Multidisciplinary Journal for Education, Social and Technological Sciences
http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

Calculator: A Hardware Design, Math and Software Programming Project Base Learning

F. Criado, L. Costero, P. Cabeza, M. Sánchez-Élez*

Dpto. de Arquitectura de Computadores y Automática. Universidad Complutense de Madrid,
Avda. Complutense s/n, 28040 Madrid, Spain.

* Corresponding author: Email: marcos@ucm.es; Phone: + 34 913947604

Received: 2013-12-25; Accepted: 2014-08-12

Abstract
This paper presents the implementation by the students of a complex calculator in hardware. This project meets hardware design goals, and also highly motivates them to use competences learned in others subjects. The learning process, associated to System Design, is hard enough because the students have to deal with parallel execution, signal delay, synchronization … Then, to strengthen the knowledge of hardware design a methodology as project based learning is proposed. Moreover, it is also used to reinforce cross-subjects like math and software programming. This methodology creates a course dynamics that is closer to a professional environment where they will work with software and mathematics to resolve the hardware design problems. The students design from zero the functionality of the calculator. They are who make the decisions about the math operations that it is able to resolve it, and also the operands format or how to introduce a complex equation into the calculator. This will increase the student intrinsic motivation. In addition, since the choices may have consequences on the reliability of the calculator, students are encouraged to program in software the decisions about how implement the selected mathematical algorithm. Although math and hardware design are two tough subjects for students, the perception that they get at the end of the course is quite positive.

Keywords
Active learning methods; Project Based Learning; Logic Design; Computer Aided Design

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

1. Introduction

Students enrolled in any of the different computer degrees they will study computer mechanics (computation, communication, coordination, automation, and recollection), design principles (simplicity, performance, reliability, resolvability, and security) and practices (programming, engineering systems, modeling and validation, innovating, and applying) (Denning 2003). These students will develop "computational thinking", which shares elements with various other types of thinking such as algorithmic thinking, engineering thinking, and mathematical thinking (Perkovic et al. 2010), in other words, they should develop the intellectual skills necessary to apply computational techniques or computer applications to problems and projects in any discipline. However, in our experience, to make it possible, in the subjects within the computer science degree, it is necessary to encourage interaction between computer science concepts and engineering and math concepts. We think this is favorable when the courses have cross-subject projects, within a project base learning (PBL) methodology, which is the main objective of this paper.

We have chosen System Design course as the main course to develop this cross-subject project. The students will design computing systems and computing components of products, develop and test their prototypes, and implement them to market. Therefore, the goal of the methodology presented in this paper is achieve that the students correctly design a complex algorithm into hardware with the help of any of the existing computer-aided design tools, without forgetting the interrelationship of this course with others subjects that are given in the degree. Hardware design is very important for undergraduate curricula in Computer Science, (Harrison et al., 1998) and (McGettrick et al., 2003), and the students need to manage concepts from many different fields.

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

Moreover, EDA industry has evolved towards the use of CAD tools which allow a rapid prototyping and manufacturing of very complex designs. It requires the management of a wide knowledge in many new areas such as, Hardware Design Languages (HDLs), FPGA-based CAD tools, use of IP cores, soft cores, advanced synchronization techniques, design of virtual systems, … (Sklyarov 2004).

Several authors have presented methodologies to deal with the challenges of teaching hardware design today. Some of them still hold a traditional point of view of hardware design, applied to a new hardware target (FPGA), such as (Velez et al 2007). In addition, most digital design texts dealing with HDLs (Ashenden 1996), (Roth 2007) present a classical view of digital design concepts merely translated into HDL. They commonly propose the structural design of a simple microcontroller as final project, which adds no significant understanding to the former schematic approach. Although in System Design courses instructors usually make use of practical examples in the form of guided lab exercises, the consequence, from the student point of view, is that the proposed guided designs and their results seem to be a black box with data to collect. This approach is far from real design problems. Instead, a PBL methodology is more suitable for these courses as it creates a course dynamics that is closer to a professional environment (Yadav et al. 2011).

In the field of apply PBL on System Design courses, the proposal of (Araujo et al 2008) is interesting, although these authors make separate simulation and synthesis projects. These are complementary aspects of a whole design process and should be analyzed together from the beginning of the course, since a correct simulation does not guarantee a correct implementation on the target hardware, FPGA. The work presented in (Cristian-Gyozo Haba et al 2011) uses PBL in a very attractive way, although they use VIP technology, which requires sophisticated laboratory equipment.

