are becoming thicker and taller, but their widths are not increasing proportionally, thereby increasing the effect of coupling capacitance on the delay of wires. The wire delay for data transfer on a communication bus depends on the data patterns transferred on the bus. The data-dependent variations in wire delay can be as large as 50% for the different switching patterns. With technology scaling, the device characteristics fluctuate to a large extent due to process variations and can cause significant variations in wire delay. Wire delay is also affected by other forms of interference such as supply bounce, transmission line effects, etc. Providing resilience from such transient delay and logic errors is critical for proper system operation. The variability in process technology and temperature distribution (thermal hotspots) and the effect of various noise sources such as power supply fluctuations and electromagnetic radiations pose major challenges for the reliable operation of current and future MPSoCs. While some of these noise sources (such as thermal effects) cause intermittent or temporary failures in the system, some others (such as process variations)can cause permanent failures of hardware components. With the increaseduncertainty of device operation, the time-to-failure period for the hardware components varies widely, with some components having a shorter lifetime than expected. Therefore, new design methodologies and architectural solutions need to be developed to ensure proper system operation. NoCs facilitate the use of error recovery schemes developed for networks to achieve a reliable system operation.

III. ALGORITHM IMPLEMENTATION

A. X-Y Routing Algorithm

Let (X1, Y1) and (X2, Y2) be the source address and destination address respectively in 2D -mesh topology. The X -Y routing is deterministic routing algorithm in which first the packet travel in X dimension until X1=X2 and then it travel in Y dimension until Y 1=Y2 to reach its destination. It is deterministic routing algorithm and cannotreroute the packet in case of defective switch. In fact it cannot tolerate even a single fault [6].

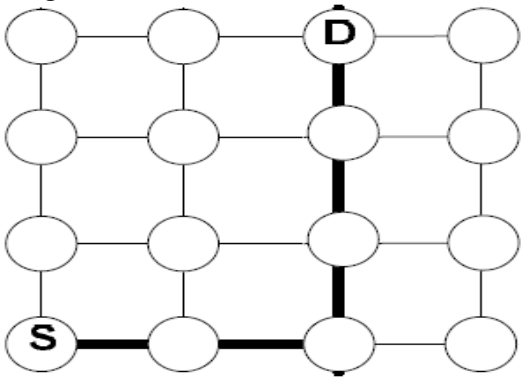

Fig.2. Example for XY Routing

B. Odd Even Turn Model

The adaptivity is needed in routing algorithm[1] to select alternate path in case of fault on the other hand the adaptivity of algorithm make it deadlock prone. In order to avoid deadlock , restrictions are applied on 2D mesh to prohibit circular movement . Typically, a switch containing four ports with four directions East North, South and West. There arefour possible turns of packet NE, ES, SW and WN for clockwise circular movement similarly SE, EN, NW and WS

are possible turns for circular anticlockwise movements. Deadlock can be avoided if the at least two turns are prohibited in both clockwise and anticlockwise circular movement. Chiu proposed an odd -even turning model consist of following rules [3]:

Rule1 (Anticlockwise Cycle): Any Packet is not allowed to take EN turn at node in even column and NW turn at node located in odd column.
Rule2 (Clockwise Cycle): Any packet is not allowed to take ES turn at node on even column and SW turn at node located on odd column.

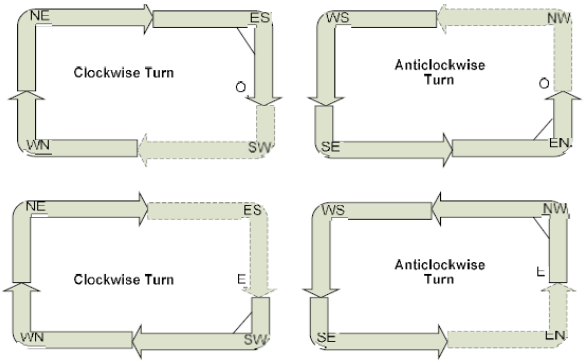

Fig.3. Dotted Arrows showing the Forbidden Turns [4]

Theorem1: Any routing Algorithm that follow s the rule of odd-even turn model is deadlock as long as 180 -degree turns are not allowed.

Proof: Consider a set of packet p1, p2, …….., pn that are deadlocked. They are waiting in circular path. So the waiting path includes rows and column. Consider the rightmost column line segment on waiting path according to Rule1 EW and NW turn is not possible. Similarly,

according to Rule2 ES and SW are no allowed in the same column .

