Designing Test Structures and Analyzing Their Impact in NoC Interconnection Functional Testing

Marcos Hervé[1], Marcelo Lubaszewski[1], Erika Cota[2], Fernanda Kastensmidt[2]

[1]Departamento de Engenharia Elétrica, [2]Instituto de Informática
Universidade Federal do Rio Grande do Sul – PGMicro – Porto Alegre – RS, Brazil

ABSTRACT

Test sequences for NoCs are usually small but the test configuration time, specially for BIST-based solutions, may become the main bottleneck for overall test time reduction. In this paper we analyze, in terms of area overhead and resulting test time, three alternatives for the implementation of a functional test strategy devised for mesh NoCs. We conclude that boundary scan can be a very interesting solution for test configuration also in NoC testing, leading to a reduced test time and a programmable and reusable strategy.

1 INTRODUCTION

Fabricating a NoC-based [1] system-on-chip (SoC) with no defects in the logic and interconnect structures is a major challenge. Ensuring the reliability of the network is important to guarantee the communication between cores not only during normal mode but also during test, since the NoC is very often used as a Test Access Mechanism (TAM) for the cores [1,2]. As a result, the testing of the communication infrastructure becomes essential to guarantee the reliability of the entire system.

Recent works have been addressing the test of the NoC infrastructure, including routers [3-5] and interconnect links [6-8]. The main challenge for NoC testing is the access to the NoC resources (links and routers) which present very poor controllability and observability. In effect, NoC inter-switch links can only be accessed through the routers, whereas the routers can only be controlled by using the NoC communication protocol, i.e., through its normal mode of operation. For this reason, NoC test methods based on BIST structures and on the functional application of the test vectors are usually preferred.

Even when BIST structures are used, some sort of parameterization and configuration is always necessary, which requires an access path from an external test equipment (ATE) to the NoC. Even when functional test is applied, network interfaces (NIs) usually need to be modified to include some test structures, i.e., test wrappers must be defined and must also be accessible from the ATE. Papers on NoC testing justifiably focus on the overall test strategy, considering fault coverage, test length, and area overhead of the test structures alone, without detailing the implementation of the test structures, their configuration, and their connection to the ATE. However, this implementation may have an important impact in the overall test cost not only in terms of area overhead, but also in test time and different strategies can be used.

In this paper, we discuss the general requirements to implement a NoC test strategy and analyze the tradeoffs of such an implementation. We then show that a test wrapper based on the boundary scan standard is an interesting alternative to reduce the test configuration time, to provide means for testing the test infrastructure, and to ensure the generality of the test solution. We further compare, as a case-study, three possible implementations of the test structures for a functional interconnection test approach previously proposed. The tradeoffs of the proposed test structures implementations are analyzed in terms of test time and area overhead.

The paper is organized as follows. Section 2 discusses some NoC test approaches proposed in the literature along with the requirements for the implementation of such solutions. Section 3 details the aforementioned case study. Section 4 proposes two different TAM/wrapper approaches for the communication of the NoC with the ATE. Experimental results are presented in Section 5, while Section 6 concludes the paper.

2. NoC TESTING STRATEGIES AND TEST INFRASTRUCTURE

The test access to the network during its own test is an important issue. In addition, for the test of the interconnect links, at-speed testing and specific paths within the NoC are usually required. On the other hand, by keeping the routers in their normal operation mode, one can avoid the inclusion of extra hardware in the system critical paths while ensuring that all network signals are effectively tested.

In terms of test infrastructure, a first possibility for the test of the network is to use a direct connection to the ATE, as assumed in [2] and [4]. The implementation of these two methods is not detailed, but it is clear that a configuration on the peripheral routers is required so that externally generated test data are inserted. These approaches may lead to a minimal test configuration time, and may be ideal for a test strategy that requires very few configuration points. However, it may be a too restrictive (not generic) option. For this reason, current works are mostly based on scan for test application (or simply configuration) or BIST for test data generation and analysis.

