Developing a Graduate Level Embedded System Programming Course Content by Using Blended Programming Methodologies: Text-Based and Graphical

Hayriye Korkmaz, IEEE Member
University of Marmara, Faculty of Technology,
Dep. of Electrical & Electronics Eng., Istanbul, TURKEY
hkorkmaz@marmara.edu.tr

Sezen Azaklar Bal, PhD Student
University of Marmara, Institute of Pure and Applied Sci.
Istanbul, TURKEY
sezenazaklar@gmail.com

Ercan Cosgun, Instructor
University of Gedik, Vocational School
Pendik, Istanbul, TURKEY
ercan.cosgun@gedik.edu.tr

Kenan Toker, Research Assistant
University of Marmara, Faculty of Technical Education,
Dep. of Electronics and Computer Education,
Istanbul, TURKEY
ktoker@marmara.edu.tr

***Abstract*—This article deals with identifying the materials and the methods that will be used in this course based on "Embedded System Programming (ESP)" course for Electrical-Electronics Engineering Graduate program of MU Faculty of Technology. When the varying levels of skill and knowledge of the students is considered, it becomes necessary to adopt a blended programming method for the contents of the course. Even though the course is regarded as 3+0 ECTS of 8 credits (Theory: 3, Applied: 0), a content supported with extensive application is recommended. The content of the course allows developing applications using Xilinx Spartan 3E Starter Kit. Students taking this course are able to choose a method best fitted to their programming skills or preferences. However, it might be possible that the lecturer may not have a wide range of programming knowledge and skills. In this case, the graduate students are expected to determine and complete their own needs or shortcomings with the accumulation of knowledge they gained in their undergraduate studies. With the lecturer's guidance, the students are free to choose whether to use graphical programming (LabVIEW, LabVIEW FPGA Module) or conventional text-based programming (Verilog, VHDL, Matlab HDL Coder) in their projects. In this way, we expect an improvement in understanding of the contents and an expansion in scope.**

Keywords— Embedded system programming, FPGAs, HDL, Verilog, LabVIEW FPGA Module, Xilinx ISE.

I. INTRODUCTION

Because of the fact that FPGAs have diverse and extensive fields of use, it became a necessity to offer this subject in universities [1-4]. Today even though FPGA is chiefly a joint subject of Computer Engineering and Electrical-Electronics Engineering, it also finds a place for itself in Electronics-Communication Engineering, Biomedical Engineering and Mechatronics Engineering. While some domestic and international universities offer this content by updating conventional Logic Circuits or Digital Design courses, some provide a brand new course under the name of Advanced Digital Design, Embedded System Programming. These courses are mostly offered at senior year of undergraduate programs or at graduate level. Table I gives the names of the courses in some of the universities based on FPGA with different names but similar content in addition to the materials used.

With the aid of FPGA training sets and programming platforms developed by various firms [5-7], students have a chance to perform practical applications. Because FPGA is such a current field, it became necessary for universities to offer courses on this subject and encourage students to use this software and relevant software components in their dissertations. This subject field is a must in engineering curriculums.

This article deals with identifying the materials and the methods that will be used in this course based on "Embedded System Programming" course for MU Electr. & Electronics Engineering Turkish Graduate program with thesis. The said graduate program accepts applications from students of various undergraduate programs. The distribution of students accepted to 2013-2014 Electrical-Electronics Engineering Turkish Graduate Program with thesis according to their Undergraduate Program are given in Table II.

Literature research shows that text-based programming method is recommended more commonly [16]; only one paper mentions the use of LabVIEW FPGA Module [17]. The reason is the earliest developed method in HDLs being text based. The first HDLs appeared in the late 1960s were looking like more traditional languages [18]. The first that had a lasting effect was described in 1971 in C. The first modern HDL, Verilog, was introduced in 1985. In 1987, a request from the U.S. Department of Defense led to the development of VHDL. On the other hand, the earliest

978-1-4799-3190-3/14/$31.00 ©2014 IEEE 3-5 April 2014, Military Museum and Cultural Center, Harbiye, Istanbul, Turkey
2014 IEEE Global Engineering Education Conference (EDUCON)

version of LabVIEW FPGA Module 8.0 is introduced to the market in 2005 [19].

