**Vedat KIRAY**

Turgut Özal University

# FPGA Based Digital Electronic Education, Data Entry Organization for a Calculator

***Abstract****. In this study, an example is presented developed under a Project named FPGA based Digital Electronics Education Project (FPGA_b_DEEP). The Project aims to develop instructive and attractive examples to teach Digital Electronics effectively and faster. In the example developed in this study, a data entry organization part of an 8 bits calculator design is dealt with. The design is prepared by considering students/new designers' knowledge level and divided into four parts for better understanding. The functions implemented in the modules are to code a number or an arithmetic operation to each button, to shift the numbers to left side with each entered digit of number, to multiply the digits with proper coefficients from decimal numbering system, to parallel add the multiplied digits with the proper coefficients, to create the number from entered digits at electronics environment (in a 32 bits register)*

***Streszczenie****. W artykule przedstawiono program szkoleniowy dla studentów, dotyczący projektowania kalkulatora w układzie FPGA (ang. Field Programmable Gate Array). (****Implementacja kalkulatora w układzie FPGA jako element edukacji w zakresie techniki cyfrowej****)*

**Keywords:** FPGA, Digital Electronics, Calculator, Engineering Education.
**Słowa kluczowe**: FPGA, technika cyfrowa, kalkulator

## Introduction

FPGA (Field Programmable Gate Arrays) technology has reached wide-spread usage on electronic engineering area. Today, many prototype applications related to the Digital electronics Design are executed on FPGAs. Naturally, the FPGAs have become an important part of electronics/Computer engineering education. Depending on these developments, some special studies related to the efficient use of FPGAs in engineering education started at lots of universities [1-13].

In the same way, our Project was initiated in 2008 aimed to use FPGAs in engineering education. The Project started at IAAU (International Ataturk Alatoo University) in corporation with ZTU/HSZ-T (Zurich Technology University (Hochschule für Technik Zürich). Nowadays, the project is carried out at Turgut Özal University. Main purpose of the Project is to use FPGAs as educational materials to teach Digital Electronics effectively and faster.

But unlike the similar studies mentioned at first paragraph, the originality of the Project is to develop instructive and attractive examples for better understanding. Because some limitations at Digital Electronics Education have been exceeded due to FPGAs at the last 10 years. For example, students/new designers achieved lots of application opportunities with large electronics component libraries. By the way, it is facilitated to set more complex circuits in shorter time. As a result, Digital electronics needs to be taught in new method and needs its own new examples suit to the new method. The examples developed under FPGA_b_DEEP are serves to this purpose. To contribute new form of Digital Electronics courses with the new examples, lots of research and study are needed. In other words, it is necessary to develop more simple and more understandable semi-professional designs which can be a sub-step to the professional designs.

The FPGA_b_DEEP aimed to introduce several examples as a case study. They are Clock-Calendar, Calculator, Basic CPU, ADC simulator, PID controller and Fuzzy Controller examples. These examples are selected by considering students interests. After introducing some developments from the Project at IKECCO conferences [14-15], the first completed example from the instructive and attractive examples' series was introduced an article at 2009 [15]. Second completed one is waiting under reviewing.

In this study, the third example among the instructive and attractive examples is introduced. An 8 bit-calculator design is divided into 3 parts and the first part of it is named as "data entry organization part." The example deals with to this part. The other parts of the calculator are arithmetic operation and display parts. They are left to other studies because full design of the calculator is too large to introduce in an article. Nevertheless given part of the calculator is % 70 percent of the full design.

For a calculator design, fix point or floating point numbering system can be used. In this developed example fix point numbering system is preferred but a process named normalization from floating point numbering system is added to the example for educational purposes. This operation does not aim a better or faster calculation but aims a better understanding and to set a bridge between fix point and floating point numbering system calculations. By the way, the important design steps for the calculation can be seen below.

- Converting the entered number to 8 digits temporary number.
- Saving the number of power to a 4 bits register. It represents relations between entered number and temporary 8 digits number.
- Making the desired operations by using 8 bits temporary numbers.
- Finding the real results by using the saved number of powers at 4 bits registers and fallowing certain basic formulas.

