

A Literature Review of on-Chip Network Design using an Agent-based Management Method

Mr. Kendaganna Swamy S
Department of Electronics and
Instrumentation
RV College of Engineering
Bangalore, India

Dr. Anand Jatti
Department of Electronics and
Instrumentation
RV College of Engineering
Bangalore, India

Dr. Uma B V
Department of Electronics and
Communication
RV College of Engineering
Bangalore, India

Abstract— As the complexity of evolving integrated circuits and the number of cores in each chip increase, reliability aspects are becoming an important issue in complex chip designs. In this paper, presents a Review of Previous works on On-Chip Network Design Using an Agent-based Management Method an on-chip network architecture that incorporates a novel agent-based management method to enhance the reliability and performance of network-based Chip Multi-Processor (CMP) and System-on-Chip (SoC) designs against faulty links and routers. In addition, to utilize the fault information required for the routing process in a scalable manner, to classify the fault information to be exploited in the distributed and hierarchical management structure.

Index Terms—Agent based, On-chip Network, CMP, SOC.

I. INTRODUCTION

Chip Multi-Processors (CMPs) have been designed to overcome the intrinsic design challenges in order to comply with the increasing processing requirements. CMPs may include hundreds of Intellectual Property (IP) cores, processing elements and embedded memory blocks which communicate with each other [1]. The best scalable interconnection infrastructure for these complex systems is the Network-on-Chip (NoC).

There are reliability, power consumption and thermal issues in NoC-based CMPs that will be more important when the number of nodes increases. In this paper, concentrate on the reliability aspect in the underlying network. This aspect includes the proposed agent-based management structure and a routing method adapted to exploit this structure to tolerate permanent faults in the nodes and links. Select fault tolerance against permanent faults since a considerable amount of device failures may occur in both manufacturing and operational phases. To tolerate permanent faults many fault tolerant routing algorithms have been designed so far. However, because of the size of CMPs, Consider only distributed and scalable routing algorithms such as the methods introduced in [2-4]

a management structure based on hardware agents inside the network components is proposed for the mesh network to optimally utilize the fault information and distribute it among the appropriate nodes. For this purpose, classify the required

fault information for the routing process in detail. The appropriate portions of the fault information will be sent to the direct and indirect neighbor nodes through the hierarchical agents to be used in the routing process. This way, a scalable and fault-aware routing algorithm is achieved with higher performance compared to methods without agent-based management.

This paper is organized as follows. Section 2 gives a Brief overview of the fault information classification needed for the routing process is presented. Section 3 presents the literature survey of different agent based management methods for on chip network design. The agent based management method is explained briefly in A conclusion is given in Section 5.

II. FAULT INFORMATION CLASSIFICATION

In this section classify the fault information needed for the routing process in the NoC routers. The fault information is provided by the fault detection part. A typical NoC router (Fig.1) includes a controller, routing unit, crossbar switch as well as input and output ports. The controller mainly includes the switch allocator and virtual channel (VC) allocator if there are virtual channels in the input ports. The input ports include a buffer for each virtual channel and the output ports directly connect to the outgoing links. Based on [5] some test and fault detection circuits can be incorporated in the NoC routers and links to detect the permanent faults in each sub-block with an acceptable hardware overhead. Therefore, we assume that appropriate signals come out from the detection circuits so that we will be aware about the faultiness of five input buffers, four direct links, the routing unit, the controller and the crossbar switch, in each router. In addition, to that aware about the faultiness of the other components inside a node that are the Network Interface (NI) and the local core or Processing Element (PE). It is worth mentioning that for simplicity, we assume the links are bidirectional and when any type of permanent fault occurs in any direction, the entire link will be considered faulty.

