Logic Emulators in Digital Systems Education

Andrej Trost
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
e-mail: andrej.trost@fe.uni-lj.si

Baldomir Zajc
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
e-mail: baldomir.zajc@fe.uni-lj.si

Abstract— **Logic emulators are extensively used for research and educational purpose in the field of digital systems design. Development of programmable devices enabled hardware emulation of complex digital circuits and systems. Comparing to the traditional digital model simulation, the physical implementation of the circuit on the emulator platform enables real application research. There are a lot of commercial programmable development boards available from the vendors of the FPGA devices. They are typically equipped with plenty of peripheral units and designed as stand-alone boards. On the other hand, a custom system can be specifically designed for prototyping and verification requirements of the user, e.g. verification of IP cores for a specific bus.**

This paper presents a custom programmable logic emulator developed in our laboratory and the design tools used in digital circuits education. The architecture of the emulator is based on two programmable devices with a PC interface. We will present some educational applications of the logic emulator. Students begin with learning the basic digital design principles using traditional tools and a novel micro-operation based design entry method. They are encouraged to study all the details of the programmable board and the design software. Later they are able to implement required course or diploma projects and master the whole design process.

Keywords: Logic emulators, FPGA, IP cores, educational tools

I. INTRODUCTION

Development of programmable technology, particularly Field Programmable Gate Arrays (FPGA), caused a huge change in digital logic design. Programmable devices enable prototyping implementation of digital systems and a rapid application development. Traditional circuit implementation in a mask-programmable technology such as mask programmable gate arrays or standard cells typically requires extensive manufacturing effort that results in a high cost and relatively long implementation period of several weeks [1].

The FPGA device is used as a replacement of the custom designed digital integrated circuit effectively emulating its functionality. The emulated circuit can be embedded in the target system working at real-time speed. The FPGA devices are able to realize complex systems consisting of HW and SW components [2]. Programmable devices are used for hardware verification of new digital designs and are due to decreasing costs also embedded in final products.

Digital circuit emulators based on programmable devices are extensively used for research and educational purpose [3]. In our Laboratory for Integrated Circuits Design at University of Ljubljana we developed the first FPGA-based logic emulator 15 years ago and designed automated hardware verification system [4]. Since the first emulator we developed different FPGA boards for each new family of FPGA devices from Xilinx. Programmable emulation boards enable realization of the complete design cycle in the laboratory practice. In this way, the students get insight in each and every aspect of the design evolution and some specific problems that occur with the prototyping realization, hardware emulation and integration of the design in the existed system. Based on this knowledge they are able to realize their final projects and begin development in the industry. We supervised more than 50 diploma and masters projects on the logic emulators during the last 10 years and several design competitions.

There are several educational programmable prototyping development boards available on the market [5]. Most of them contain programmable device, memory elements and a few standard interfaces, like 7-segment displays, pushbuttons, standard serial and parallel interfaces [6]. We found these boards useful for initial digital logic course, but rather limited for the purpose of digital systems design education. Professional general purpose logic emulators are on the other hand expensive platforms and not very suitable for preparation of a laboratory course.

In this paper we will present the design process used in digital circuits education and architecture of a custom logic emulator board. We will present some typical student projects designed and implemented on our hardware and discuss the benefits of our approach.

II. DIGITAL SYSTEMS DESIGN PROCESS

A. Design Specification

The design flow for prototyping implementation of a digital system in the logic emulator platform can be divided into 4 major steps, as presented in figure 1. The first step is the design specification. The designer's task is to understand the requirements and the desired operation of the digital system. The requirements can be in a form of an interface specification, communication protocol etc. The operation is described with equations or algorithms and an abstract

software model can be used as a reference for the design verification.