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

Although the purpose of this paper is not directly teach math, we should not forget that in the different computer degrees subjects related with math are the toughest in the opinion of students, whose main complaint was the lack of any attempt to demonstrate the applicability of those areas of mathematics which they were being taught. This may be due to the methodology usually used; it involves mainly formal lectures, which alone, are not particularly effective in teaching mathematics. Mathematical courses are built in such a way that if a student misses a key concept at the beginning of the lecture, the rest of the lecture can be lost for him or her (Sazhin 1998). Among the proposed solutions, authors in (Beaudin 2002) demonstrate that teaching engineering mathematics with technology constitutes a good opportunity to teach classical subjects with a new taste. It allows teachers to adapt their teaching methods to the new technological reality. Most important, technology helps the teacher to present live examples of what mathematics are, how beautiful they are. Students will much more appreciate theorems and general results if they can visualize concrete examples. The difficulty to teach mathematics has carried surveys during the last few years in favor of less mathematically rigorous CS curriculum (Sazhin 1998). They do not deny the importance of math, but emphasizes that if we were to include all the math that is useful for a CS major, it would result in a specialized math major, leaving us short on computer science itself. Then math is an important subject that should be interrelated with programming and hardware design concepts. However, it is most usual for programming and not at all for hardware design.

Last but not least, designing and implementing their own calculator machine as the course project is a key aspect of the methodology, as the students are able to interrelate different pieces, which have been studied in the computer science or computer engineering degree, improving the perception that different courses would no longer be compartmentalized. Moreover, the creation of their own calculator machine is a challenge

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

related to a reality that is well known for them compared to other possible hardware projects. Most students find it a little hard in the beginning because they have to handle real problems present in hardware design. By the time they are designing their own project they become very involved in the course and work really hard (Higley et al. 2001).

The rest of this paper is organized as follows: Section 2 describes the methodology implemented. Section 3 presents details of the calculator case study, Section 4 describes in detail the mathematical algorithms used in the system, and Section 5 presents the conclusions of this work.

2. Proposed methodology: integration of different teaching methods

Project Based Learning in engineering programs has been reported by several authors (Glass 2000) and (Gunstra 2001). PBL is designed to engage students in research and implementation of authentic problems and it is easily adapted to add cross-cutting concerns from other subjects to help solve the problem. Project Based Learning emphasizes learning activities that are long-term, interdisciplinary and student-centered. Unlike traditional, teacher-led classroom activities, students often must organize their own work and manage their own time in a project-based class. Project-based instruction differs from traditional inquiry by its emphasis on students' collaborative artefact construction to represent what is being learned. Projects can increase student interest because the involve students in solving authentic problems, in working with others, and in building real solutions, this makes that the PBL methodology is the perfect candidate to face problems (courses) that are hard for the students.

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

The students that going to work following a PBL methodology have passed a basic course of digital design. Therefore, they are already familiar with: laws of Boolean algebra; binary number representation; basic combinational logic design, basic gates, multi-level digital design; combinational building blocks; basic sequential building block and Finite State Machine (FSM) design. A basic knowledge of VHDL is also desirable, so a 4 - 6 hours VHDL tutorial (Sanchez-Elez 2011) is given to the students, regardless of whether or not they have studied it in previous courses.

The key focus of the methodology is on creating the expectation that at the end of the course, once they have mastered certain design challenges, the student are ready to implement a complex calculator machine. The other important aspect is to encourage students to use concepts learned in other subjects, like calculus, algebra or programming, to make design decisions that involve the realization of the most accurate calculations possible. The project is developed using VHDL as design language, Spartan 3 (Spartan 3) as prototyping platform and Xilinx ISE (ISE) as CAD tool.

The cornerstones of this methodology are the following:

- At the beginning of the course the instructors present an overall guideline of the theoretical concepts required for the implementation of different hardware modules, all of which may be combined to create any calculator machine.
- The different hardware modules that the students develop have been specifically designed to present increasing challenges in order to capacitate them to finally implement their own calculator. These modules are: simple and complex ASM; use of the FPGA internal memories; and input interfaces (PS2 Keyboard).
- There is one week exclusively dedicated to explain different HW algorithms for multiplication and division (Parhami 2000).

