AC 2009-1624: INCORPORATING SOFT-CORE AND HARD-CORE PROCESSORS IN CAPSTONE DESIGN COURSES

Wagdy Mahmoud, University of the District of Columbia

Incorporating Soft Core and Hard Core Processors in Capstone Design Courses

Wagdy H Mahmoud
University of the District of Columbia
wmahmoud@udc.edu

Abstract

This paper provides details of our electrical engineering program efforts to introduce sot and hard core processors and the concept of SoC in senior-level and senior-design courses. The paper provides details of laboratory exercises and a senior project that is implemented using both soft core and hard core processors on three different FPGA boards. Advantages and disadvantages of each of these implementations will also be presented. The paper will also detail the challenges involved in using continually-evolving embedded processing tools and the efforts made to reduce their learning times.

Introduction

The Accreditation Board for Engineering and Technology (ABET) requires providing students with a significant hands-on design experience. Graduating electrical engineering students should have the ability to design, test, and verify the correctness of operation of systems, subsystems, and components for real-time application.

The aggressive advancement of electronic design automation (EDA) tools coupled with the continual evolution of the semiconductor industry in terms of higher transistor densities and smaller process geometries has made it possible for design engineers to develop high performance systems-on-chip (SoC) designs for a verity of compute-intensive applications and to implement such systems on reconfigurable logic Field Programmable Gate Arrays (FPGA). The main characteristics of such complex system include: a) the use high level languages programs in designing complex systems, embedded controllers and applications; b) extensive use of intellectual property (IP) cores; b) employ hardware/software (HW/SW) co-design principles; and c) the use soft core and/or hard core embedded processors [6].

Contemporary Electronic Design Automation (EDA) industry projects exhibits the following trends: a) Increased design complexities, b) Shorter time to market, c) Increased dependency on Intellectual Property (IP) cores, d) Hardware/software co-design, and e) System on a Chip (SOC). The UDC electrical engineering program is increasing its effort to produce electrical engineers who can fulfill the needs and demands of the EDA industry.

The UDC electrical engineering senior projects employ many software packages and a selected set of FPGA boards. The software tools used in this research include the Xilinx Integrated System Environment (ISE), Xilinx embedded design kit (EDK), Xilinx ChipScope, and ModelSim from Mentor Graphics. The embedded processing system systems are based on PowerPC™405 hard core processors and MicroBlaze™ soft core processor. The Field programmable gate array (FPGA) boards are based on the Spartan-3E™, The Virtex II Pro™, and Virtex™ 4 FPGA chips. The ISE allows the design of complex digital systems and their implementation on FPGA boards. The EDK allows user to incorporate soft core processors such as Microblaze^{TM} or PicoBlaze^{TM} or to interface the built-in PowerPC processors with the reconfigurable FPGA resources. The communication functions of these processors are used to interface the hardware system to a variety of input/output peripheral devices needed to supply input data to the hardware system and/or to display the results of processing. The EDK allow interfaces to IP core libraries. The use IP cores allow designers to concentrate on the overall system design and performance without having to spend time to verify the correctness or performance of the system's components. ModelSim simulation package provide both functional and timing simulation for the hardware design. The use of system-level tools provides students with the technical skills needed for an integrative approach to real-time digital systems design, simulation, testing, and implementation. Students also acquire significant hands-on experience in design, development, and verification of complex digital systems that have potential societal benefits.

The FPGA boards used in this research include the Spartan 3E starter kit, the XUP-V2 Pro, and the DO-ML403-EDKISE-PC4-US. The reconfigurable resources of the Spartan 3E FPGA chip can be used to implement both PicoBlaze^{TM} and Microblaze^{TM} soft processor cores. The V2Pro board contains one V2pro FPGA chip with two IBM PowerPC 405 embedded processors, Audio Codec, and connectors to Gigabit serial input/output. The DO-ML403 board contains a Virtex-4 XC4VFX12 chip that is optimized for embedded processing and supports both the PowerPC™405 hard and MicroBlaze™ soft processors. Both the Spartan 3E and the LM 403 boards also contain a 16x2 character LCD. These boards allow designers to interface with per-designed IP cores or with custom-built IP cores. They can be used to implement a Configurable system-on-a-chip (CSoc) for DSP and embedded applications.

The following sections of this paper provide a brief overview of embedded processing tools used and present the details of some of the senior projects in the areas of embedded processing.

