Educational Digital Circuit Emulators

Baldomir Zajc Andrej Trost
University of Ljubljana
Faculty of Electrical Engineering
Trzaska 25, 1000 Ljubljana, Slovenia
zajcb@fe.uni-lj.si

Abstract

Digital circuit emulators based on programmable logic devices are used for prototyping implementation of digital circuits. The paper presents development of the digital circuit emulators for educational purposes. The emulators are composed of different families of FPGA devices and interfaces for testing the circuit with a personal computer. A software for interface synthesis and hardware verification of the emulated circuits was designed for each circuit emulator. We use schematic or behavioral tools for design and simulation of the digital circuits. The circuits are finally implemented and tested on the circuit emulators. The presented emulators are used in the regular educational process during the undergraduate study and the sample students projects are summarized.

1 Introduction

With the computational power of the todays computers and advances in CAE (Computer Aided Engineering) and EDA (Electronic Design Automation), the design cycle time is drastically reduced. Until the first appearance of Field Programmable Gate Arrays (FPGAs) in a mid-eighties [1], a certain gap between the circuit design phase and the prototyping implementation with hardware verification existed. 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.

User-programmability that is built in FPGAs brings many advances to the typical design approach. It significantly reduces the time and the cost of the prototyping realization of the design, speeds up the hardware verification phase and system integration, and allows fast redesign that is often imposed with the design errors in one design cycle. Another important aspect of FPGA's is that the designer can implement a circuit by himself, and thus complete the whole design process from the initial circuit specification to the physical implementation and system integration in one place. This is especially important for educational purposes. For example, circuit design practice courses at universities are often constrained with the tight timing frames and limited financial budget which may prevent students to complete large ASIC design unless FPGA technology is used.

Through the complete design process which completes with the working implementations, students get insight in each and every aspect of the design evolution and some specific problems that occur with the prototype realization, hardware verification and integration of the design into the existed system.

We first motivate and present basic requirements and objectives that have been evaluated in the presented circuit emulators. Developed hardware and software are presented next. In the last two sections, we discuss application of digital circuit emulators on some real designs, our experiences and directions for the future work.

2 Motivation

The basic requirements for the digital circuit emulators for educational purposes are:

- emulation of digital circuits with the complexity up to a few thousand of logic gates
- should have an interface to the PC for hardware verification of the designed circuits
- should be supported by easy to use software
- use of external I/O connectors for additional flexibility

There are a lot of programmable prototyping boards on the market today, but most of them do not meet all of our requirements. For example, FPGA provider Xilinx offers XC4000 Design Demonstration Board [2] containing an FPGA, some keys, DIP switches, two LED displays and an oscillator. The board has no interface which could be used for hardware verification, and the keys should have some debouncing circuit. Another example are XESS boards [3] with FPGA or CPLD programmable device, an 8 bit microcontroller, static memory and LED display. A PC parallel port connector is used for configuration of the programmable device, programming the microcontroller and testing the circuit. These boards are more suitable for educational purposes, but the interface to the PC is very simple and allows user control only over a few I/O signals on the programmable device. The other boards with better interfaces and software support might just be to expensive for educational purposes. For those reasons we decided to build our own programmable boards and software support.

3 Architectures of Circuit Emulators

3.1 Prototyping Board with Flexible Interface

We followed some useful hints and requirements given in [4] and [5] for builing our first prototyping board. Figure 1 illustrates the programmable prototyping board with 3 Xilinx XC3000 FPGAs [6]. The board is connected to a PC, which is used as:

- a host computer for peripheral mode programming of a daisy-chained FPGA devices,
- a test engine to perform hardware verification and
- an environment where the implemented designs can be integrated into.

