2581-4575

International Journal For Recent Developments in Science & Technology

Crossref

A Peer Reviewed Research Journal

SYSTEM ON CHIP DESIGN USING AGENT BASED METHOD

[1]SK.MUBEENA BEGUM, [2]D.SURENDRA

[1]STUDENT, A1 GLOBAL INSTITUTE OF ENGINEERING AND TECHNOLOGY

[2]ASSISTANT PROFESSOR, A1 GLOBAL INSTITUTE OF ENGINEERING AND TECHNOLOGY

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, we present 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, we classify the fault information to be exploited in the proposed distributed and hierarchical management structure. The experimental results show that the proposed architecture incurs only a small hardware overhead.

Index Terms: on-chip network, reliability, permanent fault, routing algorithm;

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-onChip (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, we 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. We select faulttolerance against permanent faults since a considerable amount of device failures may occur in both manufacturing and operational phases. To tolerate permanent faults many faulttolerant routing algorithms have been designed so far. However, because of the size of CMPs, we only consider distributed and scalable routing algorithms such as the methods introduced in [2-4]. The proposed agent-based management architecture consists of distributed agents in a hierarchical structure, which is especially suitable for large CMPs with tens or hundreds of processing elements. In this structure, the agents in each level of hierarchy have the same tasks to gather, manage and distribute the fault information. The previous works related to the hierarchical agents can be found in [5-8]. In [5] and [6] the overall structure for the agent-based management is discussed

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

without any detailed design. In [7] a NoC monitoring scheme based on hierarchical agents is addressed mainly to minimize network power consumption. System level design principles, the basic concepts of the general approach for the hierarchical agent monitoring, and the general tasks of the agents in each level of hierarchy are discussed in [8]. In addition, it includes an approach for DVFS (dynamic voltage and frequency scaling) used in power monitoring. However, it does not present any detailed or low-level design especially for faulttolerance. In this paper, 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, we 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.

The rest of the paper is organized as follows. In Section II the fault information classification needed for the routing process is presented, and in Section III the proposed agentbased management method is explained. The experimental results are presented in Section IV and finally, conclusion and future works are given in Section V.

II.FAULT INFORMATION CLASSIFICATION

In this section we 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 [9] 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, we should also be 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.We say the north direction of a router is faulty or unusable for the routing process if the north link or the north input buffer in the current router or the south input buffer in the north neighbor router is faulty. This condition can

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

be stated by (1) using the appropriate signals from the fault detection circuits:

$$N = Link_N \text{ or } Buf$$
$$Ncur\_router \text{ or } {}^{Buf}SN\_router \quad (1)$$

In (1) all terms are one-bit status data showing that if any term equals '1' its corresponding component is faulty, otherwise it is healthy. In addition, (1) is a replacement for a common assumption that declares a faulty input buffer can be modeled by its incoming link assumed faulty.

Equation (2) can be generally used for four main directions in each router:

$$X\ Link_X \text{ or } Buf_X cur\_router \text{ or } Buf\ (1X_X\_router) \quad (2)$$

In (2) X can be N, S, E or W which mean north, south, east or west directions, respectively. In this equation, $Link_X$ means the status of bidirectional link in the X direction of the current router, $Buf_X^{cur\_router}$ stands for the status of the input buffers of all VCs in the X direction of the current router, and $Buf_{(1-X)}^{X\_router}$ stands for the status of the input buffers of all VCs in the opposite direction of X in the neighbor router which is located in the X direction of the current router. In addition, (1-X) stands for the opposite direction of X, which means S, N, W and E for N, S, E and W directions, respectively.The effect of some faulty components inside a router is that the whole router and as a result the whole node should be considered faulty because the router is unable to perform its main task (sending incoming packets to the correct output ports). For this case, we dedicate a bit called Node as a part of fault information regards to this situation based on (3):