The fault information obtained should be maintained to be used in the routing process or to be sent to a higher level of the system. This local fault information is stored in a register

called the local fault register (LFR). The local fault register also stores the status of four input buffers because the neighbor nodes need them to update their own local fault registers. This means that the usability of the four output directions in a router depends on the neighbor nodes, too. The local fault register includes 10 bits; four bits for four main directions, four bits for input buffers in the main directions, and two bits for *Node* and *PE* (Fig. 2). It is essential for the routing process that a router be aware about the faultiness of its four main directions. However, it is important that a router be aware about the faultiness of all components inside a small region similar to [3] and [4] because this regional information has a substantial effect on the fault tolerance capability and the cost of the routing algorithm. Select a region smaller than the one used in [2] which was a 2-hop distance region. This region including all the neighboring links with their names is shown in Fig. 2 in which the central node is the current router. In some manner, the central node should be informed about the faults in this region. Then, the regional fault information is updated and stored in an 8-bit register called the regional fault register (RFR) (Fig. 2).

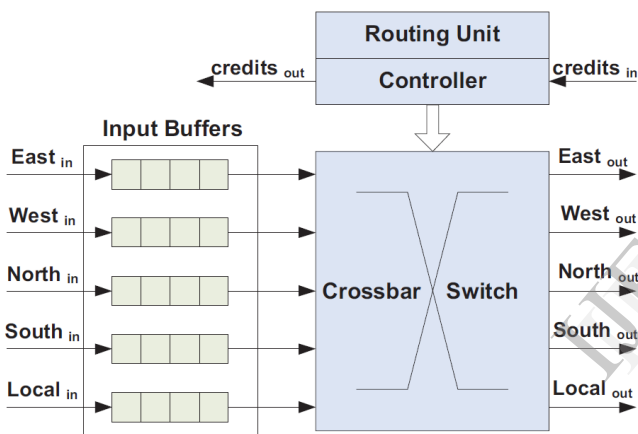


Figure 1. A typical NOC router architecture

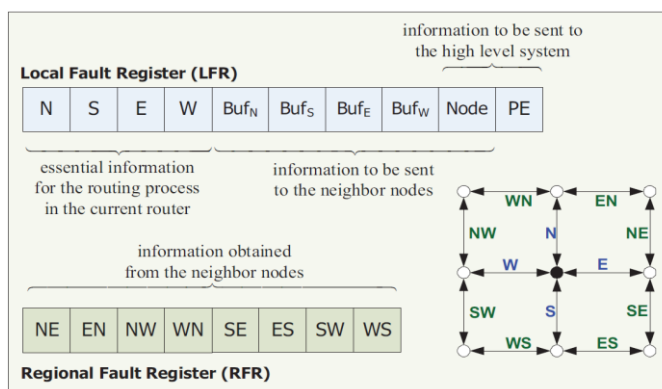


Figure 2. Fault information registers in each router, and the neighboring area

III. LITERATURE REVIEW

Hierarchical Agent Architecture is proposed to provide online monitoring services to NoC-based systems. Based on circuit conditions traced at the run-time, system settings are monitored adaptively by agents at each architectural level.

This paper explains the monitoring interaction between agent levels, and focuses on system optimization alternatives handled by different agent levels. They proposed hierarchical agent architecture with required monitoring services. This architecture adds a monitoring layer to hierarchy on NOC platform. This layer provides greater scalability and flexibility for large scale of NOC systems this architecture aims to achieve overall System performance by balancing all the on chip resources. The hierarchical approaches to various monitoring services for power optimization and fault tolerance. The trade-offs between area, energy and latency overhead, and motivates separate dedicated monitoring networks for inter agent communication [1-3].

Hierarchical agent monitoring design approach is innovative to achieve self aware and parallel computing in scalable manner. The hierarchical agents to monitors the system status during the run time, reconfigures the components to achieve better performance under the presence of errors. The manufacturing process of complex system like network on chip incurs some failures. The low cost routing algorithms are used to tolerate the permanent faulty links in the NoC[6].

