The usage of programmable logics in the education of Digital Technology

Nikoletta Tolner and András Dávid

Campus of Alba Regia Technical Faculty of Óbuda University
H-8000 Hungary Szekesfehervar, Budai str. 45, Hungary
tolner.nikoletta@amk.uni-obuda.hu
david.andras@amk.uni-obuda.hu

Abstract: The authors have been teaching Digital Technology to full time and correspondent students for several years. In recent years, the manufacturers provided newer and more advanced solutions. At present, the users can get around the physical building and actual checking of the circuitry, as appearance of programmable logics opened new possibilities for circuit design and testing. The application of these novelties in the education might facilitate both the practical and theoretical knowledge of the students. In the present paper, in parallel with an electronic university lecture learning material, the process of using programmable logics in this special context was described. The XILINX's free downloadable ISE WebPACK software was used. Digital circuitries were edited with circuit diagram editor or VHDL instructions, and the operation of the circuit was simulated with ISE WebPACK software.

I. CURRICULAR SUBJECTS

A) Digital Technology

Digital Technology is a basic subject for students in electrical engineering, informatics engineering and technical management as part of the core material. Students in electrical engineering study Digital Technology I. in the first semester for 4 credits with 2 lectures and 1 lab practice per week, Digital Technology II in the second semester for 3 credits with 2 lectures per week, then Digital Technology III with 2 lab practices per week.
Students in informatics engineering study the subject as Digital Technology with 2 lectures and 2 lab practices for 4 credits in the second semester.
Students in technical management study the subject as Analog and Digital Technology in the fourth semester with 2 lectures and 2 lab practices for 5 credits.
The aim of the subject is to achieve all the basic hardware knowledge required to design digital circuits. To this end, students deepen their theoretical knowledge through practical tasks. As lecturers, our goal is to help the students to acquire the theoretical knowledge, and to learn to use it through practical examples, like the gradual implementation of the ISE WebPACK program from XILINX environment.

B) Design of Digital Systems

Students in electrical engineering on Hardware specialization can take this course in the sixth semester. The course consists of 4 lectures and 3 lab practices for 8 credits. The goal of the course is to familiarize the students with the building blocks of digital systems, their uses, connections and diagnostic possibilities. The design possibilities of modern circuits, the basics of programmable logics, the analyzation of possible starting points of specific tasks, and design considerations are used. During laboratory practice, students use and measure the possible solutions. The main core of the course material is the detailed description of programmable logical circuits.

II. LABORATORY PRACTICE

A) Digital Technology lab

During laboratory practice, the students study the operation of sequential and combinational logic circuits; the majority of the practice is the analyzation of sequential logic circuits. Xilinx's programmable CPLD circuit (XC9572XL) (Figure 1.) was used built into a specifically designed box (Figure 2). The signals can be measured on the pins, and can be monitored with an oscilloscope. The students were capable of carrying out tasks without the need of knowing this specific programmable circuit.
Unfortunately, the following problems have arisen with this device:

- broken cables
- welding problems
- issues from multiple reprogramming

The replacement of the programmable ICs solved some of the problems for a short period of time, but has not meant permanent solution.

B) Design of Digital Systems lab

During these laboratory practises, students use the same device as in the Digital Technology lab. However, in this specific course, students not only analyse the operation of programmed circuits, but create programs as well. Using the Xilinx's ISE WebPACK circuit diagram designer, students create various circuits. Then the complete circuits are uploaded into the programmable ICs and the signals with an oscilloscope are measured. During the seminar, students get a complex task that has to be evaluated at the end of the term. The multiple uploads, unfortunately, also caused malfunctions in the ICs.

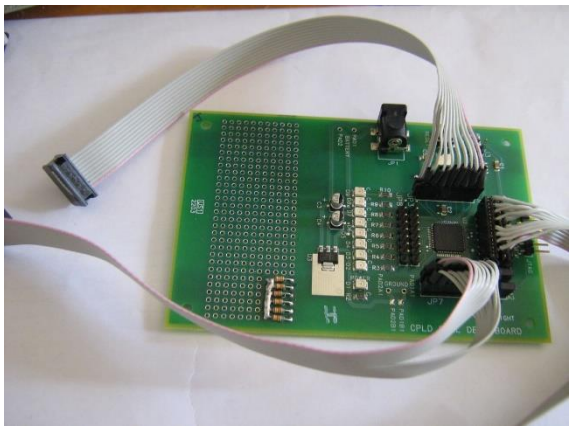

Figure 1. A board with programmable circuit

Figure 2. A measuring device

III. GOALS

In both subjects, the major goal was to substitute the classical measuring with simulations. The Xilinx ISE WebPACK has a simulation module that can be used for this purpose. However, the usage of the simulation module requires a certain level of VHDL knowledge that was integrated into the Digital Technology and Design of Digital Systems labs.