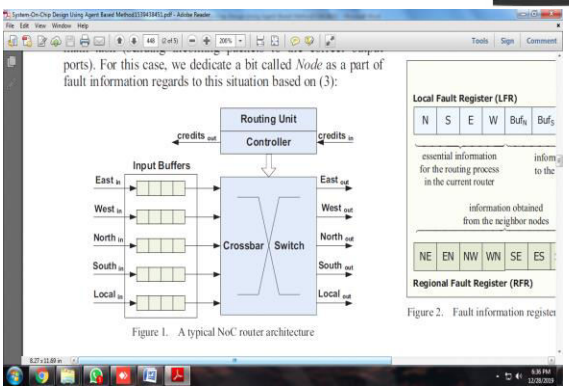

Figure 1.A typical NoC router architecture

Node routing unit_ or controlloror crossbar (3)

Equation (3) means that we should consider the whole node faulty if the routing unit, the controller or the crossbar switch is faulty.

In NoC-based CMPs, the local cores or the processing elements are connected to the routers via the network interfaces. If a processing element is unusable, the high level system should either migrate its task to other processing elements or perform a remapping process. In a NoC router we assume the local processing element is unusable if it is faulty or its network interface or the buffers located in the local port are faulty based on (4):

$$PE\ PElocal \text{ or } NI \text{ or } Buf\ local \quad (4)$$

In the equation above, PE equal to '1' means that the local core or processing element is unusable otherwise it is usable.

The fault information obtained from (2) to (4) 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

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

them to update their own local fault registers based on (2). 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 based on (3) and (4) (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 faulttolerance capability and the cost of the routing algorithm. We 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).

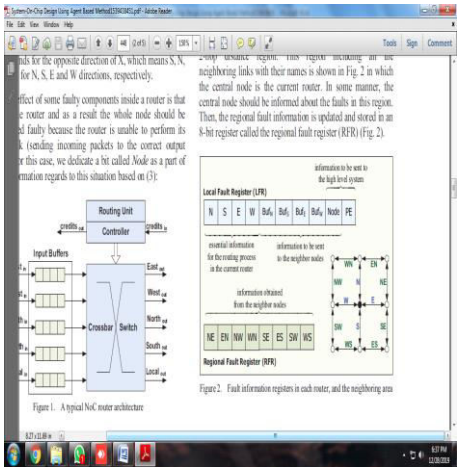

Figure 2. Fault information registers in each router, and the neighboring area

As stated before, faulty input buffers are modeled by their corresponding links assumed faulty and affect the status of the main directions of the routers. For example, based on (2), the status of the NW link (the link situated in the north of west of the central router) shown in Fig. 2 is also affected by the status of the input buffers in its two sides. This way, if its corresponding bit in the regional fault register (NW bit) equals '1', this means that in a minimal routing process the central router should not send any packet from its west direction if the destination node is top-left node.

III. 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, we propose a management method that is agent-based and hierarchical to be more profitable for scalable on-chip networks.

A. Preliminaries

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

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

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.

B. Interconnections for Hierarchical Agents

A small 3×3 network with an agent in each node is shown in Fig. 4. In this figure, R, NI and PE correspond to the router, network interface and core or processing element, respectively. In addition, the agents can be cell or cluster agents but the number of cluster agents is much less than the number of cell agents. For example, in a regular mesh network each 3×3 subnetwork can be a cluster with a cluster agent in the center. The proposed agent-based management structure uses two types of communications: a physically separate network for only peer to peer communication between the cell agents, and the baseline data network for control packet communication with a higher priority compared to data packets.