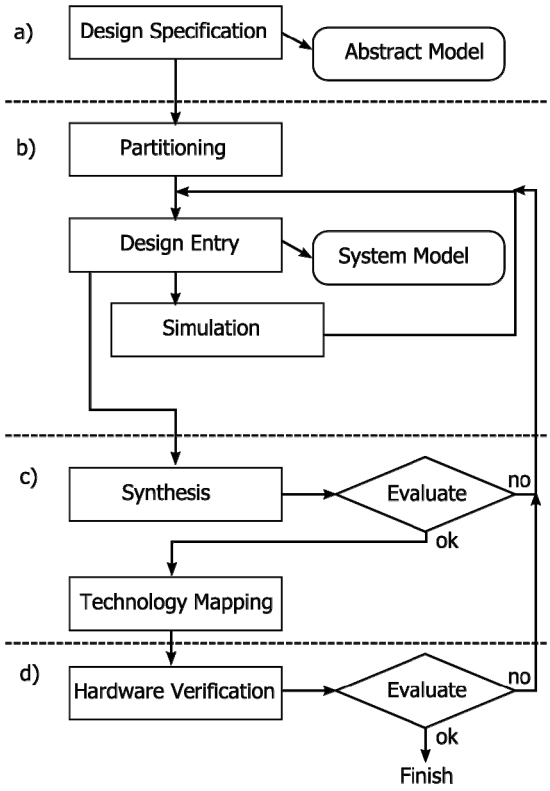

Figure 1. Design flow: a) specification, b) design entry and simulation, c) implementation and e) hardware verification

B. *Design Description*

Design description is governed by the design decisions based on previous knowledge which can be very difficult for the non experienced students. Students are familiar with a bottom-up design process starting from the basic logic structures. With this approach it is difficult to meet complex digital system specifications, and more common top-down design process should be considered.

In the top-down process, we begin by partitioning the design into smaller blocks, block description and behavioral simulation. There are several generally acknowledged design description methods supported by graphical tools (schematic or state machine) or hardware description languages (HDL). Graphical design is very intuitive, but limited by the provided elements and not practical for designing complex systems. Today it is replaced by the HDL design methods covering gate level, register-transfer level and behavioral level of the circuit description.

1) *Description of Register Microoperations*

The gate level schematic description is still used for the introductory course to digital design. In addition to schematic design entry we developed a novel web tool for design of register microoperations [7,12]. By using the basic combinational circuits and a register we can design arithmetic, logic, shift and register transfer microoperations. The web tool enables graphical design entry of microoperations by selecting the combinational operation and defining inputs and register control signals, as presented in figure 2.

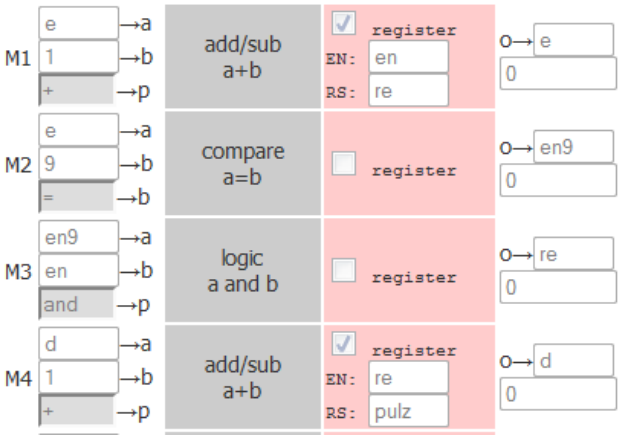

Figure 2. Web tool for development and simulation of register microoperations

The web tool includes simple cycle based simulator of the designed circuit and automatic generator of the HDL code. We present an example of automatically generated code from figure 2, which is a part of a clock counter circuit:

```
entity mikro is
 port (clk: in std_logic;
 en: in std_logic_vector(3 downto 0);
 e: std_logic_vector(3 downto 0));
end mikro;

architecture rtl of mikro is
 signal en9: std_logic;
 signal re: std_logic_vector(3 downto 0);
begin

M1: process(clk)
begin
 if rising_edge(clk) then
   if re=1 then
     e <= (others=>'0');
   elsif en=1 then
     e <= e+1;
   end if;
 end if;
end process;

 en9 <= '1' when e=9 else '0';
 re <= en9 and en;
```

In this way the students are able to design algorithms without learning the details of specific hardware description language. They can observe the generated code and learn basic principles of HDL description that will be used later for more sophisticated designs.

2) Block Level Simulation

Thorough simulation of the complete digital system is often time consuming task and can be replaced by the hardware verification process on the emulator system [8]. Students usually execute block level simulations in order to prove the behavioral correctness of the design parts. Typical simulation is driven by test benches which provide either simple signal timing or composed bus transactions.