In the case of Design of Digital Systems subject, an electronic course material was also created by the Authors, focusing on the VHDL. The students work with the circuit diagram designer and the VHDL editor.

In the Digital Technology labs students analyse – with the help of the simulation module – sequential and combinational logic circuits designed in VHDL. In this subjects, only the basics of VHDL are delivered, to use the simulation module. The oscilloscope measuring still remains part of the laboratory practises besides the simulation.

IV. THE XC9500 FAMILY

This can be considered as the standard type of the Xilinx's CPLDs [1]. The components of the XC9500 family [2] essentially work in 5V TTL and CMOS systems, but can be configured to provide and receive 3.3V signals. The highest system frequency of the device is 100MHz, and the pin-to-pin minimum delay is 5ns. The XL types requires 3.3V power supply. The input buffers fully accept 5V signals; the 3.3V output signals in 5V systems correspond with the logical "H" level that makes it possible to use it in TTL systems without additional circuits. With the proper configuration of the I/O ports the device is capable of working in 2.5V environments. In this case the pin-to-pin delay is still 5ns, but the system frequency can reach up to 222MHz. The XC9500/XL/VL family is industry's first CPLD device that builds upon a 5V FLASH memory.

The XC9572/XL's [3] features:

- 5 ns pin-to-pin logic delays
- System frequency up to 178MHz
- 72 macro cells with 1600 usable gates
- Available is small footprint packages
  - 44 pin PLCC (34 user I/O pins)
  - 44 pin VQFP (34 user I/O pins)
  - 48 pin CSP (38 user I/O pins)
  - 64 pin VQFP (52 user I/O pins)
  - 100 pin TQFP (72 user I/O pins)
- Optimized for high-performance 3.3V systems
  - Low power operation
  - 5V tolerant I/O pins accept 5V, 3.3V, and 2.5V signals
  - 3.3V or 2.5V output capability
  - Advanced 0.35 micron feature size CMOS Fast FLASH™ technology
- Advanced system features
  - In-system programmable
  - Superior pin-locking and routability with Fast CONNECT™ II switch matrix
  - Extra wide 54-input Function Blocks
  - Up to 90 product-terms per microcell with individual product-term allocation
  - Local clock inversion with three global and on product-term clocks
  - Individual output enable per output pin
  - Bus-hold circuitry on all user pin inputs
  - Full IEEE Standard 1149.1 boundary-scan (JTAG)
- Fast concurrent programming
- Slew rate control on individual outputs
- Enhanced data security features
- Excellent quality and reliability
  - Endurance exceeding 10000 program/ erase cycles
  - 20 years data retention
  - ESD protection exceeding 2000V

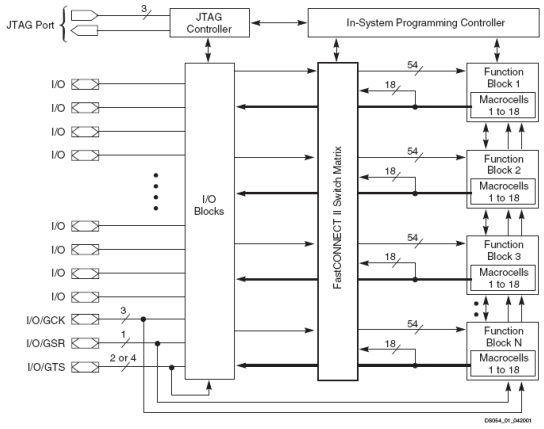

Figure 3. Structure of the XC9500XL

V. *XILINX ISE WEBPACK*

The Xilinx ISE WebPACK (Integrated Software Environment) is a free software developed by Xilinx for their FPGAs and CPLDs [4]. It can be downloaded from the company's website [5]. The software contains every tool needed for circuit diagram or hardware description language based development. Digital circuits can be created in the development environment as circuit diagram (Schematic) with the help of the circuit diagram designer; or can be written in hardware description language with the HDL editor. Supported languages: Verilog and VHDL.

A) Circuit diagram based development

The goal is to make the circuit in the Xilinx ISE WebPACK based on the circuit diagram, and to analyse it with simulation, with the help of a Schematic file (circuit diagram) and a test bench file. The Schematic file contains the circuit diagram; the test bench file is necessary to run the simulation. After the installation, the development environment can be started with the ISE Design desktop icon. After starting the program, the orienting window appears (Figure 4).