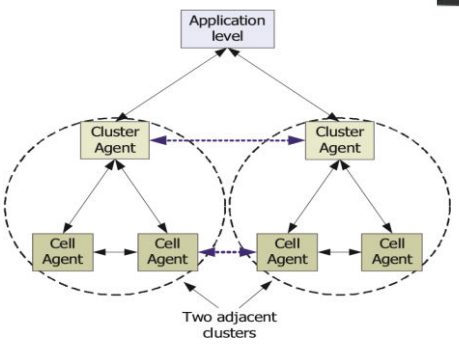

Figure 3.Hierarchical agents in two neighbor clusters

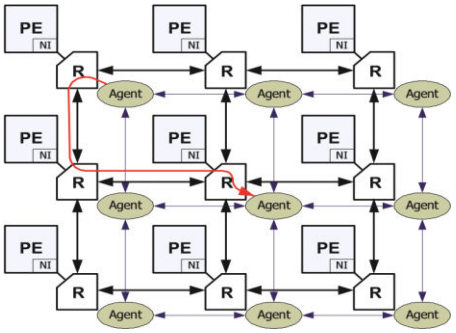

Figure 4.A 3×3 NoC with agents and interconnections

The former is shown in Fig. 4 between the agents and the latter is the network connecting the NoC routers. The control packets including the reconfiguration and fault information are exchanged between the cell agents and the cluster agents in addition to the communications between the cluster agents. In Fig. 4, the shown path is used if at least one of the source or destination agents is a cluster agent. The manner in which these communication networks are utilized is discussed in the next subsection.

C. Agent Tasks

In most previous works it is assumed that for the routing process the NoC routers are aware about the faults or failures occurred in the neighbor nodes in addition to their own components and interconnection links. However, in reality there should be a mechanism to inform the routers about

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

different faults in the network. In the proposed method, this awareness is distributed as fault information by hierarchical agents and their interconnections.

1) Cell agent

In the agent hierarchy shown in Fig. 3, all the fault information needed for the neighbor nodes (the local fault register except its right most bit) can be sent to the appropriate cluster agent, and then the cluster agent can distribute and send it to the appropriate cell agents. However, to minimize the impact of faults on the network performance especially just after fault occurrence, it is better that the cell agents themselves distribute the fault information to their neighbors because this approach is faster. This way, the fault information essential for the routing process is distributed at the lowest level of management hierarchy through the dedicated network between the cell agents. A cell agent sends a portion of LFR to each neighbor cell. In other words, only the fault information required for the routing process in a specific neighbor node is sent to its corresponding cell agent. For example, according to the neighboring area shown in Fig. 2, the cell agent situated in the central node only sends the faulty status of the north direction (N), south direction (S), west input buffer (Buf_w) and whole node (Node) to its west neighbor cell agent (if any bit equals '1'). In the west cell agent, Buf_w is used to update the E bit in LFR; N and S are used to update the NE and SE bits in RFR. The west cell agent does not receive the faulty status of the east direction (E) from the central cell agent because the east link of the central node is situated outside the defined neighboring area for the west node which is smaller than the area obtained by the 2-hop distance introduced in [2]. In addition, Node bit is used to update the NE, SE and E bits simultaneously. In other words, when the west cell agent is notified about the faultiness of the whole central node (its east neighbor node), it will set the NE, SE and E bits to '1'.

To send and receive the essential fault information, the cell agents perform simple encoding and decoding processes. Each fault occurrence should be informed only once. In addition, if more than one fault occurs simultaneously, they can be informed to the neighbors sequentially, in different clock cycles. Thus, we use three bits and as a result a three-bit unidirectional link to encode the status of two directions, one input buffer and the whole node (N, S, Buf_w and Node for the west neighbor) in addition to a non-faulty state to inform each neighbor cell agent. In the neighbor cell agents, each three-bit input is decoded to update the local and regional fault information. Therefore, the width of the links in the dedicated network shown in Fig. 4 will be six bits.

2) Cluster agent

A cluster agent informs the higher level about the critical failures (PE or whole node failures) occurred inside the cluster and receives reconfiguration commands from the higher level about remapping of the tasks on the nodes or task migration. However, it is clear that a cluster agent itself is informed by the cell agents about the critical failures. In addition, for a more effective routing process, a cluster agent should be informed about other faults inside a node such as

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

faulty output directions (which include the effect of faulty input buffers), because it should be able to send different types of fault information to other cluster agents in addition to informing other nodes inside the cluster. When the source or destination agent is a cluster agent the baseline packet-based data network is used. Due to the fact that the number of permanent fault occurrence is low when a NoC-based CMP is running some specific tasks, the amount of fault information that should be distributed is not high. Thus, the number of required packets to carry the fault information is low and as a result, the overhead is negligible. On the other hand, a cluster agent normally has a distance of more than one hop from other cluster agents and from some of the cell agents inside its cluster. Therefore, the packet-based data network is convenient for cluster agent-based communication.