This part of the calculator design is divided into four sub-modules. They are: Button Pulses Module; a number or an arithmetic operation is coded to each button of the calculator. Zero and Comma Module; several state machines are used, they are developed for exceptional usage of zero and comma buttons. Data entry Module; most important tasks are completed and a lot of state machines are developed. Shift_and_mult_with_coeffi Module; the entered numbers are shifted to left side by multiplying with proper coefficients and after that is added (parallel) to obtain real entered number at a register. Due to the importance of this module, its name is given to first part of calculator design.

In order to understand the calculator design, full electronic circuits, simulation results and some important explanations of the each module are given. In addition, the rules constitute the cornerstones of the design is also given. However some classic engineering operations such as preparing tables and algorithms used at the design and state machines are not given.

### Button Pulses Module and Devoting Numbers to Buttons

It is possible for FPGA boards to get data of button through an external keypad and put it into a register directly. But lots of students/new designers prefer to use those buttons installed on education boards. In this case, necessary circuits must be added to module design to code numbers to buttons located on boards. Considering this realty, necessary extra circuit parts were added to the module circuit to code proper numbers and functions to

buttons. These extra parts can be seen at Fig 1a. The DFFs placed at each button entry filters the entry signals. By the way, disadvantages of mechanic press operation also can be eliminated

.

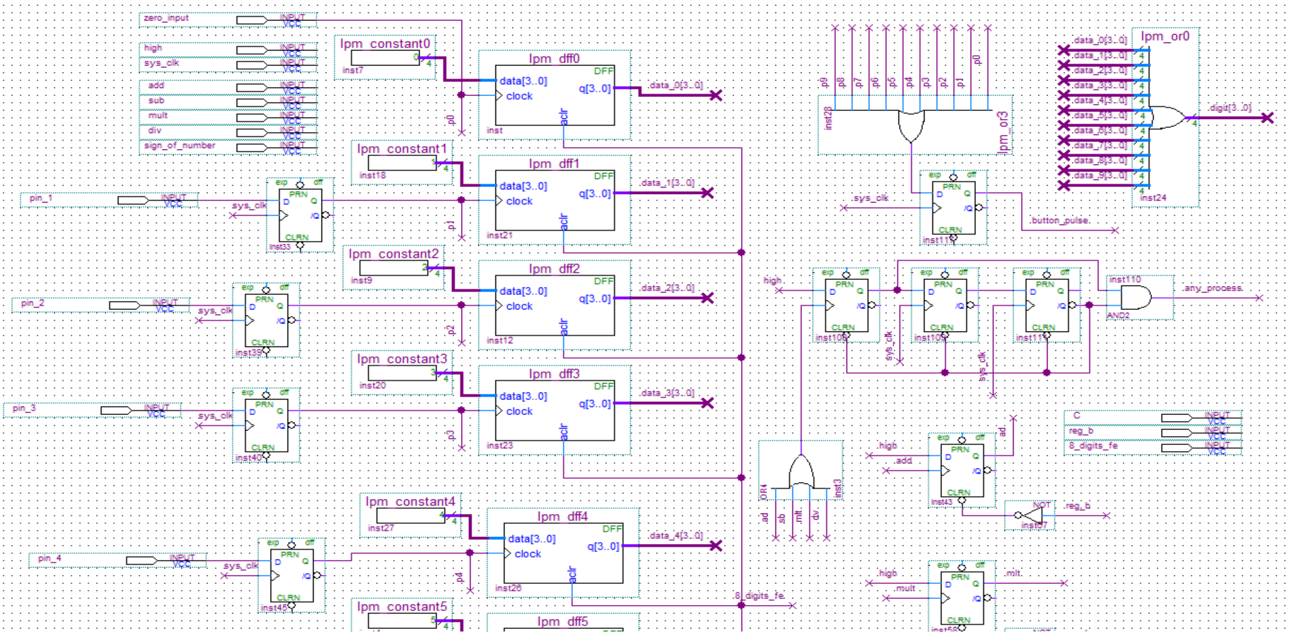

Fig 1a : Devoting numbers to buttons and saving them into 4 bits DFFs

Simulation results of the circuits can be seen at Fig 1b. To see simulation results, 3. 2. and 5. buttons are activated sequentially. After each button press operation, if related button signal is received by data entry module, another signal from this module named ‘8_digits_fe’ is sent for resetting 4 bits DFFs. Those operations also can be followed through Fig 1b.

