

Functional Diagnosis for Graceful Degradation of NoC Switches

Atefe Dalirsani, Hans-Joachim Wunderlich

Institut für Technische Informatik, Universität Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany

Email: {dalirsani,wu}@informatik.uni-stuttgart.de

Abstract—Reconfigurable Networks-on-Chip (NoCs) allow discarding the corrupted ports of a defective switch instead of deactivating it entirely, and thus enable fine-grained reconfiguration of the network, making the NoC structures more robust. A prerequisite for such a fine-grained reconfiguration is to identify the corrupted port of a faulty switch.

This paper presents a functional diagnosis approach which extracts structural fault information from functional tests and utilizes this information to identify the broken functions/ports of a defective switch. The broken parts are discarded while the remaining functions are used for the normal operation. The non-intrusive method introduced is independent of the switch architecture and the NoC topology and can be applied for any type of structural fault. The diagnostic resolution of the functional test is so high that for nearly 64% of the faults in the example switch only a single port has to be switched off. As the remaining parts stay completely functional, the impact of faults on throughput and performance is minimized.

Index Terms—Functional test, functional failure mode, fault classification, functional diagnosis, pattern generation, fine-grained reconfiguration

I. INTRODUCTION

Massive integration led to a paradigm shift of using Networks-on-Chip (NoCs) [1] as a scalable communication alternative in recent years. Due to internal redundancies, NoCs are inherently fault tolerant and can be reconfigured in the presence of defective cores, links or switches [2] to operate at a degraded performability level.

In today's massively integrated devices, test and diagnosis time has become a major bottleneck for the time-to-market [3, 4]. To maintain the profit margins, two aspects are deeply of interest: firstly, reducing the test and diagnosis time, while preserving the fault coverage and test efficiency, and secondly integrating fault tolerant features to tolerate the components' failure in defective chips [5] at a degraded functionality level, known as graceful degradation.

In NoCs, several test approaches have been proposed which reduce the test application time while preserving a high structural fault coverage [6–10]. On the other hand, fault tolerant structures such as [2, 11] support individual deactivation of defective switches, links or switch ports.

Finding a defective component is a prerequisite for graceful degradation. Concurrent error detection based on hardware redundancy, mostly employed in safety- or mission-critical applications, can be used for fault detection in the field. However, in several applications, the hardware overhead for concurrent error detection is not acceptable. Thus, available test infrastructure for manufacturing or in-field test is used for fault detection.

Functional tests offer several advantages to reduce the test costs: they shorten the test application time, lower the test hardware overhead and enable at-speed testing [12–15]. Several functional test approaches have been proposed for NoCs so far [8, 9]. Based on the functional test responses, defective switches and links are discarded. Although this information can be used for graceful degradation, such approaches suffer from the following shortcomings:

- They may pessimistically deactivate the entire switch and thus some intact communication resources of the network might be discarded.
- They cannot provide any information about the structural root cause of the malfunction.

Fault tolerant approaches like [11, 16, 17] can detect defective switch subcomponents at the cost of introducing extra hardware for testing individual blocks. The method in [18] exploits structural test information for structural diagnosis and reasons about the intact functions/ports of a defective switch. However, to the best of our knowledge, the functional test information has never been used for structural diagnosis so that the root cause of a malfunction is determined and the defective subcomponent is identified.

The paper at hand proposes a functional diagnosis approach that analyses the functional test responses and finds a set of suspected structural faults that can cause the observed functional failure. Certainly, with functional tests, the diagnosis result cannot not be narrowed down to a single culprit, but to a set of suspects which gives sufficient information for reconfiguration. To find the exact culprit, which is necessary for manufacturing diagnosis for example, additional structural test is required. With respect to the suspects set, we reason about the broken functions/ports of a defective NoC switch. Instead of completely deactivating the defective switches, in this approach only the broken parts are disabled and the intact functions are retained for data transport.

In addition to the advantages of functional tests, our approach accelerates the diagnosis and reconfiguration time, and skips the necessity for huge structural diagnosis processes in the field. The approach does not change the NoC switch structure and it allows, for the first time, fine-grained reconfiguration using only functional test information.

