

# Re-Examining the Use of Network-on-Chip as Test Access Mechanism

Feng Yuan, Lin Huang and Qiang Xu

Department of Computer Science & Engineering

The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Email: {fyuan, lhuang, qxu}@cse.cuhk.edu.hk

## Abstract

Existing work on testing NoC-based systems advocates to reuse the on-chip network itself as test access mechanism (TAM) to transport test data to/from embedded cores. While this methodology obviously reduces the routing cost when compared to the case that dedicated test buses are introduced as TAMs, it is not clear whether it is beneficial in terms of other important factors that significantly affect test cost, e.g., testing time, test control complexity and test reliability. As a result, in this paper, we re-examine the issue of using NoC as TAM in order to facilitate designers to construct a cost-effective system test architecture based on their requirements<sup>1</sup>.

## 1 Introduction

Due to the scalability limitations of functional buses with the shrinking technology feature size, network-on-chip (NoC) has become a promising alternative approach to interconnect embedded cores in giga-scale system-on-a-chip (SoC) [6]. NoC typically contains three fundamental components: network interfaces (NIs) that connect cores to the NoC, routers that transport data between NIs according to pre-defined protocol, and links that connect routers and provide the raw bandwidth.

NoC has received a lot of attention recently from both academia and industry [3]. Various NoC structures have been proposed in the literature to meet systems' different requirements on performance (throughput and latency), power consumption, reliability, and implementation cost, etc. For example, for the on-chip network (OCN) topology, there are mesh, torus, hypercube, butterfly, fat tree, octagon, and irregular topologies; for the switching mechanisms, there are circuit switching and packet switching techniques; for the routing decisions, they can be static (e.g., XY routing) or dynamic (e.g., hot-potato routing).

Efficient and effective test strategies are essential for the newly-introduced NoC-based systems to reduce their manufacturing costs and meet today's stringent time-to-market requirements. While in conventional core-based SoC testing, one of the major test challenges is the test access mechanism (TAM) design, used to connect the test sources and sinks (e.g., the ATE) to the cores under test (CUTs), and the most popular and scalable solution is the dedicated bus-based TAM [15] (denoted as *DTB-TAM*). For NoC-based systems, since all the embedded cores are already connected through the on-chip network, to the best of our knowledge, all existing work advocates to reuse the NoC itself to transfer test data (denoted as *NoC-TAM*).

New design-for-test (DfT) modules need to be developed to transfer test data in NoC-TAM scheme. Embedded cores typically use standard protocols (e.g., OCP [14]) to communicate with each other. As ATE does not understand such protocols,

Amory *et al.* introduced a so-called *ATE Interface DfT* module on-chip to conduct the protocol translation. Test wrapper in the NoC-TAM scheme is also different from the one in the DTB-TAM scheme, since it also needs to do protocol conversion and buffering in addition to balancing the wrapper scan chains [2, 10]. Test architecture optimization and test scheduling is an important research topic to reduce test cost. For NoC-based systems, Cota *et al.* [5] first tackled this problem. They assumed preemptive core testing and presented sophisticated heuristics to optimize test application time. Later, Liu *et al.* [12] considered to use dedicated NoC routing path for core testing, and their test scheduling algorithms achieved reduced testing time.

While prior work with NoC-TAM scheme obviously reduces the routing cost associated with the dedicated TAM wires, it is not clear whether this is beneficial in terms of other important test cost factors, e.g., testing time, test development cost and the reliability of the test. In addition, with so many different NoC infrastructures proposed in the literature [3], it is very likely that reusing NoC as TAM provides a good test solution for some kind of NoC-based systems, but not for the others. For example, if the NoC routing paths can be flexible configured by users during test (e.g., *Aetherial NoC* [9]), the internal NoC bandwidths can be fully utilized for test data transfer as designers can select test paths freely without resource competition. Therefore, testing such NoC-based systems with NoC-TAM scheme can achieve similar testing time as that with DTB-TAM scheme. If the NoC routing mechanisms is hard-wired and cannot be freely chosen by designers, however, reusing NoC as TAM may lead to significant larger testing time.