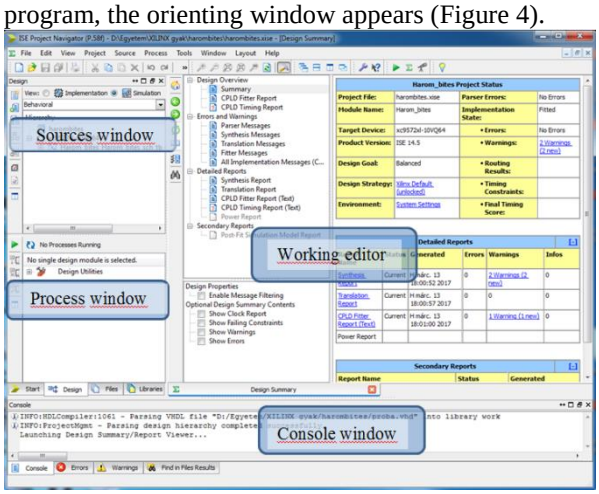

Figure 4. The program's starting window

The development environment organizes files into projects. The first step is to create a new project. We need to give the project a name and select a working directory, and the description of the project is also possible. In order to achieve a top level source, a circuit diagram based source (Schematic) has to be defined. In the next step, the tool type and attributes have to be defined. With this step, an empty project has been created. Then a new source file to the project is necessary. In the end, a circuit diagram designer window appears, and followed by the creation of the circuit diagram (Figure 5.). The symbols in order to create the circuit can be found on the Symbols panel (organized into categories). The user can also create symbols. After placing the symbols, they can be connected by clicking the pins. It's a help with complex circuit diagrams that pins don't need to be connected, it's enough to name the cables. Pins on cables with the same name are considered to be connected in reality. After creating the circuit diagram, I/O markers needs to be defined. The program automatically names the markers, but it's a good practice to rename them. If we receive a signal from outside port, or send a signal out, then Buffers needs to be installed between the I/O markers and the circuit.

After finishing with the circuit diagram, it's a good practice to run a syntax check. If it finishes without error, then we can compile the Schematic file into VHDL source code to simulate.

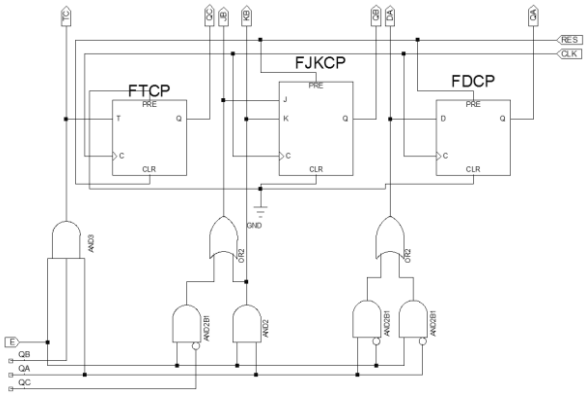

Figure 5. Example of circuit diagram creation

B) Hardware Description Language (HDL) based development

In order to test the circuitry within the HDL (VHDL), the source file and the test bench file is needed. The VHDL source file contains the code that describes the digital circuit; the test bench file is necessary to run the simulation.

The first step is to create a new project, and then to add a new source file. This is the same as with the circuit diagram designer, however, for a top-level source type the hardware description language based (HDL) source type needs to be selected, also the new source file type is now HDL based (VHDL Module). Adding a new source file, the New Source Wizard's Define Module window appears, where the ports of the device have to be defined (Figure 6).

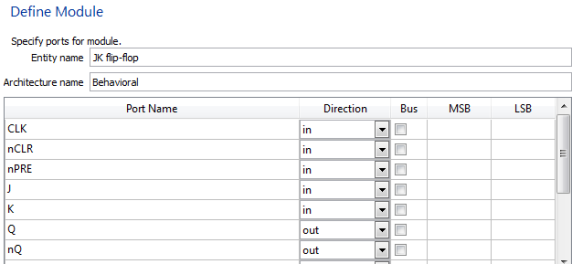

Figure 6. Definitions of the JK flip-flop's ports

After the appearance of the source editor window, the VHDL source code can be inserted (Figure 7, a source code describing a JK flip-flop).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity JK_ff_74LS76A is
    Generic (tpLHCLK_Q : time := 20 ns;
             tpHLCLK_Q : time := 20 ns;
             tpLHnPRE_nCLR_Q : time := 20 ns;
             tpHLnPRE_nCLR_Q : time := 20 ns;
             tsu_CLK : time := 20 ns);
    Port (   CLK : in  STD_LOGIC;
             J : in  STD_LOGIC;
             K : in  STD_LOGIC;
             nCLR : in  STD_LOGIC;
             nPRE : in  STD_LOGIC;
             Q : out  STD_LOGIC;
             nQ : out  STD_LOGIC);
end JK_ff_74LS76A;