TABLE I. SIMILAR COURSE EXAMPLES FROM NATIONAL AND INTERNATIONAL UNIVERSITIES

University Name	Department Name	Course Name – (Under Graduate/Graduate)	HW and SW Materials Used in Course
İTÜ [8]	Electronics Engineering	Very Large Scale Integrated Design - II	Xilinx ISE, Verilog, Cadence
Boğaziçi [9]	Computer Engineering / Electrical and Electronics Eng.	Advanced Digital Design	Xilinx ISE- Modelsim- VHDL- Xilinx Spartan II
YTÜ [10]	Electronics and Communication Eng.	Programmable Logic Circuit Design	Xilinx ISE- VHDL -Spartan-3E Starter Kit
York [11]	Digital System Eng.	Advanced Digital Design (MSc) Embedded Systems for FPGA (MSc)	VHDL- Xilinx XUP Virtex-II Pro FPGA Development System
Atılım [12]	Electrical and Electronics Eng.	Embedded System Design with FPGA (MSc)	Xilinx Spartan Starter Kit-Verilog
Maryland University [13]	Electrical and Computer Eng.	Advanced Digital Design with HDL	Verilog
The University of Arizona [14]	Electrical and Computer Eng.	Digital Logic	Xilinx Spartan 3E Starter Kit-Verilog, Xilinx ISE
University of Washington [15]	Computer Science Engineering	Advanced Digital Design	HDL

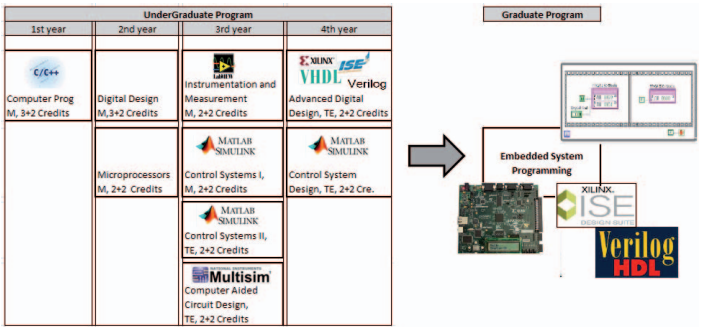
TABLE II. DISTRIBUTION OF THE STUDENTS ACCEPTED TO MASTER'S DEGREE PROGRAM IN EEE (TURKISH) ACCORDING TO THEIR GRADUATE PROGRAM.

	Electronics & Computer Education	Electrical & Electronics Engineering	Computer Engineering	Others*
Number of students	15	3	1	1

*Faculty of Arts and Sciences, Physics

II. THE STRUCTURE AND THE TEACHING METHOD OF THE COURSE

For the ESP course, the students should have certain preliminary knowledge and skills. However, the situation in Table II indicates different levels of background knowledge and skills. When the four-years of education the MU EEM Undergraduate students received are considered, the courses complementing this content are given in Fig 1. Computer programming is a course offered in freshman year and helps students to learn the logic of programming. C based programming taught in the scope of the course is quite beneficial in learning the Verilog-HDL language. This is because Verilog-HDL is a much simpler Hardware Description Language (HDL) to learn than VHDL because of its similarity to C programming language [20,21]. In sophomore year, students take the Digital Design and Microprocessors course. The design methods taught in digital design course provides a road map for this class. With microprocessors course, students get a chance to learn about I/O interfaces, timers/counters and their design. Measurement and Instrumentation course offered in junior year might seem less complementary at first glance; still it enables students to familiarize themselves for the first time with LabVIEW graphical development platform. In this way, Undergraduate students are able to gain knowledge and application skills at a preliminary level. On the other side, in Control courses, students mostly get a chance to work with Matlab/Simulink. With Advanced Digital Design course offered as elective in senior year, students gain the necessary basic knowledge and skills on HDLs. In this way, students graduated from above mentioned Undergraduate programs gain enough knowledge and skills for the graduate course of ESP. Nevertheless, students graduated from different programs of different universities will have different levels of awareness in C, LabVIEW and/or Matlab/Simulink. Besides, some of these courses listed as complementary are offered only as electives. This is why varying levels of programming knowledge and experience of the students should be considered while structuring the course at graduate level.