II. PRELIMINARIES

A. NoC Switches: Structure and Functionality

The NoC consists of several *switches* that are connected to each other via communication *links*, constituting the regular

(for example mesh or torus) or irregular network topologies [19]. Data bits are encapsulated in *data packets* and injected into the network by the system resources. Switches forward the received packets via a suitable path from a source to a destination. Each NoC switch consists of several input/output ports. Fig. 1 depicts a typical switch of the NoC mesh topology with five input/output ports. An internal control logic implements the routing algorithm, switching and scheduling policy, which manage the data flow among the switch ports. Each switch port includes a number of input data pins, output data pins and some handshake signals, which construct the interface to the neighboring switches.

No matter which switch structure and which NoC topology, the switches must fulfil the following functional properties:

- The received data is forwarded via a correct output port, determined by the internal control logic.
- The output data is left intact.
- No received data is lost.
- No new data is generated at the output ports.

B. Functional Failure Modes

Any deviation from the intended functional properties introduces a functional failure. A *functional failure mode* ω is defined by an input characteristic function (ω_{in}) and an output characteristic function (ω_{out}) for certain cycles in which the functional failure is active. For example, in the functional mode the switch receives data in packet format. This defines the input characteristic function. Moreover, the incoming data is intended to pass through the switch intact. A functional failure occurs when the outgoing data packet contains an error, for example a bit-flip. The output characteristic of such a functional failure should declare data corruption behavior at the switch outputs.

Fig. 2 presents the conceptual view of functional test pattern generation using functional failure modes. For an input vector i , let $C(i)$ be the expected functional circuit response and $C'(i)$ be the functional circuit response under a fault. Accordingly, $\omega_{out}(C(i), C'(i))$ is a Boolean formula that defines the functional mismatch between $C(i)$ and $C'(i)$. Moreover, $\omega_{in}(i)$ is a Boolean formula that defines the functional input constraints. An input vector, i.e. a functional test pattern, must satisfy the functional input constraints (ω_{in} is evaluated to true) and must cause a functional mismatch between the circuit and the faulty copy (ω_{out} is evaluated to true).

Let \mathcal{F} be the set of functional failure modes. A structural fault with respect to \mathcal{F} is *functionally redundant* when there

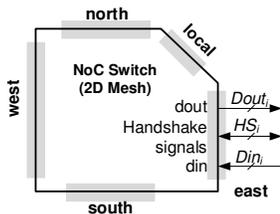


Fig. 1: A typical NoC switch with five ports

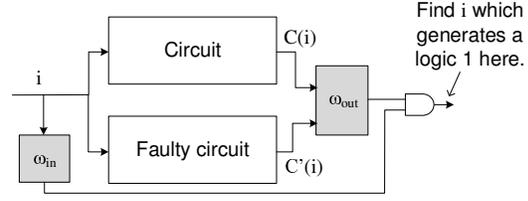


Fig. 2: The instance for functional test pattern generation

is no input vector that activates at least one functional failure in \mathcal{F} . In contrast, a fault is *functionally testable* when there is an input vector which produces one observable functional failure.

In the functional mode, switches of the network receive packets of data, thus the input characteristic function must ensure that the input data is in the packet format. For example, when a packet starts with the head flit, ends with a tail flit and fpp is the number of flits per packet, the following Boolean formula defines the input characteristic of port p at time t :

$$(din_{p,t} = \text{head}) \iff (din_{p,t+fpp-1} = \text{tail}) \quad (1)$$

In addition, the intermediate flits must be data flits, thus:

$$\bigwedge_{t < t' < t+fpp-1} din_{p,t'} = \text{data} \quad (2)$$

With respect to the switch functionality defined in section II-A, four categories of functional failures can be defined: misrouting, data corruption, packet/flit loss, garbage packet/flit. Examples for the latter functional failure mode are the so-called multiple-copy-in-space or multiple-copy-in-time failures, which produce unexpected packets at the switch outputs. The output characteristic function of each functional failure can be defined as a Boolean formula as well. For example, assume a simple switch interface with *send* as the handshake signal. *Send* is set to one whenever a valid data is sent out from the switch port. The characteristic function for a data corruption on port p of the switch is defined as follows:

$$\bigvee_t (dout_{p,t} \neq dout'_{p,t}) \wedge (send_{p,t} \wedge send'_{p,t}) \quad (3)$$

$dout'$ and $send'$ show the data output and the send signal in the presence of a fault and t is the time interval at which the functional failure is observed.