Hence the statement of the theorem is proved.

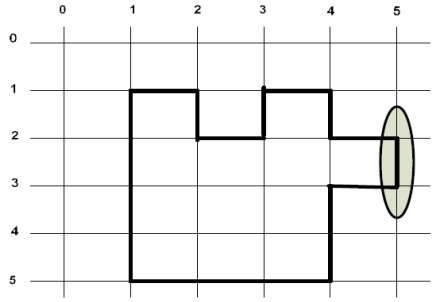

Fig. 4. The Prohibitive Turns in right most columns

C. Fault Blocks

Fault block is a set of nodes which are not used for routing a packet. The block may contain different type of nodes which are faulty, disabled and unsafe. Faulty nodes are unreachable nodes due to their defective links and switches. Disabled nodes are surrounded by faulty nodes such that they become unre achable. Unsafe nodes may cause deadlock if they are included in routing path. A fault block with faulty nodes at its upper, lower row and left and right forming a boundary is called rectangular fault block.

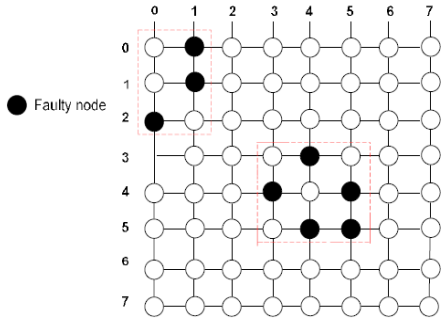

Fig. 5. Two rectangular faulty block in 7x7 2D Mesh

D. Routing Error Detection

The proposed reliable switch incorporates an online routing fault detection. This approach allows the routing error detections for adaptive algorithm based on the well-known XY routing algorithm. The main difficulty to detect the errors is to distinguish a bypass of an unavailable component in the NoC due to the use of adaptive algorithm from a real routing error due to a faulty component of the NoC. Figure 3 illustrates the challenge for such error detections. Apart from an increase of the data packet latency, the consequence of the nondetection of routing errors is the loss of data packets which can be sent either to an already detected faulty router or toward an area performing a dynamic reconfiguration. To achieve a routing error detection, the proposed reliable router relies on diagonal node state indications, additional routing information in the header flits, and routing error detection blocks in each port (see Figure 2). The basic concept of our approach is as follows: each router receiving a data packet checks the correctness of the routing decision made by the previous crossed switch. The routing scheme used in the NoC can be either static or dynamic in nature. In static routing, one or more paths are selected for the traffic rows in the NoC atdesign time. In the case of dynamic routing, the paths are selected based on the current traffic characteristics of the network. Due to its simplicity and the fact that application traffic can be well characterized for most SoC designs, static routing is widely employed for NoCs. When compared to static single-path routing, the static multi-path routing scheme improves path diversity, thereby minimizing network congestion and traffic bottlenecks. When the NoC is pre-designed, with the NoC having a fixed operating frequency, data width and hence bandwidth (bandwidth available on each network link is the product of the link data width and the NoC operating frequency), reducing congestion results in improved network performance. For most SoC designs, the NoC operating frequency can be set to match the application requirements. In this case, reducing the traffic bottlenecks leads to lower required NoC operating frequency, as traffic is spread evenly in the network, thereby reducing the peak link bandwidth needs. A reduced operating frequency translates to a lower power consumption in the NoC. When the NoC operating frequency for the schemes is set so that both schemes provide the same performance level (same average latency for tra±c streams), the multi-path

scheme results in 35% reduction in network operating frequency, leading to 22.22% reduction in network power consumption (after accounting for the overhead involved in the multi-path scheme). Another important property of the multi-path routing strategy is that there is spatial redundancy for transporting a packet in the on-chip network. A packet can be sent across multiple paths for achieving resiliency against transient or permanent failures in the network links. Many of today's NoC architectures are based on static single path routing. This is because, with multi-path routing, packets can reach the destination in an out-of-order fashion due to the difference in path lengths or due to di®erence in congestion levels on the paths. For many applications, such out-of-order packet delivery is not acceptable and packet re-ordering is needed at the receivers.In video and other multi-media applications, packet ordering needs to be maintained for displays and for many of the processing blocks in the application.With multi-path routing, packet re-order buffers can be used at the receiver to re-order the arriving packets. However, the re-order buffers have large area and power overhead and deterministically choosing the size of them is infeasible in practice.