To see the effect of these algorithms evaluate the performance, power consumption, and area over head through appropriate simulation and synthesis. They Proposed fault tolerant routing algorithm which are reconfigurable and make decision in each node based on the local fault information stored in the configuration register with respect to current and destination nodes[6]. The first routing algorithm is (FT_XY) tolerates all one faulty links. (FT_XY2) & (FT_XY3) is extension of (FT_XY) to tolerate two is more faulty links but they differ in hardware over head. According to simulation and synthesis the proposed routing algorithm does not require VCs causes a small overhead of area and power.

In paper [2] proposes a Fault-on-Neighbor (FoN) aware deflection routing algorithm for NoC which makes routing decision based on the link status of neighbour switches within 2 hops to avoid fault links and switches. Deflection routing is a non-minimal adaptive routing algorithm which can be implemented easily in hardware, since it does not need buffers for packets in transit. Fault-on-Neighbor (FoN) aware deflection routing algorithm which makes efficient routing decision based on the fault information transmission within 2 hops to avoid fault links and switches and can tolerate common convex and concave fault regions without deadlock and livelock. Fault-tolerant routing can be divided into two categories: stochastic and deterministic. Stochastic communication transfers redundant packets through different paths to avoid faults. Deterministic algorithm utilizes the structure redundancy of NoC to route packets to the destination through different paths to achieve fault tolerance. Force-Directed Wormhole Routing (FDWR) makes routing decision based on the routing table and the buffer status of neighbor switches. It uses the first flit of a packet as a look-ahead flit to investigate the buffer and fault status of neighbor switches. A resilient routing algorithm for fault-tolerant NoC based on turn model is described in. The switch can be reconfigured around faulty components while maintaining correct operation without using virtual channels. A fault adaptive deflection routing algorithm which makes routing

decision based on a cost function has been proposed in [3]. The switch implements an on-line fault diagnosis mechanism and makes routing decision based on a cost function which considers the route length and local fault status. It can not only handle link and switch faults, but also crossbar faults. Because it makes routing decision only based on the fault information of the current switch, the hop count field of the packet can easily overflow in some fault patterns.

The NoC architecture is based on Nostrum NoC, which is a 2D mesh topology. The difference from the ordinary 2D mesh is that the boundary output is connected to the input of the same switch so that the packet sent in that direction returns to the same switch. This can be used as a packet buffer. Deflection routing is used to make routing decision based on the packet priority and stress value which is the traffic load of neighbour switches in last 4 cycles. All incoming packets are prioritized based on its hop counts which record the number of hops the packet has been routed. The packet with the largest hop counts has the highest priority. The switch makes routing decision for each packet from the highest priority to the lowest [4].

Based on circuit conditions traced at the run-time, system settings are monitored adaptively by agents at each architectural level. The system setting includes resource utilization and power supplies configured hierarchically. This methodology provides systematic design approach to VLSI circuits under the influence of unpredictable variations and stringent power constraints [6]. Bio-inspired hierarchical-agent-based design methodology By Adopting the bio-inspired system architecture, a hierarchical agent monitoring NoC design method is proposed. bio-inspired approaches with multi-agent mechanisms provide network architecture with desirable properties such as scalability and adaptability the bio-inspired system architecture partitions conventional monitoring services onto different agent levels, and achieves greater scalability and efficiency than traditional dynamic management methods They propose a hierarchical monitoring agent based design approach. The system performance, including power efficiency, fault tolerance and variability, is optimized by the joint efforts of hierarchical intelligent agents which monitors, controls and adjust the NoC system at different functional levels [6]. It provides an high level abstraction for monitoring functions parallel on distributed systems. Here each level of agents performs specific monitoring operations based on their granularity. The monitoring hierarchy and operations are specified by for consistent and non-ambiguous system design.

Hierarchical Agent Monitoring (HAM): it constructs a design layer to monitor the operations this layer is separated from communication and computation. it improves the design efficiency by providing high level of abstraction. It features hierarchical communication power monitoring, in which different levels of agents make a joint effort to reduce the communication power HAM provides a systematic method applying hierarchical monitoring structure [7]. Adaptive monitors are required to monitor the system performance under the influence of variations. These monitors are nothing but the agents which are used to monitor and configure the computation and communication. it focuses on monitoring

operations reduces the complexity which is due to the other design concerns. These monitoring operations can be portioned and can be designed on to parallel systems .it does not require the count no of components can be integrated on a chip.