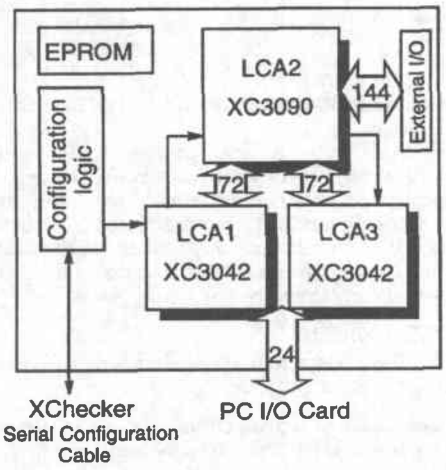

Figure 1: Programmable Prototyping Board

We built a relatively small and simple programmable prototyping board consisting of one Xilinx XC3195pg175, assigned as LCA2, and two XC3042pg132 FPGA devices, assigned as LCA1 and LCA3. The largest device, LCA2, is mainly used to implement a designed circuit, while the remainder devices implement flexible interface for hardware verification of the design. As shown in Figure 1, each of both LCA1 and LCA3 devices is connected with 72-bit wide datapaths to the LCA2, covering together all 144 user I/O pins of the LCA2. Since the interface controller utilizes relatively small portion of both FPGA devices, we rather lock I/O locations at these FPGA devices according to the connections to the LCA2.

Programmable board is connected with 24-bit wide datapath to the host PC computer with an I/O ISA card which provides parallel communication with 16 data lines, 4 address lines and 4 control lines. A 144 pin connector around the LCA2 is used to provide access to all user I/O signals and to connect additional external boards. We designed an external board for demonstrational purposes, which consists of 128k × 16 static RAM, clock generator, debounced keyboard and display. Since the FPGA board is intended to serve also as a standalone demonstration board, additional EPROM may hold the configuration data for master-mode programming of the daisy-chained FPGA devices.

Xilinx FPGA devices are selected due to their reprogrammability. The main reason for designing such a prototyping board with the given FPGA devices and datapaths relies on the requirement that the implementation of the designed circuit should be fully automatic with as few as possible manual interactions. For the design implemented in the LCA2 device, we very loosely restrict place and route software (XACT APR) [7] to efficiently and fully automatic complete the implementation. This improves the routability and consequently the average CLB utilization that can be achieved in the largest device.

3.2 Programmable Prototyping Card

Since the communication between the digital circuit emulator and PC is very importat for hardware verification, we integrated the communication interface and programmable devices and designed a programmable prototyping PC card. The hardware components of the programmable prototyping card are presented in Figure 2. The FPGA devices (LCA1 and LCA2) on the programmabe card are used for implementation of the prototyping design and interfaces. We used Xilinx XC4010 FPGA devices with the capacity of up to 10.000 equivalent logic gates. The programmable card is inserted in a PC ISA bus.

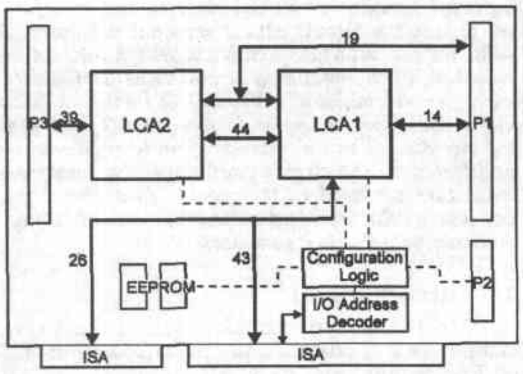

Figure 2: Prototyping Card

The PC is used for sending test vectors and reading design responses through the 16 bit ISA bus and the interface circuit inside FPGA. The interface circuit for hardware verification of emulated design is implemented in LCA1 and typically occupies around 10% of FPGA resources. Smaller designs are implemented entirely in the remaining logic resources of LCA1, while larger designs have to be partitioned between LCA1 and LCA2 device.

External I/O connectors are provided if additional logic or memory boards are required in the current design.

3.3 Compact Prototyping Board

Figure 3 presents a prototyping board based on XC4000E family of FPGA devices. The board was designed as a small prototyping system for educational and demonstrational purposes. An interface on the board is composed of TTL logic and is used for connection of the board to the PC through parallel port. The interface logic provides computer control over 24 input signals and 32 output signals on the FPGA. The FPGA I/O signals are also connected to a keyboard with debouncing circuit, DIP switches and LED display.

The FPGA circuit on the board can be any smaller XC4000E FPGA, ranging from XC4003E with capacity of 3.000 logic gates to XC4010E with capacity of 10.000 logic gates. For the configuration of the FPGA we can use serial PROM, XChecker serial cable or paraller port interface.