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

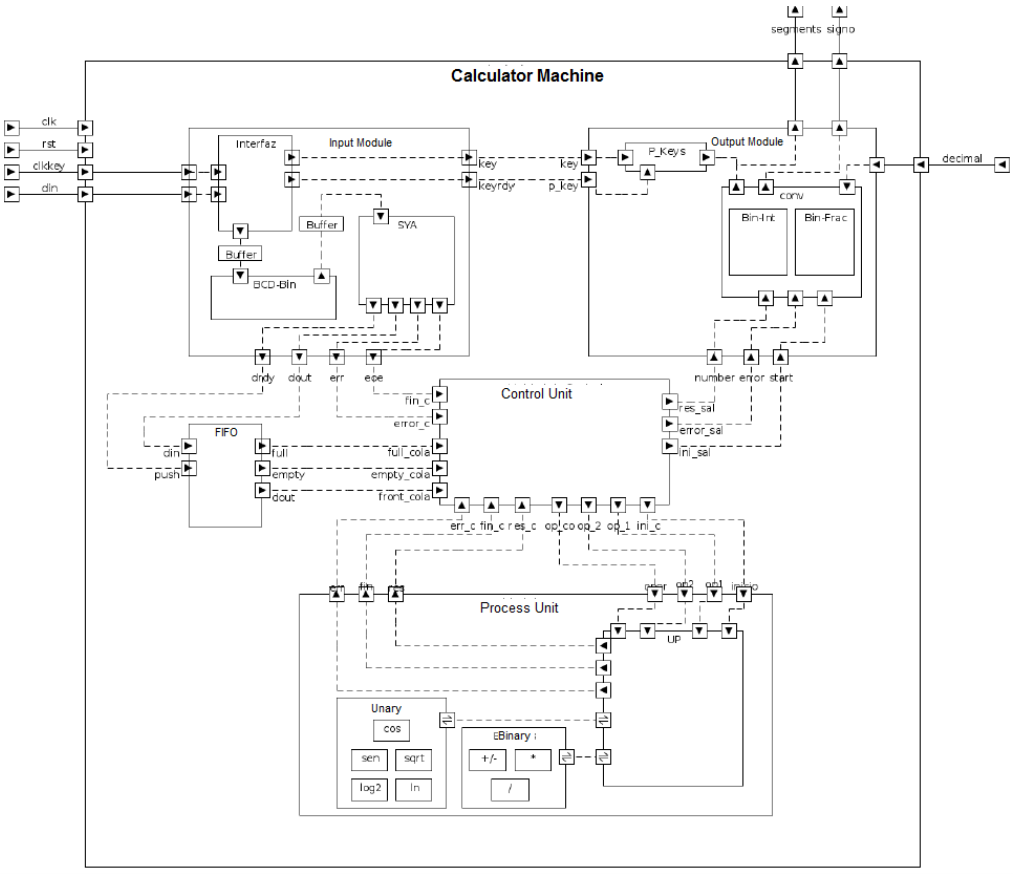


Figure 1. Calculator Machine Block Diagram

- Midterm group discussion: students are requested to suggest calculator functionalities that may be implemented with the hardware modules developed along the course. This should be presented as a realistic project, that is, they have to evaluate the difficulty of the math functions implementation by identifying the necessary hardware elements, the amount of reusable code and deadlines. In some cases the teacher addresses the students to either simplify or increase the complexity of their project, in order to meet the educative goals and deadlines of the course.

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

- Along the course, think critically over the numerical approximations done. The students should present a revision of the accuracy of the math functions implemented. They may study the differences using Matlab or programming their hardware approximation in other high level languages. They have to discuss the different notation algorithm and choose one. They could use recursive software algorithm for resolve math problems that have to translate to hardware. They should use Backus Normal Form …

3. Case study

Since the students face the project working in groups, it is useful done a division of different parts that form the calculator as well as the implementation of a communication protocol between the different modules. Thus students within each group can distributed in a more simple way their workload, because implement a complete calculator of this kind by a single student could be a sufficiently complicated work that only would help to discourage them.

The design of the calculator is formed by four modules as is shown in Figure 1:

- **Input Module:** data and operations are introduced into the calculator following postfix notation (reverse Polish), it is a mathematical notation in which every operator follows all of its operands and it is usually parenthesis free. Additionally, users of reverse Polish calculators made fewer mistakes than for other types of calculator (Agate et al. 1998). The input module is divided into three sub-modules (interface, BCD-Bin and SYA). The interface directly translates the key pressed into a number or an operator. The communication protocol between the modules is based on an intermediate buffer that stores the output of the previous module

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

and some control signals to implement a basic handshake. In order to process the input, three different errors are taking into account: overflow (the integer is very large), parenthesis bad balanced, and syntactically incorrect numbers (i.e. more than one decimal point). If an error occurs a signal is sent to the Control Unit.