Grecu et. al. [3] propose a test strategy for the NoC routers where FIFO buffers, once tested, are reused to transport test data to the routing logic blocks (RLBs) of the switches. The FIFOs are tested through a BIST structure that implements a functional test. The test patterns for the combinational logic, on the other hand, are injected from an external ATE through one injection port that is connected directly to one of the NoC switches. The authors assume that scan insertion was performed as a DFT strategy for the routers, although the number and configuration of this(ese) scan chain(s) are not given. From the router connected to the ATE, test vectors can be transmitted to other routers using an unicast or multicast approach.

Grecu et. al. [5] also propose a built-in self-test methodology for testing the NoC interconnect links. The proposed methodology is based on two BIST blocks, the test data generator (TDG) and the test error detector (TED). The test vectors are launched on the link under test from the transmitter side of the link, and then sampled and compared for logical consistency at the receiver side of the link by the TED circuit. Although not detailed by the authors, those strategies require a configuration step before the test, to change to operation mode of the routers and to connect the test structures to the ATE.

Cota et.al. [6] propose a functional testing method to detect pairwise shorts in interconnects of 2-D mesh NoCs . A set of test configurations based on 2x2 sub-NoCs is implemented to detect

short faults inside the refereed neighborhood in larger networks. This set of test configurations implies that the test structures must be configurable.

Analyzing the approaches mentioned above one can observe three test access mechanism (TAM) models for the network: i) an external TAM, where the peripheral routers are directly connected to the ATE. In this case, the system configuration is minimal; ii) a scan-based TAM, where one or multiple scan chains are used to control and observe the routers. In this case, these scan chains must be connected to the signals in the routers' interface; iii) a BIST-based TAM, where test data is internally generated and/or analyzed. In this case, the test data generators and analyzers must be configured through the scan chains.

We note that the scan chains play different roles in the test scheme depending on the test access mechanism used. When using a scan-based TAM (item "ii" above), the scan chains are used to deliver and remove the test vectors from and to the ATE, where the test sequence is generated and analyzed. In BIST-based approaches (item "iii" above), the scan chains are needed to configure the BIST circuits. In this article, different schemes for the configuration of the BIST-based solution are evaluated.

3. CASE STUDY

We consider in this work, the test of the NoC interconnects as proposed in [6], which consists in detecting all pairwise shorts within a 2x2 NoC (defined as a basic NoC for fault model definition) by sending packets through this basic network. Assuming the routers in functional mode, the communication protocol (packets organized as header, payload, and tail) must be used to apply test vectors to the NoC interconnects. The packets are sent in such a way that all links are filled up at the same time. The test sequence (Walking One Sequence) is placed on the packet payload.

The number of test vectors of the Walking One Sequence is equal to the entire number of wires under test. Test Data Generators (TDGs) and Test Response Analyzers (TRAs) are assumed to be connected to the network interfaces to send and receive the test packets.

Figure 1 illustrates the application of the test sequence of [6] in the basic NoC topology. Considering the XY routing strategy, this means that 4 packets can be sent across the 2x2 mesh network. The packet paths are shown in Figure 1(a) by the four different lines, one solid, one dashed, one with lines and points, and one with triple lines. For instance, for Packet 00 (solid lines) the routing path is given by one step east and one step south, from NI 0 to router 00, to router 01, to router 11, to NI 3.

All nodes use the same test sequence. The payload, however, is shifted in time so there is only one flit that contains only one bit at '1' at a time in all data wires to ensure the detection capabilities of the Walking-One sequence. Nodes are numbered from 0 to 3 and node 0 is assumed to be the first one to send test vectors, as shown in Figure 1(b).

