An Undergraduate Course and Laboratory in Digital Signal Processing With Field Programmable Gate Arrays

Uwe Meyer-Bäse, *Member, IEEE*, Alonzo Vera, *Member, IEEE*, Anke Meyer-Bäse, *Member, IEEE*, Marios S. Pattichis, *Senior Member, IEEE*, and Reginald J. Perry, *Senior Member, IEEE*

***Abstract*—In this paper, an innovative educational approach to introducing undergraduates to both digital signal processing (DSP) and field programmable gate array (FPGA)-based design in a one-semester course and laboratory is described. While both DSP and FPGA-based courses are currently present in different curricula, this integrated approach reduces the number of electives students would have to take and at the same time provides a hands-on DSP experience. Developing such a new course with no textbook available is challenging. Therefore, the educational materials developed, the software tool evaluations, and topics to be covered in lectures and laboratories are described. Detailed evaluations of the selection of appropriate software and hardware platforms, topics to cover, and student feedback are provided.**

***Index Terms*—Digital signal processing (DSP), DSPbuilder, field programmable gate array (FPGA), MatLab/Simulink, System-Generator.**

I. Introduction

MANY undergraduate curricula include an introduction to digital signal processing (DSP). This introductory course often serves as a substitute for the usual signal and system courses using books like [1]–[3] or a more advanced course based on previous signal and system and communications courses using advanced books like [4] and [5].

An advanced undergraduate course or graduate course on DSP usually covers more advanced topics like fast Fourier transforms (FFTs) [6] or filter design [7], which are highly desirable from an industrial perspective. To provide hands-on experiments, textbooks often take advantage of the demos and toolboxes provided by MatLab/Simulink [8]–[10]. Innovative demonstrations using non-real-time multimedia data processing with audio or video output are also used to give students a better feeling for the data processing [11], [12].

Field programmable gate arrays (FPGAs) provide a unique platform for the students to design and implement real-time DSP architectures. To appreciate some of the topics associated with developing real-time hardware, a basic understanding of computer arithmetic [13], [14] and FPGA hardware and software is also required.

Currently, activities to teach DSP are often seen taught in combination with DSP microprocessors [15], [16] or VHDL/Verilog design [17]–[19]. In these previously reported approaches, the designs are most often first simulated in MatLab/Simulink and then coded by hand into HDL. This labor-intensive and error-prone task can be avoided if the FPGA Simulink library from the FPGA vendors is used. The FPGA vendors provide basic library elements that allow both the simulation in MatLab/Simulink and a "push-button" approach to generating the bit stream used to program the FPGA. More complicated IP blocks like finite-impulse response (FIR) filters or FFTs can then be instantiated via the vendor IP core libraries into Simulink.

There are many reasons why this is a promising approach, but a new software design flow also brings risk. All software tools applied in an undergraduate classroom setting need to undergo a careful review process. Properties that are important in a professional setting within the industry may differ from the requirements in a teaching lab environment. Compilation, simulation time, or device utilization for large designs may be an important factor to consider within industry projects, while in a teaching lab, issues like ease of use, the students' learning curve, or board cost and library support may be most important. A detailed study of the choice of an appropriate development tool and hardware is provided in Section II.

Among the advantages of the approach, it is noted that the use of Simulink allows the interconnections in large DSP systems to be visualized while providing a representation that is somewhat closer to representations in standard textbooks [3]. As a result, the class can focus much more time on teaching basic issues associated with teaching fundamental issues associated with DSP architectures and much less time focusing on teaching HDL. Clearly though, while the "push-button" approach can significantly help in developing complicated real-time systems, it can also limit the student's understanding of fundamental concepts. These issues are addressed by providing a detailed evaluation of the lecture and laboratory topic selection process in Section III.

In the remainder of this paper, a summary of the findings is provided in Section IV, and acknowledgments are made in Section V.

Manuscript received February 23, 2009; revised December 01, 2009. First published January 19, 2010; current version published November 03, 2010.

U. Meyer-Bäse, A. Meyer-Bäse, and R. J. Perry are with FAMU-FSU, Tallahassee, FL 32310 USA (e-mail: umb@eng.fsu.edu; amb@eng.fsu.edu; perry@eng.fsu.edu).

A. Vera is with Micro-RDC, Albuquerque, NM 87110 USA, and also with the Electrical and Computer Engineering Department, University of New Mexico, Albuquerque, NM 87131 USA (e-mail: alonzo@ece.unm.edu).

M. S. Pattichis is with the Electrical and Computer Engineering Department, University of New Mexico, Albuquerque, NM 87131 USA (e-mail: pattichis@ece.unm.edu).