The above has motivated us to provide a comprehensive comparison for the two test access schemes in this paper. Their main difference lies in the fact that test data are transferred through on-chip network in functional mode in NoC-TAM scheme and hence are constrained by the NoC working mechanism (e.g., routing scheme and error control mechanisms); while for the DTB-TAM scheme, however, designers have full controllability on how to transfer test data to the CUTs. In terms of testing time, instead of presenting new NoC-TAM optimization algorithms to compare with existing DTB-TAM solutions [15], we derive its theoretical lower bound and compare with the ones for DTB-TAM presented in [4, 8]. In addition, we also compare the two test strategies in terms of other test cost factors, e.g., DFT area, test control complexity and test reliability.

The remainder of this paper is organized as follows. Section 2 then details the theoretical lower bound of the testing time in NoC-TAM scheme and compare it with the one in DTB-TAM scheme shown in [4, 8]. Next, a comprehensive comparison of test cost factors is shown in Section 3. Finally, Section 4 concludes this paper.

<sup>1</sup>This work was supported in part by the Hong Kong SAR RGC Earmarked Research Grants 2150503 and 2150558.

## 2 Lower Bound on Testing Time

The modular test architecture optimization and test scheduling in DTB-TAM scheme have been subject to extensive research [15]. Existing test scheduling techniques for NoC-based systems based on the NoC reuse methodology (e.g., [5, 12]), however, are still immature. They mainly target a single type of NoC model (namely SoCIN [16]), and are difficult, if not impossible, to port to the other NoC structures. In practice, it might be necessary to design NoC-specific heuristics for different types of NoC infrastructures. Therefore, to quantify the testing time in NoC-TAM scheme and compare with the one in DTB-TAM test scheme, instead of presenting new NoC-TAM optimization algorithms for a particular NoC, we derive theoretical lower bound for generic types of NoCs in this section.

### 2.1 Problem Definition

Different from the NoC model used in some prior work (e.g., [10]) that employs dedicated test pins to connect to the ATE, we assume to reuse functional input and output (I/O) ports of some external cores as test I/O ports to deliver test data between the ATE and the NoC-based system, and the test data are first multiplexed to the closest router, i.e., the router that connect to the external cores<sup>2</sup> (as shown in Fig. 1). The problem addressed in this paper can be formulated as follows: Given is the test parameters of a set of cores  $C$ , the number of test input ports  $N_i$ , test output ports  $N_o$ , and the maximum external test bandwidth  $B_{max}$ . Furthermore, the on-chip communication network characteristics are also given, including the NoC topology, the link sharing property (e.g., non-shared or TDMA), and the routing mechanism. Derive the testing time lower bound.

### 2.2 Lower Bound in NoC-TAM Scheme

Chakrabarty [4] and Goel *et al.* [8] presented two lower bound formulations for SoC testing time when dedicated bus-based TAM is used, denoted as  $LB_{dtb}^1$  and  $LB_{dtb}^2$ , respectively. The testing time lower bound in DTB-TAM scheme can then be calculated as:  $LB_{dtb} = \max(LB_{dtb}^1, LB_{dtb}^2)$ .

There are two reasons that the testing time in NoC-TAM scheme can be larger than the one in DTB-TAM scheme. First, as the NoC is utilized for test data transfer, the external test bandwidth and the internal NoC bandwidth might not match with each other, which leads to under-utilization of the available test bandwidth and excessive testing time. Secondly, even if the internal NoC bandwidth exceeds the external test bandwidth, due to the NoC infrastructure itself, there might exist resource competition that prevents concurrent test of certain cores. In this section, we take the above into account and derive the testing time lower bound in NoC-TAM test scheme.

Generally speaking, for an embedded core  $i$ , when more test bandwidth  $b_i$  is allocated to it, its testing time  $T_{i,b_i}$  decreases. However, as shown in previous work [11], when  $b_i$  increases to a certain point that saturates all scan chains, its testing time can no longer decrease. We denote this bandwidth value as core  $i$ 's biggest effective bandwidth ( $BEB_i$ ). The lower bound for core  $i$ 's testing time can be written as follows.

$$LB_{i,noc} = \begin{cases} T_{i,BEB_i} & BEB_i \leq B_{max} \\ T_{i,B_{max}} & BEB_i > B_{max} \end{cases} \quad (1)$$

<sup>2</sup>Without this assumption, the NoC-TAM scheme may incur large package cost and also involves non-trivial routing cost for test data transfer.