- **Output Module:** is very simple, the output has two modes, display input and display output. If a key is pressed it is passed to the first, if it is doing a calculation it is changed to the second.

- **Control Unit Module:** the control is fully dedicated to the evaluation of expressions in post-fix notation. Communication with the input module is through the intermediate buffer, with signals of coordination between both (handshake algorithm). The communication with the output module is very simple: signals are sent with the result or the error to be displayed. The control unit sends the operators to the Processing Unit and waits for the result or the error signal. The Control Unit, manages Input Module errors, and also handles buffer overflow (math expression too long) or stack overflow (too many nested operators). Moreover the Processing Unit send an error signal to the Control Unit when there is a failure in the execution of any math algorithm (i.e. try to calculate -1 square root).

- **Processing Unit Module:** is responsible for collecting the operands and the corresponding math function from the Control Unit. And lead the inputs to the module in charge of the operation. In order to make this task it uses a state machine. Each module is responsible for managing their own mistakes and only processing unit redirects the error signal to the Control Unit. The Processing Unit is design allows to add new operations quickly and almost effortlessly, since each

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

math operation is implemented in a differentiated module inside the Processing Unit.

4. Algorithm study

The most complicated part of the calculator implementation is related with the number representation and the math algorithms.

4.1. Number representation

The first problem to be solved by students involves the representation of numbers, since the system is a calculator, the numbers are introduced in BCD code and have to be translated to two complementary code. However, students have to take into account that the numbers have an integer part and a decimal part, which are translated separately by different algorithms.

The integer part is translated following the algorithm: $10N + x = (N<<1) + (N<<3) + x$. This implies the use of two adders and two shifters.

When the students attempt to convert the decimal part different algorithms arose, they may use the standard algorithm in which the decimal part is multiplied by two … However, the students who have participated in this project decided use the following algorithm:

1) The decimal part is converted to binary as if it were an integer.
2) The new integer is multiplied by two, if the result is lower than one then a zero is added to the representation and the algorithm repeats the step two.

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

3) The results is larger than one then a one if added to the representation and a new value is used, the new value is the result of subtracting the result obtained in the step two minus one.

4) If the process has been repeated fewer times than the number of decimal digits the algorithm repeats the step two.

The algorithm works properly when one is the binary representation of 10N, where N is the number of decimal places in BCD. Moreover, a counter is required to implement step four.

4.2 Input Algorithm

The user enters numbers and operators, to transform these inputs into a recognizable expression the system uses the Dijkstra's Shunting Yard Algorithm (SYA), it transforms infix notation into post-fix notation, which is the notation used by the system as explained before. The algorithm works as follows:

- If the input is a number it goes to the following module
- If the input is (, it is pushed in the operators' stack
- If the input is) , then the operators are popped until find a (.
- If it is an operator, depending on the operator precedence it is pushed or it goes to the following module
- It is necessary an End of Expression symbol (EoE), in this case all the operators are popped.

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

4.3 Arithmetic Algorithms

For the most basic calculator operations, namely, addition, subtraction, multiplication, two's complement, we have implemented them by using the arithmetic operations of IEEE libraries, properly encapsulating to manage the arithmetic errors.

The problems arise when the students try to implement most complex arithmetic functions.

In the case of the division there are several sequential algorithms as can be shown in (Parhami 2000), the students can use any of them. But they need to take into account the number of decimal places of the representation to truncate the division algorithm when the decimal places are completed. Moreover, overflow and division by zero have to be referred. It is very important, when implementing this module in the FPGA, studying the delay and the hardware resource use.