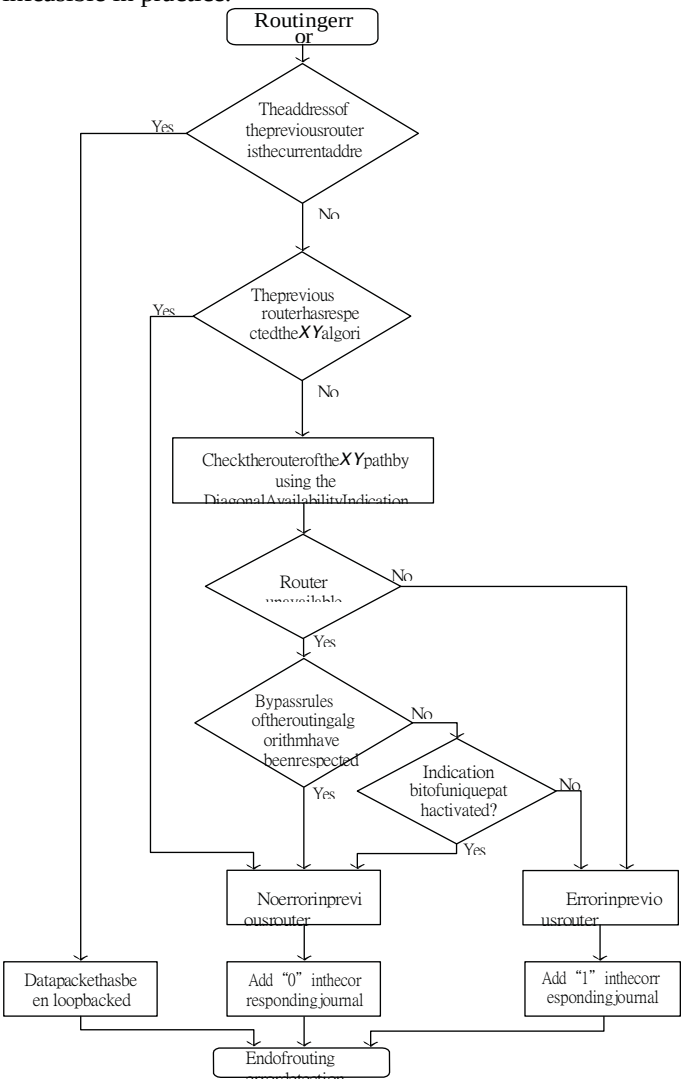

Fig. 6. Routing Error Detection Algorithm

E. Adaptive Routing Algorithm

An XY-based adaptive routing algorithm primarily uses the rules of the XY algorithm [12] to route data packets into the network when the required components are available. In the case of an unavailable component, a specific routing path is locally chosen to bypass its position. When a router receives a data packet, it checks the correctness made by the routing decision of the previous node, using the routing error detection algorithm. From address comparisons, the router checks if the previous routing decision obeyed the XY routing algorithm. If it is the case, then the previous decision is correct. Otherwise, the router decides whether the previous decision is a bypass decision or a routing error. The detection algorithm is required to check the availability of the router through which the data packets should have passed according to the XY algorithm. This verification is performed thanks to the DAI links. If the router in the XY path is unavailable, the previous router decision was a correct bypass. If it is available, the previous router decision is a routing error. In the latter case, the router adds one "1" to the error journal associated with the faulty routing logic block. The position of the faulty block is deduced from the address of the penultimate router in the SG field. If three consecutive errors are performed by the same faulty routing logic block, a permanent error is considered. In this situation, a specific data packet is generated towards the switch generating the routing errors. This specific one-flit data packet indicates the faulty input port of the considered router that must be disconnected. For NoCs based on multiflit data packets, it may happen that a flit is received without being preceded by a header flit, which is an erroneous situation. In the proposed RKT-NoC, there is a bit in each flit indicating whether the flit is a header flit or a data flit. When a router receives the first flit of a data packet, it checks after the hamming decoding whether it is a header flit. If not, the flit is destroyed. Therefore, when receiving a data packet, the destination IP or PE counts the number of received flits. If this number does not match the number indicated in the header flit, the packet is destroyed and a retransmission request is sent back to the emitter IP or PE.

Step 1 – Each router checks the correctness of the routing decision made by the previous crossed switch

Step 2 –The router checks if the previous routing decision obeyed the XY routing algorithm

Step 3 – If it is the case, then the previous decision is correct

Step 4 – Otherwise the router decides the previous decision is a bypass or a routing error

Step 5 – If the router in the XY path is unavailable, the previous router decision is a correct bypass

Step 6 – If the router in the XY path is available, the the previous router decision is a routing error

Step 7 – If three consecutive errors are performed by the same faulty routing logic block, a permanent error is considered.