A cluster agent can be placed in the processing element of a node as a software agent which is implemented entirely in software (SW). It can also be a hardware (HW) component similar to the proposed cell agent or can be implemented in HW/SW co-design manner. In our method a cluster agent is implemented in hardware and it includes the cell agent of the node in which it is located. For packet-based communication and to use existing resources in a node, the cell and cluster agents exploit the local port to send and receive packets. Thus, some extra logic for multiplexing is required to separate data and control packets and then to direct packets to the agents or the local core (PE) or to accept packets from them.

D. Fault-Tolerant Routing Algorithm

The fault-tolerant routing algorithm to incorporate the agent-based method and different types of fault information is a modified version of the method introduced in [4]. This routing algorithm is a low-cost, adaptive and congestion-aware method and since it does not use routing tables and acts in a distributed manner, it is scalable and thus suitable for large NoC-based CMPs. For better utilization of agent-based management method some extra fault information besides the information introduced in Section II is used in the routing algorithm. In each cell agent, this information is obtained from the cluster agent and includes the fault status of the main directions in some nodes inside the cluster. We call it cluster-dependent fault information and it is highly dependent on the size and the shape of a cluster in addition to the type of the routing algorithm.

For a realistic example, we assume that each cluster in the network has a regular structure and includes a 3×3 sub-network similar to the network shown in Fig. 2 in which the central node includes the cluster agent. In such a cluster, if the source and destination nodes are top-right and bottom-left nodes, respectively, the source node should be aware about the status of the links with the labels S, W, SW and WS in Fig. 2 around the destination. These links are located outside of the neighboring area of the top-right node that means this node cannot be aware about the status of the mentioned links from the neighbor cell agents; thus, it should be informed by the cluster agent. The designed routing algorithm acts in such a way that it can correctly deliver all the

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

packets to their destination by selecting the shortest or non-shortest paths if it is not aware about the status of the links with the labels S, W, SW and WS links. However, if it is aware, it definitely selects a shortest path from the source to the destination. This example justifies the usage of the cluster agent that manages and distributes the appropriate fault information related to each node inside the cluster.

IV.EXPERIMENTAL RESULTS

A. Performance Evaluation

To demonstrate the effect of the proposed agent-based management method on the network performance, we simulated 3×3 VHDL-based NoCs with the input buffer size of four flits in each virtual channel and the packet length of 16 flits under the uniform traffic pattern. The number of incorporated virtual channels equals two thus it is minimum. In addition, two different methods are used: The Main-RAFT routing algorithm [4] that does not use any agents, and a modified version of the Main-RAFT that uses the agent-based management method including cluster-dependent fault inform-ation. It is assumed that each cluster is a 3×3 sub-network in which the center node includes the cluster agent. In the 3×3 NoC 25% of the links are faulty (three faulty links) as depicted in Fig. 5a. The average packet latency for each traffic load is measured upon all the packets when each local core generates 2000 packets. As shown in Fig. 5a, for the source node S and the destination node D there are two different paths P1 and P2 that can be traversed by the packets based on [4]. P1 is a minimal path but P2 is a non-minimal path. However, the routing based on the proposed agent-based management only selects the minimal path P1 for the packets. In addition, similar paths can be obtained for other source-destination pairs in the mentioned 3×3 NoC. The load-latency diagram for this network is shown in Fig. 5b. As shown in this figure, the proposed agent-based routing method has a better performance and a lower saturation point compared to the method introduced in [4].