Embedded Processing Tools and Projects

Embedded systems are typically composed of interconnecting hardware and software components to perform a specific application. According to the 2006 annual embedded market survey, embedded processors accounts for over 98% of the processor market and that most new embedded system design are now using 32 bit processors. Unlike general purpose processors that are optimized for a mix of I/O and compute-intensive applications, embedded processors are low-cost processors that are optimized to perform a specific function. Embedded processors are extensively used in wide verities of applications such as mobile phone systems, automotive applications, office and home equipments, aerospace applications, and defense systems. Therefore, embedded system design courses are used to teach interdisciplinary designs.

The MicroBlaze^{TM} Architecture

The MicroBlaze is a 32-bit RISC Harvard soft processor core that can be embedded in the reconfigurable logic of an FPGA chip [1]. The MicroBlaze processor is based on the IBM CoreConnect bus architecture. The user can instantiate as many MicroBlaze soft cores inside the FPGA fabric as required by the application [4]. Associated peripheral devices for each processor such as the counter timer, interrupt controller, memory controller, and receiver/transmitter are typically implemented using IP cores. Soft core processors, such as the MicroBlaze, are typically slower and simpler than hard core processors, however, they are much easier to implement and users can acquire them for free. The MicroBlaze is a true 32-bit processor. It has thirty-two 32-bit general-purpose registers, up to 14 special-purpose registers, Arithmetic and Logic Unit (ALU), 32-bit data bus, 32-bit address bus, 32-bit instruction words, two levels of interrupts, and a single issue pipeline that can be configured as 3-stages or 5-stages. The MicroBlaze core is parameterized to allow enabling a set of configurable features. This basic design can be configured with advanced options such as: barrel shifter, memory management/memory protection unit, IEEE-754 compatible single precision floating-point unit (FPU), caches, exception handling, and debug logic. The MicroBlaze has an extensive instruction set with many of the operations have large number of variants for register, immediate, constant, signed and unsigned operands [1]. Some of these instructions such as floating point multiplication and division are optional in various hardware configurations. Optional support is provided for privilege mode, memory protection and virtual address translation. The maximum performance of the MicroBlaze processor depends on the targeted FPGA chip, and on the processor configuration. On-chip memory can be accesses using a local memory bus (LMB). or by the on-chip peripheral bus (OPB). The OBP can also be used to connect the MicroBlaze to off-chip memories and peripherals. The Fast simplex link (FSL) of the MicroBlaze allows users to interface an embedded or external hardware co-processor to accelerate the execution of compute-intensive, time-critical functions. The MicroBlaze soft core processor is a major component of the Xilinx Embedded Development Kit (EDK). The EDK also include the Xilinx Platform Studio^{TM} Tool Suite (XPS), and a library of peripheral IP soft cores. More information about the EDK tools and design techniques are found in [2].

MicroBlaze Laboratory Exercises and Senior Projects

The following set of MicroBlaze laboratory exercises aims as making student familiar with the EDK and MicroBlaze. These exercises target the Spartan-3E and the Virtex II pro boards. The list of exercises includes:

- Create a simple hardware and software processor system using the Base System Builder (BSB) and the Xilinx Intellectual Properties (IPs) available in the EDK. This exercise familiarizes students with the EDK environment and how to use it to design a reconfigurable embedded processing system and how to connect it to peripheral devices. The I/O interface uses a Universal Asynchronous Receiver/Transmitter (UART) for serial communication. The UART is designed using an IP core. Other peripheral devices are also created using available IP core library. The MicroBlaze generates the implementation netlists of the processor, and the FPGA implementation bit stream. Downloading the implementation bit stream onto the development FPGA board, the processor system is realized.
- Design an information display system to display user defined messages and information using a hyper-terminal.

The list of senior projects includes:

- Design a MicroBlaze embedded processor with multiple peripherals using both vendor-supplied and user-created custom IP cores, each of which has a different level of priority. This exercise enables students to work with interrupt in the MicroBlaze environment. Implement the design in the FPGA board.

- Design a controller for an external flash memory. The flash memory can be used to store the processor program that is used to configure the FPGA chip automatically whenever the power is turned on.
- Design a controller for the LCD display. In this project, student need to design an customized on-chip peripheral bus to connect the MicroBlaze processor to the LCD display This project also allows student to understand the timing issues involved in the display of characters on the LCD display.

The HW/SW debug of the designed systems are performed using the Xilinx ChipScope tool.