In the following section, we describe our diagnosis approach based on the preliminaries.

III. DIAGNOSIS FLOW

Fig. 3 illustrates the proposed diagnosis flow for a circuit under test (CUT) using functional tests. The **functional test patterns** (detail in section III-B) are applied to the circuit under test and the **failure signature** is extracted. The failure signature is then looked up in the **classification dictionary** (section III-A) and the suspected structural fault candidate(s) is saved for further analysis for volume diagnosis for example. On the other side, with respect to the

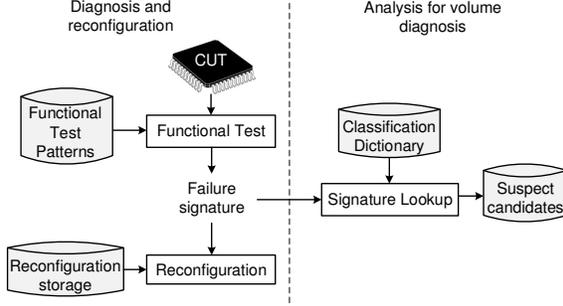


Fig. 3: Overall flow of functional diagnosis

failure signature, the reconfiguration information is picked up from the **reconfiguration storage** (section III-C). With this information the defective subcomponents are bypassed and the intact functions can be used for system operation.

In the rest, we will introduce the failure signature and discuss how the classification dictionary is filled up. In section III-B, we discuss how to generate the functional test patterns to achieve a high diagnosis resolution, and finally in section III-C, signature analysis for reconfiguration is discussed and we will show how the reconfiguration storage is filled up.

A. Classification Dictionary

Let \mathcal{S} be the set of structural faults in the circuit which are functionally testable with respect to an ordered set of the functional failure modes \mathcal{F} . For each $s \in \mathcal{S}$, the *failure signature* $\Omega_s \subseteq \mathcal{F}$, is the set of all functional failures, which may occur in the presence of the fault s is in the circuit.

Accordingly, for a set of functional failure modes of size $|\mathcal{F}|$, there exist $2^{|\mathcal{F}|}$ failure signatures. However, some signatures may never be produced by any structural fault. On the other hand, several structural faults may produce the same signature.

To extract the signature for every structural fault, searching the entire input space and all functional failure modes is not feasible. Instead, a satisfiability (SAT) solver can be employed to figure out whether there exists an input assignment to activate a functional failure under a structural fault or to prove that such an input does not exist.

The SAT instance is made based on the block diagram of Fig. 2. It includes the Boolean representation of the circuit (Tseitin transformation), the faulty copy (including the injected structural fault) as well as the input and output characteristic functions of the functional failure mode in Conjunctive Normal Form (CNF). The general fault model, Conditional Line Flip (CLF) is used to introduce structural faults to the SAT instance [10] and therefore the approach can be applied to a wide range of structural fault types that can be modeled by CLF. For a structural fault s and a functional failure mode $\omega \in \mathcal{F}$, and I^n being the input space, the SAT is satisfiable iff the following Boolean formula is true:

$$\exists i \in I^n : \omega_{in}(i) \wedge \omega_{out}(C(i), C'(i)) \quad (4)$$

Here, $C'(i)$ is the circuit response under fault s . Accordingly, the failure signature of the fault $s \in \mathcal{S}$ is defined as:

$$\Omega_s = \{\omega \in \mathcal{F} \mid \exists i \in I^n : \omega_{in}(i) \wedge \omega_{out}(C(i), C'(i))\} \quad (5)$$

As the satisfiability solver is used to find the failure signatures, it is proved that a functional failure in $\mathcal{F} \setminus \Omega_s$ is never produced by the fault s (assuming no *time out* occurs). In other words, it is proved that the fault s with respect to $\mathcal{F} \setminus \Omega_s$ is functionally redundant.

The SAT instance is constructed for all structural faults in the circuit, which are functionally testable, and the failure signature is extracted. Although this is a computationally expensive procedure, the classification dictionary must be filled up once in the design phase and stores the failure signature for structural faults in the circuit. All identical switches in the network use the same classification dictionary. Having a failure signature, the set of structural faults that produce the given signature can be also extracted from the classification dictionary.