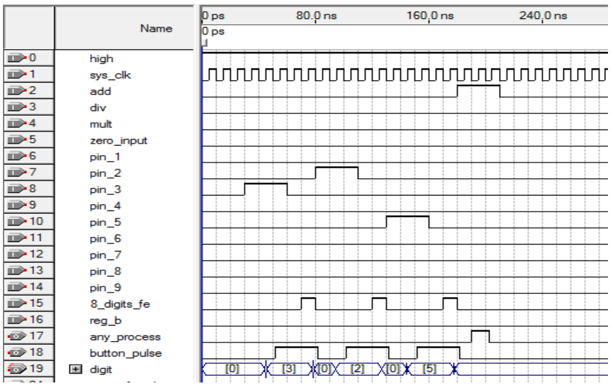

Fig 1b: Simulation results for Button Pulses Module

### Design of the Zero-Comma Module

There are some exceptions about the usage of zero and comma buttons at calculators. For better understanding, related state machines are developed under a separate module named zero-comma module. In order to develop necessary state machines following rules are used. Obtaining and using of this rules is very important for students/new designers to understand logic of the digital design.

- When you make the display of the calculator active, number zero must appear on it. (This situation, for today, is not a compulsory result of the design, but it is a practical usage to indicate that calculator is ready for the usage.
- The zero button must be passive if there is no any data entry from the other buttons. After any other button is used on the list one time, zero button must return to its normal function.
- If the comma button is pressed while only number zero is appearing at display, first zero signal, then comma signal must be generated.
- If the comma button is pressed after any other button, it must work in its normal function. It means there is no need to generate zero signal before generating comma signal.

The circuit prepared according to the rules given above, can be seen at Fig 2a. It consists of several circuits parts. Some of the functions of these parts are: Generating necessary pulses in proper time points, arranging the duration of high position of some pulses, Controlling the activation of zero and comma pulses by using fallen edge of the ‘button _pulse’ signal as a reference pulse.