architecture Behavioral of JK_ff_74LS76A is
signal J_act : STD_LOGIC := '0';
signal K_act : STD_LOGIC := '0';
signal next_state : STD_LOGIC := '0';
signal not_next_state : STD_LOGIC := '1';
begin
 JK_ff : process(J, K, CLK, nCLR, nPRE)
  begin
   if NCLR = '0' then
     if next_state = '1' then
        next_state <= '0' after tpHLnPRE_nCLR_Q;
        not_next_state <= '1' after tpLHnPRE_nCLR_Q;
     end if;
   elsif NPRE = '0' then
     if next_state = '0' then
        next_state <= '1' after tpLHnPRE_nCLR_Q;
        not_next_state <= '0' after tpHLnPRE_nCLR_Q;
     end if;
   elsif J /= J_act or K /= K_act then
       J_act <= J after tsu_CLK;
       K_act <= K after tsu_CLK;
   elsif rising_edge(CLK) then
      if J_act = '1' and K_act = '0'
          and next_state = '0' then
            next_state <= not(next_state) after tpLHCLK_Q;
            not_next_state <= not(not_next_state)
                                    after tpHLCLK_Q;
   elsif J_act = '0' and K_act = '1'
         and next_state = '1' then
           next_state <= not(next_state) after tpHLCLK_Q;
           not_next_state <= not(not_next_state)
                                    after tpLHCLK_Q;
   elsif J_act = '1' and K_act = '1' then
      if next_state = '0' then
           next_state <= not(next_state) after tpLHCLK_Q;
           not_next_state <= not(not_next_state)
                                    after tpHLCLK_Q;
      else
           next_state <= not(next_state) after tpHLCLK_Q;
           not_next_state <= not(not_next_state)
                                    after tpLHCLK_Q;
      end if;
   end if;
  end if;
 end process JK_ff;
 Q <= next_state;
 nQ <= not_next_state;
end Behavioral;
```

Figure 7. Source code of a JK flip-flop

After writing the VHDL source code, the check of syntax and the compilation is taken place.

C) Simulation

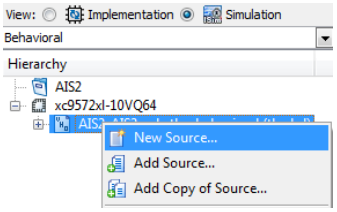

Figure 8. Adding new source to the project

The operation of logic circuits that has been created either as circuit diagram or with hardware description language can be tested with the ISIM simulator. In order to achieve a simulation, a test bench file has to be created, and then added to the project. The test bench describes the signals sent to each input. If the circuit contains clock signal, then a clock signal generator processor has to be added to the test bench file (Figure 9.).

```
-- Clock period definitions
constant CLK_period : time := 10 ns;
-- Clock process definitions
CLK_process :process
begin
   CLK <= '0';
   wait for CLK_period/2;
   CLK <= '1';
   wait for CLK_period/2;
end process;
```

Figure 9. Definition of clock signal

The input signals definitions should to be written in the appearing editor (Figure 10).

