AC 2009-1140: VHDL PROJECT TUTORIAL ON ALTERA DE2 BOARD

Mouna Nakkar, University of Sharjah

VHDL Project Tutorial on Altera DE2 Board

Abstract

This paper presents a project tutorial designed for junior/senior students specializing in embedded systems. The project and tutorial provides a practical introduction to system-on-chip (SoC) design and general knowledge of FPGA and reconfigurable computing. The design is implemented on Altera DE2 board. The board is a small cost FPGA-based SoPC system designed for educational use. It has a wide range of I/O interfaces typically found in standard PC. This design can be also used in embedded system laboratory courses as well as a workshop on SoC. The tutorial is organized to be completed within 6-8 hours. A sample group of junior/senior students took this survey and the initial results are positive. The design and tutorial are available for public at the University of Sharjah website.

Introduction

A. SoC revolution

The last ten years witnessed a revolutionary shift in the traditional design of VLSI to a more modern approach which is SoC; i.e. a board with reconfigurable hardware (FPGA) and other added chips and I/O features. As a result, the embedded systems market in general increased drastically. Jackson estimated that 98% of processors are used in embedded systems[1]. Currently, the market for SoC is a $2 billion industry and is expected to grow more in the next few years.

To reflect on this, several leading universities are incorporating alternative teaching methods of Embedded Systems[1-5]. This change is of an agreement to proposals made by chief industry engineers. For example, G. Martin[6] mentioned that few universities are changing its curriculum to reflect on industry's needs. Further, he added that the industry have a shortage of SoC engineers that universities are not providing.

B. Teaching Embedded Systems/SoC/FPGA design

Despite the improvement of reconfigurable hardware, FPGA, and EDA tools associated with them, FPGA/SoC design is still a difficult pedagogical task especially for undergraduate courses. The design requires a good understating of the fundamentals of Digital Logic Design as well as advanced knowledge of systems and interface. Teaching a board's interface with several different peripherals is not an easy task. The matter gets worse with the increase of flexibility of embedded chips. The more features added to the chip, the more difficult the teaching process. Perhaps, the most effective way to teach SoC is through laboratory and well guided tutorials[7-12]

C. Design project and guided tutorial

This paper offers a tutorial design project which is geared towards junior and senior students. The project is implemented on Altera DE2 board[13]. The project is offered in three phases. These phases reflect three design steps of embedded systems: digital logic design and implementation using Hardware Description Language (HDL) utilizing the on-chip FPGA memory, interface with on-board memory and clock, and testing the system at a high frequency rate.

Hardware platform

The Altera DE2 FPGA educational board[13] shown in Fig. 1 has Cyclone II FPGA, 512 KB of SRAM, 8MB SDRAM, and 4MB of Flash and full range of I/O interfaces. The large Cyclone II FPGA has 33,216 Logical Elements and on-chip memory of 105 4K RAM blocks. These are used for internal storage and configuration. The EDA tool that comes with this chip is Quartus II 6.1 software. It is provided with the DE2 board kit. The board is designed for senior/graduate and small research projects.

Description of design project

The design problem is described in the first part of the tutorial. The goal of this project is to design a system to monitor traffic around high secure areas such as military camps or governmental departments. The system will have a record of all allowed vehicles. When a foreign vehicle passes through the vicinity, a signal is sent to the nearby police to stop this particular vehicle. The policeman will either grant permission or reject the car. If the new vehicle is permitted, the system stores its data in memory. When Memory gets full, the device will replace an old entry with the new, implementing Least Frequently Used (LFU) algorithm. The system is not limited only to traffic security; it can be used for company personnel with different ID cards or a parking lot system.

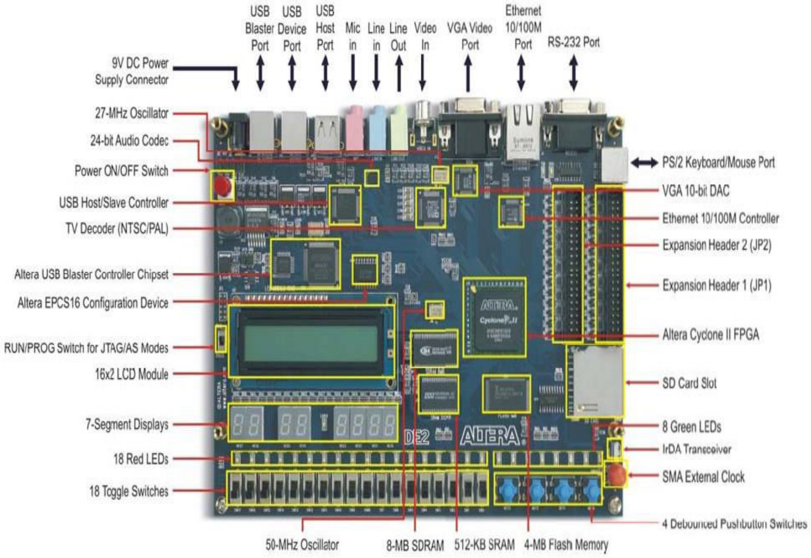

Fig. 1. A picture of the Altera DE2 Board.