B. Functional Test Patterns for High Diagnosis Resolution

Functional test patterns are generated with respect to the functional failure modes \mathcal{F} . To ensure the test quality, the test pattern set must have a high fault coverage, that is every nonredundant structural fault must be detectable by activating at least one functional failure mode. A constrained ATPG with fault dropping is conducted to generate a minimal functional test pattern set with full coverage of functionally testable structural faults. The input and output characteristic functions of the target functional failures are introduced to the ATPG via constraints similar to [20].

Yet with the generated functional test patterns, the structural faults may not produce exactly the same signature as stored in the classification dictionary. Therefore, the diagnosis cannot narrow down the result to exact culprits, i.e. the diagnosis resolution is low. Moreover, the succeeding reconfiguration step will not be able to avoid all the fault effects, if it is based on functional tests with an incomplete signature. The example below explains this situation:

In the sample classification dictionary of Fig. 4, the failure signature of s_4 is $\Omega_{s_4} = \{\omega_1, \omega_2, \omega_3\}$. Let us assume the case, where the functional test patterns cannot produce the correct signature for s_4 (e.g. only one or two of functional failure modes get activated). If only ω_1 and ω_3 are activated by the test patterns, then s_3 will be also recognized as a suspect. If only one functional failure is activated by the test patterns, several structural faults including s_4 are recognized as suspects. In both cases, the diagnosis resolution has been reduced.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
ω_1	X	X	X	X					
ω_2				X			X	X	X
ω_3			X	X	X	X	X		

Fig. 4: A sample classification dictionary: A set of structural faults s_1 to s_9 and the functional failures $\omega_1, \omega_2, \omega_3$

To increase the diagnosis resolution, an exhaustive search is needed to extract the signature for each structural fault with respect to the test pattern set. Similar to classification, a SAT instance is constructed to extract the signature for each structural fault. Here, the inputs are constrained to the values defined by the test patterns. The signature for structural fault s with respect to the functional test pattern set T is named $\Omega_{s,T}$.

For the faults with non-matching signature, i.e. $\Omega_{s,T} \subset \Omega_s$, the ATPG is performed one more time. Here, for each functional failure mode in $\Omega_s \setminus \Omega_{s,T}$, extra patterns are added to get the same signature for each structural fault as stored in the classification dictionary. This ensures that by applying the completed functional pattern set to a circuit and in the presence of fault s , the failure signature Ω_s is achieved.

C. Signature Analysis for Reconfiguration

Let us consider a reconfigurable system that includes a set of components K_1 to K_m . Each component drives certain functional outputs. As the output characteristic functions of the functional failure modes are defined over the functional outputs, it can be shown which component(s) has produced an observed functional failure. In other words, the observed functional failure can be masked by avoiding the respective components.

Fig. 5 gives an example of a system consisting of four components K_1 to K_4 . Consider the set of functional failure modes $\mathcal{F} = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. Let us assume that the output characteristic functions of ω_1 , ω_2 , ω_3 , and ω_4 , are defined over the outputs of the components K_1 to K_4 , respectively. If after a functional test the failure signature is $\Omega_s = \{\omega_1\}$, then by avoiding only the component K_1 the fault effect is masked. If the failure signature is $\Omega_s = \{\omega_1, \omega_2\}$, the fault effect can be masked by avoiding the components K_1 and K_2 . With a test pattern set with high diagnosis resolution, which ensures generating the failure signature for every structural fault, ω_3 and ω_4 are proven to be intact, thus K_3 and K_4 can be used without problem.

In a similar manner, as outputs of the switch ports are independent, functional failure modes are defined for individual switch ports. For example, a data corruption may occur on the outputs of the west port of the NoC mesh switch and it is easily distinguishable from a data corruption at the east port. The correspondence between the functional failures

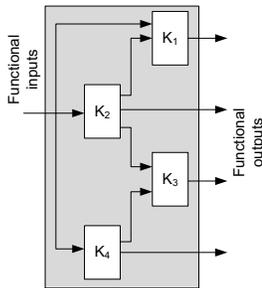


Fig. 5: A sample system with four components

and the ports of the switch to be discarded can be extracted while defining the functional failure modes. This information is saved in a storage, called *reconfiguration storage*. The only requirement is to ensure the functional test patterns generate the complete failure signature for every structural fault in the circuit, as explained in section III-B. Once the functional test patterns are applied, the failure signature is looked up from the reconfiguration storage and the respective ports to be avoided are determined. Many reconfigurable NoC architectures [2, 11] support discarding individual switch ports and offer routing algorithms to prevent deadlock or livelock.