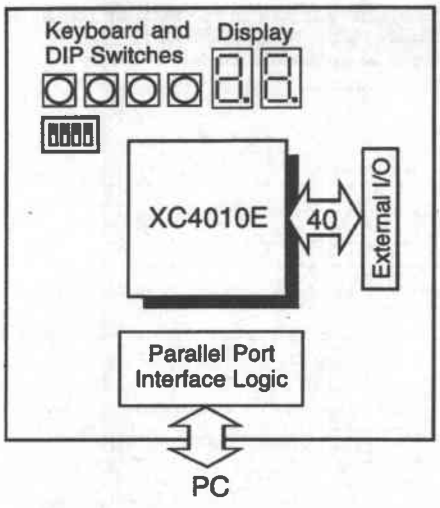

Figure 3: Compact Prototyping Board

The external 40 pin I/O connector is used mainly for additional memory boards since only small amounts of static memory can be emulated with the FPGA device.

3.4 CPLD Based Prototyping Board

Smaller digital designs can be effectively emulated with CPLD (Complex Programmable Logic Device) devices. The CPLD devices are used today mainly for glue logic and interfaces while FPGAs are used for actual computational tasks.

Figure 4 presents a prototyping board based on Xilinx XC95108 CPLD device. The CPLD has Flash configuration memory and can be configured through the JTAG connector on the prototyping board.

An interface on the board is designed for connection of the data and control signals to the parallel port of the PC computer.

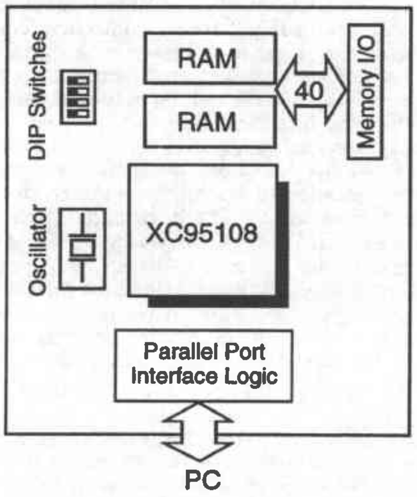

Figure 4: CPLD Prototyping Board

The prototyping board has 64k × 16 bit static memory, an external memory connector, DIP switches and an oscillator.

4 Software Support for the Circuit Emulators

The hardware verification procedure is embedded into the design flowchart as illustrated in Figure 5.

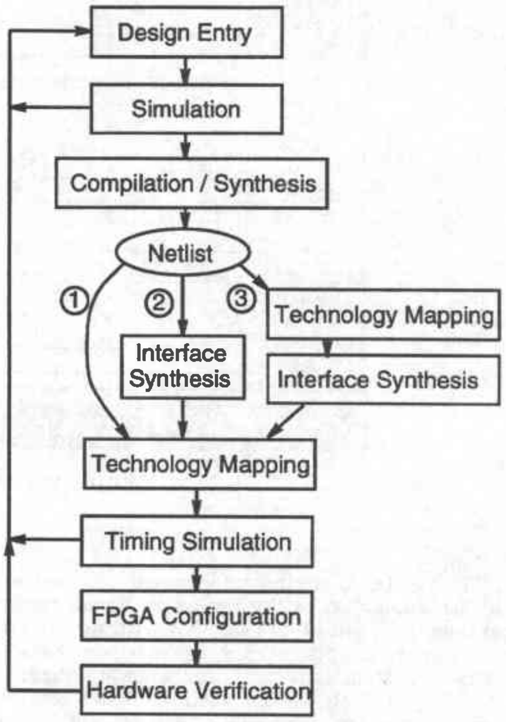

Figure 5: Design Flow

The design entry is either schematic or behavioral (VHDL). We use Xilinx Foundation software [8] for schematic entry and functional simulation or Aldec Active-HDL [9] for design and simulation in VHDL. The schematic design is compiled to XNF netlist while the VHDL design needs to be synthesized. We use Exemplar Logic Leonardo or Synopsys FPGA Express software for synthesis of VHDL designs into XNF netlist.

The next design step depends on type of interface on the programmable system. For the systems with the simple parallel port interface, we only have to define pin locations on the FPGA or CPLD device before technology mapping (path 1 in the Figure 5).

Interface logic is synthesized automatically for the design on the prototyping card and the interface netlist is merged with the design netlist. The final netlist is then input to the technology mapping step (path 2).

On the prototyping board with the flexible interface, we have to perform technology mapping first. Then the interface circuits are automatically synthesized according to the used I/O locations and technology mapping for the interface circuit in LCA1 and LCA3 is performed next (path 3).