According to [7], Equation 1 below defines the size of the test packet. The resulting test time (in number of cycles) for the defined test sequence is given by Equation 2, for *L* the latency for the test packet to arrive at the target node.

$$S = h + z_1 + 4.p + t \quad (1)$$

$$Total\ test\ time = S + L \quad (2)$$

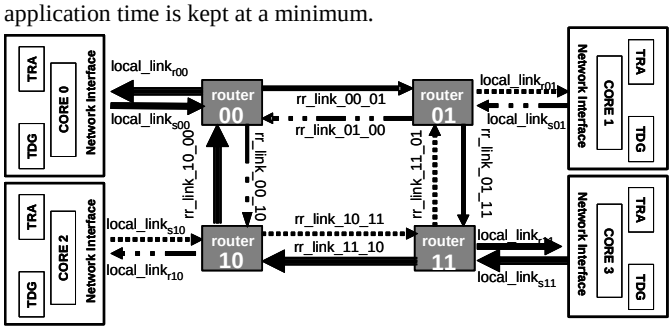
In (1) and (2), *h* is the number of flits of the header, z_1 is the number of cycles required to transmit the header from the source node to the target one, *t* is the number of flits in the tail, and *p* is the payload size given by:

$$p = w \,.\, (1 + z_3) \quad (3)$$

where *w* is the number of data wires in the link and z_3 is the number of clock cycles needed to send a payload flit from the source to the target node.

For the case-study network in [7,8], h=1, z_1= 9, z_3= 4, t=1, *w* = 8, and L = 11, which gives a total of 171 flits per test packet and a test time of 182 clock cycles for the entire 2x2 Mesh NoC.

The 2x2 basic NoC topology is used to define the minimum test configuration topology for the detection of realistic pairwise short faults affecting an on-chip network layout neighborhood. If short faults inside the refereed neighborhood are to be detected in larger networks, a set of test configurations based on 2x2 sub-NoCs must be implemented. The test configurations that can run in parallel are grouped in the same test round so as the test application time is kept at a minimum.



(a) Test strategy for the basic NoC

Node 0	Header	Z1	Payload (P flits)	P (zeros)	P (zeros)	P (zeros)	Tail
Node 1	Header	Z1	P (zeros)	Payload (P flits)	P (zeros)	P (zeros)	Tail
Node 2	Header	Z1	P (zeros)	P (zeros)	Payload (P flits)	P (zeros)	Tail
Node 3	Header	Z1	P (zeros)	P (zeros)	P (zeros)	Payload (P flits)	Tail

Time

(b) Test packet organization

Figure 1: Test strategy proposed in [6]

To implement the proposed test strategy, Test Data Generators (TDGs) and Test Response Analyzers (TRAs) must be included in the logic that connects an IP core to the network, i. e., the network interface (NI), and be activated in test mode. The TDG generates the header, a number of zero flits followed by the payload flits, and then another series of zero flits followed by the tail. The numbers of flits of zeros depend on the node and are inputs of the TDG. Data such as the link width (*w*), the tail flit, and the number of cycles a payload flit takes to traverse the network (z_3 as defined before) are constants. Although for the basic 2x2 NoC all packets have a fixed target address, for larger NoCs the actual address is variable. Thus, the header data is also an input of the TDG.

The Test Response Analyzer has a similar structure. TRA waits for the router control signal that indicates the presence of incoming flits from the network. Once the first flit arrives, the TRA reproduces the data generated by the TDG and compares each flit with the incoming flit. Comparisons are based on data coming from the ATE (the same as for the TDG) and on internal constants (modified header, tail, z_3, and *w*). If the generated data

differs from the incoming flit, an error is signaled. If the router control signal does not arrive within a certain time limit, a timeout signal is triggered, resulting in a timeout error indication.

The configuration of TDGs and TRAs shall be loaded from the ATE before the test begins and test results can be extracted from TRAs to the ATE after the test is concluded.

4. PROPOSED APPROACHES FOR TEST APPLICATION

To implement the NoC interconnect test strategy described in Section 3 one must define the structures that connect the BIST blocks (each TDG and each TRA) to the ATE, i.e., a test access mechanism for the configuration and test of those blocks. In this section, two test application methods are discussed.