The PowerPC 405 Processor Architecture

The PowerPC 405 is a 32-bit embedded hard core RISC processor core that is designed for low-end system-on-a chip (SoC) applications [4]. Such applications include GPS devices, digital cameras, fax machines, DMA controllers, and memory controllers. Multiple PowerPC405 cores are embedded inside the Virtex-II Pro, and selected families of the Virtex-4 FPGA devices. The PowerPC 405 processor is a Harvard architecture with thirty-two 32-bit General Purpose registers, 5-stage data path pipeline with single cycle execution for most instructions, ALU with multiply and divide units, Auxiliary Processing Unit (APU), 16Kb, two-way associative Data and instruction caches, IBM CoreConnect bus architecture, embedded memory management unit, dedicated on-chip memory interface to Block RAMs (BRAMs), and debug and trace support. It is a true little-endian format. The PowerPC 405 can be integrated with other peripherals and design components using the CoreConnect^{TM} bus architecture [3]. The PowerPC 405 is marketed as a low-power consumption processor with operating frequency up to 450 MHz in the Vitex-4 FX and 400 MHz in the Virtex-II Pro devices.

Inside the Xilinx FPGA chips, the PowerPC instructions are executed using a Xilinx IP core called Fabric Co-Processing Module (FCM). Each cycle, the instruction cache unit (ICU) can supply up to two instructions to the fetch and decode units. More information about the PowerPC405 can be found in [5].

PowerPC 405 Laboratory Exercises and Senior Projects

The PowerPC 405 (PPC) laboratory exercises are similar to those of the MicroBlaze. The following set of PowerPC exercises aims as making student familiar with the PowerPC405 architecture and its instruction set. They also help student gain experience in designing useful hardware/software embedded applications that targets the Virtex-II Pro or the Virtex-4 FX devices. The set of exercises include:

- Creating a simple PPC processor to display user-defined messages and information using a hyperterminal. A UART is used to send displayed data to the hyperterminal.
- Add multiple peripherals to the PPC processor system using both vendor-supplied and user-created custom IP cores, each of which has a different level of priority. This exercise enables students to work with the PPC interrupt. The Generated netlist for the design is used to implement the design in the FPGA chip.
- Writing a basic software application to access the IP peripheral.
- Adding a timer to the designed system
- Writing a software application that uses the timer.

- Performing HW/SW system debug using the Xilinx ChipScope debugging tool.
- Creating a core using Xilinx System Generator and adds it to a PPC 405 using the On-Chip Peripheral Bus (OPB) and the Processor Local Bus (PLB) interfaces.

After completing the above exercises, students are able to employ the PowerPC 405 processor to implement a significant design project.

The senior project assigned was the development of an embedded system for image processing using the LM403 board. The system takes a grey-scale image as input using the RS233 serial cable. The image is processed using a variety of edge detection algorithms including Sobel, Roberts, and Prewitt methods. The system output processes results on VGA monitor. A discussion of various edge-detection algorithms their operators, the sequence of operation needed for their use, and the choice of parameters can be found in [7, 8]. Edge detection algorithms are basic and essential tools used in a variety of digital signal processing applications such as feature extraction, target tracking, video surveillance, improving the appearance of blurred video streams, automated inspection of semiconductor chips finding the boundaries of organs in medical images, etc.

The designed system used many of the software components available in the reference designs provided by Xilinx. Some of these components need to be modified to take advantage of the IP cores available in the current version of the EDK These modification included the use of UARTlite, the generation of addresses to connect the UARTlite to the PowerPC 405, and to modify the microprocessor software specification (MSS) to reflect the use of the UARTlite. A Screen shot of the original image is shown in Figure 1. The detected edges are shown in figures 2-4.