Digital Object Identifier 10.1109/TE.2009.2039216

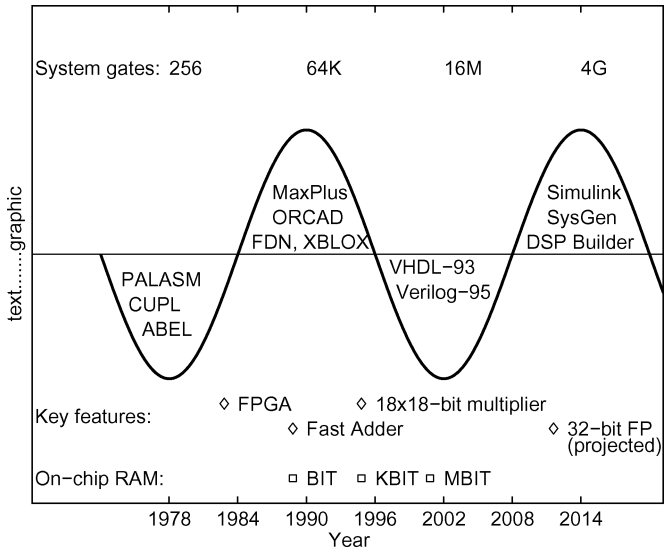

Fig. 1. FPGA design waves. (FP = Floating point).

II. Laboratory Software and Hardware Setup

Gordon Moore's Law, that the number of transistors on a chip quadruples every three years, has proven accurate for over five decades now. While this has had a monumental impact on the equipment used today, ranging from low-power portable devices to supercomputers, it has also influenced the design tools used to design integrated circuits. As a consequence, roughly every 12 years a paradigm change in the design tools of programmable logic devices (PLDs) has been seen, as shown in Fig. 1, which shows the design waves along with key features of PLDs/FPGAs.

In the past, PLDs had only a few gates and were used as "glue logic" and programmed by text-based tools such as PALASM, ABEL, or CUPL. As the devices grew larger, PLD vendors provided design libraries that contained larger logic blocks such as 16-bit adders. These libraries used typical graphic tools such as Altera's MaxPlus or Xilinx FDN or Xblox. One major problem with this graphic design was that a control unit typically designed as a finite state machine (FSM) was not feasible for graphic blocks, and a "black box" PAL assembler was instantiated in the graphic tool. Another major design method change, that of using hardware description language (HDL), such as VHDL or Verilog, then allowed the combination of a control and data path within one description file by using sequential as well as data-flow coding styles. Currently, with high-speed, multimillion FPGA system gates, the newest design wave is based on system-level tools based on MatLab/Simulink. The FPGA vendors provide library elements that allow both the simulation in MatLab/Simulink as well as a "push-button" approach to generating the bit stream used to program the FPGA. The most important reasons why this is a promising approach can be summarized as follows:

1) Without detailed knowledge of a HDL or FPGA devices, the system engineer (and, in particular, the 1 million MatLab users) can simulate the design precisely, and the conversion into the FPGA circuit is done automatically with the vendor tool.
2) In contrast to earlier system level tools like Xblox, the quality of the vendor libraries is excellent and can exceed hand-based HDL coding.
3) The tedious and error-prone task of specifying test benches can now be done much more easily in Simulink, therefore allowing higher design productivity.
4) Many high-end FPGA applications today are simulated in MatLab/Simulink first anyway, so conversion by hand into HDL is no longer necessary.

These claims are corroborated in FPGA vendor advertisements. It seems useful, therefore, to first look at some important properties of the tools and to also suggest workarounds for deficiencies. Since Altera and Xilinx both lead the FPGA market by a wide margin, the following discussion concentrates on their MatLab/Simulink interface called System Generator 7.1 and DSP Builder 5.0, respectively. A brief summary of an evaluation and possible solutions on how to avoid any shortcomings is described.

Altera and Xilinx libraries have 171 and 84 **library blocks** arranged in 11 and 16 subgroups, respectively. Altera has more blocks since it supports each development board with about 20 blocks. To verify that all major DSP designs are supported, the circa 50 design examples from a HDL DSP design book [20] were used. Only standard blocks, already in this library, needed to be used. In conclusion, it can be argued that both vendor libraries are equally suited for an undergraduate course.

Regarding the **library organization**, it was found that Altera's blocks are sometimes more difficult to locate due to the naming and organization in the subgroups. The "downsampling" block, for instance, is located in the storage group but not in the multirate subgroup. Also, the standard adder is called a "Parallel Adder," not simply an "adder," and the index group is also missing in Altera's DSPbuilder. The organization of the Xilinx library avoids these shortcomings and is therefore preferred. In the laboratory, these shortcomings are avoided by letting the students locate the basic DSP builder blocks (such as adder, multiplier, delay, etc.) in the very first lab.

In the category of **Simulink design support**, again the Xilinx tool offers more benefits to the students. One particular time-consuming aspect in designing the FPGA circuit with the potential of optimizing the size, speed, and/or power is the detail design at the bit-level. The Xilinx ports clearly show the data type and format, while the Altera nets always show double precision, and the specified bit-width within the block needs to be checked for each net. The control signal in the Xilinx System Generator can be turned on and off, which is one major advantage when compared with older tools like Xblox or the Altera tools, and therefore avoids the addition of a control signal connected to GND or VCC. Pipelining is shown in Xilinx blocks using the z-transform symbol, while in the Altera block the mask of the block needs to be checked. As a consequence, it can be said that the Xilinx library support is much better and saves students much design and debugging time. These shortcomings in the Altera software could only be avoided by writing very detailed lab instructions on each block and describing the internal mask values.