M: Mandatory, TE: Technical Elective

Fig.1. Supporting courses for Embedded System Programming in a 7 semester + 1 semester in industry curriculum

Because today most of the electronic devices contain

embedded systems, we aim to introduce and teach the embedded system structure and its programming in addition to the necessary skills for problem focused embedded system designing with the help of applied projects and assignments.

In the 14 weeks of curriculum given in Table III, the contents of the first 8 weeks are taught by the lecturer. Later, the students will start to learn by practice with various applications. In the last 2 weeks, students will have an opportunity to share their project applications with their classmates in class presentations. In this way, students will be able to eliminate their shortcomings by exchanging information and improve themselves by learning different methods.

TABLE III. COURSE CONTENT

Subjects	Duration (3 hours in a week)
Introduction to Embedded Systems and FPGA architecture, VHDL / Verilog Hardware Description Languages and syntax, Structural, dataflow and behavioural styles of Verilog	3 weeks
Introduction to Xilinx ISE software tool for synthesis of various levels of design, simulation, verification environment, Testing Program (Test Bench) Development and Behavioural Simulation (Xilinx ISIM), real-time debugging and verification-(ChipScope Pro)	3 weeks
LabVIEW Graphical Development Platform and LabVIEW FPGA Module	2 weeks
Midterm Exams	1 week
Combinational and sequential logic circuit design, Hand out of the term project (individual or group)	1 week
Applications: Processor design, memory blocks and timing, I/O peripheral designs, serial and parallel protocols.	3 weeks
Presentation and evaluation of projects	2 weeks
Final Exams	2 weeks

The learning outcomes of Embedded System Programming course can be listed as follows:

- Comprehends the basic facts about embedded system and FPGA.
- Effectively uses Embedded System Programming design tools and software.
- Designs advanced digital system and embedded system by using text based hardware description language or graphical programming.
- Simulates the digital design and operates it on FPGA.
- Understands the methods for project design, development, testing and debugging.

The theory and the application of ESP course cannot be separated from each other. For this, a laboratory containing FPGA training sets, computer and a projector is necessary. Our university already has Xilinx ISE Design Suite and LabVIEW FPGA Module as software, and Digilent Xilinx Spartan 3E Starter Kit as hardware.

III. MODEL APPLICATION

The design and comparison of a simple 4 bytes down counter circuit in Verilog and LabVIEW environment are handled in this section.

A. Implementation by using Verilog

4-bit down counter example has 4 blocks as shown in Fig. 2: Clock Divider: It divides the Kit's clock by a user-defined constant to get a new system clock, Down counter: It starts from 1111 and counts down by 1, BCD Converter: Binary-to-BCD conversion to drive the LCD display control, LCD Display Control: The LCD is a practical way to display a variety of information using standard ASCII and custom characters. However, these displays are not fast. Scrolling the display at half-second intervals tests the practical limit for clarity. Compared with the 50 MHz clock available on the board, the display is slow. A PicoBlaze processor efficiently controls display timing plus the actual content of the display [22, 23].

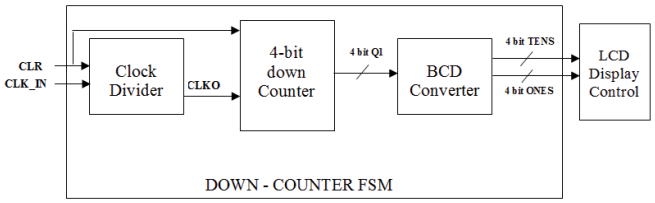


Fig.2. Block diagram of 4-bit down counter.

Since Xilinx Spartan 3E Starter Kit has a 50 MHz Clock signal, counter circuit will work at every 20 ns. Because it is impossible to monitor the outputs in this case, "Clock Divider" should be set to give output at a slower pace. To be able to display the outputs of the counter circuit on LCD, BCD conversion should be performed.

