Teaching SoC Design in a Project-Oriented Course based on Robotics

Abner Barros, Péricles Lima, Juliana Xavier, Manoel E. Lima
Centro de Informática,Universidade Federal de Pernambuco, Pernambuco, Brasil
{abc, pls, jmcx, mel@cin.ufpe.br}

Abstract

The fast growing complexity and short time-to-market of embedded systems designs, besides the great increase in capacity of today's chips, are mobilizing the industry towards to the development of System-on-Chip (SoC) solutions and platform based designs. However, as the complexity of such systems grows, the problems associated with the integration of the IP-cores necessary to form the system arise as a major problem in the design flow. In a case study approach, based on robotics, teaching students the synthesis design flow in a rapid prototyping platform for SoCs, provides them the understanding and skills they need to perform such function in their subsequent employment. This paper describes a design flow, for a project-oriented course, to help designers to gain productivity by using the SoC approach on FPGAs, while avoiding integration problems by introducing automatic tools. This methodology has been used in under-graduate and graduate digital system design classes.

1. Introduction

System-on-a-chip (SoC) is a revolution in the design of integrated circuits due to the unprecedented levels of integration possibility. The key concept in SoC design is that a chip can be designed rapidly using third party and proprietary IP-cores, where IP-core refers to pre-designed behavioral or physical descriptions of a standard component [1]. However, the increasing complexity of such systems has brought a major problem to engineers: integration of heterogeneous cores has become a bottleneck of the design flow, compromising the tight time-to-market constraints. In this way, the teaching of techniques concerning the development and use of IP-cores, standard interfaces [2,3], combined with modern CAD tools, that automate the integration process is essential to successful embedded systems designs. Besides, the adoption of reconfigurable chips, like Field-Programmable Gate Arrays (FPGAs), as prototyping platforms, can help the system designer to make improvements to the system and ensure that a design meets its requirements before it is implemented as an Application Specific Integrated Circuit (ASIC), if it is needed. SoCs, implemented in such reconfigurable platforms, are commonly known as System-on-Programmable-Chip (SoPC)[4].

In this one-semester project-oriented course, a top-down SoC design flow is presented, aiming the development of a controller to a small robot, based on the Altera Nios Processor [5]. This methodology goes from the initial system specification, synthesis, validation, followed by IP-cores integration on Altera SoPC Builder tool [4]. The prototyping is performed on an Altera APEX20KE FPGA family platform board (Excalibur) [6].

In section 2 a design flow overview is presented. Sections 3 and 4 present a case study and conclusions respectively.

2. Design Flow

Initially, the course introduces a VHDL design methodology for FPGA and ASIC, including synthesis, validation techniques and technology [7-9] (Figure 1a). In the second part, a class project, base on a real application, is presented. Its specification is done in a natural language. After, a manual hardware/software partitioning is performed, according to design requirements.

An Excalibur platform, based on the Altera Nios processor, and Avalon Bus [2], is the project prototyping platform. The Avalon bus interface has a very simple protocol, allowing us to easily add new cores to the system with a minimum overhead. The SoPC Builder Tool [4] is used for integration of the IP-cores in the design. The cores can be proprietary, from Altera, or developed by users. In this class, new cores

Proceedings of the 2005 IEEE International Conference on Microelectronic Systems Education (MSE'05)
0-7695-2374-9/05 $20.00 © 2005 **IEEE**

are specified in VHDL (Figure 1b), synthesized and validated by Quartus II design tool [6]. All IP-cores implement an Avalon bus interface protocol.

After synthesis and core verification, they are integrated on the Nios CPU soft core system by using SoPC Builder tool. After synthesis, the system (SoC) is created as an unique module (hardware), ready to be tested in the platform. At the same time, a C code, the software component is developed and compiled for the Nios Processor.

At the end, the hardware bitstream is download to the platform via a JTAG interface. A shell (SDK) transfers the executable C code to the platform.

The course topics are shown in figure 1.

