EFFICIENT TECHNIQUES AND METHODOLOGIES FOR EMBEDDED SYSTEM DESIGN USIGN FREE HARDWARE AND OPEN STANDARDS

J.I. Villar, J. Juan, M.J. Bellido

ID2 Group / Department of Electronic Technology,
University of Seville
E.T.S. de Ingeniera Informtica, Avda. Reina Mercedes s/n, 41012 Sevilla
email: jose@dte.us.es, jjchico@dte.us.es, bellido@dte.us.es

1. INTRODUCTION AND MOTIVATION

In an embedded system hardware design process, there are three main factors that have a great impact on quality, cost and development time of the final design. These factors are the availability of a library of well proven building blocks (also known as IP cores); computer tools for the design, implementation and verification of the system, and finally, the adoption of efficient methodologies well suited to the design environment and to the available set of cores and tools.

During the last few years FPGAs have experienced a great increase with regards to the achievable integration level and performance, what have made it possible to fit in a single programmable device all the parts of a microprocessor based computer or a complex digital system. This approach, so-called SoC [1] (System-on-Chip) or SoPC (System-on-Programmable-Chip) has made it necessary the adoption of new methodologies to help hardware developers with the management of the growing complexity of tackled designs.

On the other hand, the rising success of programmable devices, has also made hardware development extremely popular, reaching new target audiences as small/medium enterprises, academic environments and even individual developers that until FPGA popularization could not afford large designs with ASIC implementations. This growing popularity has promoted the creation of initiatives that try to bring to the hardware side, the success that Free Software [2] has experienced in software. Between these initiatives, one of the most relevant is Opencores [3], which offers more than 500 projects ready to use, including CPUs, Arithmetic cores and peripheral controllers among others.

In this scenario, SoC developers find that Free Hardware offers a wide range of solutions for almost any purpose, but such a great effort has been driven by individual projects with almost no coordination between them. This fact makes that Free Hardware cores are very heterogeneous regarding to multiple characteristics. The most important differences between cores can be found in the packaging and distribution methods; implementation languages; how do synthesis, implementation and simulation tools manage the sources; customization methods; interfacing with the rest of elements of the SoC, etc...

Also, open hardware designs usually suffer lacks of component testing. The number of detected bugs depends largely on the number of users and testers. Free cores often remain in the dark not only because of the absence of publicity but due to the fact that they were conceived for a particular application and later released to the community with little or no documentation. It makes really difficult for other developers to integrate and reuse several of these cores in a straightforward way into their designs.

In addition to the above, to build a complete SoC, developers not only have to deal with third party cores. There are parts that are highly dependent on concrete SoC implementations and specifications and that developers have to build for every specific design such as interconnection matrices for a defined number of masters/slaves/channels, multiple clock frequency synthesizers (phase related or not) to fulfill the needs of every component, reset generators, glue logic, etc...

This PhD project tries to develop solutions for the abovementioned problems.

2. RESEARCH LINES

This research tackles the previously described problems from four points of view, giving a comprehensive set of solutions to produce efficient design methodologies, work flows and tools. These solutions are applied to the following fields: packaging and distribution, automatic core generation, system integration and methodology.

This thesis has three main objectives:

1. Design of a core packaging and distribution standard

This work has been partially supported by the Spanish Government's MEC HYPER (TEC2007-61802/MIC) and SEPIC (TSI-020100-2008-258) projects.

978-1-4244-3892-1/09/$25.00 ©2009 IEEE

in which, as well as the source code in an HDL, packages contain machine readable descriptions of other important aspects to let automate the process of acquisition and integration of these components in larger designs. These aspects ranges from the semantics of its interfaces to implementation and functional constraints, dependencies with other cores, configuration and parametrization capabilities and all other aspects needed by other tools to integrate cores with almost no human interaction. This may be seen as an analogy to the way that software components are distributed in several GNU/Linux distributions using package repositories [4].

2. In order to fulfill the above mentioned glue logic requirements we aim at the development of tools for automatic core generation using an approach based on high level general purpose languages with code generation backends independent on the output HDL. This approach not only fills the gap of glue logic requirements, but it makes possible to raise the abstraction level of designs and allow metaprogramming techniques that let designers go one step further to where they can get using just parameterizable VHDL or Verilog. This kind of tools, combined with the proposed core packaging methods, opens a series of promising new methodological vias for SoC design.

3. Finally we aim at the definition of new work flows, methodologies and tecniques for SoC and core design, taking as main ideas the use of plug-and-play packaged cores and auto generated cores as building blocks.

A clear transverse objective of this thesis project, is the development of all the infrastructure and tools needed to implement and verify the proposed ideas. It includes the following developments:

- Development of tools and infrastructure to deploy online package repositories.
- Development of core description, packaging and distribution tools.
- Development of an automatic core generation tool based on high level languages such as Python or Java to achieve metaprogramming capabilities.
- Development of an automatic SoC generation tool that taking a simple description of the requirements and configuration as input, is able to produce an output design in synthesizable HDL. This tool is a real implementation of the methodology described in the third objective, using core repositories and autogenerated cores described in the first and the second objectives.

3. CURRENT WORK

In the current stage, the first phases of the research have been completed.

- The state of the art and and possibilities of the Free Hardware market of cores have been deeply prospected and evaluated to determine its maturity and applicability to critical designs.
- Particular co-design methodologies have been proposed [5] for several Wishbone [6] based SoC designs using Free Hardware cores and Free Software tools.
- These methodologies have been verified [7] applying them to the development of different SoCs based on Free Hardware softcore microprocessors such as OpenRisc 1200 and aeMB (Microblaze compatible) from Opencores [3], and Mico32 from Lattice [8].
- Several high level HDLs based on general purpose programming languages have been studied as a base for the core generator implementation, where promising results have arisen using the Python based MyHDL [9] approach.
- Development of a peripheral test platform based on OpenRisc 1200 applying it to the characterization of high performance SDR SDRAM and DDR SDRAM soft memory controllers.

4. REFERENCES

[1] U. Kamth and R. Kaudin, "System-on-chip designs: Stategy for success," June 2001.

[2] R. Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Free Software Foundation, 2002.

[3] Opencores, "Opencores website: http://www.opencores.org."

[4] I. Murdock, "How package management changed everything," July 2007.

[5] J. Villar, M. Bellido, E. Ostua, D. Guerrero, J. Juan, and A. M. noz, "Metodologia de diseño de soc con openrisc sobre fpga," in *Proc. 5th International Conference on Telecommunications, Electronics and Control*, vol. 1, July 2008, pp. 247–256.

[6] R. Herveille, *WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores*, 2002.

[7] J. Villar, M. Bellido, E. Ostua, D. Guerrero, J. Juan, and A. M. noz, "Metodologia de diseño de soc basada en openrisc sobre fpga con cores y herramientas libres," in *JCRA, Workshop on Reconfigurable Computing and Applications*, vol. 1, Sept. 2008, pp. 247–256.

[8] *LatticeMico32 Processor Reference Manual*. Lattice Semiconductors, 2007.

[9] J. Decaluwe, *MyHDL Manual 0.6*, 2009.