4.1 Configuring TDGs and TRAs through Scan-Chains

In this option, a single scan chain can connect all test structures (TDGs and TRAs). Let us evaluate the length of the scan chain for each test module. According to the abovementioned test method, the maximum number of bits accounting for the zero flits previous to the payload and the flit with zeros after the payload will be the same for the TDG and the TRA and will be given by:

$$\text{Zeros\_before\_payload(max)}: \lceil (\log_2(z_1+3.w+3.w.\ z_3) \rceil \quad (4)$$

$$\text{Zeros\_after\_payload (max)}: \lceil (\log_2(3.w+3.w.\ z_3) \rceil \quad (5)$$

For the TDG configuration, the following additional bits are needed in the scan register: start_TDG (1 bit) and header (w+2 bits). For the TRA configuration, additional bits are also needed in the scan register, as follows: start_TRA (1 bit), time-out flag (1 bit), and error flag (1 bit).

Equations 6 and 7 give the lengths (as a function of w) of the TDG and TRA scan chains, assuming $z1$=9, $z3$=4, h=1, and L=11, as it applies to our NoC and our test sequences.

$$sc_{TDG} = 3 + w + \lceil \log_2(9+15.w) \rceil + \lceil \log_2(15.w) \rceil \quad (6)$$

$$sc_{TRA} = 3 + \lceil \log_2(9+15.w) \rceil + \lceil \log_2(15.w) \rceil \quad (7)$$

For the single scan chain configuration approach, the total NoC interconnect test application time will be given by:

$$T = 2 . m^2 . sc + C, \quad \text{for } m=2 \quad (8)$$

$$T = (tr + 1) . (m^2 . sc) + tr . C, \quad \text{for } m>2 \quad (9)$$

where $tr=4$ (number of test rounds), m^2 is the number of cores in a mxm NoC, sc is the scan length of the register used for the configuration of the TDG and TRA (sc_{TDG} + sc_{TRA}), and C is the number of test cycles required by the sequence (in the case study 182 clock cycles).

4.2. Configuring TDGs and TRAs through Boundary Scan

In this section we propose the use of a test wrapper based on the boundary scan structure for the configuration of TDGs and TRAs. The configuration bits of these structures are accommodated in this boundary scan register of the wrapper. TDI and TDO are the input and output serial ports for the test instructions and configuration bits. The test wrapper complies with the Boundary Scan Test Std. 1149.1 and, in addition to TDI and TDO, has other 3 Test Access Ports - TMS, TCK and TRST, that connect the wrapper to the ATE. The procedure to configure the TDG and the TRA consists in loading the instructions to the instruction register and loading the configuration data to the boundary scan registers. This is done by applying test signals to the Test Access Port (TAP) Controller and the correct data to the TDI port. Also, a single bit bypass register is used to pass the data through TDGs or TRAs that compose the scan chain, but are not in use for a particular test round (considering a NoC topology larger than 2x2, as in Fig.2).

The control signals for these registers are managed by the TAP Controller's Finite State Machine. The test signals for the TAP controller are driven by the ATE. Four test instructions are implemented in our wrapper: SAMPLE/PRELOAD (mandatory, but has no use in the test application), INTEST (applies the configuration bits to the TDG/TRA FSM), BYPASS (reduces the scan length when the TDG/TRA is not part of the test round to execute), and SCAN_TEST (makes it possible the scan test of the TDG/TRA).

Since in the boundary scan wrapper the time-out and error flags are embedded into the IR status register, Equations 4, 5 and 6 still apply, but Equation 7 changes to (assuming $z1$=9 and $z3$=4 as it applies to our specific NoC and test sequences):

$$sc_{TRA} = 1 + \lceil \log_2(9+15.w) \rceil + \lceil \log_2(15.w) \rceil \quad (10)$$

To compute the total NoC interconnect test application time imposed by this approach, we shall account for the scan length reduction due to the BYPASS boundary scan instruction. The number of bypassed TDGs/TRAs (Bypass) in each test round for a mxm NoC is given by Equations 11, 12, and 13 below.