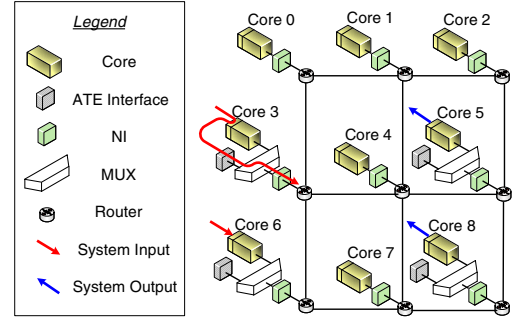


Figure 1. A NoC-Based System Example.

As mentioned above, it is possible that multiple embedded cores cannot be tested simultaneously due to the resource competition when using NoC to transfer test data. Naturally, we can conclude that the total SoC testing time cannot be smaller than the sum of the minimum testing times of these incompatible cores (this lower bound is denoted as  $LB_{noc}^1$ ). The problem to calculate  $LB_{noc}^1$  can be formulated as a graph problem. That is, we construct a *test incompatibility graph*  $G = (V, E)$ , in which each node  $v_i$  denotes an embedded core and its weight equals  $LB_{i,noc}$ , and we add an edge  $e_{ij}$  between two nodes  $v_i$  and  $v_j$  if these two cores cannot be tested simultaneously no matter which test I/O ports are utilized for them. It is obvious that the incompatible cores will form cliques in the graph. Our objective is to find the clique with the largest weight in this graph.

Consider the  $3 \times 3$  mesh NoC shown in Fig. 1, assuming X-Y routing and non-shared NoC links; functional inputs of core 3 and core 6 are reused as test input ports, while functional outputs of core 5 and 8 as test output ports. We construct the corresponding test incompatibility graph and find five cliques, i.e.,  $\{0, 1, 2\}$ ,  $\{3, 4\}$ ,  $\{6, 7\}$ ,  $\{5\}$ , and  $\{8\}$ . The total testing time must be greater than any of their total weights.

When calculating  $LB_{noc}^1$ , we mainly target the incompatibility of core tests in NoC-TAM scheme and we do not consider how test data can be allocated to the CUTs. We next derive another theoretical lower bound from the test bandwidth utilization standpoint when considering compatible core tests. The basic idea is that, the external test bandwidth might not be fully utilized during test data transfer due to resource competition. Again, consider the above example with 8-bit test I/Os, and 100MHz test frequency, thus the available test bandwidth is  $B_{max} = 800Mbps$ . Suppose the BEB of each core is shown as follows:

Core No.	0	1	2	3	4	5	6	7	8
BEB ( $\times 100Mbps$ )	3	4	6	2	3	4	3	3	2

Let us consider the test of core 0 with  $BEB_0 = 300Mbps$ . As there are only two pairs of test I/O ports, we can test one more core together with it. Due to the resource competition in NoC-TAM scheme, cores 1 and 2 cannot be tested at the same time, in the best case, when core 0 and core 5 (with  $BEB_5 = 400Mbps$ ) are scheduled to be tested simultaneously, the total utilized bandwidth is  $B_t = 700MHz$  and  $B_{max} - B_t = 100MHz$  bandwidth is wasted without being able to transfer test data.

Based on the above, for each core  $i$ , we can identify a set of cores  $S_i$  out of all its compatible cores, satisfying the following constraints: (1)  $|S_i| + 1 \leq \min(N_i, N_o)$ ; (2) all cores in  $S_i$  are compatible; (3)  $BEB_{comp}^i = \sum_{j \in S_i} BEB_j$  is the maximum.