IV. EXPERIMENTAL RESULTS

The approach is applied to a five-port switch (8-bit data width) of the mesh NoC. Fault tolerant features of the NoC allow fine-grained reconfiguration of the switches such that in case of a fault only the defective switch ports are masked.

We consider five functional failures, data corruption, flit lost, invalid flit, invalid packet and packet lost for every port of the switch, thus for a five-port switch we have 25 functional failure modes. The experiments have been conducted for stuck-at faults. For the examined cycles in the experiments, we did not observe any fault leading to misrouting. Table I shows the output characteristic functions for the functional failures observed at port p of the switch. In these formulas, T is the number of cycles in which we define the functional failure. The SAT instance for classification and test pattern generation employs the standard technique of the time frame expansion [21] to transport the sequential behavior of the switch (for T cycles) from the time domain to the space domain.

A. Classification

With respect to the functional failure modes, 5313 stuck-at faults in the switch are functionally testable. This is 83.72% of the total structural faults in the switch logic. We first construct the classification dictionary and extract the failure signature for every fault. Faults in the circuit produce 221 failure signatures. The largest failure signature class includes 263 stuck-at faults.

TABLE I: Output characteristics of the functional failure modes for port p of the switch

Functional Failure	ω_{out} for port p
Data corruption	$\bigvee_{t=1}^T (dout_{p,t} \neq dout'_{p,t}) \wedge (send_{p,t} \wedge send'_{p,t})$
Flit lost	$\bigvee_{t=1}^T (send_{p,t} \wedge \overline{send'_{p,t}})$
Packet lost	Flit lost holds for the packet length
Garbage flit	$\bigvee_{t=1}^T (\overline{send_{p,t}} \wedge send'_{p,t})$
Garbage packet	Garbage flit holds for the packet length

Studying the classification dictionary, we have determined the portion of covered stuck-at faults by each functional failure mode. The result is depicted in Fig. 6. In the horizontal axis, we have five main categories corresponding to the functional failure modes in table I. In each category, we observe the result for the ports local (L), north (N), west (W), south (S) and east (E), respectively. It is observed that data corruption and flit lost classes have covered the highest number of structural faults and thus are the most probable functional failures that might be observed due to a stuck-at fault in the switch. The vast majority of structural faults (97.76%) produce more than one functional failure mode, that is sets of structural faults in each class have some overlaps. This has the advantage to reduce the set of suspected structural faults by observing a failure signature. When some functional failures are observed, the suspect candidate belongs to the intersection of the structural faults in these functional failure classes.

Further analysis of classification dictionary shows that 142 failure signatures are generated by less than 10 structural faults, from the targeted fault model. It means that by observing one of these failure signatures, one can ensure that the suspect candidate belongs to the group of less than 10 structural faults. This already provides a good diagnosis resolution with functional tests, and can be served as a good starting point for further physical analysis, e.g. for volume diagnosis. 52 stuck-at faults are uniquely mapped to one failure signature.

For n functional failure modes, an n -bit register is used to store an observed failure signature after applying functional tests to a switch. In this register, each bit refers to a functional failure mode. If a functional failure mode is observed, a '1' is stored at the corresponding bit position in the register. For the defined functional failure modes in the experiments a 25-bit register is required. This representation is used to store the failure signature of each structural fault in the classification dictionary as well. The memory size for the classification dictionary of the target switch is 40 KB. It is noted that all identical switches in the network have the same classification dictionary.