The structural redundancy on the chip network can be exploited by adaptive routing algorithms in order to provide connectivity even if network components are out of service due to faults [8]. Distributed fault diagnosis method facilitates determining the fault status of individual NoC switches and their adjacent communication links. In order to take measures against NoC faults, first detect and diagnose them. we make use of EDC-error detecting codes..i.e to equip data packets with (EDCS).The focus the network layer, which implements switching and routing functionality. The proposed techniques aim to support the transport layer, which is responsible for fragmenting messages into packets at the sender side, providing a reliable end-to-end communication by resending corrupted packets and finally assembling the received packets into the original message at the receiver side [8].

Dynamic XY (DyXY) routing is proposed for NoCs to provide adaptive routing and ensure deadlock free and livelock free routing at the same time. A new router architecture is developed to support this algorithm. It is observed it can achieve better performance than compared to static XY routing and odd even routing. In NOCs routing algorithm are used to determine the path of packet traversing from source to destination. Routing algorithms are classified as deterministic routing and adaptive routing. Deterministic routing its simplicity benefits in router design. It suffers from throughput degradation when the packet injection rate increases. Adaptive routing determines the routing paths based on the congestion conditions in the network .the adaptiveness reduces the chance of packets entering the faulty components and thus reduces the blocking probability of packets. virtual channels are introduced to assist design of adaptive and non adaptive routing algorithms for variety of network architectures A static XY routing algorithm for 2 dimensional mesh networks has been proposed .with static XY routing first the packet traverses in X dimension and then traverses in Y direction. This algorithm is dead lock free but provides adaptiveness [9].

With the DyXY routing algorithm, each packet only traverses along the shortest path between the source and the destination. Suppose if there are multiple shortest paths available the router will help the packet to choose of one of the path based on the congestion condition of the network. In a routing algorithm odd even turn was proposed based on the turn model. It restricts some locations where turns can be taken so that dead lock can be avoided. DyXY was proposed this is the combination of both deterministic routing algorithm called oe-fix and adaptive routing algorithm called odd even. The router can switch between these two routing modes based on the network congestion conditions.

ABOUT NOC: the continuing reduction in feature sizes of digital very large scale integration (VLSI) circuits enables the integration of dozens, and in the future hundreds of processing elements (cores, resources) on a single chip.

Traditional onchip buses can no longer sustain the increasing demand for communication between these cores. In order to overcome the performance gap, networks on chip (NoC) are being researched. A NoC is an on-chip communication infrastructure that implements multi-hop and predominantly packet-switched communication. Through pipelined packet transmission, NoC permit a more efficient utilization of communication resources than traditional on-chip buses. Regular NoC structures reduce VLSI layout complexity compared to custom routed wires [10].

IV. BRIEF ABOUT AGENT-BASED MANAGEMENT METHOD

To enhance the performance of a fault-tolerant on-chip network with a large number of components, a scalable management method can be beneficial. Thus, propose a management method that is agent-based and hierarchical to be more profitable for scalable on-chip networks.

There are two types of agents in the proposed management structure:

Cell agent: Each node or cell includes an agent called the cell agent which collects, manages and distributes the fault information related to the components of its node. In addition, it updates the LFR and RFR.

Cluster agent: Each cluster that includes a number of nodes is controlled by a cluster agent. A cluster agent configures the cell agents inside the cluster by sending the new fault information which is obtained from the other cell agents inside the cluster or other cluster agents.

The incorporated agent hierarchy is shown in Fig. 3. This agent hierarchy differs from that of proposed in the previous works ([5-8]). This is due to the fact that in the proposed structure, for faster reconfiguration the cell agents communicate with their neighbor cell agents even if they are situated inside different clusters. This is a real case because in general, in a CMP, a task may require more than a cluster to be run. On the other hand, the clusters running a common task are not necessarily neighbor clusters. However, the routers should be aware about their neighbors to select the best path for sending the packets to their destinations, and for faster awareness their cell agents should exchange the required fault information.

