This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Project-Based Learning in Embedded Systems Education Using an FPGA Platform

Akash Kumar, *Member, IEEE*, Shakith Fernando, and Rajesh C Panicker, *Member, IEEE*

***Abstract*—With embedded systems becoming ubiquitous, there is a growing need to teach and train engineers to be well-versed in their design and development. The multidisciplinary nature of such systems makes it challenging to give students exposure to and experience in all their facets. This paper proposes a generic architecture, containing multiple processors, that allows easy integration of custom and/or predefined peripherals. The architecture allows students to explore both the hardware and software issues associated with real-time and embedded systems. Furthermore, the architecture can be extended to train students in advanced concepts in embedded multiprocessor systems. This generic architecture has been used for two courses at the National University of Singapore—one on real-time embedded systems and the other emphasizing the hardware aspects of embedded systems. The project in the real-time embedded systems course has students develop a five-a-side soccer system on multiple field-programmable gate array (FPGA) boards using embedded processors. In the embedded hardware design course project, students use an embedded processor-based system to perform decryption of a block encrypted image, accelerated through a custom co-processor. The use of displays gives students a visual/interactive experience and a sense of accomplishment, while reinforcing the theoretical concepts. Both qualitative and quantitative assessment results are presented, showing how students perceived these projects and met the learning objectives.**

***Index Terms*—Embedded systems, field-programmable gate array (FPGA), hardware-software co-design, project-based learning, real-time systems.**

I. Introduction

MODERN embedded computing platforms are fast becoming more heterogeneous and therefore more complex to design and program. Multicore-based mobile phones (e.g., Tegra) and hybrid platforms with both hard multicore processors and reconfigurable area (e.g., Xilinx Zynq) are examples of such systems in the consumer electronics domain. The market for embedded systems is ever increasing, with the consumer electronics domain expected to expand by 41% in volume and other domains showing similar growth, with a forecast of 35% in automotive and 37% in telecommunications. This means that the current 16 billion devices (roughly three embedded devices for every living person) are expected to grow to 40 billion by 2020 [1]. Driven by shorter time-to-market demands, commercially available off-the-shelf (COTS) components are used more and more in the hope of reducing the overall system development time and cost [2]. Using COTS components can often increase system-integration work and dependency on third-party vendors. Hence, a good embedded curriculum should not only teach and train engineers to design such complex and heterogeneous embedded computing systems, but also teach them industrially relevant design challenges in using COTS components.

Teaching *embedded systems* as an integrated topic is a difficult task since it can be very diverse and multidisciplinary [3]–[6], ranging from micro-controller basics and real-time concepts to hardware/software co-design, distributed processing, reconfigurable computing, and system-level architecture design [7]. The projects in such courses must span multiple learning objectives; to motivate students to acquire embedded system design skills, these must also be objectives to which the students can relate. Furthermore, these projects must simulate industrial relevance by including component-integration with COTS components; the time required for the necessary learning curve must therefore be built into the course structure.

The main contribution of this paper is a generic architecture for design projects using COTS hardware and IP with lab assignments, whose purpose is to accelerate the student learning curve in this area. This generic architecture is built upon multiprocessor systems allowing easy integration of custom and/or predefined peripherals. The flexible architecture allows exploration of both hardware and software issues pertaining to real-time and embedded systems. Furthermore, hardware/software co-design-based projects can also be built upon this architecture. This generic architecture for use in project-based learning has already been used at National University of Singapore (NUS), Singapore, in two embedded systems courses: a real-time embedded course (EE4214) and an embedded hardware systems design course (EE4218). Furthermore, the architecture can be extended to train students and engineers in advanced concepts in embedded systems such as networks-on-chip, mapping, and scheduling of tasks onto multiprocessor architectures. This paper extends previously published work on the real-time embedded course (EE4214) [8] that emphasized a single project and proposed a fixed architecture limited to software-based systems and not applicable to hardware-intensive projects. In the current paper, the proposed generic architecture is also extended to the embedded hardware

Manuscript received August 01, 2012; accepted October 11, 2012. This work was supported by the Xilinx University Program.

A. Kumar is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583, Singapore (e-mail: akash@nus.edu.sg).

S. Fernando is with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven 5612 AZ, The Netherlands (e-mail: s.fernando@tue.nl).

R. C. Panicker is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: rajesh@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2013.2246568

0018-9359/$31.00 © 2013 IEEE

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

systems design course (EE4218). Detailed qualitative and quantitative survey results are presented to evaluate students' perception of these projects.

Students receive four modular credits for each course, which is equivalent to approximately 130 h of work over one 4-month semester. The courses are offered as electives for final-year undergraduate students and are attended by about 80 students each year. The students are expected to be familiar with the basics of computer architecture and to be comfortable with C/C++ and VHDL (for EE4218). Concepts are taught through a series of lectures, tutorials, lab exercises, and a project. The labs and project form a very important part of the course, determining 50% (EE4214) or 60% (EE4218) of the final student grade. The labs are intended to help students appreciate the theory taught in the lectures and gain the knowledge and experience required for the project.

