J. Parallel Distrib. Comput. 71 (2011) 675–686

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

A new test scheduling algorithm based on Networks-on-Chip as Test Access Mechanisms

Alexandre M. Amory[a,*], Cristiano Lazzari[b], Marcelo S. Lubaszewski[c], Fernando G. Moraes[a]

[a] PPGCC-Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brazil
[b] INESC-ID, Lisbon, Portugal
[c] PGMICRO-Departamento de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil

ARTICLE INFO

Article history:
Received 2 March 2010
Received in revised form
14 July 2010
Accepted 14 September 2010
Available online 9 November 2010

Keywords:
SoC test
Test scheduling
Networks-on-Chip

ABSTRACT

Networks-on-Chip (NoCs) can be used for test data transportation during manufacturing tests. On one hand, NoC can avoid dedicated Test Access Mechanisms (TAMs), reducing long global wires, and potentially simplifying the layout. On the other hand, (a) it is not known how much wiring is saved by reusing NoCs as TAMs, (b) the impact of reuse-based approaches on test time is not clear, and (c) a computer aided test tool must be able to support different types of NoC designs. This paper presents a test environment where the designer can quickly evaluate wiring and test time for different test architectures. Moreover, this paper presents a new test scheduling algorithm for NoC TAMs which does not require any NoC timing detail and it can easily model NoCs of different topologies. The experimental results evaluate the proposed algorithm for NoC TAMs with an exiting algorithm for dedicated TAMs. The results demonstrate that, on average, 24% (up to 58%) of the total global wires can be eliminated if dedicated TAMs are not used. Considering the reduced amount of dedicated test resources with NoC TAM, the test time of NoC TAM is only, on average, 3.88% longer compared to dedicated TAMs.

1. Introduction

With the scaling of microchip technology, computation is becoming cheaper than communication. The main reason is that *global wires* do not scale as transistors and local wires [13] because they communicate across the chip. Global wires can be found in the chip-level communication infra-structures like buses and Networks-on-Chip (NoCs). NoCs [4] may replace global buses in the near future due to scalability, parallel communication features, and shorter global wires.

Modular testing has been proposed as a solution to test such complex SoCs [12]. The conceptual model for modular testing consists of test wrappers (used to switch between functional and test modes), test sources and sinks (used, respectively, to generate test stimuli and to compare the actual test responses to the expected responses), and Test Access Mechanisms (TAMs) (used to transport test data from/to the test pins to/from the Core-Under-Test (CUT)). The most common practice for TAM design is to include dedicated and global test buses used only for test data transportation. Since these TAMs consist of long global wires, dedicated test buses are also subject to the same interconnect problems such as signal integrity, delay, and power dissipation. In an attempt to avoid the long global wires related to the TAMs, Cota et al. [9] proposed the use of the NoC to transport test data, avoiding *long global wires* both in functional and in test modes.

The main motivation to use a NoC as TAM is to avoid the extra long global wires required to implement the dedicated TAMs. It has several potential benefits like reducing area (wiring area and buffers), layout congestion, and power dissipation over long wires with buffers. Additional motivation includes the large internal bandwidth of NoCs that can be used to optimize the test time and the support for globally asynchronous and locally synchronous systems. Testing systems with multiple clocks is a noticeable design challenge. The use of this type of NoC as TAM, which is an open research topic, could simplify the test data transportation for systems with multiples clocks or asynchronous systems.

The *original contributions* of this paper consist of an adaptable test scheduling algorithm for NoC TAM, a test environment used to quickly evaluate different test architectures, and a method to estimate the wire length required by dedicated TAMs. This paper proposes a *new test scheduling algorithm for test architecture based on NoC TAM* that requires only topology and channel bandwidth information of the NoC. This feature eases the modeling of other NoCs, for example, with different topologies, where most NoC reuse approaches existing so far are specific for mesh-based NoCs. The *test environment* has been built to ease the generation and

[*] Corresponding author.
E-mail addresses: alexandre.amory@pucrs.br, amamory@gmail.com (A.M. Amory), lazzari@inesc-id.pt (C. Lazzari), luba@ece.ufrgs.br (M.S. Lubaszewski), fernando.moraes@pucrs.br (F.G. Moraes).

0743-7315/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2010.09.008

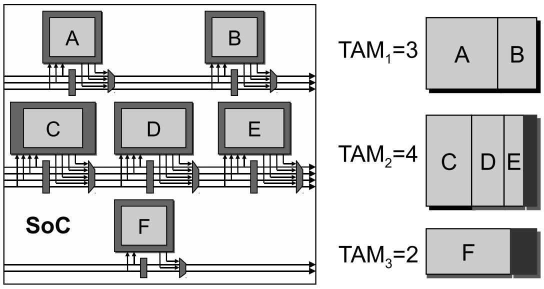

Fig. 1. Test architecture based on dedicated TAMs [12]. It presents the architectural view (left side) and test scheduling view (right side). The test scheduling has three TAMs of width 3, 4, and 2, respectively, executing the test in parallel. Cores A and B are assigned to TAM 1; cores C, D and E are assigned to TAM 2; and core F is assigned to TAM 3. The height of the box (core) is the test bandwidth and the width is core test time. The black part at TAM 2 and 3 represents idle time.

analysis of results. It consists of a set of NoC-based SoCs used as benchmarks, scripts to automate the execution and analysis of hundreds of cases, the proposed test scheduling for NoC TAM, a conventional test scheduling approach used for comparison, and the *proposed wire length estimation method*. As far as we know, this environment is the first to support both NoC-based and dedicated TAMs test architectures and is the first to numerically evaluate the wire length of dedicated TAMs. This type of environment is important because, as demonstrated in the results, both test architectures have advantages and drawbacks, thus, the designer needs an environment like this to quickly evaluate and select the most appropriate solution.