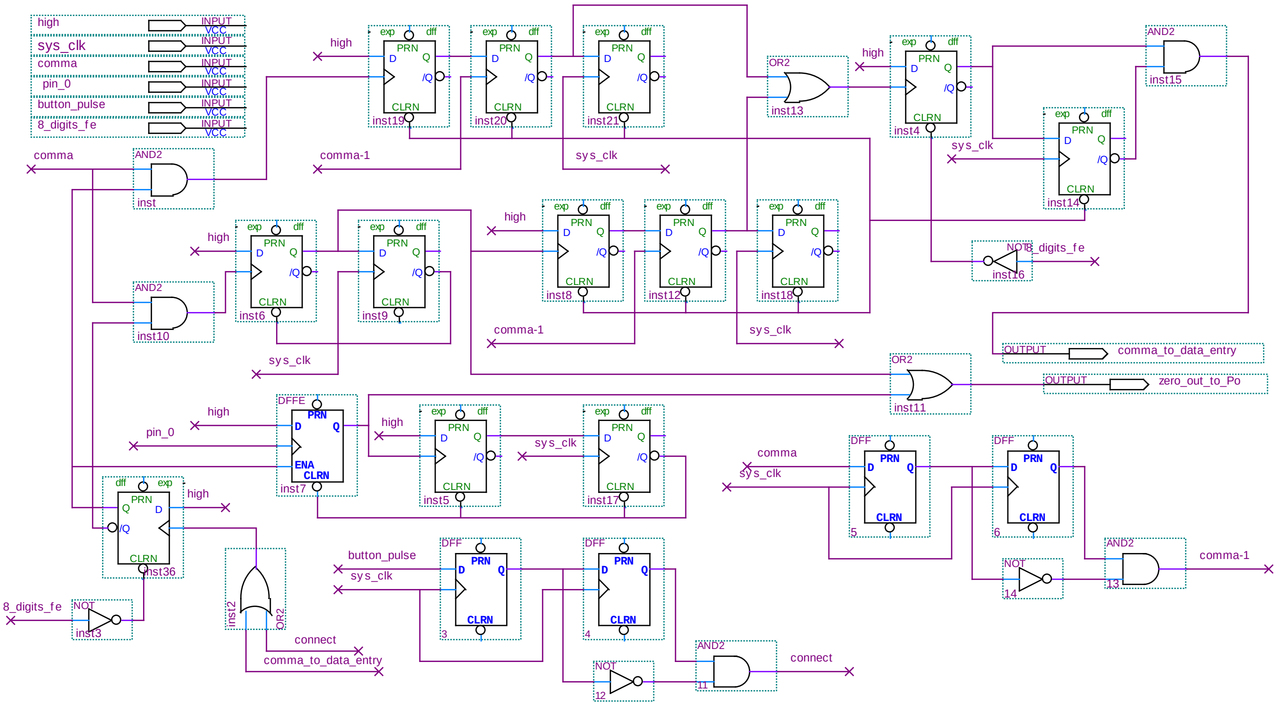

Fig 2a : The circuit which is organized for exceptional work of the zero and comma buttons.

The simulation result of the circuit can be seen at Fig 2b. The first two pulses applied to system are 'Pin_0' Pulses of 0_button. At this period, there is no any signal on the output called "zero_out_to_P0." After pressing comma button, first zero signal then comma signal is generated.

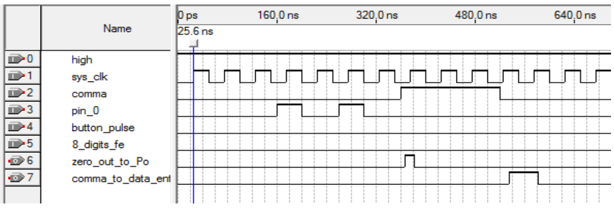

Fig 2b : Simulation results related to usage of zero and comma buttons.

## Design of Data Entry Module

This module is the main module of the Data entry Organization part of the calculator. Several important operations related to preferred method of calculation are completed here. Some of these operations are as follows.

- Converting each entered numbers to 8 digits (32 bits) temporary numbers,
- Saving the difference between entered number and 8 digits temporary number in as power of the number.
- Generating triple pulses for Shift Taps Component after entering each digit of the number
- Generating some important signals to be used in other modules.

At first glance this module can be found complicated, Zero-comma module before this module provides a great convenience. These two modules help students/new designers to understand how to develop a design and how to convert rules to electronics circuits. These rules are very important in term of educational purposes. There are rules below related to data entry module from the last design study.

- When you press a button, the 'Button_pulse' signal must be generated
- Following the 'Button_pulse' signal, Triple pulses must be generated and sent to the output called 'clik_for_shift_taps'
- Following the first 'Button_pulse' signal, down counter (from 8 to 0) must be active and it must continue to count after each 'Button_pulse' signal.
- Following the first 'Button_pulse' signal, the 'Any button' signal must be high and it must keep its level till the digits of the number reache to 8
- 'Any Process' signal must be generated when any button related to arithmetic operations is pressed.
- Following the 'Any Process' signal after its becoming High, if the digit of the number is not equal to 8, the rest digits must be completed to 8, meanwhile, the value at the out of down counter must be written to 4 bits DFF register.
- IF comma button is pressed, down counter must go on counting (to protect data loss at output 'clk_for_shift_taps' ), but the value at the out of down counter must be written to 4 bits DFF register with comma signal.
- If the entered number has reached to 8 digits, the 'Any Process' signal must be low.
- The number saved at 4 bits DFF must sent to output q[3..0]. This value also must be saved to another register at out of the module. For his purpose, the value of 4 bits DFF must be sent to output 'save _number_mpower'.