IV. SIMULATION AND RESULTS

A. Design And Analysis Of Data Transmission Module

Data Flow Graph is nothing but the general block diagram of the inner network in the chip.

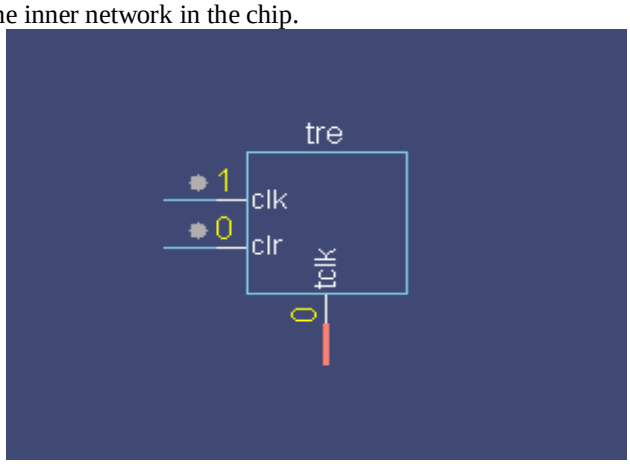

Fig. 6. Data Encryption

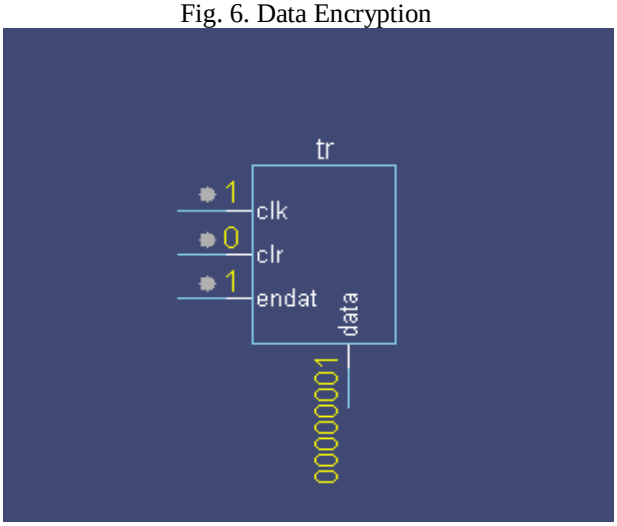

Fig. 7. Giving data Enable Signal

B. Simulation Result

Snapshot depicts each moment of the application while running. It gives the clear elaborated explanation of the application. It will be useful for the new user to understand for the future steps.