Using the Altera **design flow**, students can gain back some of the time they lost in the longer design and debugging time

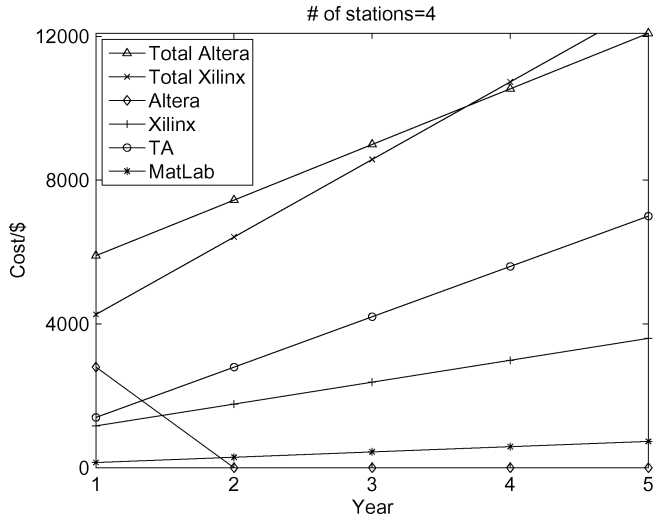

Fig. 2. Lab costs including scopes, MatLab, and TA costs at 14 weeks with 10 h/week and $10 per hour.

because the Altera tool controls the full circle of compilation from Simulink, including conversion into HDL, synthesis and fitter, and device programming. In Xilinx, it is necessary to use a separate tool for these steps: a pin description file (`.ucf`) needs to be added to the project before synthesis. Overall, the evaluation favors the Altera design flow. To lower the burden for the Xilinx tool, the first lab should include a compilation-ready design so that students can run a working example through the tool chain without having the uncertainty of a potentially incorrect design.

A student lab requires a **DSP board** so that students can experience a hands-on experiment and see that the simulation in the Simulink "Scope" can be reproduced with a conventional scope after synthesis and downloading of the bit stream to the board. At a minimum, such a board should include an FPGA with DSP features such as embedded multipliers and memory blocks, as well as peripheral components such as an A/D converter, a D/A converter, switches, and LEDs. Altera supports several boards and provides Simulink blocks to connect peripheral components. Xilinx has no direct board support, but very capable boards are available at a low price ($100) from Digilent, for example. However, the Simulink blocks provided by Altera simplify the design for the student, and Altera boards are therefore preferred. For the Nexys Digilent [21] board, the lab teaching assistant (TA) should provide macro blocks to translate the serial bit stream from A/D or D/A into a parallel bitstream.

The required **cost** of building and running a DSP with FPGAs lab can be prohibitive. Fortunately, the software tools in their student versions are currently free or available at a substantial discount. Donation of boards and software is also possible with limited quantities. Appropriate boards from Altera (Cyclone II) and Nexys (extended with A/D and D/A submodules) are available at $700 and $138, respectively. The break-even point for four workstations, including a TA, board, and software cost is at four years. Fig. 2 provides an overview of the overall cost, including a scope. By adding a portable scope, a complete portable lab environment can be built, and multiple workstations

TABLE I
AVERAGE VHDL ↔ SIMULINK COMPARISON FOR 17 DESIGNS FOR THE XILINX NEXYS BOARD AND ALTERA CYCLONE II DSP BOARD (Mul. = Embedded multipliers; LEs = Logic elements, i.e., 4-INPUT LUT)

Tool	Altera			Xilinx		
	LEs	Mul.	MHz	LEs	Mul.	MHz
VHDL average	112.9	0.7	240.7	86.3	0.7	130.0
Simulink average	99.2	1.3	284.2	63.9	1.3	149.0
Gain to VHDL in %	13.8	-50	18.1	34.9	-50	14.6

can share the boards and scopes. Both labs are about equivalent when it comes to the required costs.

Usually in a university lab, the concern is less with the **quality of results** than in the case of an industry setting where the overall product cost is in question. However, assuming that the aim is to prepare the students for careers in this field, such a flow should be taught only if the design flow gives competitive results when compared with HDL-based designs, and the flow is successful within the industry. Table I shows such a comparison for 17 different designs used in the lectures. The following designs from [20] have been used for comparison since they are also used in the lecture notes: Sine function generator: `fun_text`; 1-, 2-, and 3-stage pipelined adder: `add_1p`, `add_2p`, `add_3p`; serial/parallel multiplier: `mul_ser`; generic FIR filter: `fir_gen`; constant coefficient FIR filter using symmetry, CSD coding, and/or adder tree: `fir_srg`, `fir_sym`, `fir_csd`, `fir_tree`, `fir_csd_sym`, `fir_csd_sym_tree`; distributed arithmetic serial and parallel filter: `dasign`, `dapara`; IIR filter: `iir`; Daubechies polyphase filter: `db4poly`; complex multiplier: `ccmul`; and FFT butterfly processor: `bfproc`. As can be seen from Table I, both vendor tools outperform the HDL-based design in most cases.

In a laboratory setting, another concern is often the **compilation time**. The measured compilation time for both tools is very reasonable, with an average of less than a minute for each design.

In conclusion, the Simulink design flow for FPGAs is a useful alternative both for a university lab and for professional developers in the industry. Based on the evaluations, it was possible to avoid several shortcomings and to arrive at better options using Simulink.

