STEPS: Experimenting a New Software-based Strategy for Testing SoCs Containing P1500-compliant IP Cores

M. Benabdenbi A. Greiner F. Pêcheux E. Viaud M. Tuna
Laboratoire LIP6, Dept ASIM
12, Rue Cuvier, 75252 Paris Cedex 05, France
{mounir.benabdenbi,alain.greiner,francois.pecheux,emmanuel.viaud,matthieu.tuna}@lip6.fr

Abstract

This paper presents STEPS, an innovative software-based approach for testing P1500-compliant SoCs. STEPS is based on the concept that the ATE is not considered as an initiator applying vectors to the SoC test pins but rather as a target, a huge repository of 32-bits test data and control commands. The ATE is connected to the functional SoC external RAM controller interface. The only additional test component in the SoC is a P1500 test processor that converts test data into serial P1500 streams. This paper applies the STEPS methodology to SoCs containing a VCI-compliant interconnect, a microprocessor, P1500 compliant IP cores and an external RAM controller interface. Using the ITC02 SoC benchmarks a comparison is done between the STEPS architecture and a classical bus-based strategy.

1 Introduction

The gap between SoCs complexity and Automated Test Equipment (ATE) capabilities is continuously growing. Thus, testing new SoCs requires expensive ATEs with higher frequency, greater accuracy and larger memory. This is mainly due to the fact that in traditional testing the ATE is considered as an active part while the SoC is the passive component.

We propose in this paper to switch the roles of the ATE and the SoC in order to reduce the need for expensive ATEs. For that purpose, we have developed a new SoC test methodology called STEPS (Software-based Test Environment for P1500 compliant SoCs), that replaces the ATE by a simple RAM of huge capacity containing the test program. The test program is actually executed on-chip by the SoC and includes test data (stimuli and expected responses) and test instructions defining what to do with the test data.

Many benefits appear using this approach. First, the test of an SoC does not depend on the ATE frequency: as the tester is a simple addressable RAM, the test frequency is the SoC one, and is only limited by the chip bond pads (test at speed). Second, a great accuracy of the ATE is no more needed since the comparison between test responses and expected ones are erformed on-chip. Third, this approach offers great flexibility in developing the test program, and moreover it induces low area overhead since SoC functional resources are reused.

The concept of reusing the embedded resources to execute a program for SoC test purpose is not new. In [1] A. Krstic et al. give a survey of the embedded software-based self testing paradigm. However, to our knowledge, none of the published approaches consider the SoC as a master and the ATE as a target. In the following, we briefly describe the STEPS basics and two kinds of architectures based on STEPS. We present the first results obtained when applying this approach to ITC'02 SoC benchmarks.

2 STEPS basics

The STEPS methodology (figure 1) can be applied to test SoCs that include: a main 32-bit microprocessor available for test purpose, a system bus or a network interconnect, an external RAM controller with a 32-bit interface and P1500-compliant IP cores [2]. Each IP (local or third-party) test informations and test patterns are also supposed available. As depicted in [1], we assume that the main microprocessor, the system bus and the RAM controller have been previously tested. STEPS first keypoint consists in introducing a single new hardware component: a P1500 processor, dedicated to SoC testing. This test processor is in charge of converting 32-bit test data into P1500 streamed test data to be applied to the right IP at the right moment. It allows IP cores to be tested concurrently. Three test modes can be selected to test the SoC depending on the requested level of accuracy. These test modes allow identifying the faulty IP, the faulty pattern or the faulty bit.

1530-1591/04 $20.00 (c) 2004 IEEE

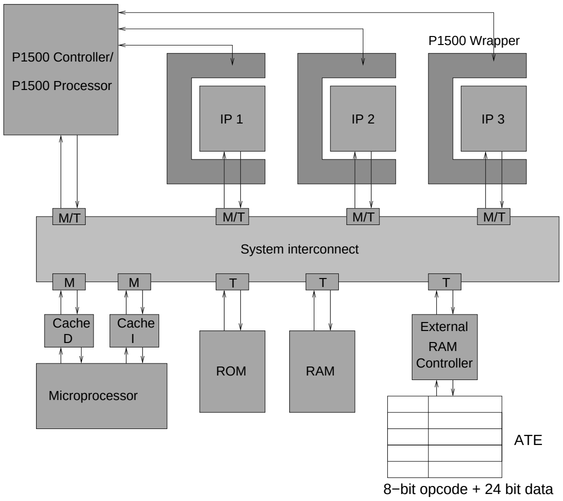

Figure 1. STEPS Architecture overview.