```
-- *** Test Bench - User Defined Section ***
   tb : PROCESS
   BEGIN
      E <= '1';
      RES <= '0'; wait for clk_period;
      RES <= '1'; wait for clk_period; RES <= '0';
      wait for 2*clk_period; E <= '0';
      wait for clk_period; E <= '1';
      WAIT; -- will wait forever
   END PROCESS;
-- *** End Test Bench - User Defined Section ***
```

Figure 10. Enter excitation of input signals

Before running the simulation, the setup of the simulation process properties is required. This can be done within the menu shown on Figure 11. The Simulation Running Time is probably the most important property.

After the clock signal and the input signal setup is done, the ISIM simulator can be started. The most important part of this is the waveform windows, where the timing diagram of the circuit can be seen (Figure 12.). The timing diagram describes the operation of the circuit shown on Figure 5.

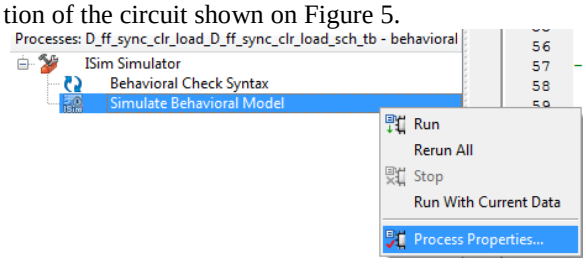

Figure 11. Setup of the properties of the simulation process

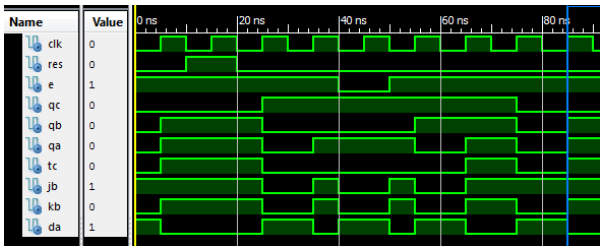

Figure 12. Timing diagram of the tested circuit

VI. ELECTRONIC COURSE BOOK

A. Goals

There are many great literature of the subjects for the students [6-8], albeit these references do not fully satisfy our criteria in the curriculum. To this order, a new course book will be made for the Design of Digital Systems subject. The book, additional to the theoretical material, would contain a good number of examples, to help deeper and practical understanding of the subject. At the labs, unfortunately there is no time to start from the basics, and there is a need for a course book that the students can use to prepare in advance. The simulation related parts of the course book also can be used in the Digital Technology labs. We aim to keep the theoretical descriptions at the necessary minimum, because we want to emphasize the practical examples.

B. The structure of the electronic course book

The course book is made up of 4 main chapters. In the first part, there are theoretical descriptions, and the second part would contain a good number of examples.
The first chapter would be a summary about programmable logics. The second chapter is the introduction of the Xilinx ISE WebPACK program suit, with a detailed description from the downloading of the program till its usage. In the third chapter would contain a description about the VHDL. An extensive material into the course book from the language elements to its syntax has been incorporated. Here only some examples and program parts can be found. In the fourth chapter, examples of combinational and sequential logics are described. For every examples, only a short description would be addressed, because those enrolling the course already possess a certain level of Digital Technology knowledge (the successful completion of the Digital Technology course is a requirement of this course). After the description of the examples, the description of VHDL would occur the test bench file for the simulation, and the simulation result timing diagram. These examples will help the students to solve more complex tasks later. In most cases, the whole test bench file is provided in order to use them in Digital Technology labs. In the fourth chapter, we added exercises after each topic that we would like to give as homework to the Design of Digital Systems course's students. We discuss the solutions at the laboratory practices. Later we plan to make the exercises' solutions available to the students in electronic format. Figure 13. shows a part of the course book.

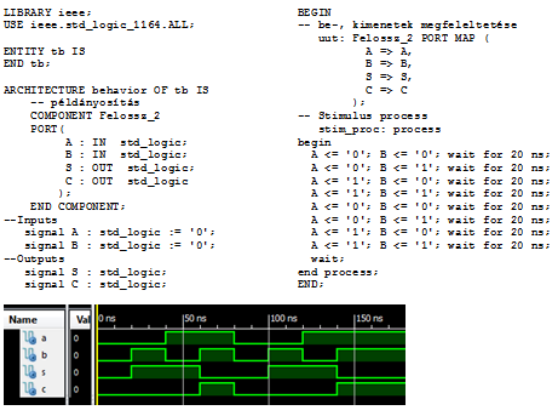

Figure 13. Part of the Design of Digital Systems course book

C. Expectations, hopes

We hope that our course book will help the students to gain the necessary knowledge in the subject. Of course it would be impossible to include everything into one course book, but we try to demonstrate the basics with a great number of examples. Our goal always was to help the understanding with as many practical examples as possible. This electronic course book was also made with the same aim in mind. We hope that it will live up to the expectations. Our future plan is to create an online course in the subject.

VII. REFERENCES

[1] https://www.xilinx.com/cpld/
[2] https://www.xilinx.com/support/documentation/data_sheets/DS063.pdf
[3] https://www.xilinx.com/support/documentation/data_sheets/ds057.pdf
[4] https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html
[5] https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools/archive.html
[6] László Varga: FPGA development in Xilinx ISE Webpack https://www.hobbielektronika.hu/forum/getfile.php?id=118958
[7] Gergely Nagy, Péter Horváth, András Poppe: Practical aspects of thermal transient testing in live digital circuits (Proceedings of the 19th International Workshop on THERMal INvestigation of ICs and Systems, ISBN:978-1-4799-2271-0) https://repozitorium.omikk.bme.hu/handle/10890/5088
[8] Szabolcs Szilágyi: Multiway Switching Controller Design using FPGA (ICAI 2014: Proceedings of the 9th International Conference on Applied Informatics) http://icai.ektf.hu/icai2014/papers/ICAI.9.2014.2.57.pdf