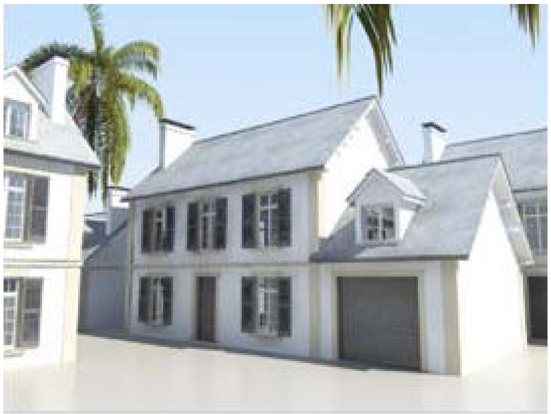

Figure 1 – Original image

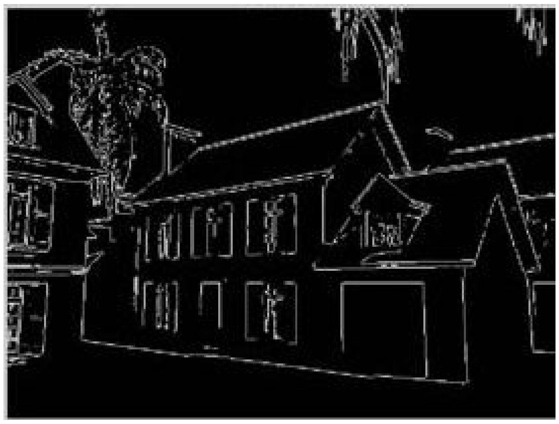

Figure 2 – edges using Sobel method

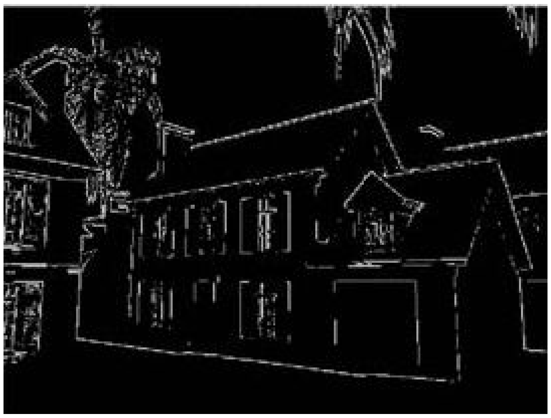

Figure 3 – edges using Robert method

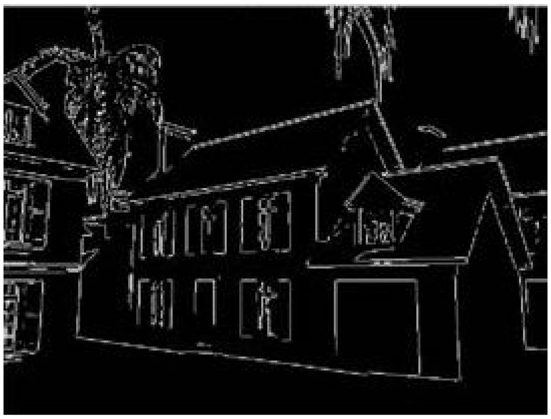

Figure 4 – edges using Perwitt method

Major Challenge and Results

Introducing system-level and embedded processing design tools into undergraduate courses creates major challenges to the instructors of these courses. These challenges include:

a) Frequency of tools updates and modifications: currently, most of the software tools are being updated twice a year. FPGA chips and boards become obsolete in a few years. Such updates usually modify the functionalities of many of the IP cores used in the design projects. In many cases, new versions of the software do not support some of the functions or boards that were supported in previous versions. This requires constant updates to the contents of the projects and the associated handouts and tutorials needed to help student complete the designs in a timely fashion.
b) Documentation sizes: Most of these tools come with lengthy user guides, references, and getting started manuals.

Student feedback and conclusions

The authors have been working on developing system-level design projects for the past few years. The introduction of soft core and hard core processors were introduced in the past year. Initial reactions and comments from students were extremely positive. Students have mastered the use of both soft core and hard core processors and gained valuable experiences in hardware/software co-design. However, more data need to be collected in the next semester to perform a meaningful analysis of the results. Such Analysis will be the subject of another paper.

References

[1] MicroBlaze™ Processor Reference Guide, Xilinx Inc., http://www.xilinx.com/ise/embedded/edk92i_docs/mb_ref_guide.pdf

[2] EDK Concepts, Tools, and Techniques—A Hands-on Guide to Effective Embedded System Design, Xilinx Inc., http://www.xilinx.com/ise/embedded/edk92i_docs/edf_ctt.pdf

[3] IBM PowerPC® ISS Reference Guide, Xilinx Inc., http://www.xilinx.com/ise/embedded/edk92i_docs/oslib_rm.pdf

[4] Clive Maxfield, “The Design Warrior’s Guide to FPGA”, Prentice Hall, 2004, ISPN 0-750676043.

[5] http://www.xilinx.com/ise/embedded/ppc405block_ref_guide.pdf

[6] Jason G. Tong, Ian D. L. Anderson, and Mohammed A. S. Khalid, “Soft-Core Processor for Embedded System,” Proceedings of the 18th International Conference on Microelectronics (ICM) 2006, Dhahran, Saudi Arabia, 2006.

[7] Rangarajan, Srikanth. “Algorithms for Edge Detection.” Stony Brook University. Web Document link: www.ee.sunysb.edu/~cvl/ese558/s2005/Reports/Srikanth%20Rangarajan/submission.doc

[8] Mohsen Sharifi, Mahmoud Fathy, Maryam Tayefeh Mahmoudi. “A Classified and Comparative Study of Edge Detection Algorithms.” IEEE International Conference on Information Technology. January 2002.