$B_{max}$ (Mbps)	$LB_{dtb}$	6 × 5 Mesh											
		3 I/O pairs I: core 0, 6, 23 O: core 11, 17, 18				2 I/O pairs I: core 6, 23 O: core 11, 17				2 I/O pairs I: core 6, 23 O: core 11, 12			
		$LB_{noc}^1$	$LB_{noc}^2$	$LB_{noc}^3$	$\Delta(\%)$	$LB_{noc}^1$	$LB_{noc}^2$	$LB_{noc}^3$	$\Delta(\%)$	$LB_{noc}^1$	$LB_{noc}^2$	$LB_{noc}^3$	$\Delta(\%)$
1600	419937	155166	<b>419937</b>	134012	0	351396	<b>419937</b>	190106	0	145417	<b>419937</b>	190106	0
3200	209968	102965	<b>209968</b>	117485	0	<b>271404</b>	209968	160122	29.26	102965	<b>209968</b>	160122	0
4800	139979	102965	<b>139979</b>	117485	0	<b>271404</b>	170591	160122	93.89	102965	157762	<b>160122</b>	12.7
6400	104984	102965	104984	<b>117485</b>	11.91	<b>271404</b>	178279	160122	158.52	102965	<b>165863</b>	160122	57.99
8000	102965	102965	93842	<b>117485</b>	14.1	<b>271404</b>	190304	160122	163.59	102965	<b>180372</b>	160122	75.18
$B_{max}$	$LB_{dtb}$	Fat-Tree											
1600	419937	197973	<b>419937</b>	134012	0	197973	<b>419937</b>	190106	0	<b>476374</b>	419937	190106	13.44
3200	209968	117977	<b>209968</b>	117485	0	125274	<b>209968</b>	160122	0	<b>356402</b>	209968	160122	69.74
4800	139979	117977	<b>139979</b>	117485	0	117977	<b>160582</b>	160122	14.72	<b>347360</b>	163071	160122	148.15
6400	104984	<b>117977</b>	110132	117485	12.38	117977	<b>167673</b>	160122	59.72	<b>342838</b>	180448	160122	226.56
8000	102965	<b>117977</b>	107105	117485	14.58	117977	<b>181622</b>	160122	76.39	<b>340125</b>	191842	160122	230.33

$$\Delta = \frac{LB_{noc} - LB_{dtb}}{LB_{dtb}} \times 100\%: \text{Difference ratio between } LB_{noc} \text{ and } LB_{dtb}$$

**Table 1. Experimental Results for p22810.**

If  $BEB_i + BEB_{comp}^i < B_{max}$ , it is inevitable that some test bandwidth cannot be used for test data transfer. The wasted test data volume when testing core  $i$  is thus at least:

$$WDV_i = \max \{0, (B_{max} - BEB_i - BEB_{comp}^i) \times LB_{i,noc}\} \quad (2)$$

It is important to note that the  $WDV$  for a particular core might be counted several times. For instance, in the above example, when testing core 0, core 5 is chosen to be concurrently tested and  $100Mbps$  bandwidth is wasted. When we calculate  $WDV$  for core 5, it is likely to choose core 0 this time. However, because these two cores are tested simultaneously, the wasted test data volume can exist at most once. Therefore, the minimum total wasted test data volume for the system is:

$$WDV_{total} = \frac{\sum_{i \in C} WDV_i}{\min(N_i, N_o)} \quad (3)$$

We assume the total testing time for the NoC-based system equals  $LB_{dtb}^2$  when no test bandwidth is wasted<sup>3</sup>. A new testing time lower bound in NoC-TAM scheme can be calculated as follows:

$$LB_{noc}^2 = LB_{dtb}^2 + \frac{WDV_{total}}{B_{max}} \quad (4)$$

Lower bound calculations  $LB_{noc}^1$  and  $LB_{noc}^2$  do not reflect the test I/O port constraint directly. For non-shared link NoCs, the number of cores that are concurrently tested cannot exceed that of the test I/O ports. For example, when testing the NoC-based system shown in Fig. 1, as there are only two pairs of test I/O ports, any combination of three cores cannot be tested concurrently. Therefore, the testing time must be no less than the sum of the testing times of the smaller two cores in any three-core combination. From the above, we can sort all embedded cores based on their  $LB_{i,noc}$ , and a new lower bound  $LB_{noc}^3$  can be calculated by adding up the testing times of the second and third largest cores. More generally, for a system with  $N = \min(N_i, N_o)$  test I/O pairs,  $LB_{noc}^3$  can be calculated as the sum of the  $N$ th and  $(N + 1)$ th largest cores' testing times.

$LB_{noc}^1$ ,  $LB_{noc}^2$  and  $LB_{noc}^3$  complement with each other to provide an improved lower bound on testing time in NoC-TAM scheme:  $LB_{noc} = \max(LB_{noc}^1, LB_{noc}^2, LB_{noc}^3)$ .

<sup>3</sup>This depends on the wrapper design in NoC-TAM scheme and it is an approximate estimation.

### 2.3 Lower Bound Comparison

To compare the testing time lower bounds in DTB-TAM scheme and the one in NoC-TAM scheme, we use a ITC'02 benchmark SoC [13], p22810, and map it to two kinds of network topologies with non-shared links: 2-D mesh and fat-tree. For the 2-D mesh topology, every link is bi-directional and we assume X-Y routing scheme. For the fat-tree topology, it is a hierarchical architecture and every node has more than one parents. Embedded cores are at the lowest level and connect to routers at the above levels. For both topologies, we assume the NoC internal bandwidth is  $8000Mbps$ .