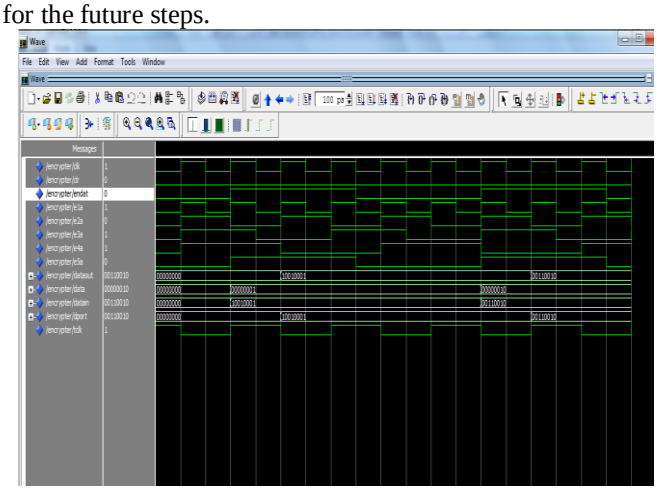

Fig. 7. Data Transmission Simulation result

C. Design And Analysis Of Nodes

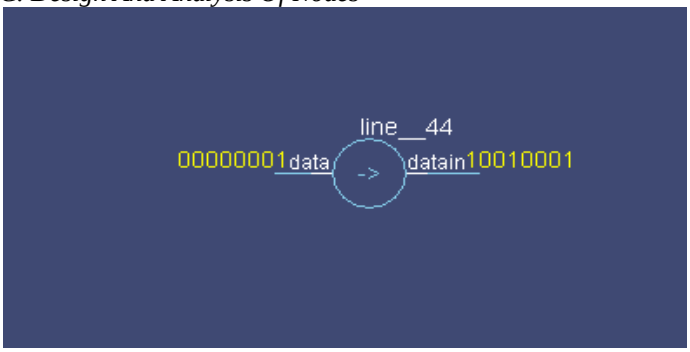

Fig. 8. Data Transmission - 1

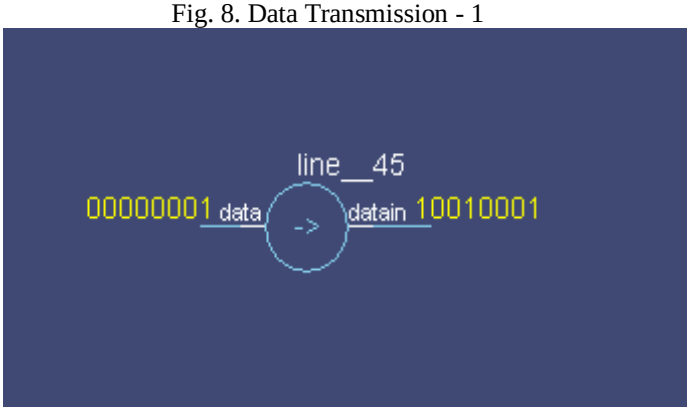

Fig. 9. Data Transmission - 2

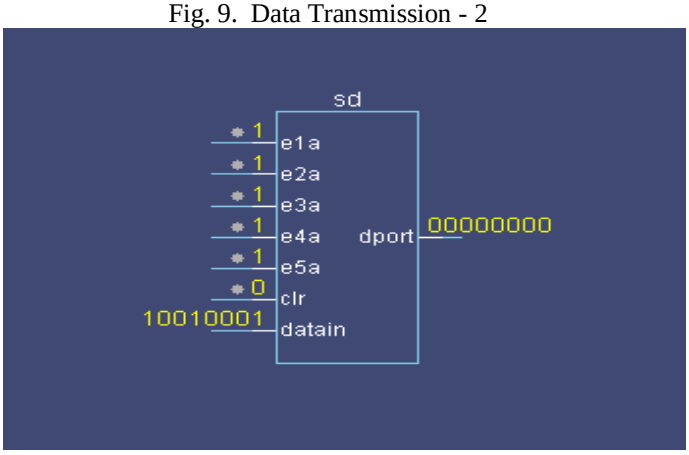

Fig. 10. Giving Input Values

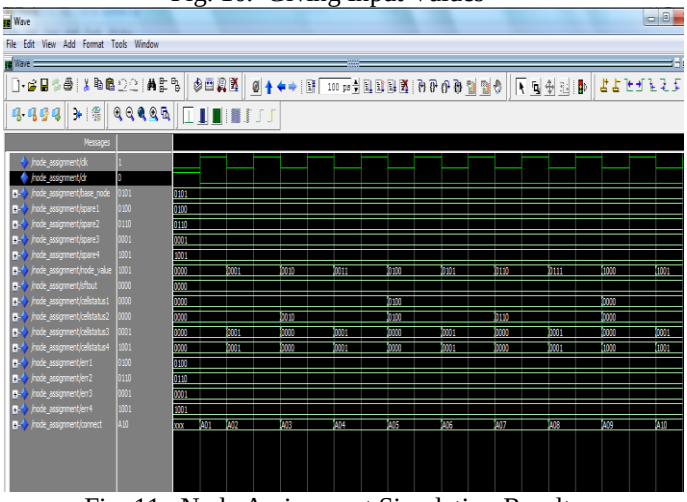

Fig. 11. Node Assignment Simulation Result

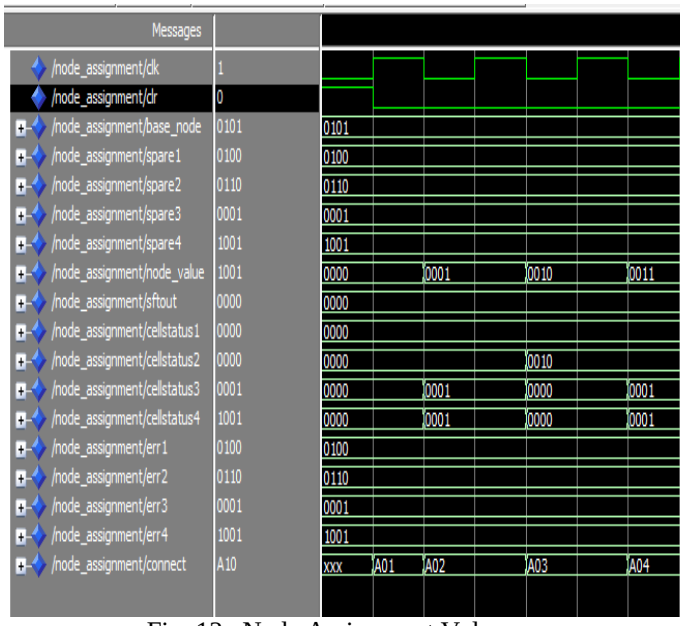

Fig. 12. Node Assignment Values

V. CONCLUSION AND FUTURE WORK

In smart reliable network on chip, first the transmission module for transmit the data or information to the destination network is designed. Before transmitting the data, the transmitting datas are encrypted using encryption module. After encryption, the data is transmitted based on the enable signal. In second the nodes for interconnecting the source and destination are designed. After design of nodes, the nodes to create the network for proper transmission of data are interconnected. In this project, the adaptive routing algorithm for selecting the data and detecting the faulty nodes in the network is designed . Finally all the sub modules in the design are integrated and the output is analyzed. In the future enhancement, validation simulations of our proposed routing error detection showed a routing error localization close to 96% for routing errors on an adaptive algorithm based on XY in a 6 × 6 NoC. Regarding the proposed data packet error localization mechanisms, the simulations presented in this paper clearly show the efficiency of our techniques, which can localize permanent sources of errors more accurately than the switch-to switch or code-disjoint mechanisms. The impact of faulty detection blocks are evaluated accurately and improves the routing error detection mechanisms, by protecting the DAI links and routing detection blocks against errors.

REFERENCES

[1] G.-M. Chiu, "The odd-even turn model for adaptive routing," IEEE Trans. Parallel Distrib. Syst., vol. 11, no. 7, pp. 729–738, Jul. 2000.
[2] Y. M. Boura and C. R. Das, "Efficient fully adaptive wormhole routing in n-dimensional meshes," in Proc. 14th Int. Conf. Distrib. Comput. Syst., Jun. 1994, pp. 589–596