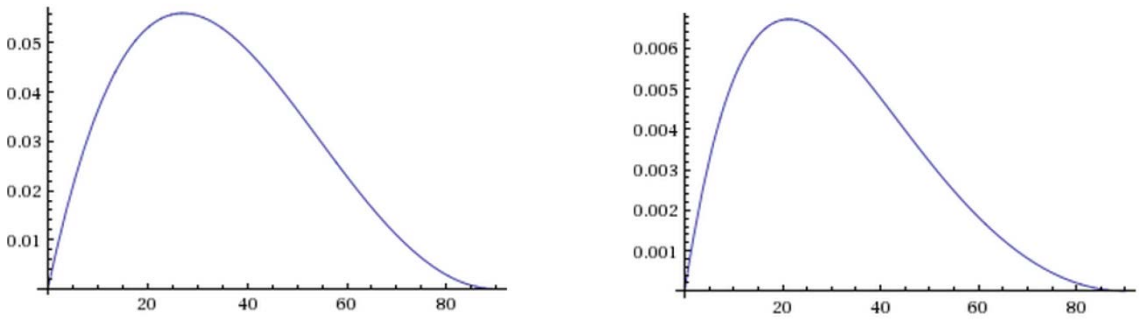
For the trigonometric operations the students can use the CORDIC algorithm (Parhami 2000), but the instructor encourage them to test their students-made option. Is this part of the development of the calculator one in which most clearly may need use concepts learnt in other courses. For example students could implement in software the trigonometric approximation and compare with the math results, and/or students could study the polynomial representation of the trigonometric functions and choose an approximation to implement in the FPGA. The students that have participated in this project decided use a

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

polynomial approximation with sexagesimal degrees instead of radians. The applied formula is: $sin(x) = \left(2 - \left|\frac{x}{90}\right|\right) \cdot \left(\frac{x}{90}\right)$.

The error of this approximation is shown in Figure 2.a, the maximum error value is 0.05. At this point the instructor can encourage the students to review their calculus concepts to reduce the error. The error is estimated as: $\epsilon=\alpha \bullet \beta(1-|\beta|)$ wherein β stands for the result of the previous approximation and α is a parameter. This parameter is with which students can play (program sin results for different α values). With α value of 0.25, which is easily implemented with a shifter, the error is shown in Figure 2.b.

Finally, since the algorithm always divides by ninety, the division is transformed to a multiplication after calculating its binary value.



(a) (b)

Figure 2. Absolute error of the different approaches

The students may develop other mathematical functions. In the book (Parhami 2000), there are algorithms enough to implement math function in hardware. The student should check if their numeric representation suits with the algorithm described and check, as was

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

done for sin function, if a change in the original algorithm would be required. Most of the students also implement natural logarithm, and square root.

5. Analysis and conclusions

The results presented in this section are related to relate to the students' feelings about course development. The course is held at Universidad Complutense de Madrid (Spartan 3) with an average of 25 students per class. A greater number of students complicate the correct development of the course.

This course in a university program meant to evaluate teaching performance. Within this program, the results of teaching activities are translated into terms of progress in student learning and in the assessment expressed as perceptions or opinions by students, graduates, academic leaders and the academic staff. This evaluation is coordinated within a program of the Spanish Agency in Quality Assurance in Higher Education, ANECA. This agency is a full member of European Association for Quality Assurance in Higher Education (ENQA) and a full member of the International Network for Quality Assurance Agencies in Higher Education (INQAAHE). The results obtained for this course in this program are shown in Table I (questions are made on a rating scale of 5).

Table 1. External evaluation of the teaching activities

	Mean
Lessons are well organized and prepared	4.24
The concepts are explained clearly and understandably	4.00
Arouses our interest in the course	4.35
Used properly teaching resources	4.00
Degree of fulfillment of the course syllabus	4.24
Course workload	3.53
Degree that have achieved the objectives of the course	4.13

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

Additionally, we have carried out a specific test about students' perceptions and opinions. The results are shown in table II. This test shows that students are satisfied with the course and feel confident on their skills for designing complex hardware, although for some of them the effort required was significant. It was quite surprising to us that although students consider mostly to study the subject following this methodology requires more effort, 100% of the students would return to attend it following this methodology.

Table 2. Internal evaluation of students' perceptions

This method implies that the course work is	
Much higher than a traditional method	56%
Higher than a traditional method	38%
Similar to a traditional method	6%
Lower than a traditional method	0%
Once all the implications of the methodology are known, would you enroll in this course again?	
Yes	100%
No	0%
Which of these options reflects your perception of the course?	
I was surprised positively	25 %
I liked it	56 %
I found it tedious	19 %
I feel disappointed	0 %
My perception about the design skills acquired is that	
I feel capable of designing any other complex system	31 %
I feel capable of designing systems of a similar nature	50 %
I feel capable of designing systems of a lower complexity	19 %
I still do not feel capable of facing the design of a complex system without assistance	0%
I feel that choosing this course for my curriculum	
Is not significant	0 %