III. LECTURE AND LABORATORY TOPIC SELECTION

After it had been determined that the Simulink design flow for FPGAs had become a viable design path, the next step was to develop lectures and laboratory materials that were closely coordinated so that topics covered in lectures were reinforced by hands-on laboratory experiments within, at most, a two-week time period. Using the HDL textbook on DSP with FPGAs [20], the appropriate topics for an undergraduate course and selected appropriate lab exercises were identified.

Eleven consultants were asked to evaluate appropriate topics, and the overall evaluation is shown in Fig. 3. Only topics with

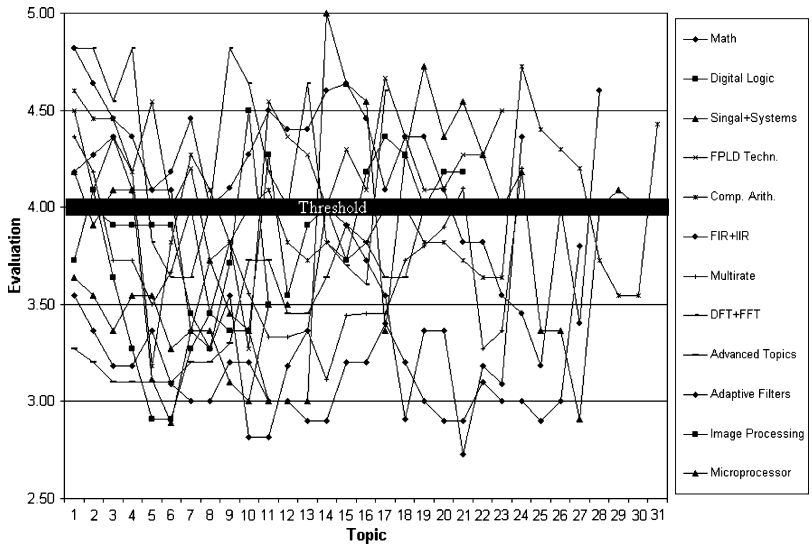

Fig. 3. Topic selection by 11 evaluators with threshold at 4.0.

TABLE II
DEVELOPED AND ADOPTED LECTURE NOTES, NUMBER OF CLASSES (25 LECTURES PLUS MLK HOLIDAY AND TWO TEST REVIEWS = 14 WEEKS), QUIZZES, AND AUDIO/VIDEO DEMOS. LISTED CHAPTER NUMBERS REFER TO [20]. THE SIGNAL+SYSTEMS REVIEW USES THE BOOK DSP FIRST [1]

Chapter + topic	# of classes	Altera # of slides	Xilinx # of slides	Other materials developed/adopted
Review logic	2	16	16	Getting started quiz; logic quiz;
Review Signal+Systems uses book [1] slides	3	–	–	From [1] Ch. 1-7,22,23,14,15 Signal+System quiz
Matrix+Matlab	1	11	14	MatLab quiz
1) Intro. DSP	1	21	21	Quantization + Sampling audio demo
1) Intro. FPGAs	2	20	21	FPLD quiz; Xilinx Video;
2) Computer Arithmetic	4	32	31	CSD quiz; DA quiz
3) FIR filter	4	36	36	3D FIR quiz; 2DSP first videos; RAG quiz; interactive pole/zero placement;
4) IIR filter	2	16	17	3D IIR quiz; 2 DSP first video; LP interactive design
6) DFT and FFTs	4	22	23	DFT quiz; FFT quiz;
5) Multirate systems	2	15	16	2 Channel quiz;
Total	25	237	243	

evaluation scores of 4 and higher were considered for the course. Based on this evaluation, the topics, quizzes, and laboratory experiments to be covered in class were selected. The lecture notes and quizzes developed and adopted are listed in Table II. A typical sequence of lectures modules is shown in Fig. 4. The thick arrow shows the sequence used in the Spring 2008 course: All tutorials, all core topics (Ch. 1–4 and 6), and, from the advanced topics, Ch. 5 on multirate systems was taught. In general, tutorials and other advanced topics can be skipped at the instructor's discretion. Typically, labs would begin with a prelab section, explaining and going through exercises on the theory, followed by the design lab, which usually used incomplete schemes that were to be completed during the supervised lab time. See Figs. 5 and 6 for a snapshot from the Web page for Lab 1 [22]. The labs are comprised of a key `lab1.mdl`, incomplete Simulink files `lab1inc.mdl`, and instructions (*.pdf). The Xilinx lab also contains the pin description file *.ucf. The following is a short summary of the topics covered in the eight developed laboratory experiments:

- **Lab 1**: Introduction to Simulink and DSP Builder (Objectives: to associate components with the library, to understand the Simulink/DSP builder design flow, and to design and simulate a circuit using Simulink).
- **Lab 2**: Number Systems and Quantization (Objectives: to understand the difference of signed and unsigned numbers systems, to determine minimum and maximum values in integer and factional number systems, to compute quantization error, and to design and simulate a circuit using Simulink).
- **Lab 3**: Introduction to Signal Flow Graphs (Objectives: to characterize systems by linearity, stability, causality, and time invariance; to understand the difference between FIR and IIR systems; and to design and simulate nonlinear, FIR, and IIR systems using Simulink).