Table 1 compare the theoretical testing time lower bound in DTB-TAM scheme and that in NoC-TAM scheme, under various test configurations (i.e., maximum test bandwidth, network topology, number and placement of system I/O ports<sup>4</sup>). From this table we can observe that  $LB_{noc}^1$ ,  $LB_{noc}^2$  and  $LB_{noc}^3$  complement with each other to provide an improved lower bound and can be significantly higher than  $LB_{dtb}$ .

When the external test bandwidth is small (i.e., when  $B_{max} = 1600Mbps$  or  $3200Mbps$ ), there is not much difference between  $LB_{noc}$  and  $LB_{dtb}$ . However, when the external test bandwidth is getting larger,  $LB_{noc}$  becomes significantly higher than  $LB_{dtb}$  and the difference increases with the increment of external test bandwidth. This is because, the bandwidth mismatch between the external test bandwidth and NoC internal bandwidth is the main reason for the difference between the two lower bounds. When the external test bandwidth is small, the NoC internal bandwidth competition does not affect much and the system testing time is mainly constrained by the external test bandwidth. With the increase of external test bandwidth, however, the on-chip network resource competition limits the test data flow and hence significantly increases its testing time. We can also observe that when the external test bandwidth continues to increase to be close to the NoC internal bandwidth (i.e., when  $B_{max} = 8000Mbps$ ), it is possible that more bandwidth is wasted and  $LB_{noc}$  can even increase (see the bottom three cells of Column 8).

Given the same external test bandwidth, the number and position of test I/O ports also significantly affect the testing time. Generally speaking, more test interfaces imply that each core has more test access pathes and more embedded cores can be

<sup>4</sup>The test I/O ports are chosen randomly in our experiment.

	DTB-TAM	NoC-TAM
Testing Time	architecture-independent — low	NoC-dependent — maybe quite high
Routing Cost	dedicated test buses — high	wires to connect to external routers — low
DfT Area Cost	regular test wrapper — low	test wrapper with buffers; ATE interfaces — high
Test Reliability	high	easily affected by electrical noises & soft errors — low
Test Control Complexity	low	high

**Table 2. Comparison of Test Access Schemes for NoC-Based Systems.**

tested simultaneously, and hence leading to reduced test application time in NoC-TAM scheme (e.g., see Columns 3-5 and 7-9). The positions of the I/O ports available for test purpose affect the selection of routing paths and hence also influence the testing time significantly (e.g., see Columns 7-9 and 11-13).

### 3 Re-Examining NoC-TAM Test Cost

In this section, we compare the two TAM schemes in terms of other important test cost factors, as summarized in Table 2.

**Routing Cost:** In DTB-TAM test scheme, dedicated test buses are introduced to the system to connect all embedded cores, which obviously results in large routing cost and designers should carefully route test buses in order to avoid congestion. In NoC-TAM test scheme, as we reuse the on-chip network itself to transfer test data, we get significantly lower routing cost and this is the one of the main advantages to prefer NoC-TAM scheme. At the same time, we should be aware that, even in NoC-TAM scheme, there is still some routing cost to connect the ATE to the on-chip network.

**DfT Area Cost:** Test wrappers are required in both DTB-TAM scheme and NoC-TAM scheme to isolate embedded cores during test. The DfT area cost of the two schemes, however, is quite different.

One issue to be addressed when reusing NoC as TAM is the “language” barrier, i.e., NoC uses protocols like OCP, while ATE does not understand. To tackle this problem, ATE interfaces need to be introduced to conduct protocol translation and bandwidth matching [1]. Similarly, the test wrappers in NoC-TAM need to have the above functionalities in addition to the ones in conventional test wrapper designs in DTB-TAM scheme.

In addition, as discussed in [1], ATE generates continuous test data, and CUTs expect the same traffic shape with zero-jitter requirement. However, if using NoC-TAM scheme, shared channels, shared routers, and load fluctuation (i.e., test data are condensed into bursty format during transmission) render traffic jitter an inevitable phenomenon. It is therefore essential to introduce buffers into core test wrappers to eliminate jitter. The size of the buffer is determined by the test traffic jitter bound and it may dramatically increase the test wrapper area cost if the jitter bound is not well-controlled.