Binary to BCD Conversion block uses a well-known "Shift and Add-3 Algorithm" [24]. This algorithm can be used on a binary number with any number of bits. The most important point that you should notice is that the shift number must be equal to bit number. If the number in the "ONES" column is greater than four, then it must be added to three.

1) Clock Divider:

```
module clkdiv(CLR, CLK_IN, CLK_OUT);
input CLR, CLK_IN;
output CLK_OUT;
reg CLK_OUT;
reg[24:0] bol;
reg clk;
initial begin
clk=0;
bol=0;
end
```

```
always @ (posedge CLR or posedge CLK_IN)
begin
if( CLR )begin
bol = 0;
CLK_OUT = 0;
clk = 0;
end
else begin
if( bol == 25000000) begin
CLK_OUT = ~clk;
clk = ~clk;
bol = 0;
end
else begin
CLK_OUT = clk;
clk = clk;
bol = bol + 1;
end
end
end
endmodule
```

2) 4-Bit Down Counter:

```
module countdown (CLK, CLR, Q);
input CLK, CLR;
output [3:0] Q;
reg [3:0] tmp;
initial
 tmp = 4'b1111;
  always @(posedge CLK or posedge CLR)
    begin
      if (CLR)
              tmp = 4'b0000;
      else
        tmp = tmp - 1'b1;
      end
  assign Q = tmp;
endmodule
```

3) Binary to BCD Converter:

```
module BCD (
input  [3:0] binary,
output reg [3:0] tens,
output reg [3:0] ones
);
integer i;
always @(binary)
begin
              tens=4 'd0;
              ones=4 'd0;
              for ( i=3; i>=0; i=i-1)
              begin
                            // add 3 to column >=4
                            if (tens>=4)
                                          tens=tens+3;
                            if (ones>=4)
                                          ones=ones+3;
                            // shift left one
                            tens=tens <<1;
                            tens[0]=ones[3];
                            ones=ones << 1;
                            ones[0] = binary [i];
              end
end
endmodule
```

4) Down - Counter Finite State Machine (FSM) Circuit:

```
module COUNTER(CLR,CLK_IN,TENS, ONES);
input CLR, CLK_IN;
output [3:0] TENS, ONES;
wire CLKO;
wire [3:0] Q1;
clkdiv m1(CLR, CLK_IN, CLKO);
countdown m2(CLKO,CLR, Q1);
BCD m3(Q1, TENS, ONES);
endmodule
```

B. Implementation by Using Graphical Programming

When the same design is implemented in LabVIEW environment as shown in Fig. 3, it is possible to create a design without any knowledge of syntax, by wiring the objects (symbols) to each other through the use of data stream logic only. Table IV gives a detailed comparison of the same content developed by using text based Verilog-HDL to the content developed by using LabVIEW FPGA Module as the graphical based programming method. When the workloads are considered, the most obvious difference takes place in the LCD driving process. SPARTAN 3E XUP LabVIEW driver which was developed to enable educators to use LabVIEW FPGA to teach digital and embedded design concepts simplifies the process. However ASCII code conversion is a common task for both methods.

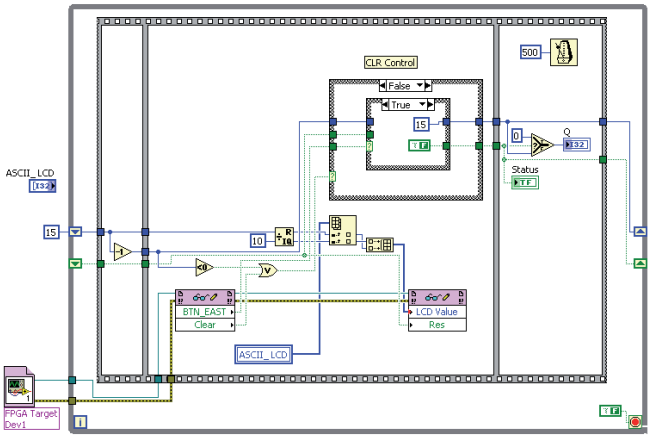


Fig. 3. 4-bit down counter example by using Graphical Programming method

Here, it is a rather relative comparison to refer one method as more simple or difficult than the other. It is best to make a selection based on the level of knowledge and skills the lecturer and the student possess as well as the requirements of the application to be developed. Forcing a single method may lead to failure of the students. Therefore, the opportunity to select is considered a facilitating and a success boosting factor.