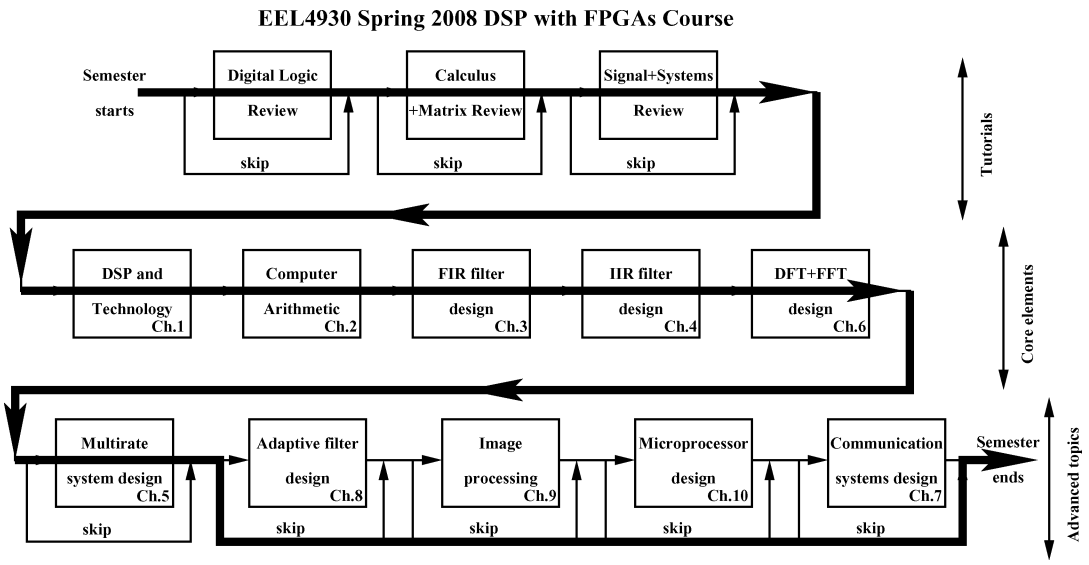

Fig. 4. Sequence of topics used for the Xilinx-based DSP with FPGAs course in Spring 2008.

- **Lab 4**: Introduction to MatLab M-File Scripts (Objectives: to write simple M-file scripts; to define tables and use predefined functions; to use the MatLab help, demo, and function library; to design and simulate complex multiplier systems using Simulink).
- **Lab 5**: Introduction to FIR Filters (Objectives: to design and simulate a moving average filter, to understand the difference between direct- and transpose-form FIR filters, to design and simulate a reduced adder graph FIR filter using Simulink).
- **Lab 6**: Introduction to IIR Filters (Objectives: to design and simulate a first-order IIR filter; to determine the magnitude, phase, and pole-zero diagram of IIR filters; to design a third-order elliptic low pass filter; to compare IIR and FIR design parameters).
- **Lab 7**: Introduction to Discrete Fourier Transform (Objectives: to develop a basic Goertzel IIR loop and process test data, to configure a subdesign with I/O ports, to instantiate a previously developed block, to design and simulate a selected DFT component).
- **Lab 8**: Introduction to Fast Fourier Transform (Objectives: to develop a radix-2 FFT and process test data, to understand the difference between DFT and FFT, to design and simulate an FFT using the principle of decimation-in-frequency using Simulink).

The labs were developed for the Altera Cyclone II DSP board and Xilinx Nexys boards provided by Digilent [21]. The Altera boards are more costly but include more features, including a larger FPGA and a high-speed A/D and D/A converter. The Nexys boards can be programmed and powered by USB cables only. Together with a Velleman [23] scope, a complete mobile lab environment can be built.

A. Web-Based Approach and Student Feedback

All eight Altera and eight Xilinx labs and supporting files are available online [22]. The lecture slides could not all be posted since they include copyrighted material; see Table II. Metadata

Fig. 5. Altera Lab 1 files from [22].

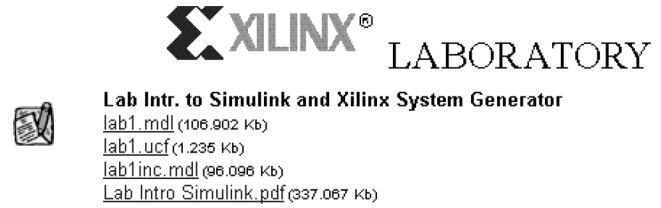

Fig. 6. Xilinx Lab 1 files from [22].

are used to assist the Web spider to more easily access extra key information included in the documents. The Dublin Core Metadata Initiative was used to identify the best tools. The DC-assist and DC-dot were used to generate HTML metadata text for items like the title, creator, subject, description, publisher, and contributor and to include these metadata into the HTML code.