**Test Reliability:** One of the key assumptions when reusing NoC as TAM is that the on-chip network itself is error-free. Even though we can test the NoC first before testing embedded cores, with the ever-decreasing feature size of today’s VLSI technology and ever-increasing circuit operational frequency, failures caused by electrical noise such as crosstalk and transient errors [7] might happen during test data transfer in NoC functional mode and can render the test useless if not taking into account.

As NoC is inherently a fault-tolerant communication scheme, we generally do not expect it to function without any errors af-

ter passing manufacturing test. Different from functional mode, however, a single error happened during test data transfer will invalidate the entire test process as test data requires uncorrupted and lossless transmission. What we need to do is therefore to let the test be aware of the fault-tolerant features of the NoC. When error happens, the NoC might drop the erroneous packet or re-transmit it, etc. We need new DfT modules to inform the ATE and control the test process in order to achieve reliable testing.

As for dedicated test buses, as its operational speed is usually slow and there is not much logic existing on the buses (usually only buffers), the possibility to be affected by electrical noises and soft errors is much lower than the NoC-TAM scheme.

**Test Control Complexity:** To control the embedded core test in DTB-TAM scheme, we only need to provide test clock and scan enable signals for embedded cores. While for the NoC-TAM scheme, because the test data are broken into test packets and transmitted to the embedded cores using on-chip network in functional mode, the traffic jitter and the possible soft error require more complex test control in order not to invalidate the test results.

### 4 Conclusion

In this paper, we re-examine the cost of using NoC as TAM and compare to the one with dedicated bus-based TAM in terms of testing time, DfT area cost, test reliability and test control complexity. Our analysis facilitates designers to construct cost-effective test architectures for NoC-based systems based on their test requirements.

### References

- [1] A. M. Amory, et al. DfT for the Reuse of Networks-on-Chip as Test Access Mechanism. In *Proc. VTS*, pp. 435–440, 2007.
- [2] A. M. Amory, et al. Wrapper Design for the Reuse of Networks-on-Chip as Test Access Mechanism. In *Proc. ETS*, pp. 213–218, 2006.
- [3] T. Bjerregaard and S. Mahadevan. A survey of research and practices of network-on-chip. *ACM Comput. Surv.*, 38(1):1–54, 2006.
- [4] K. Chakrabarty. Optimal Test Access Architectures for System-on-a-Chip. *ACM TODAES*, 6(1):26–49, Jan. 2001.
- [5] E. Cota, L. Carro, and M. Lubaszewski. Reusing an on-chip network for the test of core-based systems. *ACM TODAES*, 9(4):471–499, 2004.
- [6] W. J. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proc. DAC*, pp. 18–22, 2001.
- [7] A. P. Frantz, et al. Crosstalk- and SEU-Aware Networks on Chips. *IEEE D&T*, 24(4):340–350, Apr. 2007.
- [8] S. K. Goel and E. J. Marinissen. Effective and Efficient Test Architecture Design for SOCs. In *Proc. ITC*, pp. 529–538, 2002.
- [9] K. Goossens, J. Dielissen, and A. Rădulescu. The Æthereal network on chip: Concepts, architectures, and implementations. *IEEE D&T*, 22(5):21–31, Sept-Oct 2005.
- [10] F. A. Hussin, T. Yoneda, and H. Fujiwara. Optimization of NoC Wrapper Design under Bandwidth and Test Time Constraints. In *Proc. ETS*, pp. 35–42, 2007.
- [11] V. Iyengar, K. Chakrabarty, and E. J. Marinissen. Co-Optimization of Test Wrapper and Test Access Architecture for Embedded Cores. *JETTA*, 18(2):213–230, Apr. 2002.
- [12] C. Liu, et al. Test Scheduling for Network-on-Chip with BIST and Precedence Constraints. In *Proc. ITC*, pp. 1369–1378, 2004.
- [13] E. J. Marinissen, V. Iyengar, and K. Chakrabarty. A Set of Benchmarks for Modular Testing of SOCs. In *Proc. ITC*, pp. 519–528, 2002.
- [14] Open Core Protocol Specification. <http://www.ocpip.org>.
- [15] Q. Xu and N. Nicolici. Resource-Constrained System-on-a-Chip Test: A Survey. *IEE PCDT*, 152(1):67–81, Jan. 2005.
- [16] C. Zeferino and A. Susin. SoCIN: A Parametric and Scalable Network-on-Chip. In *Integrated Circuits and Systems Design*, pp. 169–174, 2003.