IV. CONCLUSION AND DISCUSSION

This article deals with identifying the materials and the methods that will be used in the "Embedded System Programming" course taught at the Graduate level of MU Faculty of Technology EEE curriculum based on the varying levels of knowledge and skills of the students. We find it more appropriate to teach the class contents in two different ways using either conventional text based programming (Verilog) or Graphical programming (LabVIEW and FPGA Module).

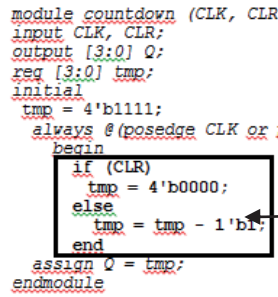
When compared with a student model that has a certain level of preliminary Graphical programming knowledge, it

might be better to emphasize the use of LabVIEW FPGA Module not to lose time to teach Verilog syntax in order to be able to focus more on system design concepts [2]. Or alternatively, at the initial stage, getting start with the ISE Design Suite platform's Schematics option (Project>New Source>Schematics) is recommended to overcome some challenges of text-based programming. Then, remembering that testbench files which drive the simulations are also Verilog files, necessary code can be created automatically by using "Verilog Test Fixture" option for the Schematics type source files [25]. Thus, student success can be increased by using a combination of schematics and Verilog to specify a design.

Even though LabVIEW enables ease of use without necessitating any knowledge of HDL languages, it also has certain limitations. For example, some of the functions may not be available for FPGA HW. In addition, when the compilation reports are considered as shown in Table V, it is obvious that text based programming has a remarkable and clear advantage over FPGA LabVIEW Module in terms of hardware resources' usage.

TABLE-IV. COMPARISON OF THE FUNCTION LIST DEVELOPED FOR BOTH METHODS.

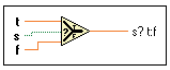
<table>
<tr><th>Functions</th><th>Method 1-Verilog-HDL</th><th>Method-2 LabVIEW FPGA Module</th></tr>
<tr><td>Clock Generation</td><td>You should write a 31-Line code. Please see Clock Divider section given above for details.</td><td> 

Find the icon from Function →Time and Dialog Palette and place it on the block diagram. After that, place a Numeric constant and wire it to the input terminal of "Wait Until Next ms Multiple" object.</td></tr>
<tr><td></td><td colspan="2">Calculation T= 1/(50MHz/25000000)=0.5s.</td></tr>
<tr><td>Count down by 1</td><td> 

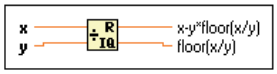
*Defining CLK Input: Explained as in the above row.

*Defining CLR Input: Find the icon from FPGA IO →BTN-EAST and place.

*Defining Q output: See Fig 3.

*If BTN-EAST is pressed, then Q is equal to "0", else it starts from "1111" and counts down by 1. "Select" is used for if-then-else structure. For details see Fig 3.</td></tr>
<tr><td>Binary to BCD Conversion</td><td><table>
<tr><th>Operation</th><th>Tens</th><th>Ones</th><th>Binary</th></tr>
<tr><td>Start</td><td></td><td></td><td>1111</td></tr>
<tr><td>Shift</td><td></td><td>1</td><td>111</td></tr>
<tr><td>Shift</td><td></td><td>11</td><td>11</td></tr>
<tr><td>Shift</td><td></td><td>111</td><td>1</td></tr>
<tr><td>Add-3</td><td></td><td>1010</td><td>1</td></tr>
<tr><td>Shift</td><td>1</td><td>0101</td><td></td></tr>
<tr><td>Decimal</td><td>1</td><td>5</td><td></td></tr>
</table>
You should write the codes shown in Binary to BCD Converter section above.</td><td>*Quotient & Remainder: Separates the ONEs and TENS digit. Here x is 15, y is 10.