Blackboard (BB) is the main software used to set up Web-supported and online classes at Florida State University (FSU), Tallahassee [24]. Using a Blackboard course Web site, instructors can deliver content and coordinate online activities. Blackboard is the most widely used and feature-rich online education software available. FSU participated in its development and continues to use it because the software is relatively easy to learn and manage. At the most basic level, the BB Web site is one more way to make course content available to students. Just as in an on-campus class, instructors are able to hand out class materials such as the course syllabus, assignments, the class calendar or schedule, written lectures, articles, and so on. Instructors can post class materials on the course Web site for students to view or print. The material posted remains available throughout the

TABLE III
EVALUATION REGARDING EDUCATIONAL MATERIAL USED IN FIG. 7

Fig. 7 no.	SPOT item	Question text
1)	B2	The course materials (e.g., textbook) helped me better understand the subject matter
2)	B3	The course assignments helped me better understand the subject matter
3)	D1	Description of course objectives and assignments
4)	D7	Facilitation of learning
5)	D8	Overall assessment of instructor

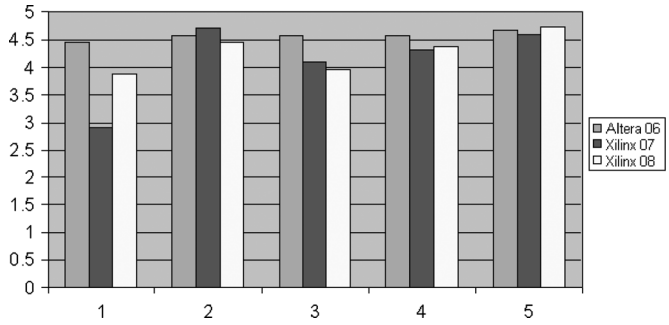

Fig. 7. Selected results for SPOT evaluation to the five questions listed in Table III from 16 students of the Altera (2006) course and the Xilinx prototype course from 2007 with 21 students and 2008 with 14 students.

semester (or until removed by the instructor), eliminating the expense and labor of making copies for each student.

Evaluation of the course by enrolled students and by outside evaluators both from academia as well as industry is performed on a continuous basis. During the course, Blackboard enables feedback; for instance, a threaded discussion board, student homepages, a digital drop box, group pages, and class statistics [24] were often used.

At the end of the semester, students evaluate the course in full detail using FSU's "Student Perception of Teaching" (SPOT) system. The SPOT has six more sections than the simple "State University System Student Assessment of Instruction" (SUSSAI) forms, with a total of 30 questions ranging from (A) Demographics, (B) Course & Instructor Details, (C) Overall Course & Instructor Assessment, (D) SUSSAI (E) Additional Questions, to (F) Additional Questions, Instructor. Students have a five-level response scale ranging from strongly agree/excellent to strongly disagree/poor. Many of the questions relate to the style, performance, and skills of the instructor, but some of the questions provide feedback about the quality of the educational material in use. Related questions are shown in Table III.

Student evaluations are conducted at the end of each semester. The collected evidence is then used to improve the material. Fig. 7 shows the results related to the five educational material development (EMD)-related questions in the course. As can be seen from the overall evaluation, the students were very satisfied with the overall material and the learning progress that was provided. However, the greatest weakness was the textbook material, the only SPOT evaluation that scored, on average, lower than 4.0. The development of a professional textbook description should be one of the main concerns and highest aims for the next phase of the project. The improvement of SPOT feedback B2 will be a major indicator of the progress that is made in the next phase from the students' perspective.

TABLE IV
EMD EVALUATION BY THREE ALTERA AND FOUR XILINX EXPERTS FROM INDUSTRY (3) AND ACADEMIA (4). (Grading 5 = full coverage ... 1 = all missing)

Topic	Altera	Xilinx	Total
Logic Review	4.7	4.3	4.5
Matrix + Calculus	4.3	4.1	4.2
Signal + Systems	4.8	4.0	4.4
Overview DSP	4.5	4.9	4.7
FPGA Technology	4.0	4.5	4.3
Computer Arithmetic	4.5	4.5	4.5
FIR filter	4.7	4.5	4.6
IIR filter	4.3	4.0	4.2
Fourier Transform	4.7	5.0	4.8
Multirate DSP	4.5	4.0	4.3
Laboratory	5.0	4.8	4.9
Syllabus	5.0	4.3	4.6
Mid term	4.7	5.0	4.8
Final Exam	4.7	5.0	4.8
BB Webpage	5.0	5.0	5.0
Overall	4.62	4.52	4.57

B. External Expert Reviewer Feedback

One of the greatest benefits of the NSF-sponsored CCLI project is the possibility of having the experience of paid consultants to improve the EMD. Input from four consultants from the industry as well as four from academia was sought. All consultants had substantial experience in the DSP with FPGA field.

After the EMD was completed, the consultants were asked whether or not they thought the material was complete or if there were any improvements that possibly needed to be made. Since the instructors themselves may be considered biased toward the evaluation of their own work, only the outside reviewers' feedback is reported in Table IV. For the Altera EMD and the Xilinx EMD, three and four feedback responses, respectively, were received. Overall, the evaluations were very encouraging.

IV. SUMMARY AND EXTENSIONS

The findings regarding a DSP with FPGAs Simulink-based course and lab can be summarized as follows.

- The Simulink design flow of DSP systems for Xilinx and Altera FPGAs has become a viable design path. The Simulink design flow for FPGAs is an interesting and viable alternative for introducing students at an early level to DSP hardware design concepts.
- Both vendors provide a sophisticated toolbox of library elements. The quality of results (QOR) is excellent, and suitable boards for teaching the lab are available.
- The strength of the tools and alternative paths provided to avoid their weaknesses are described.