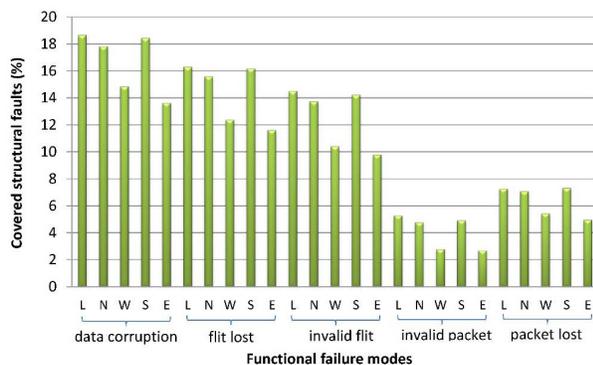


Fig. 6: The portion of covered stuck-at faults by each functional failure mode

B. Functional Test Patterns

We generate the test pattern set with high diagnosis resolution to extract the failure signature for all structural faults. For functional test of the switch, 3814 test packets are required which are applied in 36042 cycles. The test patterns set has 100% fault efficiency with respect to the targeted functional failures.

The cores surrounding a switch apply the functional test patterns to the switch under test and gather the test responses similar to [10]. The memory size for the functional patterns is 432 KB. The same set of functional patterns is used to test all identical switches of the network. The system memory or chip internal or external FLASH can be used to store this information for in-field functional diagnosis.

C. Reconfiguration

For fine-grained reconfiguration, the fault tolerant NoCs like [11] allow independent deactivation of any port of the switch. Even designs targeting only deactivation of complete switches need to support individual deactivation of ports to properly isolate a defective switch. To deactivate individual switch ports, a single bit for every port indicates the faulty / non-faulty status of that port. The reconfiguration storage is implemented by a small look up table as shown in Fig. 7, which extracts the faulty switch ports out of the observed failure signature. For the target switch and functional failure modes, the cell area for the reconfiguration storage (including the registers) is 280 unit in lsi_10k library, where the cell area of a two input NAND gate in the library is one area unit.

We analyse the functional failure signatures and find the faults which can be masked by discarding a single switch port. The result is depicted in table II. This analysis took 1.495 seconds on a 2 GHz processor. It is done once after the design phase and the information shows the quality of the defined functional failure modes for fine-grained reconfiguration.

The first row of table II declares that 12.74% of structural faults generate functional failures which are defined over the 'north' port of the switch. It is noted that these faults may produce more than one functional failure, but all of them are defined over the outputs of the 'north' port and thus can be masked by avoiding only the 'north' port. This argument holds for the other ports of the switch as well. The slight

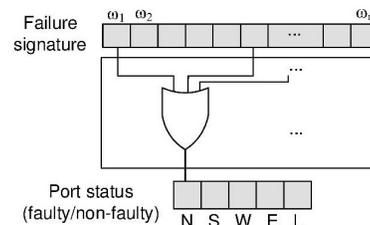


Fig. 7: A lookup table implementation of the reconfiguration storage. Input: functional failure signature, output: port status register, '0': non-faulty, '1': faulty

TABLE II: Port deactivation based on functional diagnosis results

Port	Structural faults (%)
north (N)	12.74
west (W)	12.61
south (S)	12.93
east (E)	12.27
local (L)	12.84
sum	63.93

difference of the structural faults in different switch ports comes from the logic optimization during the synthesis.

As shown in the last row of table II, all in all 63.93% of the structural faults in the switch affect a single switch port. That is the fault effect in 63.93% of defect situations in the switch can be masked by discarding a single switch port instead of deactivating the defective switch entirely. In the rest 36.07% cases the entire switch has to be deactivated. As already shown in [18] such fine-grained information improves the performability of defective NoCs, on the one side by improving the performance and on the other side by increasing the number of cores that can communicate to each other bidirectionally. In our work here, this fine-grained information is extracted only by analysing the functional test responses and without modifying the circuit structure. Moreover, our approach skips the need for huge structural diagnosis process in the field which is required for fine-grained reconfiguration.

V. CONCLUSION

This paper presented a functional diagnosis approach which extracts structural fault information for fine-grained reconfiguration. The failure signature of each structural fault is extracted and stored in a classification dictionary. The signature specifies the set of functional failure modes that are produced in the presence of a structural fault in the circuit. Moreover, the set of broken switch ports for each failure signature is extracted and saved in a reconfiguration storage.

The functional test patterns are extended to generate the predefined failure signature for each structural fault. Thus, in the presence of defects and by applying the functional test patterns to the switch under test, the failure signature is extracted. With respect to the observed signature, the structural root causes of the observed malfunctions can be looked up from the classification dictionary. Moreover, the respective switch ports to be deactivated are looked up from the reconfiguration storage. This enables a fine-grained reconfiguration of the defective NoCs by using only functional tests.