The technology mapping involves several steps: translation and mapping of the logic into the internal structure, in-chip place and route, timing analysis and bitstream generation for the prototyping

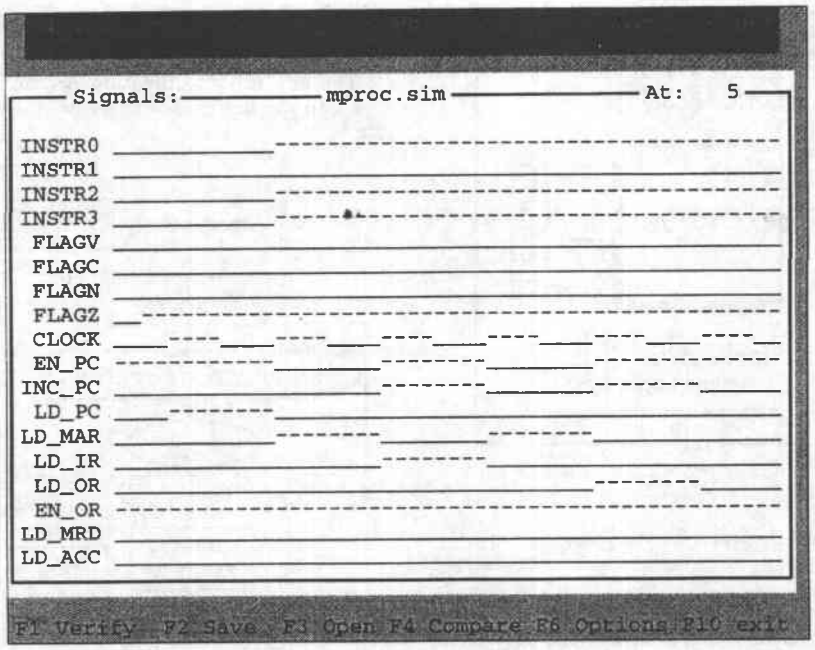

Figure 6: Hardware verification

design.

The produced timing information can be used for the simulation of the design with real timing parametrs. When we are satisfied with the results, we configure the programmable device on the prototyping system and perform hardware verification. Figure 6 shows the window of the verification program which was designed for each prototyping system. The user can interactively set the test vectors, perform hardware verification and observe and compare verification results with the simulation results.

5 Applications of the Prototyping Board

In this section, the sample student projects are described. Most of them are the results of the diploma work after 4 years of electrical engineering courses covering digital electronics, CAD and digital signal processing. Some of the typical working designs are described.

The first project deals with the design of the simple 8-bit CPU [12]. This simple, hypothetical CPU is introduced in the first year of the electrical engineering study to cover basic operations in computers. It is composed of five 8-bit registers, arithmetic logic unit and 256 bytes of RAM. The registers are: program counter, accumulator, state register, memory address register and memory data register. CPU is realized within LCA2 and 256 bytes of RAM is addressed at the external memory board. Operation of the CPU is the following: the program, written in the assembly language developed for this CPU is compiled to the binary instructions and stored in the RAM. The CPU can then operate in two modes, *interactive* or *automatic*. In the first mode, the input parameters of the program are interactively inserted by the user via external I/O card, and the results are displayed on a 7-segment displays. In the automatic mode, all input parameters are stored in the external RAM. Once the program is completed, the results in RAM are transferred to the PC. Rather than simulating the operation of the CPU on the computer with the required software, students can verify their programs in a real time with the programmable hardware.

The traffic controller is a typical student project which illustrates the hierarchy of design, steps of the functional verification and the integration of the real design into the environment. A crossroad of 4 roads including the signals for pedestrians and cyclists is considered [13]. Special attention is paid to the possibility of dominated phase and timing variability of each phase in order to regulate traffic density. The problem of the traffic controller was split into three subproblems: controller, counter and the control logic to interface both parts. The controller and the counter were described as a State Transition Graph (STG) and synthesized using SIS [10], while the interface logic was designed at the schematic-level using OrCAD [14]. Once the basic traffic controller had been designed, two controllers were cascaded to demonstrate the green phase over two crossroads. The environment, i.e. the crossroads with the required traffic lights, is emulated graphically on the PC driven by the signals from the prototyping board. Since we have a good experience with these illustrative projects, we are currently working on the similar projects such as washing machine and elevator controller.