- No HDL knowledge is necessary to develop sophisticated DSP designs such as the designs used in the eight lab experiments, i.e., RAG-based FIR filters, elliptic IIR filters, low-complexity Goertzel DFTs, or fully pipelined FFTs. These labs give the DSP student much more experience than do the simple "DSP first" [1] book topics.
- Teaching an undergraduate course without a suitable text book is challenging, though well-structured quizzes, homework assignments, and hands-on labs can still provide good learning success. Feedback through SPOT and blackboard shows that a textbook would be a welcome addition and would help the EMD.
- Eight Altera and eight Xilinx labs, including all descriptions and supporting Simulink files, are posted on the Web [22].

Improvements, such as adding textbook-quality descriptions, public domain videos in the style of DSP First, capstone design projects, e-learning training materials, and professor workshops, will be considered in the future.

ACKNOWLEDGMENT

Phase 1 of the CCLI project "Digital Signal Processing with Field Programmable Gate Arrays: An Undergraduate Prototype Course with Companion Lab" was funded by NSF (ID 0442600) from June 2005 to May 2008. Additional results can be found in three SPIE papers [25]–[27] totaling 33 pages, available online [22]. The help of the following consultants and program manager is acknowledged: B. Esposito, M. Phipps, R. Maroccia (Altera), J. Weintraub, A. Acevedo, C. Sepulveda, C. Dick (Xilinx), P. Athanas (Virgina Tech), H. Hamel (Raytheon), W. Moreno (USF), S. Connors (Boeing-SVS), J. Hallman (Harris Corp.), E. Manolakos (Northeastern University), E. Zurek (USF). Thanks goes to the following research assistants: A. Hoover and E. Kim for the Altera laboratory development, and A. Canosa, D. Murphy, and S. Canosa for the Xilinx laboratory development. Many thanks to C. Neff for the English editing of this article. The authors thank the anonymous reviewer for helpful comments. Products and company names used in this article may be trademarks of their respective owners. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] J. McClellan, R. W. Schafer, and M. Yoder, *Signal Processing First*. Upper Saddle River, NJ: Prentice-Hall, 2003.
[2] R. Lyons, *Understanding Digital Signal Processing*. Upper Saddle River, NJ: Prentice-Hall, 2001.
[3] A. Oppenheim, A. Willsky, and S. Hamid, *Signals and Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.
[4] S. D. Stearns and D. R. Hush, *Digital Signal Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[5] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[6] H. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. Heidelberg, Germany: Springer, 1990.
[7] F. Taylor, *Digital Filter Design Handbook*. New York: Marcel Dekker, 1983.
[8] C. Chen, *Digital Signal Processing*. New York: Oxford Univ. Press, 2001.
[9] L. Tan, *Digital Signal Processing*. Amsterdam, The Netherlands: Elsevier, Academic Press, 2008.
[10] S. Mitra, *Digital Signal Processing: A Computer-Based Approach*, 3rd ed. Boston, MA: McGraw-Hill, 2006.
[11] A. Spanias and V. Atti, "Interactive online undergraduate laboratories using J-DSP," *IEEE Trans. Educ.*, vol. 48, no. 4, pp. 735–749, Nov. 2005.
[12] U. Rajashekar, G. C. Panayi, F. P. , and A. C. Bovik, "The siva demonstration gallery for signal, image, and video processing education," *IEEE Trans. Educ.*, vol. 45, no. 4, pp. 323–335, Nov. 2002.
[13] I. Koren, *Computer Arithmetic Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[14] K. Hwang, *Computer Arithmetic: Principles, Architecture and Design*. New York: Wiley, 1979.
[15] L. DeBrunner and V. DeBrunner, "The case for teaching DSP algorithms in conjunction with implementations," in *Proc. IEEE 2nd Signal Process. Educ. Workshop*, Oct. 2002, pp. 195–200.
[16] W.-S. Gan and M. Kuo, "Teaching DSP software development: From design to fixed-point implementations," *IEEE Trans. Educ.*, vol. 49, no. 1, pp. 122–131, Feb. 2006.
[17] T. Sansaloni, A. Perez-Pascual, V. Torres, V. Almenar, J. Toledo, and J. Valls, "FFT spectrum analyzer project for teaching digital signal processing with FPGA devices," *IEEE Trans. Educ.*, vol. 50, no. 3, pp. 229–235, Aug. 2007.
[18] F. Ahamed and F. Scarpino, "An educational digital communications project using FPGAs to implement a BPSK detector," *IEEE Trans. Educ.*, vol. 48, no. 1, pp. 191–197, Feb. 2005.
[19] T. Hall and D. Anderson, "A framework for teaching real-time digital signal processing with field-programmable gate arrays," *IEEE Trans. Educ.*, vol. 48, no. 3, pp. 551–558, Aug. 2005.
[20] U. Meyer-Baese, *Digital Signal Processing With Field Programmable Gate Arrays*, 2nd ed. Berlin, Germany: Springer-Verlag, 2004.
[21] Nexys board, Digilent Inc., Dec. 2009 [Online]. Available: http://www.digilentinc.com/Products/Detail.cfm?Prod=NEXYS
[22] U. Meyer-Bäse, "Online CCLI project page posting," Dec. 2009 [Online]. Available: http://www.eng.fsu.edu/~umb/
[23] HPS40, Velleman Inc., Dec. 2009 [Online]. Available: http://www.vellemanusa.com/us/enu/product/view/?id=351409
[24] *"6 Manual,"* Blackboard, Dec. 2009 [Online]. Available: http://library.blackboard.com/docs/r6/6_1/instructor/bbls_r6_1_instructor/
[25] U. Meyer-Baese, A. Vera, A. Meyer-Baese, M. Pattichis, and R. Perry, "Discrete wavelet transform FPGA design using MatLab/Simulink," in *Proc. SPIE Int. Soc. Opt. Eng.*, Apr. 2006, pp. 624 703-1–624 703-10.
[26] U. Meyer-Baese, A. Vera, A. Meyer-Baese, M. Pattichis, and R. Perry, "Smart altera firmware for DSP with FPGAs," in *Proc. SPIE Int. Soc. Opt. Eng.*, Apr. 2007, pp. 65 760T1–65 760T12.
[27] U. Meyer-Baese, A. Vera, A. Meyer-Baese, M. Pattichis, and R. Perry, "DSP with FPGAs: A XILINX/Simulink-based course and laboratory," in *Proc. SPIE Int. Soc. Opt. Eng.*, Mar. 2008, pp. 697 907-1–697 907-12.