B.Area Overhead

To evaluate the area overhead of the proposed method we implemented DyXY [10] as the basic adaptive routing algorithm, Main-RAFT [4], and the routing algorithm based on the proposed agent-based management method in a state of the art router architecture using VHDL synthesized with a standard cell library. A medium width of 32 bits has been selected for the links and flits. The size of input buffers is four flits or eight flits. Table I shows the areas for the 5-port routers using the mentioned methods. In addition, this table shows the area overhead of the proposed agent-based router compared to other routers. Based on the obtained results, the area overhead of the proposed router compared to [4] is only 1.1% or 1.6% when the size of the input buffers is eight or four flits, respectively. It is worth mentioning that the DyXY method does not have any means to reliably convey all the packets to their destinations in the fault situations. In addition, the wire overhead of the proposed method (3 bits in each direction) is less than 5% and 10% for 64- and 32-bit flits, respectively.

V.CONCLUSION

In this paper, a scalable agent-based management architecture for fault-tolerant

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

NoC-based CMPs is proposed. To exploit the proposed agent-based architecture for the faulttolerant 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. The simulation and synthesis results reveal that the proposed architecture improves the network performance with a small hardware overhead. In future, we will investigate the agents in different levels of hierarchy with a more efficient management process. In addition, we will classify and manage the fault information useful for the routing process in the neighbor clusters.

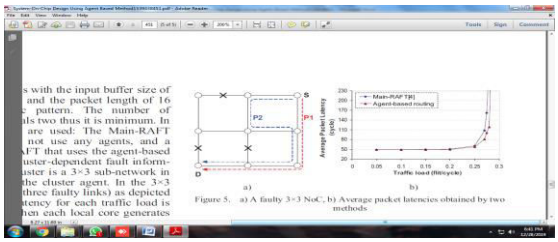

Figure 5.a) A faulty 3×3 NoC, b) Average packet latencies obtained by two methods

TABLE I. HARDWARE COST AND OVERHEAD

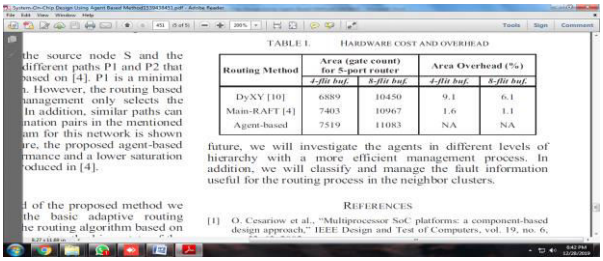

TABLE I. HARDWARE COST AND OVERHEAD

Routing Method	Area (gate count) for 5-port router		Area Overhead (%)	
	4-flit buf.	8-flit buf.	4-flit buf.	8-flit buf.
DyXY [10]	6889	10450	9.1	6.1
Main-RAFT [4]	7403	10967	1.6	1.1
Agent-based	7519	11083	NA	NA

REFERENCES

- O. Cesariow et al., "Multiprocessor SoC platforms: a component-based design approach," IEEE Design and Test of Computers, vol. 19, no. 6, pp. 52–63, 2002.
- C. Feng, Z. Lu, A. Jantsch, J. Li, and M. Zhang, "FoN: Fault-onNeighbor aware routing algorithm for Networks-on-Chip," Proc. 23th IEEE Int. System-on-Chip Conf. (SOCC), pp. 441–446, 2010.
- 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.
- M. Valinataj, S. Mohammadi, J. Plosila, P. Liljeberg, and H. Tenhunen, "A reconfigurable and adaptive routing method for fault-tolerant meshbased networks-on-chip," Elsevier, Int. J. Electronics and Communications (AEÜ), vol. 65, no. 7, pp. 630–640, 2011.
- P. Rantala, J. Isoaho, and H. Tenhunen, "Novel agent-based management for fault-tolerance in network-on-chip," Proc. 10th Euromicro Conf. on Digital System Design (DSD), pp. 551–555, 2007.
- W. Yin et al, "Hierarchical agent monitoring NoCs: a design methodology with scalability and variability," Proc. 26th NORCHIP Conf., pp. 202–207, 2008.
- 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.
- 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.
- 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.
- 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.