For (m = 2):

$$Bypass = 0 \quad (11)$$

For (m > 2) and (m even):

$$Bypass = \begin{cases} 0 & \textit{Test round 1} \\ 2 \cdot m & \textit{Test round 2} \\ 2 \cdot m & \textit{Test round 3} \\ 2 \cdot m + 2 \cdot (m-2) & \textit{Test round 4} \end{cases} \quad (12)$$

For (m > 2) and (m odd):

$$Bypass = (2.m - 1) \qquad \text{All rounds} \quad (13)$$

Then,

$$Conf\_time = 4 + 2.m^2.IR + 4 + [m^2 . sc - 2 . bypass . (sc - 1)] + 3 \quad (14)$$

where IR is the number of bits of the instruction register (in our case IR=4) and sc is the scan length of the register used for the configuration of the TDG and TRA (sc_{TDG} + sc_{TRA}, according to Equations 6 and 10). The constant values in Equation 14 represent the number of clock cycles to move from one state to another in the TAP Controller FSM. The term $2.m^2.IR$ represents the total scan length associated to all instruction registers of TDGs and TRAs.

Finally, the total test time will be given by:

$$T = Conf\_time + C + (3 + 2.m^2.IR), \text{ for } m=2 \quad (15)$$

$$T = tr. (Conf\_time + C) + (3 + 2.m^2.IR), \text{ for } m>2 \quad (16)$$

where $tr=4$ (number of test rounds) and C is the number of test cycles required by the test sequence (in the case study, 182 clock cycles)

5. EXPERIMENTAL RESULTS

In order to evaluate the proposed technique, a 2x2 SoCIN NoC [8] was implemented in VHDL along with the Test Data Generators and Test Response Analyzers. The circuits were synthesized using the Encounter RTL Compiler (with a 0.35µm technology library) and the area results are presented in Table 1.

The second and third columns of Table 1 show the area results for the TDG and TRA, respectively, for the single scan chain implementation described in Section 4.1. The fourth and fifth columns represent the area results for the test structures including the test wrappers described in Section 4.2. The sixth column shows the area result for the TAP Controller and the last column shows the area results of the SoCIN's router RASoC [8], for comparison.

Table 1 – Test infrastructure area (number of equivalent gates)

Circuit	TDG	TRA	TDG + Test Wrapper	TRA + Test Wrapper	TAP Controller	RASoC
Size	341	401	680	684	116	1688

From the results above, it is clear that the greatest impact on the area overhead of the boundary scan wrapper comes from the TDG and TRA structures. The boundary scan infrastructure used in the test wrapper has lower impact, and mainly due to the TAP Controller as shown in Table 1. Notice also that the TDG and the TRA structures together represent from 44% (single scan chain approach) to 80% (boundary scan wrapper approach) of the area of a complete RASoC router (with 5 ports). The boundary scan implementations already include scan chains for the test of the TDG and TRA logic (finite state machines). These overheads may seem too high, however one should consider that the routers alone have a rather small contribution to the whole system area, whereas the cores and the interconnections are responsible for most of the chip occupation. Thus, we believe that the gains in terms of detection capabilities and test time provided by the proposed implementation compensate its cost.