Uwe Meyer-Bäse (S'93–M'95) received the M.S.E.E. and Ph.D. (summa cum laude) degrees from the Darmstadt University of Technology, Darmstadt, Germany, in 1989 and 1995, respectively.

He is currently a Professor with the Electrical and Computer Engineering Department, Florida State University, Tallahassee. In 1994 and 1995, he held a Postdoctoral Position in the "Institute of Brain Research," Magdeburg, Germany. In 1996 and 1997, he was a Visiting Professor at the University of Florida, Gainesville. From 1998 to 2000, he worked as a Research Scientist in the ASIC industry, where he was responsible for development of high-performance architectures for digital signal processing. He holds three patents, has published over 80 journal and conference papers, and supervised more than 60 Master's thesis projects in the DSP/FPGA area. He is author of the best-selling textbook *Digital Signal Processing With Field Programmable Gate Arrays* (Berlin, Germany: Springer-Verlag, 2004).

Dr. Meyer-Bäse is a Member of the IEEE Biomedical Engineering, Signal Processing, and Circuits and Systems societies. He was a recipient of the "Habilitation" (venia legendi) from the Darmstadt University of Technology in 2003, the Max-Kade Award in Neuroengineering in 1997, the Humboldt Research Award in 2006, and FAMU-FSU College of Engineering Teaching Award 2007.

Alonzo Vera (M'96) received the B.S degree in electrical engineering from the Pontificia Universidad Catolica del Peru, Lima, Peru, in 1998, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of New Mexico (UNM), Albuquerque, in 2004 and 2008, respectively.

He is currently a Research Engineer with Micro-RDC, Albuquerque, NM, and a Research Assistant Professor with UNM. His research interest include reconfigurable computing, digital signal processing, and spaceborne digital computer systems.

Anke Meyer-Bäse (M'94) received the M.S.E.E. and Ph.D. degrees from the Darmstadt University of Technology, Darmstadt, Germany, in 1990 and 1995, respectively.

She is an Associate Professor with the Department of Scientific Computing, Florida State University, Tallahassee. She has published over 150 papers in several areas ranging from intelligent systems, medical image processing, computational biology, and neural networks. She is author of the book *Pattern Recognition in Medical Imaging* (Amsterdam, The Netherlands: Elsevier/Academic Press, 2003) and the forthcoming book *Biomedical Signal Processing: Contemporary Methods and Applications* (Cambridge, MA: MIT Press, 2010). Her research areas include theory and applications of neural networks, medical image processing, pattern recognition, and parallel processing.

Marios S. Pattichis (M'99–SM'06) received the B.Sc. (high honors and special honors) degree in computer sciences and the B.A. (high honors) degree in mathematics, the M.S. degree in electrical engineering, and the Ph.D. in computer engineering from the University of Texas at Austin in 1991, 1993, and 1998, respectively.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering (ECE), University of New Mexico (UNM), Albuquerque, where he is also an Associate Professor with the Department of Radiology. He has authored or coauthored more than 20 journal papers, 15 book chapters, and over a hundred conference papers. His current research interests include digital image, video processing and communications, dynamically reconfigurable computer architectures, and biomedical and space image processing applications.

Dr. Pattichis received the 2004 ECE Distinguished Teaching Award and the 2006 School of Engineering Harrison Faculty Recognition Award at UNM. He is an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and an Associate Editor of *Pattern Recognition*. He is a member of the Editorial Board of the *International Journal of Experimental and Computational Biomechanics*. He was the General Chair of the 2008 IEEE Southwest Symposium on Image Analysis and Interpretation.

Reginald J. Perry (M'85–SM'99) received the B.E.E. (CO-OP, Highest Honors), M.S.E.E, and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 1982, 1983, and 1989, respectively.

He worked for IBM from 1983 to 1986 in the area of high-performance CMOS SRAM development. Since 1989, he has been a Professor of electrical and computer engineering at Florida A&M University, working at the joint Florida A&M University-Florida State University College of Engineering, Tallahassee. He served as ECE Department Chair between 1999 to 2004. He has served as the Engineering Associate Dean for Student Affairs and Curriculum since 2004. He is also the Project Director and a Co-Principal Investigator of the NSF-funded STEM Learning Community at Florida A&M University project.