The circuit of data Entry Module prepared according to the rules can be seen at Fig 3a. There are only DFFs connected to the DFF rows at the missing part of the picture (right part) and its simulation results can be seen at Fig 3b.

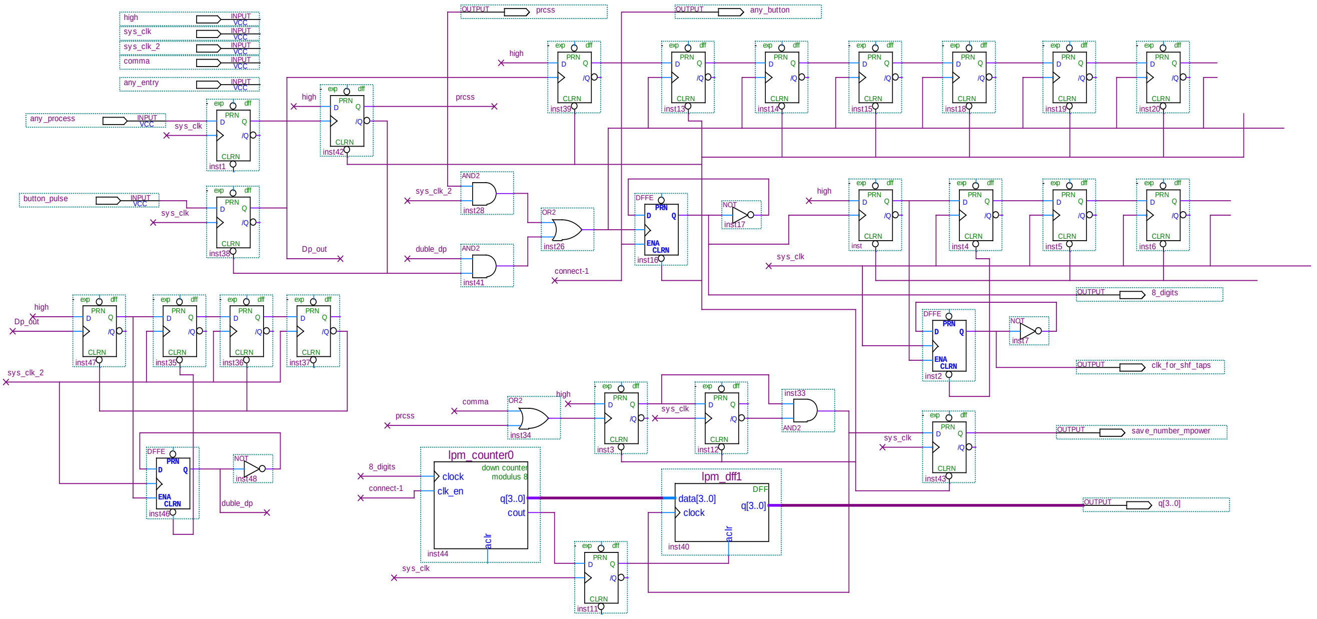

Fig 3a : Data Entry Modulü

As can be seen at Fig 3b, there are two data entry period at simulation report page. At first period (till to point of 24.2944 us) there is only comma operation while button pulses are going on. When the comma button is pressed at the point of 10.24 us, the number at the output of down counter is sent to 4 bits DFF register but meanwhile, number entering and generating of triple pulses for Shift Taps Component goes on. At second period, ‘any process’ button is pressed after two ‘button_pulse’ signals and the rest digits are generated automatically (point of 30.72 us).

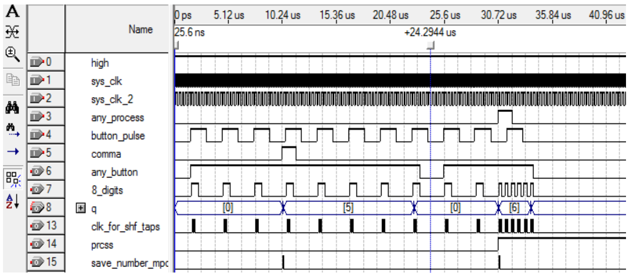

Fig 3b : The simulation results for Data entry Module.