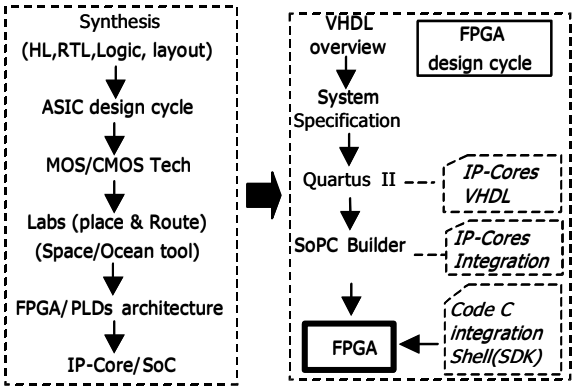


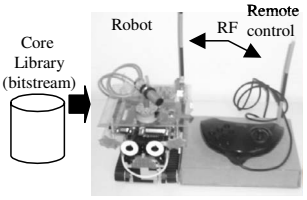
Figure 1. Course topics

3. Projects

As a project-oriented course, a robotic application (Figure 2.b) has been chosen.

In the classes, several cores have been developed in VHDL, as presented in Figure 2a.

Core	VHDL (lines)
IR	411
Dc motor	562
Stepper motor	522
RF (wireless)	1463
LCD	409



(a)VHDL library b) Ip-core library/Robot

Figure 2. Case study

In each one-semester new class, at least, a new core is added to a library core. So, depending on the semester, students can reuse core or create new ones, according to robot functionality. A C program controls the robot. Single read command to I/O devices, like wireless transmitter (RF) and sensors, are decoded by the Nios CPU, that controls the dc motor and stepper motor cores, moving the robot according to a control algorithm.

The physical structure of the robot was developed using a Lego MindStorms kit [10] and other apparatus from third party sources. In this example, the cores of controllers for two dc motors, a step motor, a digital camera, a RF unit, an infrared proximity detector, together with a Nios CPU and a Avalon bus soft core have been integrated to a FPGA APEX200KE from Altera. The C code has been downloaded into a SRAM board memory. As the results, 3000 logic elements and a 1108 lines of C code, running at 33.333 MHz have been implemented. The Excalibur board is connected to the robot via a flat cable. The robot is controlled by a RF core application implemented in an UP1 board[6].

4. Conclusions

A methodology for teaching of digital system designs based on SoPCs case study development has been presented. This works shows how easy and productive is the development of such systems base on practical case studies, by using modern design CAD tools. These practices have motivated students and enhanced learning. New cases and IP-cores are under development.

ACKNOWLEDGMENTS

We would like to thank Altera Inc. for supporting the classes through the Altera University program.

REFERENCES

[1] Wilton, S.J.E.; Saleh, R., "Programmable logic IP cores in SoC design: opportunities and challenges", Custom Integrated Circuits 2001, IEEE, pp. 63-66, 2001.

[2] Andy S. Lee, Neil W. Bergmanu., "On-chip Communication Architectures for Reconfigurable System-on-Chip", FPT2003, pp. 332 – 335, 2003.

[3] Avalon Bus Specification Reference Manual. http://www.altera.com/literature/mnl_avalon_bus.pdf.

[4] SoPC Builder User Guide. http://www.altera.com /literature/ug/ug_SoPCbuilder.pdf., 2004.

[5] Nios 3.0 CPU Datasheet. http:// www.altera.com/literature/ds/ds_nioscpu.pdf., 2004.

[6] Altera Development Kits and Design Software. http://www.altera.com, 2004.

[7] West, N., Principles of CMOS VLSI Design - A System Perspective, Addison-Wesley Publishing Company, 2000.

[8] Jerraya, A., Ding, H., Kission, P., Rahmouni, M., *Behavioral Synthesis and Component Reuse With VHDL*, Kluwer Academic Publishers, 1997.

[9] Salcic, Asim Smailagic, *Digital Systems Design and prototyping*, Kluwer Academic Publishers, 2000.

[10] Lego Mindstorms Kits http/mindstorms.lego.com, 2004.