[3] S. Jovanovic, C. Tanougast, S. Weber, and C. Bobda, "A new deadlock-free fault-tolerant routing algorithm for NoC interconnections," in Proc. Int. Conf. Field Program. Logic Appl., Aug.–Sep. 2009, pp. 326–331
[4] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Trans. Comput., vol. C-36, no. 5, pp. 547–553, May 1987
[5] M. Majer, C. Bobda, A. Ahmadinia, and J. Teich, "Packet routing in dynamically changing networks on chip," in Proc. 19th IEEE Int. Parallel Distrib. Process. Symp., Apr. 2005, p. 154b
[6] Yong Zou, "NARCO : Neighbor aware Turn Model-Based Fault Tolerant Routing for NOCs ", IEEE Embedded Systems Letters, Sep.2010
[7] Jingcao Hu, Radu Marculescu, " DyAD – Smart Routing for Networks-on-chip", Department of Electrical and computer Engineering, apr.2004
[8] Yu-Hsin Kuo, Po-An Tsai, Hao-Ping Ho, En-Jui Chang, Hsien-Kai Hsin, and An-Yeu (Andy) Wu1, ," Path-Diversity-Aware Adaptive Routing in Network-on-Chip Systems", Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan
[9] Myong Hyon Cho, Mieszko Lis, Keun Sup Shim, Michel Kinsy, Tina Wen and Srinivas Devadas, "Oblivious Routing in On-Chip Bandwidth-Adaptive Networks ", Computer Science and Artificial Intelligence Laboratory,Massachusetts Institute of Technology, Cambridge, MA
[10] Cedric Killian, Camel Tanougast, Fabrice Monterio and Abbas Dandache, " A New Efficient and Reliable Dynamically Reconfigurable Network-on-Chip", in Journal of Electrical and computer Engineering, June 2012
[11] K. Sekar, K. Lahiri, A. Raghunathan, and S. Dey, "Dynamically configurable bus topologies for high-performance on-chip communication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 16, no. 10, pp. 1413–1426, Oct. 2008
[12] J. Wu, "A fault-tolerant and deadlock-free routing protocol in 2d meshes based on odd-even turn model," IEEE Trans. Comput., vol. 52, no. 9, pp. 1154–1169, Sep. 2003.
[13] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C. Das, "Exploring fault-tolerant network-on-chip architectures," in Proc. Int. Conf.Depend. Syst. Netw., Jun. 2006, pp. 93–104.