This project is ideal for teaching SoC where it allows students to fully design, implement, and test a whole design. Further, this project gives the students a sense of implementing a software algorithm LFU on reconfigurable hardware.

Fig. 2 shows a block diagram of the system. It shows that the system has an interface to memory. Memory is needed to store the vehicle's license plate and the number of accesses for this vehicle to be used for the LFU algorithm.

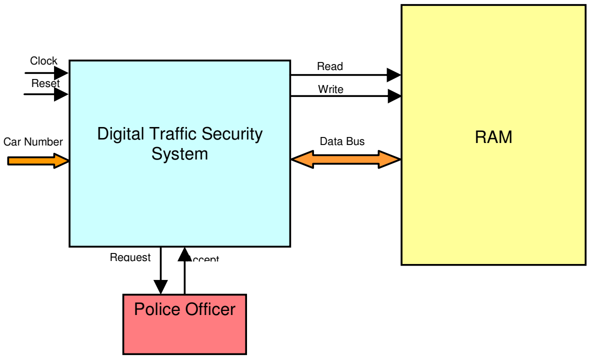

Fig. 2. A block diagram of the tutorial design project showing Traffic Security system.

Implementation and Tutorial

Phase I: basic digital logic design and synthesis.

The idea of phase I of this tutorial project is to build a small prototype of the system. This step should be easily implemented for students who took two courses of Digital Logic Design.
The design has three sequential parts: search for matching vehicle's number, search for empty memory space, and search for least frequently used vehicle as shown in state diagram in Fig. 3. A more detailed state diagram is provided in the tutorial. The design is Moore type machine where all the outputs are a function of the present state. Block diagram of the design of phase I is shown in Fig. 4. The system is implemented using VHDL on Quartus II 6.1 software provided with the Altera DE2 Board.

For simplicity the project represents the licenses plate number as an 8-bit number and the memory initially is chosen to be 16X16 bit. It will be scaled up later in phase III. In this part, the memory is designed using a generic RAM model. The on-board RAM is used in later phases. It is always easier for the student to understand small systems with small numbers and scale it up becomes an easy task.

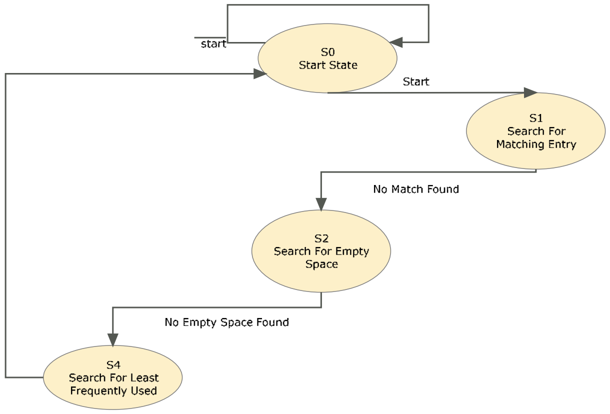

Fig. 3. State Diagram of the Traffic Security System Design.

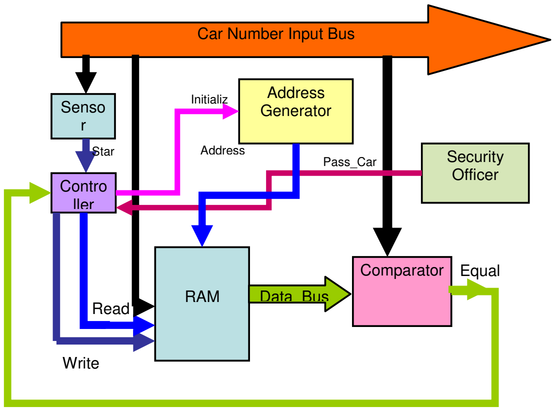

Fig. 4. State Diagram of the Traffic Security System Design.

The design is implemented, synthesized, and simulated with Quartus II 6.1 software. The simulation of this design is done with clock frequency equals to 50 MHz which is the highest on-board clock frequency. The three sequential searching methods are tested: search for a matching number in memory, search for an empty space, and search for LFU used. Pin assignment is carried out such that inputs and outputs are merely push-buttons and switches.

The design is implemented on Cyclone II processor. Table 1 shows the total number of Logical Elements used and the memory usage of this design. Fig. 5 shows Quartus II 6.1 timing simulation.