The area of signal processing has been found very promising for demonstrating the powerful use of FPGA technology [15]. Since the Xilinx FPGAs exploit the symmetrical array structure, we have automated the design of systolic arrays operating as Finite Impulse Response (FIR) filters [16]. For the required specifications of the filter, the circuit netlist is automatically generated. Final technology mapping, placement and routing is then performed by XACT tool. Due to the regular structure of the systolic arrays, placement and routing in FPGA is relatively simple and it takes few minutes on Sun SPARC2 workstation from the initial specification to the working implementation.

The other project considers the adaptive FIR filters based on the Least Mean Square (LMS) algorithm [17]. The goal of the project was to establish the merits of FPGAs for the realization of these filters. We have experimentally confirmed that the filters ranging from the first order with the 16-bits input word length up to the ninth order with the 4-bit input word length can be realized within one XC3195 (LCA2). We have demonstrated the noise removal from the speech signal as a typical application of this adaptive FIR filter.

Current project in this area is related to pitch detection in a speech signal. After validating the algorithm for pitch detection with Matlab, we implemented the algorithm in hardware. Speech signal is sampled and digitalized with the sound blaster and then processed with the hardware implementation of the algorithm in the prototyping environment. Preliminary results show that the speech signal can be processed in a real time, comparing to several minutes required with Matlab on a Pentium based PC with 32Mb RAM computer.

6 Conclusions

In this paper, we have presented prototyping programmable systems for very fast implementation and hardware verification of digital electronic circuits, used as a standard tool in the courses covering principles and design of digital circuits given at University of Ljubljana, Department of Electrical Engineering. The main advantage of the systems are their simplicity which provide fully automated implementation and functional verification of the designed circuit. Their efficiency for the educational purposes has been proven through the various student projects. Based on our current experiences, we are focusing our efforts to design modular based hardware prototyping environment, consisting of multi-FPGA based programmable boards. The new environment will keep all the features of the current systems, as well as providing an additional flexibility to implement larger designs.

References

[1] Stephen D. Brown, Robert J. Francis, and Jonathan Rose. *Field-Programmable Gate Arrays.* Kluwer Academic Publishers, Boston, 1992.

[2] Xilinx. *XACT Development System.* Xilinx Incorporation, 2100 Logic Drive, San Jose, California, 1993.

[3] XESS. Manuals and Docs. http://www.xess.com/.

[4] M. Wendling and W. Rosenstiel. A Hardware Environment for Prototyping and Partitioning Based on Multiple FPGAs. In *European Design Automation Conference '94*, pages 77–82, September 1994.

[5] Zycad. Top 10 Rapid Prototyping Hints. http://www.zycad.com/.

[6] Xilinx. *The Programmable Gate Array Data Book.* Xilinx Incorporation, 2100 Logic Drive, San Jose, California, 1995.

[7] Xilinx. *User Guide and Tutorials.* Xilinx Incorporation, 2100 Logic Drive, San Jose, California, 1995.

[8] Xilinx. *Foundation Series Quick Start Guide 1.4.* Xilinx Incorporation, 2100 Logic Drive, San Jose, California, 1997.

[9] Aldec. *Active-HDL Quick Start Guide.* Aldec, Inc., 2230 Corporate Circle, Henderson, NV, 1999.

[10] SIS – Release 1.2. UC Berkeley Soft. Distr., July 1994.

[11] ViewFPGA. User's Guide, 1994. Viewlogic Systems, Inc.

[12] F. Bratkovič and V. Guštin. *Fundamentals of Computers.* Faculty of Electrical Engineering, Ljubljana, 1995.

[13] B. Zupan, A. Žemva, and B. Zajc. Programable Integrated Circuit for Traffic Light Controller. In *International Symposium on Traffic Electronics*, pages 317–322, October 1994.

[14] OrCAD/SDT. User's Guide, 1992. Hillsboro.

[15] L. Mintzer. FIR Filters with Field-Programmable Gate Arrays. *Journal of VLSI Signal Processing*, 6(2):119–127, August 1993.

[16] H. Froehlich, B. Zajc, and M. Lopez. Implementation of FIR Filters as Systolic Arrays with FPGAs. In *Proc. of the Third Electrotechnical and Computer Conference ERK'94*, pages 93–96, September 1994.

[17] R. Kranjc. Possible Realizations of Adaptive LMS FIR Filters with FPGAs. In *M.Sc. Thesis, University of Ljubljana*, July 1995.