## Obtaining the Number From Digits: Shifting, Multiplying with Proper Coefficients and Parallel Adding

The operations mentioned at the title are implemented under ‘shift_and_mult_with_coeffi’ module. The digits of the number are shifted by Shift Taps Component (Fig 4a.) This component needs to triple pulses after 4 bits BCD data entry. These necessary external pulses called ‘clk_for_shifttaps’ are generated at the data Entry Module and sent to the ‘Shift_and_mult_with_coeffi’ module.

Eight multiplier components are used at the circuit called ‘Lpm_Mult.’ Each ‘Lpm_mult’ component has got two inputs. One input of each multiplier is connected to the corresponding output of the ShiftTaps component. The other input is connected to the corresponding constant value component (a coefficient at decimal numbering system such as 1, 10, 100, 1000, …. 1000000). It is important where the shift outputs and the constant outputs must be connected to obtain the number in true form. The ‘Lpm_Mult’ components can be seen in Fig 4a; one under the other.

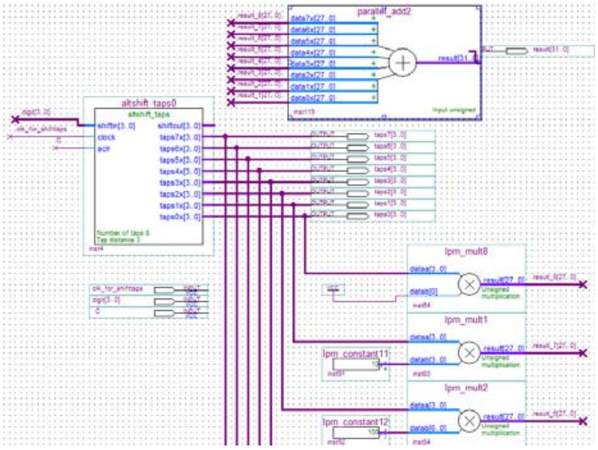

Fig 4a: The Circuit of ‘Shift_and_mult_with_coeffi’ module

Parallel adding operation is implemented by using Parallel_Add component. Since the designing calculator is eight digits calculator the component is arranged to do 8 parallel add operations. The digits that are shifted continually at the outputs of the Shift_Taps component are multiplied with the proper coefficients and then entered to Parallel_Add component. For example when we entered the number 345, the digit 5 places at the first left and is multiplied with coefficient 100, sequentially 4 is with 10 and 3 is with 1. (100.3+10.4+1.5=345). The simulation results can be seen at Fig 4b. The digits 12345678 are entered to the module then the number appears at the output of the ‘parallel_add’ component or at the output named result.

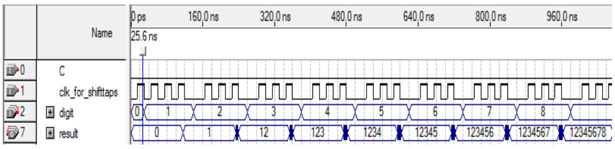

Fig 4b: The simulation results of ‘Shift_and_mult_with_coeffi’ module.

## Conclusion

In this study, data entry part of an 8 bits calculator is designed by dividing to sub modules for better understanding. First version of this example was used as a **course material at** IAAU electronics department and got positive results. By using the feedbacks from students, this design was developed and the example presented at this study was prepared. The example was included to instructive and attractive examples' series under the FPGA_b_DEEP.

The most important conclusion extracted from the study is that a calculator design includes lot of sub circuits suit to curriculum of Digital Electronics courses especially in term of those topics such as numbering systems, combinational circuits and sequential circuits.

There are two important originality at the developed example. First, the calculation method used at the design is a method which is implemented under the fix point numbering system however the logic of method is very close to floating point numbering system. The method is preferred to students to feel ready themselves to floating point calculations. Second is to make some extra arrangements in the sub module designs to students/new designers comprehends their design logic. For example, Lots of rules are generated for each module design. These rules make the course easy, interactive and understandable.

