

Title: Bambu: an open-source framework for research in high-level synthesis

Webpage: http://panda.dei.polimi.it/?page_id=644

Abstract:

Hardware accelerators are becoming a key element in modern architectures to deliver energy-efficient high performance. To this end, they feature a specialized micro-architecture for both the computational logic and the storage elements to exploit spatial parallelism and efficiently execute the most computationally intensive parts of the applications. However, the design of these components requires hardware designers with specific skills to port a functionality from a software-level specification into the corresponding digital circuit. The creation of hardware accelerators becomes even more complicated when the original specification contains complex data structures or memory-related operations, which require efficient solutions for the implementation of data accesses. To tackle the design complexity, designers often rely on high-level synthesis (HLS) tools. HLS tools can automatically translate specification described in a high-level language (e.g., C, C++, or SystemC) into a circuit description in a hardware language (e.g., Verilog or VHDL) ready for the synthesis. Such tools initially required a thorough understanding of the hardware design and an extensive and error-prone code rewriting to optimize the generated micro-architecture. Their evolution, however, has reduced more and more these requirements, making them suitable to be adopted by software designers.

This tutorial presents Bambu, an open-source framework for research in high-level synthesis (http://panda.dei.polimi.it/?page_id=81). It leverages the GCC compiler to automatically generate hardware accelerators directly from the C language. Bambu is organized in a modular way so that it can be easily extended with custom passes for specific optimizations. It also features a novel memory architecture that supports a wide range of C constructs, limiting as much as possible code rewriting and, thus, simplifying the system-level integration. The tutorial is oriented to both hardware and software designers. Hardware designers will learn how to create efficient accelerators and, eventually, integrate custom modules in the generated circuits. Software designers will learn how to accelerate their applications with limited effort. We will thus present an overview of the high-level synthesis process, the tool and how it can be used to generate accelerators for several platforms (from embedded to high-performance architectures) and application domains. We will also present how to simulate and validate the accelerators generated with Bambu, as well as complex systems including third parties IPs.

Format:

This full-day tutorial will have two parts: An introduction to high-level synthesis and Bambu, and a hands-on session with access to the tool with several examples. A preliminary program is:

[Morning]

- Introduction to high-level synthesis (20 mins)
- Presentation of Bambu and compiler-based optimizations (40 mins)
- Tuning and customization of generated accelerators (45 mins)
- Synthesis and optimization of memory accesses (45 mins)
- Debugging and automated bug detection (30mins)

[Afternoon]

- Hands-on session (2h30), covering the following topics:
 - Installation and configuration of Bambu
 - Tuning of the Bambu design flow and accelerator customization
 - Optimization of the memory architecture
 - Verification of accelerators generated with Bambu and designs obtained by integrating third parties IPs

- Integration of the support for new boards
- System-level integration with standard bus protocols

What you will learn:

In this tutorial, we will explore ways of generating hardware accelerators with Bambu. At the end, you will be able to:

- Use compiler-based and HLS-specific optimizations to improve the generation of accelerators;
- Understand how to optimize the synthesis of memory accesses, including the optimization of the local accesses (e.g., by exploiting multi-bank memory modules) and the accesses to the external memory (e.g., DRAM);
- Perform automatic bug identification, especially for multi-vendor IPs;
- Generate synthesis scripts for targeting different FPGA boards;
- Generate simulation scripts and verify the results;
- Integrate the generated accelerators with standard bus protocols.

Prerequisites:

The Bambu tutorial assumes some basic knowledge about C programming (e.g., primitives and complex data types, attributes, etc.) and digital design (e.g., finite state machines, etc.). Each participant is expected to bring his/her own laptop with VirtualBox installed. A virtual machine with the installed framework will be provided.

Presenters:

- Pietro Fezzardi, PhD student, Politecnico di Milano, Italy
- Marco Lattuada, Postdoctoral Researcher, Politecnico di Milano, Italy
- Christian Pilato, Postdoctoral Researcher, University of Lugano, Switzerland
- Fabrizio Ferrandi, Associate Professor, Politecnico di Milano, Italy

Organizers and Short Bios:

- **Christian Pilato**, *Postdoctoral Researcher, University of Lugano, Switzerland*

Christian Pilato received the Laurea degree in computer engineering and the Ph.D. degree in information technology from the Politecnico di Milano, Milan, Italy, in 2007 and 2011, respectively. From 2013 to 2016, he was a Post-Doctoral Research Scientist with the Department of Computer Science, Columbia University, New York, NY, USA. He is currently a Post-Doctoral Researcher at the ALaRi institute of Università della Svizzera italiana (USI), Lugano, Switzerland. He has been visiting researcher at NanGate, Chalmers University of Technology, and Delft University of Technology.

His current research interests include high-level synthesis, reconfigurable systems and system-on-chip architectures, with emphasis on memory aspects, publishing more than 50 papers on these topics. He has actively participated to several projects sponsored by European Union, a research center supported by Semiconductor Research Corporation (SRC), and DARPA. Dr. Pilato served as the Program Chair of the Embedded and Ubiquitous Conference, in 2014. He is currently involved in the program committees of many conferences on embedded systems, CAD, and reconfigurable architectures, such as FPL, DATE, and CASES. He is a Member of the Association for Computing Machinery.

- **Marco Lattuada**, *Postdoctoral Researcher, Politecnico di Milano, Italy*

Marco Lattuada received the Master and the PhD degrees in Computer Engineering from Politecnico di Milano, Italy, in 2006 and 2010, respectively. In 2012 and 2013 he was visiting researcher at European Space Agency. Since 2010, he has been temporary researcher and lecturer at Dipartimento di Elettronica, Informazione e Bioingegneria of Politecnico di Milano.

His research interests include methodologies for embedded system design and in particular high-level synthesis, performance estimation and automatic code generation for multiprocessor heterogeneous architectures. He has actively participated to several projects sponsored by European Union and by European Space Agency. He is a Member of the Association for Computing Machinery.

- **Fabrizio Ferrandi**, Associate Professor, Politecnico di Milano, Italy

Fabrizio Ferrandi received his Laurea (cum laude) in Electronic Engineering in 1992 and the Ph.D. degree in Information and Automation Engineering (Computer Engineering) from the Politecnico di Milano, Italy, in 1997. He joined the faculty of Politecnico di Milano in 1999 as "Ricercatore" and later in 2002 as Associate Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria. His research interests include synthesis, verification simulation and testing of digital circuits and systems. Fabrizio Ferrandi is a member of the IEEE Computer Society since 1995, the Test Technology Technical Committee, and the European Design and Automation Association.