 
</td></tr>
<tr><td>Displaying the results on the Kit-LCD</td><td>This paper does not include the control of the LCD Display in Verilog HDL. It is not an easy task and you should write long-codes as explained in Section A.</td><td> 

* The shown icon enables you to write the array content on the display.
* Fig.3 shows the complete tasks including LCD control and displaying the results.</td></tr>
</table>

TABLE-V. COMPILATION/DEVICE UTILIZATION SUMMARY

		LabVIEW FPGA Module		Verilog	
	Available	**Used**	**Utilization**	**Used**	**Utilization**
Number of Slice Flip Flops	9312	2077	22%	62	1%
Number of 4 input LUTs	9312	2048	21%	259	2%
Number of Occupied Slices	4656	1724	37%	169	3%
Number of Bounded IOBs	232	88	37%	9	3%
Number of BUFGMUXs	24	2	8%	1	4%

REFERENCES

[1] A. J. Arauqo, J. C. Alves, "A Project Based Methodology to Teach a Course on Advanced Digiral Systems Design", WSEAS Transactions on Advanced in Engineering Education, Issue 6, Volume 5, June 2008.

[2] W. Balid, M. Abdulwahed, "A novel FPGA educational paradigm using the next generation programming languages case of an embedded FPGA system course", IEEE Global Engineering Education Conference, pp. 23-31, March 2013

[3] T. Tuma, I. Fajfar, A New Curriculum for Teaching Embedded Systems at the University of Ljubljana", Information Technology Based Higher Education and Training, pp. 14-19 July 2006.

[4] K. Ferens, "Teaching Embedded System Design", Proceedings of the Canadian Engineering Education Association Conference, 2012.

[5] http://www.xilinx.com/products/boards-and-kits/EK-S6-SP605-G.htm

[6] http://www.digilentinc.com/Products/Catalog.cfm?NavPath=2,400&Cat=10&FPGA

[7] http://www.altera.com/devices/fpga/fpga-index.html

[8] http://www.vlsi.itu.edu.tr/ituvlsi/webs/courses/undergraduate/ele425/course-information/ELE425-VLSI2-ECTS

[9] http://www.ee.boun.edu.tr/Graduate/GraduateCourses/EE540/tabid/318/language/en-US/Default.aspx

[10] http://www.bologna.yildiz.edu.tr/index.php?r=course/view&id=281&aid=5

[11] http://www.elec.york.ac.uk/gsp/mscDigital/content.html#ADD

[12]http://eee.atilim.edu.tr/academicprogramcourses/view/id/1439?lang=en

[13] http://www.ece.umd.edu/undergrad/courses/300-level#ENEE359V

[14]http://www2.engr.arizona.edu/~slysecky/courses/ece274_s13/index.html

[15] http://courses.cs.washington.edu/courses/cse467/13au/

[16] V.Sklyarov, I. Skliarova, "Teaching Reconfigurable Systems: Methods, Tools, Tutorials, and Projects", IEEE Transactions on Education, Vol. 48, No. 2, 290-300, May 2005.

[17] W. Balid, M. Abdulwahed, I. Alrouh, "Constructivist Multi-Access Lab Approach in Teaching FPGA Systems Design with LabVIEW", International Journal of Engineering Pedagogy, vol.3, Issue 3, EDUCON 2013.

[18] http://en.wikipedia.org/wiki/Hardware_description_language

[19] http://www.ni.com/pdf/manuals/371572a.pdf

[20] http://www.pearsonhighered.com/samplechapter/0136392539.pdf

[21] http://eng.upm.edu.my/~kmbs/teaching/digitaldesign/tutorial/tutorial3.pdf

[22] www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf

[23] http://www.xilinx.com/products/boards/s3estarter/files/s3esk_startup.pdf

[24] http://en.wikipedia.org/wiki/Double_dabble

[25] ISE 12.2 and Spartan3E Board Tutorial, v12.2.1-August 2010