VI. ACKNOWLEDGMENT

Part of this work was supported by the German Research Foundation (DFG) under grant WU 245/10-3 (OTERA).

REFERENCES

[1] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip," *IEEE Circuits and Systems Magazine*, vol. 4, no. 2, pp. 18–31, 2004.

[2] M. Radetzki *et al.*, "Methods for fault tolerance in networks-on-chip," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, p. 8, 2013.

[3] International technology roadmap for semiconductors 2013. [Online]. Available: <http://www.itrs.net/reports.html>

[4] A. Jutman, M. Reorda, and H.-J. Wunderlich, "High quality system level test and diagnosis," in *IEEE 23rd Asian Test Symposium (ATS)*, Nov 2014, pp. 298–305.

[5] J. H. Collet *et al.*, "Comparison of fault-tolerance techniques for massively defective fine-and coarse-grained nanochips," in *16th IEEE International Conference on Mixed Design of Integrated Circuits & Systems (MIXDES)*, 2009, pp. 23–30.

[6] A. M. Amory *et al.*, "A scalable test strategy for network-on-chip routers," in *IEEE International Test Conference (ITC)*, 2005, pp. 1–9.

[7] C. Grecu *et al.*, "Testing Network-on-Chip Communication Fabrics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 12, pp. 2201–2214, 2007.

[8] J. Raik, V. Govind, and R. Ubar, "Design-for-Testability-based External Test and Diagnosis of Mesh-like Network-on-a-Chips," *IET computers & digital techniques*, vol. 3, no. 5, pp. 476–486, 2009.

[9] A. Ghofrani *et al.*, "Comprehensive online defect diagnosis in on-chip networks," in *IEEE VLSI Test Symposium (VTS)*, 2012, pp. 44–49.

[10] A. Dalirsani, M. E. Imhof, and H.-J. Wunderlich, "Structural software-based self-test of network-on-chip," in *32nd IEEE VLSI Test Symposium (VTS)*. IEEE, 2014, pp. 1–6.

[11] D. Fick *et al.*, "Vicis: A reliable network for unreliable silicon," in *46th ACM/IEEE Design Automation Conference (DAC)*, 2009, pp. 812–817.

[12] P. Maxwell, I. Hartanto, and L. Bentz, "Comparing functional and structural tests," in *IEEE International Test Conference (ITC)*, 2000, pp. 400–407.

[13] J. Zeng *et al.*, "On correlating structural tests with functional tests for speed binning of high performance design," in *IEEE International Test Conference (ITC)*, 2004, pp. 31–37.

[14] H. Fang, K. Chakrabarty, and H. Fujiwara, "RTL DFT techniques to enhance defect coverage for functional test sequences," *Journal of Electronic Testing*, vol. 26, no. 2, pp. 151–164, 2010.

[15] A. Riefert *et al.*, "An effective approach to automatic functional processor test generation for small-delay faults," in *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, March 2014, pp. 1–6.

[16] K.-C. Chen *et al.*, "A Scalable Built-In Self-Recovery (BISR) VLSI Architecture and Design Methodology for 2D-Mesh Based On-Chip Networks," *Design Automation for Embedded Systems*, vol. 15, no. 2, pp. 111–132, 2011.

[17] A. Strano *et al.*, "Exploiting network-on-chip structural redundancy for a cooperative and scalable built-in self-test architecture," in *Design, Automation Test in Europe (DATE)*, 2011, pp. 1–6.

[18] A. Dalirsani *et al.*, "Structural test for graceful degradation of NoC switches," in *16th IEEE European Test Symposium (ETS)*, 2011, pp. 183–188.

[19] P. Pande *et al.*, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Transactions on Computers*, vol. 54, no. 8, pp. 1025–1040, 2005.

[20] A. Dalirsani *et al.*, "On covering structural defects in NoCs by functional tests," in *IEEE 23rd Asian Test Symposium (ATS)*, 2014, pp. 87–92.

[21] M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital, Memory, and Mixed-signal VLSI Circuits*. Kluwer Academic, 2002.