At the following studies, it is aimed to prepare new examples dealing with to another calculator and a basic CPU structures.

## REFERENCES

[1] Y. Zhu, T. Weng, and C.-K. Cheng, Enhancing Learning Effectiveness in Design Courses Through the Use of Programmable Logic Boards, *IEEE Trans. on Education.* 2009. *vol. 52, no. 1, 151–156,.*

[2] Z. A. Obaid, N. Sulaiman and M. N. Hamidon, FPGA-based Implementation of Digital Logic Design using Altera DE2 Board. *IJCSNS International Journal of Computer Science and Network Security.* July 2009. *VOL.9 No.8,*

[3] Hoffman M, An FPGA-Based Digital Logic Lab For Computer Organization And Architecture, *Journal of Computing Sciences in Colleges.* 2004. *19(5):214–227.*

[4] Li Y, Chu W, Using FPGA for Computer Architecture/Organization Education. *IEEE Computer Society Technical Committee on Computer Architecture Newsletter, IEEE Computer Society Press.*1996. *pp.31-35.*

[5] S. Sothong, and P. Chayratsami, Design of Combinational Logic Training System Using FPGA. *IEEE Frontiers in Education Conference (FIE).* 2010. *pp. F4F-1 – F4F-4,*

[6] Rey Juan, Project Based Learning Experience in VHDL Digital Electronic Circuit Design. *International Conference on MICROELECTRONIC SYSTEMS EDUCATION*, MSE '09. IEEE, 2009

[7] T. Weng, Y. Zhu, and C.-K. Cheng, Digital Design and Programmable Logic Boards: Do Students Actually Learn More? *in Proc. ASEE/IEEE Frontiers in Education Conference,* 2008.

[8] T. S. Hall and J. O. Hamblen, Using FPGAs to simulate and implement digital design systems in the classroom, *In Proceedings of the American Society for Engineering Education Southeastern Section Annual Meeting, Tuscaloosa,* April 2006.

[9] Quintans C, Valdes MD, Moure MJ, Ferreira LF, Mandado E, Digital Electronics Learning System Based on FPGA Applications. *35th ASEE/IEEE Frontiers in Education Conference.* 2005.

[10] V. Sklyarov and I. Skliarova, Reconfigurable Systems in Education. *13th International Conference ( FPL),* Lisbon, Portugal, September, 2003.

[11] Holland M, Harris J, Hauck S, Harnessing FPGAs for Computer Architecture Education. *Proceedings of the 2003 IEEE International Conference on MICROELECTRONIC SYSTEMS EDUCATION (MSE'03).*

[12] Michael A. Soderstrand, Role of FPGAs in undergraduate project courses. *Proceedings of the 1997 International Conference on MICROELECTRONICS SYSTEMS EDUCATION (MSE '97)*

[13] Koch A, Golze U, FPGA Applications in Education and Research. *proceedings of 4th eurochip workshop,* toledo, 1993. *pp.260-26.*

[14] Kiray V, Jambulov A, FPGA based Digital Electronic Education. *International Kyrgyz-Kazak Electronics and Computer Conference Organization.* Bishkek 2006*, pp.69-74.*

[15] Kiray V, Jambulov A, Satibaldiev D, FPGA based Digital Electronic Education - Assessment for first period. *International Kyrgyz-Kazak Electronics and Computer Conference Organization.* Almaty 2009, pp.17-22.

[16] Kiray V, Jambulov A, (2009). FPGA based Digital Electronic Education and a simulator core design for A/D communication. *Scientific Research and Essays - Academic Journals (SRE)* November 2009, pp.111-120

[17] Altera, "DE2 Development and Education Board," available online at http://www.altera.com.

[18] Altera, "Quartus II Web Edition Software," available online at http://www.altera.com.

***Author***

*He is working at Turgut Özal Unv. Engineering Faculty, Electrics and Electronics Engineering Department. Ankara, Turkey.*

*vkiray@turgutozal.edu.tr*