

A Low-cost Emulation System for Fast Co-verification and Debug

*Original*

A Low-cost Emulation System for Fast Co-verification and Debug / LAGOS BENITES, JORGE LUIS; Grosso, Michelangelo; Sterpone, Luca; SONZA REORDA, Matteo; Audisio, G.; Pipponzi, M.; Sabatini, M.. - (2011), pp. 212-212. ( IEEE European Test Symposium Trondheim (N) May 23-27, 2011) [10.1109/ETS.2011.32].

*Availability:*

This version is available at: 11583/2413925 since:

*Publisher:*

IEEE

*Published*

DOI:10.1109/ETS.2011.32

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# A Low-cost Emulation System for Fast Co-verification and Debug

J. Lagos-Benites, M. Grosso, L. Sterpone, M. Sonza Reorda  
DAUIN – Politecnico di Torino – Torino, Italy  
{jorge.lagos, michelangelo.grosso, luca.sterpone,  
matteo.sonzareorda}@polito.it

G. Audisio, M. Pipponzi, M. Sabatini  
Pirelli Tyres S.p.A. – Milano, Italy  
{giorgio.audisio, mauro.pipponzi.ex,  
marco.sabatini}@pirelli.com

**Abstract**— A flexible system for SoC co-verification is proposed, built around an Infrastructure Microprocessor (IM), providing improved controllability and observability in a fast self-contained FPGA-based emulation environment. In addition, software debug is supported by enabling observation of critical signals, breakpoint setting and step-by-step execution with total memory accessibility. Experimental results in an industrial case study confirm the effectiveness of the approach for validating and debugging hardware and software.

## SUMMARY

Guaranteeing the implementation correctness of today's integrated systems, which may include microprocessor and logic cores as well as several software layers, is becoming a daunting task. Finding design or implementation bugs requires the activation of the devices in functional mode while feeding them with input data and control signals, replicating complex and possible long operating sequences, and observing critical signals. Single hardware modules and small low-level code segments can be efficiently reproduced and analyzed in simulation, employing testbenches, or verified through formal techniques. For larger and more complex systems, simulators can hardly be employed, due to the huge requirements in terms of computational power and time. Emulation represents a viable improvement, enabling faster system behavior replication, but often with reduced system observability [1-6]. Commercial platforms are available for several architectures, based on simulators, emulators and/or programmable hardware, each one offering a specific set of features and possibly introducing a significant setup cost in the design flow.

A co-verification and debug system is here proposed (Fig.1), based on full FPGA emulation of the circuit design and on a dedicated microprocessor-based interface. The emulation environment is self-contained and runs at high speed. Complete controllability on the inspected device is achieved, including I/O signals, clocks and memories, so as to perform hardware validation/verification and supporting software debug, while the development of tests relies on writing high-level code routines. In addition, the adaption to different circuits takes minimal redesign effort, making the system particularly suitable to follow the different development steps of a new complex device. To enable effective use, a supporting software environment was developed, ranging from HDL simulators, ad-hoc flow management tools, up to high-level system models. The platform presented herein extends the capabilities of our previous work [7] by providing additional control and observation features in order to interact more deeply with the emulated system. Also, we demonstrate the possibility of adding purpose-specific verification support modules to the emulation system in order to perform specific tasks, such as bus protocol conversions or noise injection.

The proposed platform has been implemented on a Xilinx Virtex-5 LX development board, using the Xilinx MicroBlaze soft core as IM. It is currently being employed for co-verification and debug of an innovative automotive system under development by Pirelli Tyres [8]. The main integrated device, located inside a car's tire, is intended to acquire accelerometric data from the inner liner and transmit them to the main car body. Such data will then be elaborated and employed for improving braking and stability control systems. The proposed platform enabled performing noise immunity characterization of the implemented DBB radio modules and general system debugging, granting up to a 100x speed-up with respect to RTL simulation.

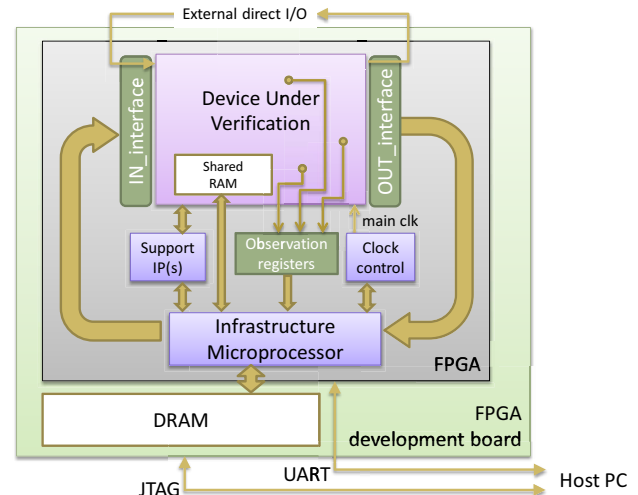


Figure 1. The proposed FPGA-based co-verification and debug platform.

## REFERENCES

- [1] R. Turner, "System-level verification – a comparison of approaches", IEEE Int'l. Workshop on Rapid System Prototyping, 1999, pp. 154-159.
- [2] Y. Nakamura, "Software verification for system on a chip using a C/C++ simulator and FPGA emulator", IEEE Int'l. Symposium on VLSI Design, Automation and Test, 2006, pp. 1-4.
- [3] C. Chuang, W. Cheng, C. Liu, D. Lu, "Hybrid approach to faster functional verification for full visibility", IEEE Design & Test of Computers, Vol. 24, N. 2, March-April 2007, pp. 154-162.
- [4] K. Oh, S. Yoon, S. Chae, "Emulator environment based on an FPGA prototyping board", IEEE Int'l. Workshop on Rapid System Prototyping, 2000, pp. 72-77.
- [5] J. Langer et al., "Debug-Tool for Embedded Real Time Systems", IEEE EUROCON Int'l. Conference on Computer as a Tool, 2005, pp. 599-602.
- [6] J. Liu, M. Zhu, J. Bian, H. Xue, "A debug sub-system for embedded-system co-verification", IEEE Int'l. Conf. on ASIC, 2001, pp. 777-780.
- [7] M. Di Marzio et al., "A novel scalable and reconfigurable emulation platform for embedded systems verification", IEEE Int'l. Symposium on Circuits and Systems, 2010, pp. 865-868.
- [8] S.C. Ergen et al., "The tire as an intelligent sensor", IEEE Trans. on CAD, Vol. 28, N. 7, 2009, pp. 941-955.