C. Synthesis and Technology Mapping

Although the circuit synthesis is an automatic process, it is important for the students to run the synthesis on each block in order to get the first feedback of the design. When using HDL, one can quickly describe a hardware that can be well simulated but not synthesized. The synthesis results should be carefully evaluated against common design mistakes: inferred latches, unspecified states, etc.

D. Hardware Verification

In the hardware verification process we test operation of the design block or the complete system on the prototyping board. We can first apply the same test vectors as used by simulation to the logic emulator and verify the responses. Since the logic emulation runs at or near real-time speed, we can perform thorough testing of the design.

Even if the circuit passed behavioral simulation it can fail during hardware verification due to timing problems or weak design specification or test bench. Completing the design process is important in educational process in order to get familiar with these issues.

III. Logic Emulator

The custom logic emulator board is designed for hardware verification of the IP components. The section presents our design requirements, block diagram, design considerations and custom software tools.

A. Design Requirements

- Two programmable devices: one middle sized FPGA for custom logic and one CPLD for verification interface.
- The FPGA device is connected to a Wishbone bus, controlled by external PC.
- Configuration and verification control is performed through USB interface.
- Peripheral modules (e.g. memory, A/D) should be mounted on external connectors.

B. Block Diagram

Figure 3 presents block diagram of the custom logic emulator. There are 3 main integrated circuits: an FPGA Xilinx XC3S200 or XC3S400 [9], a CPLD XC2C256 and an USB interface. The combination of two programmable devices provides a lot of possible configurations for educational purpose. Instead of a lot of common input and output units (e.g. keyboard, display) the CPLD can emulate virtual interfaces and use the PC for signal control and visualization.

For the purpose of IP core development, the CPLD can emulate different bus protocols. In a typical configuration the CPLD implements 8 or 16 bit Wishbone bus master controlled by the external PC through the USB port. The CPLD uses 135 out of 256 macrocells in this configuration and has still enough cells for potential extensions.

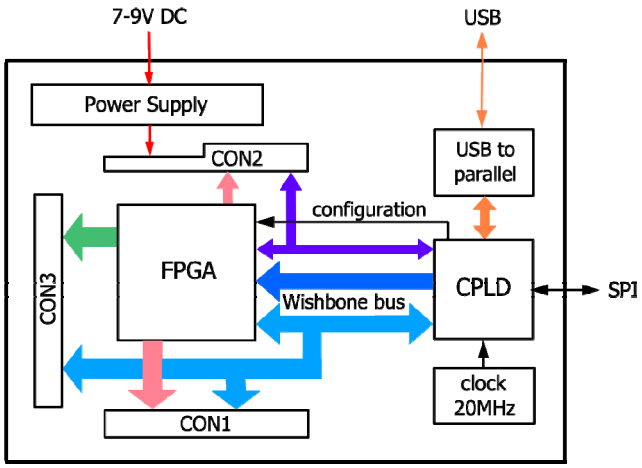

Figure 3. Block diagram of a custom logic emulator

The emulator board contains also a reference clock source and power supply units, which should be carefully designed in order to meet the requirements of programmable devices [10]. We minimized the number of components on the emulator board in order to cut the board costs. The simple board schematic is also a good reference to the students who are able to design their own boards if required.

C. Wishbone Bus

The digital system design tools provide a variety of IP core designs specifically tailored for the selected FPGA device. The circuit design process can be less error prone and simplified if we are using standard interfaces to data processing IP cores.

There are a lot of standards for system data busses, which are usually tied to specific microprocessor architecture. Even if we do not use a microprocessor, we benefit by using common bus architecture, selected by the following considerations:

- parallel synchronous buses are best candidates for FPGA architecture,
- a bidirectional bus is implemented inside the device with two single directional busses.

For the educational purpose it is important that the bus specification is open and not too complex. Based on these requirements we selected a Wishbone standard, for which we can obtain a lot of free IP core components from Opencores organization [11].

D. *Configuration and Control Software*

We developed the emulation board control software presented in figure 4. The program is used to download the configuration bitstream to the FPGA device, set-up the interface and perform basic data transfer. The basic transfer consists of setting the address and sending or receiving data words. The students can use data transfer functions for design of their own test bench and application specific graphical interfaces.