V. CONCLUSION

In this Paper surveyed about agent-based architecture for the fault tolerant routing process, different types of required fault information are classified to be distributed in the network. Each level of agent hierarchy manages and distributes a specific type of fault information. This way, a higher performance can be achieved in the networks of different sizes.

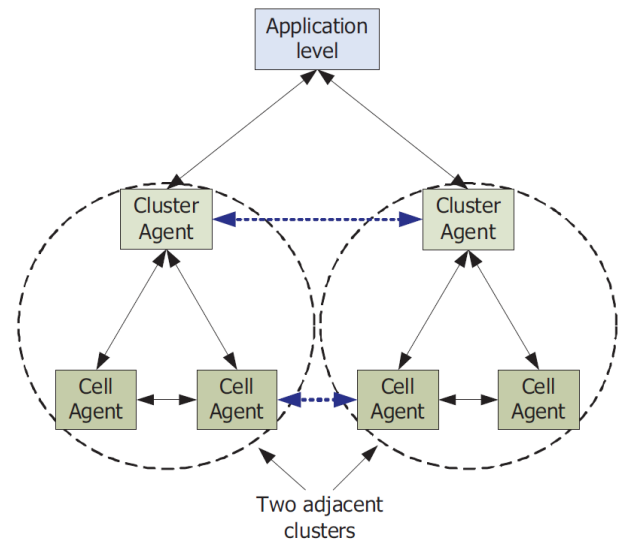


Figure 3. Hierarchical agents in two neighbor clusters

REFERENCES

- [1] O. Cesariew et al., "Multiprocessor SoC platforms: a component-based design approach," *IEEE Design and Test of Computers*, vol. 19, no. 6, pp. 52–63, 2002.
- [2] C. Feng, Z. Lu, A. Jantsch, J. Li, and M. Zhang, "FoN: Fault-on-Neighbor aware routing algorithm for Networks-on-Chip," *Proc. 23th IEEE Int. System-on-Chip Conf. (SOCC)*, pp. 441–446, 2010.
- [3] M. Valinataj, S. Mohammadi, and S. Safari, "Fault-aware and reconfigurable routing algorithms for Networks-on-Chip," *IETE Journal of Research*, vol. 57, no. 3, pp. 215–223, 2011.
- [4] M. Valinataj, S. Mohammadi, J. Plosila, P. Liljeberg, and H. Tenhunen, "A reconfigurable and adaptive routing method for fault-tolerant mesh based networks-on-chip," *Elsevier, Int. J. Electronics and Communications (AEÜ)*, vol. 65, no. 7, pp. 630–640, 2011.
- [5] A. Kohler, G. Schley, and M. Radetzki, "Fault tolerant network on chip switching with graceful performance degradation," *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no.6, 2010.
- [6] A. W. Yin et al, "Hierarchical agent monitoring NoCs: a design methodology with scalability and variability," *Proc. 26th NORCHIP Conf.*, pp. 202–207, 2008.
- [7] L. Guang, B. Yang, J. Plosila, K. Latif, and H. Tenhunen, "Hierarchical power monitoring on NoC - a case study for hierarchical agent monitoring design approach," *Proc. 28th NORCHIP Conf.*, 2010.
- [8] L. Guang, E. Nigussie, P. Rantala, J. Isoaho, and H. Tenhunen, "Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip," *ACM Trans. on Embedded Computing Systems*, vol. 9, no. 3, article 25, 2010.
- [9] A. Kohler, G. Schley, and M. Radetzki, "Fault tolerant network on chip switching with graceful performance degradation," *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 6, 2010.
- [10] M. Li, Q. Zeng, and W. Jone, "DyXY- a proximity congestion-aware deadlock-free dynamic routing method for Network on Chip," *Proc. 43th Design Automation Conference (DAC)*, pp. 849–852, 2006.