3 Two STEPS architectures

The second keypoint of the STEPS methodology consists in mixing test data and commands into fine-grain P1500-instructions (P1500-insts, for short), stored in the external RAM. P1500-insts are 32-bits words that contain all the required information to access the right P1500-compliant IP test interface with the right test data at the right time, according to the test protocol. As shown in Fig. 1, P1500-insts correspond to load-immediate instructions of a general purpose microprocessor, with a 8-bits opcode and 24-bits immediate data field. A sequence of successive P1500-insts form a single P1500-executable. Fig. 2 presents the two retained architectures for the P1500 controller/processor.

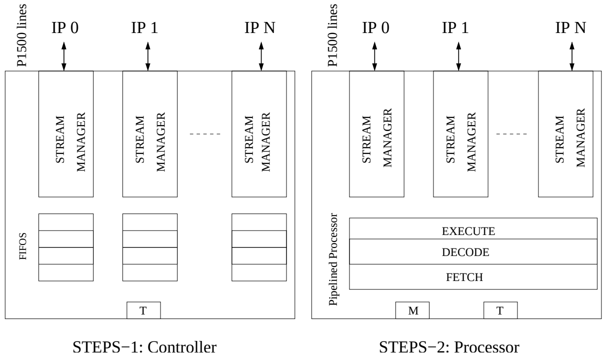

Figure 2. The two STEPS Architectures.

In STEPS-1, the P1500 controller is a target component only, and the P1500-insts located in the external RAM are interpreted by a small program, the P1500-insts interpreter, run by the embedded microprocessor. This interpreter fetches the P1500-insts from the external memory, decodes and executes them, and then feeds the appropriate P1500 controller internal fifos with correct test data. The P1500 controller main task consists in extracting the test data from fifos and converting them into P1500 streams, thanks to P1500 stream managers. P1500 Stream managers contain all the required logic to serialize/deserialize test data and compare test/expected responses. Overall performance of STEPS-1 relies on the fact that the embedded microprocessor keep feeding the fifos in such a way that it prevents the P1500 controller from data starvation.

STEPS-2 is an hardware implementation of the STEPS-1 interpreter. The P1500 controller is replaced by a P1500 processor, which is both a target and an initiator on the system bus. Once it has been correctly initialized (start address, code length, and run order) thanks to the target interface, the P1500 processor executes exactly one P1500-inst instruction per system cycle. To reach optimal performance, the P1500-insts are pipelined according to a fetch/decode/execute scheme. With a smart instruction interleaving, several IP cores can be tested in parallel.

4 Results

To evaluate the test performances in terms of test speed, we applied the STEPS methodology to ITC'02 SoC benchmarks, using System C level models and IP cores wrapped according to [2]. The results show that STEPS-2 is much more efficient than STEPS-1 (test time and area overhead). A comparison has been made between STEPS-2 and TR-Architect, a classical bus-based test strategy as described in [3]. In the case of wrapped cores equipped with the Serial Interface Layer (SIL), the STEPS-2 architecture consumes 3 to 5 time more test cycles than TR-Architect. Using cores wrapped with the Parallel Interface Layer (PIL), some encouraging preliminary results show a test cycles overhead reduced to 50 percent. However, with STEPS, the test process is executed at system speed and the test application time is not tied anymore to the ATE frequency.

References

[1] A. Krstic, L. Chen, W.-C Lai, K.-T Cheng, and S. Dey. Embedded Software-Based Self-Test for Programmable Core-Based Designs. *IEEE Design and Test of Computers*, pages 18–26, july-august 2002.

[2] Erik Jan Marinissen et al. On IEEE P1500's Standard for Embedded Core Test. *Journal of Electronic Testing: Theory and Applications*, 18(4/5):365–383, August 2002.

[3] Sandeep Kumar Goel and Erik Jan Marinissen. Effective and Efficient Test Architecture Design for SOCs. In *Proceedings IEEE International Test Conference (ITC)*, pages 529–538, Baltimore, MD, October 2002.