Table 1: Phase I Compilation Report

Family	CYCLONE II
Device	EP2C35F672C6
Total Logic Elements	109
Total Registers	35
Total Pins	42

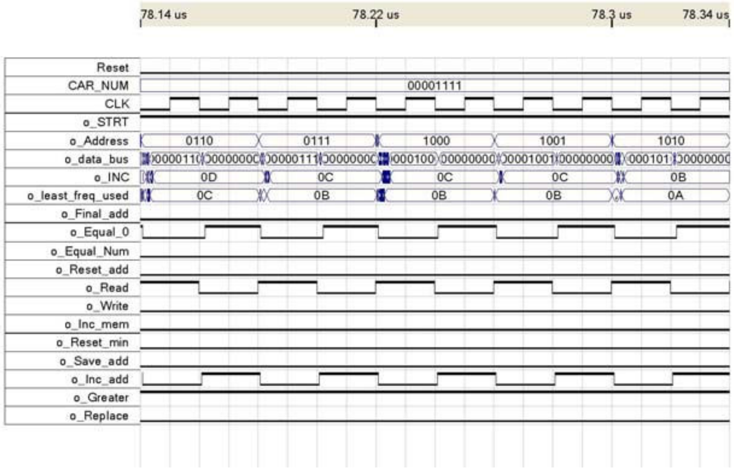

Fig. 5. Simulation results for Traffic Security System.

The Cyclone II processor is configured using Quartus II 6.1 software. The system is tested using displays and push-buttons provided on the board. It is shown in the tutorial that the system testing results match the simulation properly.

In this phase of the tutorial the students review FSM design and learn pin assignments, simulation, and testing of Altera DE2 board using Quartus II 6.1 software.

Phase II: memory interface, I/O, and timing.

Phase II of the project teaches students design interface with memory and external clock. The student must first understand memory specifications and RAM pin configurations. It is provided in the tutorial. The student will then remove the old memory module entity from the VHDL design and replace it with an interface. Fig. 6 shows block diagram of memory interface. The student then link the design with the on-chip memory using Quartus II 6.1 pin assignment.

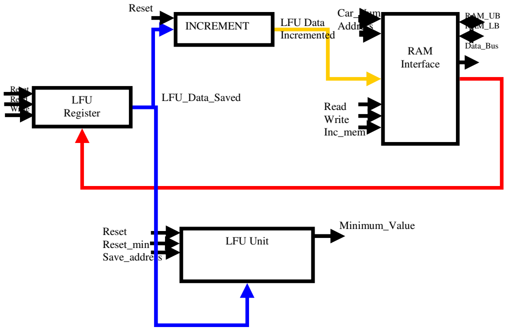

Fig. 6. Replacing of memory interface unit.

There are three types of memory provided on DE2 Board: 4-Mbyte Flash Memory, 512-Kbyte SRAM, and 8-Mbyte SDRAM. The student may choose any of the above. However, the tutorial provides guidelines to using 512K byte X 16 SRAM. The system will only use 16 entries of this big memory to simplify the testing part.

The design is synthesized and configured to the board and Table 2 shows the status of this design. The system's clock at this stage is changed to use the external clock set at a very low frequency for testing purposes.

Table 2: Phase II Compilation Report

Family	CYCLONE II
Device	EP2C35F672C6
Total Logic Elements	65
Total Registers	20
Total Pins	73

Phase III: use of internal clock, testing, and scaling.

In this phase, the students will learn how to use the internal board clock of the system. The clock used is the 50 MHz on-board clock. With this frequency, there need to be a testing method. Testing is done by running Control Panel Software provided with the Altera DE2 kit.

In addition, the students will learn in this phase scaling-up the design to utilize 256K X 16 RAM and test it. Table 3 shows Quartus II compilation report.

Table 3: Phase III Compilation Report

Family	CYCLONE II
Device	EP2C35F672C6
Total Logic Elements	205
Total Registers	34
Total Pins	102

Assessment

The design tutorial was tested on a group of 11 junior/senior students. Almost all of the students finished the three phases of tutorial within 6-8 hours, some of them took more time due to some problems with software setup, but the overall average was about 6-8 hours. All students taken this tutorial had only one course of digital design, Digital Logic Design, and most of them didn't have any exposure to the DE2 board before. The students took the tutorial without the help of the instructor or any other teaching assistants. Only one student had knowledge of FPGAs and DE2 Board.

The final questionnaire is divided into two parts: assessing the understanding of the design problem and assessing the overall quality of the tutorial.