The test time was also evaluated for three different test structure implementations: the conventional scan path with a single TDG-TRA chain, the boundary scan wrapper with a single TDG-TRA chain and the boundary scan wrapper with separate chains for TDGs and TRAs. The test time T was calculated using Equations 8, 9, 15 and 16, for mxm NoCs ($2 \leq m \leq 5$), considering w=8, 16, 32 and 64 data wires (while computing sc), IR = 4 bits, C = 182 clock cycles and tr = 4 test rounds (this is a constant value), for the three mentioned implementations. The curves are shown in Figure 2. In the Figure, from (a) to (c), it is clear that, no matter the values of w and m, the boundary scan wrapper is always a better solution in terms of test time. This is mainly due to the use of the bypass capability and the extraction of the test responses during the instruction register scan out. The best results achieved, around 50% test time reduction when compared to the conventional single scan chain, were obtained when two boundary scan chains are separately implemented for the TDGs and TRAs. In this case, the further time reduction, when compared to the implementation in Figure 2(b), is due to the fact that the lengths of the TDG and the TRA chains are no longer added, only the longest among the two chains now accounts for calculating the final configuration time.

6. CONCLUSIONS

We have analyzed three alternatives for the implementation of a functional test strategy devised for mesh NoCs. The impact in test time and area overhead of the test structures have been presented. We conclude that boundary scan can be a very interesting solution for test configuration also in NoC testing, leading to a reduction in test time of around 50%, and to a programmable and reusable strategy.

7. REFERENCES

[1] Vermeulen, B. et. al. Bringing Communication Networks on a Chip: Test and Verification Implications, IEEE Communications Magazine, Volume 41, Issue 9, Sept. 2003, pp. 74-81.

[2] Stewart, K.; Tragoudas, S.; Interconnect testing for networks on chips, VTS, 2006.

[3] Grecu, C.et al. Methodologies and algorithms for testing switch-based NoC interconnects, DFT, 2005, pp. 238 – 246.

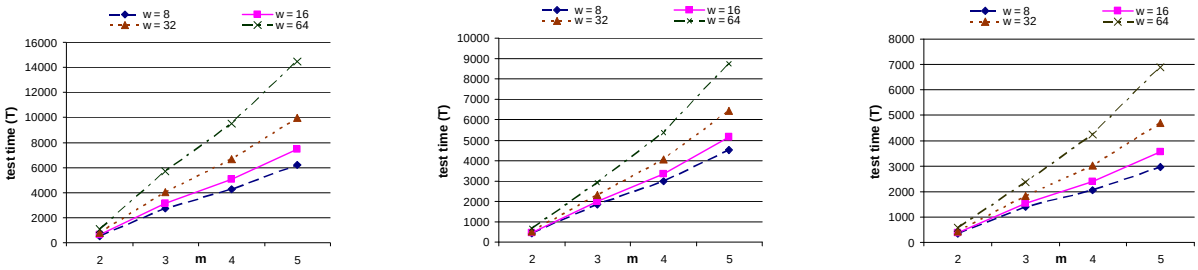
[4] Raik, J.; Govind, V.; Ubar, R.; An External Test Approach for Network-on-a-Chip Switches, ATS, 2006, pp. 437 – 442.

[5] Grecu, C. et al. .BIST for network-on-chip interconnect infrastructures, VTS 2006.

[6] Cota, E.; Kastensmidt, F. L.; Cassel, M., Meirelles, P., Amory, A., Lubaszewski, M., Redefining and Testing Interconnect Faults in Mesh NoCs. ITC, 2007.

[7] Cota, E.; Kastensmidt, F. L.; Cassel, M., Hervé, M., Alemida, P., Meirelles, P., Amory, A., Lubaszewski, M., A High Fault Coverage Approach for the Test of Data, Control and Handshake Interconnects in Mesh Networks-on-Chip. IEEE Transactions on Computers (Preprint version).

[8] Zeferino, C.A.; Kreutz, M.E.; Susin, A.A.; RASoC: a Router Softcore for Networks-on-Chip, Design, Automation and Test in Europe Conference, DATE, 2004, pp. 198-203.



(a) conventional single chain approach; (b) boundary scan wrapper with single TDG-TRA chain; (c) boundary scan wrapper with separate TDG and TRA chains

Figure 2. Test time for different link widths (*w*) and NoC sizes (*mxm*).