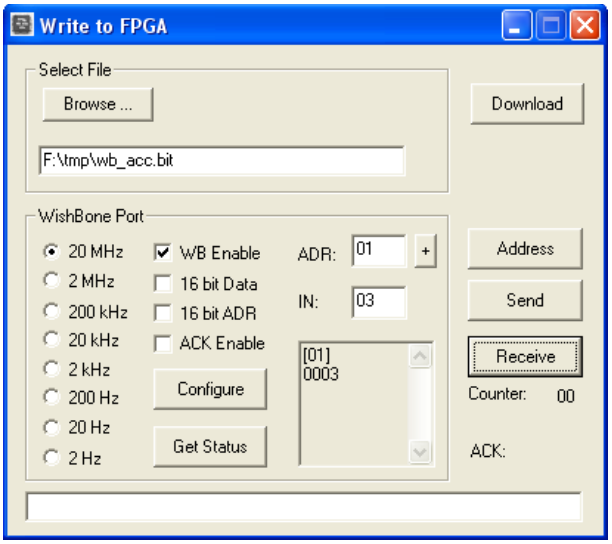

Figure 4. Configuration and control software

IV. Design Project Examples

In the introductory course we design a digital clock with multiplexed display and incremental encoder. The students use graphical design method for encoder finite state machine and web based register microoperation design for the clock counters. The design blocks can be separately simulated and verified and then schematically connected on the system and checked on board with virtual interfaces.

In digital integrated circuit design courses the students typically design one IP core on the Wishbone bus. A common core is a serial interface (UART, LIN, SPI). Students are faced with problems of clock synchronization and metastability and have to solve them in order to get reliable design.

In a second level (postgraduate) course we discuss HW/SW integration. The students design a communication and data processing IP core on Wishbone bus and test the cores on the emulator platform. The SW part is implemented with embedded 8-bit processor core (Xilinx Picoblaze) with a Wishbone wrapper. After separate verification of each core, they make the complete digital system and test the operation on the emulator board.

Here is a list of some of the student diploma projects based on programmable devices:

- Digital Audio Mixer
- JPEG Compressor
- Clock-Data Recovery Circuit
- VGA Image Rotator
- SD Micro Interface
- Wishbone Bridge
- PAL Video Generator
- Digital Storage Oscilloscope
- Embedded Logic Analyzer

The final student projects range from design of data interfaces and signal processing cores to development of custom boards for digital oscilloscope and audio analyzer.

V. Conclusions

We presented our educational design experience on a custom digital logic emulator based on two programmable devices. The CPLD device can be used for emulation of virtual interfaces or interconnection systems. Currently we developed Wishbone bus emulator with software support for control and data transfer.

The students are able to realize a variety of projects on the same hardware platform. Since they are able to participate in each step of the design cycle, they get the knowledge that can be readily used in industrial application development.

References

[1] W. Wolf, FPGA-Based System Design, Prentice Hall, New Jersey, 2004

[2] A. Zemva, M. Verderber, FPGA-oriented HW/SW implementation of the MPEG-4 video decoder. Microprocessors & microsystems., vol. 31, no. 5, pp. 313-325, August 2007.

[3] D. Hanna and R. E. Haskell, "Learning Digital Systems Design in VHDL by Example in a Junior Course," Proceedings of the ASEE North Central Section Conference, Charleston, West Virginia, March 2007.

[4] A. Zemva, A. Trost and B. Zajc, "Educational Programmable System for Prototyping Digital Circuits," International Journal of Electrical Engineering Education, vol. 35, no. 3, pp. 236-244, March 1998.

[5] R. H. Hosking: Choosing the Right FPGA Board, "More Features Mean Tougher Decisions," RTC Magazine, October 2006.

[6] R. E. Haskell and D. M. Hanna, Learning by Example Using VHDL - Advanced Digital Design, LBE Books, Rochester, MI, 2007

[7] M. M. Mano, Logic and Computer Design Fundamentals, Prentice Hall, 2007.

[8] R. Wilson, Verifying FPGA designs, EDN, February 2009

[9] Xilinx: FPGA, Spartan Series, http://www.xilinx.com/products/silicon_solutions/fpgas/

[10] J. Falin, J. Pham: Tips for successful power-up of today's high-performance FPGAs, Texas intrumenets, 2005, http://www.ti.com/

[11] WishBone Specification, http://www.opencores.org

[12] Microoperation Design Tool, http://lniv.fe.uni-lj.si/celldesign.html