Almost all of the students believed that the design problem is well presented. Further, they believed that this tutorial increased their sequential logic knowledge. However, when asked "can you create a simple VHDL design," the average score was 2.5 on a 5.0 scale as seen in Table 4. This is due to the fact that students were taking this tutorial on their own without an instructors' help. The written comments of the students were that this tutorial will be more of use if it was followed by an assignment of a similar design. In the future, the tutorial will incorporate some VHDL design exercises following the current tutorial and a one hour tutorial session.

Table 4: Assessment of problem description

Statement to be rated	Average *
How do you rate the problem description of this tutorial	3.6
How do you rate the increase your sequential knowledge	3.1
How do your rate the description of the simulation part	3.0
Are you able to create a simple VHDL design	2.5
*on scale of 1 to 5 (where 1 = Unsatisfactory, 2 = Average, 3 = Good , 4 = Very Good, 5 = Excellent)	

Table 5 shows the student's evaluation of the quality of the tutorial. Almost all the students thought that the tutorial is well written and organized. In addition, almost all of the students strongly recommended this tutorial.

Table 5: Assessment of tutorial quality

Statement to be rated	Average *
How do you rate the flow of the tutorial	4.9
Language used in the tutorial	4.4
Organization of the tutorial	4.2
Overall evaluation of the tutorial	3.9
Recommendation of the tutorial	4.3
*on scale of 1 to 5 (where 1 = Unsatisfactory, 2 = Average, 3 = Good , 4 = Very Good, 5 = Excellent)	

Conclusion

A design project with tutorial is presented. This project combines: digital design and implementation using VHDL, interfacing with memory and other peripherals, scaling-up, and testing. The project is implemented and tested on Altera DE2 board utilizing its memory and I/O Interface.

This tutorial was taken by a small group of students and the initial results are encouraging. Almost all of the students recommend this tutorial to others. However, there is still room for improvement especially in the design part. For example, the author feels that the students should be able to design a simple VHDL module after completing Phase I of the tutorial, but the students didn't have a strong confidence in design. This problem can be easily fixed when the tutorial is given in as a laboratory procedure followed by small similar exercises. Updates to the tutorial are in progress.

The current tutorial and design project database are available for download at this UOS website: http://www.sharjah.ac.ae/Pages/Altera_Tutorial.aspx

Bibliography

1. D. J. Jackson and P. Caspi, "Embedded systems education: future directions, initiatives, and cooperation," SIGBED Rev., vol. 2, no. 4, pp. 1-4, 2005.
2. J. O. Hamblen, "Using a Low-Cost SoC Computer and a Commercial RTOS in an Embedded Systems Design Course," *IEEE Trans. Education*, vol. 51, no. 3, Aug. 2008.
3. K. G. Ricks, D. J. Jackson, W. A. Stapleton, "An embedded systems curriculum based on the IEEE/ACM model curriculum," *IEEE Trans. Education*, vol. 51, no. 2, pp. 262-270, May 2008.
4. A. Hoover, "Computer Vision in Undergraduate Education: Modern Embedded Computing," *IEEE Trans. On Education*, vol. 46, No. 2, May 2003.
5. A. Bindal, S. Mann, B. N. Ahmed, and L.A. Raimundo, "An undergraduate system-on-chip (SoC) course for Computer Engineering students," *IEEE Trans. Education*, vol. 48, no. 2, May 2005.
6. G. Martin, "Industry needs and expectations of SoC design education," *in Proc. Int. Conf. Microelectronics Systems Education*, Anaheim, CA, Jun. 2003, pp. 146-147.
7. T. S. Hall and J. O. Hamblen, "System-on-a-programmable-chip development platforms in the classroom," *IEEE Trans. Education*, vol. 47, no. 4, pp. 502-507, Nov. 2004.
8. V. Sklyarov and I. Skiarova, "Teaching reconfigurable systems: methods, tools, tutorials, and projects," *IEEE Trans. Education*, vol. 48, no. 2, May 2005.
9. R. B. Foist, C. S. Grecu, A. Ivanov, and R. F.B. Turner, "An FPGA design project: creating a PowerPC subsystem plus user logic," IEEE Trans. Education, vol. 51, no. 3, 2008.
10. T. S. Hall and J.O. Hamblen, "Using and FPGA processor core and embedded Linux for senior design projects," *in Int. Conf. Microelectronic Systems Education*, Jun. 2007, pp. 33-34.
11. T.S. Hall and J.O Hamblen, "Engaging undergraduate student with robotic design projects," *in Sec. IEEE Int. Electronic Design, Test and Applications*, Jan. 2004, pp. 140-145.
12. J.O. Hamblen, T.S. Hall, and M. Furman, *Rapid Prototyping of Digital Quartus II edition*, Springer, August 2005.
13. Altera Corporation, Embedded Processor Web Site, http://www.altera.com