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

Was interesting but I do not think will help me find a job	38 %
Will provide more professional opportunities for me	38 %
Will help me find the type of job I am interested in	25 %

5. References

Agate S.J., Drury C.G. (1980). Electronic calculators: which notation is the better? Applied Ergonomics, Volume 11, Issue 1, March 1980, Pages 2-6, ISSN 0003-6870

Araujo A. J. and Alves J. C. (2008), A Project Based Methodology to Teach a Course on Advanced Digital Systems Design. WSEAS Transactions on Advances in Engineering Education, Vol.5, No.6, pp. 437-446.

Ashenden P.J (1996). The designer's guide to VHDL, Morgan-Kaufmann, 1996

Beaudin, M. (2002). "Teaching mathematics to engineering students with hand-held technology". In 2nd International Conference on the Teaching of Mathematics.

Cristian-Gyozo Haba, A., Liviu Breniuc, A. (2011). Teaching FPGA embedded system design by course correlation and with the help of VIP systems. Proceedings of the 8th FPGAWorld Conference 2011.

Denning, P. J. (2003). Great principles of computing. Communications of the ACM, 46(11), 15-20.

Gajski D. (1996) Principles of digital design, Prentice-Hall

Glass R. L. (2000) A new Answer to 'How Important is Mathematics to the Software

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

Practitioner'?. IEEE Software, November/December 2000, pp.135-136.

Gunstra N. (2001). Universities aren't serving the IT workforce. Potomac Tech Journal, July 9, 2001, available at http://www.potomactechjournal.com

Harrison C.G., Jones P.L. (1998). Experiences with FPGA teaching. The Teaching of Digital Systems (Digest No. 1998/409), IEE Colloquium on , vol., no., pp.3/1-3/4, 18 May 1998, doi: 10.1049/ic:19980570

Higley, K.A., Marianno, C.M. (2001). Making Engineering Education Fun. Journal of Engineering Education, Vol 90, No. 1, pp105-107, January 2001, ISBN-13: 9780135103821

ISE Design Suite Software Manuals and Help - PDF Collection
www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/manuals.pdf

Keleman C. and Tucker A.B. () ITiCSE audience survey, available at http://www.cs.geneseo.edu/~baldwin/maththinking/ITiCSE-survey.html

Lethbridge T. (2007) Software Engineering Education Relevance survey, available at http://www.site.uottawa.ca/~tcl/edrel/

McGettrick A., Theys M.D, Soldan D.L, Srimani P.K. (2003) Computer Engineering Curriculum in the New Millennium, IEEE Transactions on Education, vol. 46, no. 4, November 2003

Parhami, B. (2000). Computer Arithmetic: Algoritms and Hardware Design. Oxford University Press. 2000

Perkovic L., Settle A., Hwang S. and Jones (2010), J. A Framework for Computational

http://dx.doi.org/10.4995/muse.2015.2275
EISSN: 2341-2593

Thinking across the Curriculum, Proceedings of the 2010 Conference on Innovation and Technology in Computer Science Education, 123-127.

Roth C.H.Jr. (2007), Digital Systems Design Using VHDL, 2nd Edition, University of Texas, Austin, ISBN-10: 0534384625 ISBN-13: 9780534384623

Sanchez-Elez M. (2011). Introduccion a la programación en VHDL. Available at http://eprints.ucm.es/26200/

Sazhin, S. S. (1998). Teaching mathematics to engineering students. International Journal of Engineering Education, 14(2), 145-152.

Sklyarov V., Skliarova I. (2004). Teaching reconfigurable systems: methods, tools, tutorials, and projects. Education, IEEE Transactions on, vol.48, no.2, pp. 290- 300, May 2005 doi: 10.1109/TE.2004.842909

Spartan-3 Generation FPGA User Guide, (2011). Available at www.xilinx.com/support/documentation/user_guides/ug331.pdf

Velez I., Sevillano J.F. (2007). A Course to Train Digital Hardware Designers for Industry. Education, IEEE Transactions on , vol.50, no.3, pp.236-243, Aug. 2007, doi: 10.1109/TE.2007.900027

Yadav A., Subedi D., Lundeberg M. A., Bunting C. F., (2011) Problem based Learning: Influence on Students' Learning in an Electrical Engineering Course, Journal of Engineering Education, Vol. 100,No.2, pp. 253-280.