This paper is organized as follows. Section 2 summarizes the basic concepts of SoC testing for dedicated TAMs and presents an example of a test scheduling algorithm. The proposed test architecture for NoC TAM is introduced in Section 3. Section 4 presents previous papers about test scheduling for NoC TAM and compares them with the proposed approach. Section 5 presents the proposed wire length estimation method, motivating the use of the existing NoC as TAM. Section 6 presents the proposed test scheduling for NoC TAMs. Section 7 presents four sets of experiments: wire length, test time when the NoC is faster than tester, test time when the NoC is as fast as the tester, and test time for several NoC topologies. Section 8 presents the conclusion of the paper.

2. Background on SoC testing

2.1. Test architecture and test scheduling for SoCs

Given the previously presented components of a SoC test architecture (test wrapper, test source/sink, and TAM), the test scheduling and test architecture optimization can be defined as: given a set of modules and a given number of test pins, a test designer has to determine (1) the test architecture type, (2) the number of TAMs, (3) the widths of these TAMs, (4) the assignment of modules to TAMs, and (5) the wrapper design for each module, such that (a) test costs (test length, silicon overhead, performance overhead) are minimized and (b) the number of test pins is not exceeded.

One of the major challenges for SoC testing is to design an efficient TAM to link the test sources and sinks to the CUT. The research on SoC testing has been focused on dedicated test bus access, as illustrated in Fig. 1, due to its modularity and scalability. Some features of this test architecture are:

- Test application time depends on the test data bandwidth (number of test pins × transfer rate), and on the TAM distribution to cores;
- The SoC testing time can be minimized by distributing the width of each individual TAM proportionally to the amount of test data that needs to be transported to and from the cores connected to the TAM;
- The test data bandwidth might be distributed among several independent TAMs of different width;
- Multiple TAMs on a SoC operate independently and cores on a TAM are typically tested sequentially;
- Cores might be soft or hard cores. In soft cores the number and depth of scan chains can be decided during system integration, enabling balanced scan chains, improving the core test length. In hard cores the internal scan chains are fixed.

2.2. Example of test scheduling algorithm for dedicated TAMs

This section gives an example of a test scheduling algorithm for dedicated TAMs, called TR-Architect [12]. This example is detailed since the proposed test scheduling for the NoC TAM is built on top of TR-Architect. This algorithm is extended due to its simplicity and feasibility. Other test optimization algorithms are surveyed by Xu and Nicolici [24].

TR-Architect executes the following procedures in sequence: *CreateStartSolution*, *OptimizeBottomUp*, *OptimizeTopDown*, and *Reshuffle*. Fig. 2 illustrates these procedures.

The procedure *CreateStartSolution* creates the initial test architecture such that it sorts the modules of the SoC in decreasing order of test data volume, then it assigns the ordered modules to one-bit TAMs. If there are more modules than test pins, then it assigns the remaining modules to the TAM with the shortest test length. Otherwise, if there are more test pins than modules, it assigns test pins to the TAM with the longest test length.

The procedure *OptimizeBottomUp* tries to merge the TAM with the shortest test length to another TAM such that the freed test pins are allocated to the longest TAM to reduce the SoC's test length. Fig. 2(a) shows that the shortest TAM (the one with cores A and D) is merged to another TAM and its w_3 test pins are assigned to the longest TAM, reducing the SoC's test length.

The procedure *OptimizeTopDown* tries to merge the TAM with the longest test length to another TAM such that the resulting TAM receives the test pins of both original TAMs. Fig. 2(b) shows that the shortest TAM is merged to the longest TAM that receives w_3 test pins, reducing the SoC's test length.

The procedure *Reshuffle* tries to reduce the current SoC's test length by moving one module of the TAM with the longest test length to another TAM. Fig. 2(c) shows that the module D, inside the longest TAM, is moved to another TAM, reducing the SoC's test length.

3. Proposed test methodology

This section presents the proposed test approach for NoC TAMs.

3.1. Test architecture and DfT modules

Fig. 3 shows the proposed test architecture, its DfT modules, and the protocol conversions along the test data flow. The ends of the test (ATE and CUT) require test streaming, the NoC interface requires a standardized protocol such as OCP (Open Core Protocol, http://www.ocpip.org/), and the NoC internals use some network protocol, such as handshake, which is transparent to the test. The DfT modules, called ATE interface and wrapper, do all the required protocol and width conversions such that both the ATE and the CUT are not aware of the NoC [2,1].