The real-time embedded course includes a major design project that is carried out on a popular COTS hardware (Xilinx Spartan 3E board from Digilent [9]). The aim of the project is to design a system for five-a-side soccer. The system comprises: 1) a client strategy controller, and 2) a server to referee and display the game in real time. At the end of the semester, all teams compete against each other to determine the winner. The competition element motivates the students and brings out the best in them.

The aim of the embedded hardware design project is to design a hardware accelerator for compute-intensive tasks of two encryption algorithms—Advanced Encryption Standard (AES) and PRESENT [10], [11]. The students are also required to explore the design space of these accelerators and analyze the performance and area tradeoffs. The encoded and decoded images are displayed on a monitor to give a visual and interactive experience.

The distinguishing features of both projects are that they:

- have a good balance of breadth (real-time concepts, multiprocessor architectures, and FPGA exposure) and depth (theory and implementation of real-time and hardware design concepts);
- give an opportunity for students to get hands-on experience in using real hardware;
- provide lab assignments on the generic architecture to accelerate the student learning curve in mastering the complicated tool-chain and other aspects necessary for the projects;
- expose students to the analysis of the design space tradeoffs, given hardware area constraints in real industrial COTS hardware;
- provide a visual and interactive design experience;
- include a fun and competitive element to motivate students and enhance the learning experience;
- encourage students to share ideas, work in teams, and manage time and resources effectively.

Furthermore, from the first week of the course, weekly consultations with the teaching assistants encourage student discussion of the projects. This not only helps them with problems related to tools and hardware, but also helps ensure that they are on track to complete the projects on time. Two Wiki pages set up for project management and dissemination of information [12], [13] are very useful in making available resources such as datasheets, detailed project specifications, and updates. The Wiki also allows students to share know-how in solving technical problems related to hardware and electronic design automation (EDA) tools.

This paper is organized as follows. Section II summarizes the related work on embedded systems education. Section III gives a brief overview of the generic embedded architecture used for both courses at NUS. Sections IV and V describe the projects and the lab assignments given to students in EE4214 and EE4218, respectively. Section VI describes the evaluation criteria, highlights some innovative projects, and summarizes student feedback. Section VII concludes the paper with a discussion on the insights gained from these projects.

II. Related Work

One of the earliest papers to propose a multidisciplinary approach for the analysis and design of complete embedded systems was by Wolf *et al.* [14]. Until then, teaching in this area was largely ignored by academics because it had not thrown up sufficient challenges [15]. Recently, there has been a growing interest in discussing various teaching methodologies for embedded systems education, as shown by the number of papers published in the IEEE TRANSACTIONS ON EDUCATION in this area [3], [16]–[19]. Furthermore, to address the challenges in this area, publications devoted to embedded systems education have appeared, such as special issues on embedded systems education in the *ACM Transactions on Embedded Computing Systems* [4], [5] and the Workshop on Embedded Systems Education (WESE) [6], [7], [20], [21]. The industry has also organized fun embedded design competitions for students to promote embedded computing, e.g., the embedded design track at the Microsoft Imagine Cup [22], the Intel Undergraduate Embedded Design Contest [23], and the Xilinx OpenHW Contest [24].

A. Projects Teaching Real-Time Embedded Systems Design

Design projects used to teach real-time embedded systems are compared here; a comparison of those for embedded hardware design follows in Section II-B. For real-time embedded systems, the comparison examines the kind of problems tackled in design projects, their real-time constraints, whether a uniprocessor or a multiprocessor system is used, and whether a field-programmable gate array (FPGA) is used as a teaching tool. The features of various design projects are summarized in Table I and explained in what follows. A railroad control system real-time project, described by McCormick *et al.* [25], is a very good example of a problem with real-time constraints. The use of model trains provides a fun aspect for students. Mehdi *et al.* [26] also propose a soccer system. However, their game play is run on a PC, while the projects described here require students to implement the system on a real embedded platform. Neither project focuses on the use of multiprocessors or FPGAs. Hansson *et al.* [21] propose a design project where students partition and map the JPEG decoder on to a multiprocessor platform running on an FPGA. It emphasizes the diverse aspects of designing embedded systems with multiprocessors, FPGA, and network-on-chip communications. However, their

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
COMPARISON OF FEATURES IN DESIGN PROJECTS TO TEACH EMBEDDED SYSTEMS

	Contains a real life problem with real-time constraints?	Embedded Hardware	Use of Multi-processor	Use of FPGA as a teaching tool
McCormick *et al.* [25]	Yes	No	NA	NA
Mehdi *et al.* [26]	Yes	No	NA	NA
Hansson *et al.* [21]	Maybe	Yes	Yes	Yes
Ttofis *et al.* [16]	No	Yes	Yes	Yes
Edwards *et al.* [20]	Dependent on the proposal	Yes	NA	Yes
Previous NUS [27]	Dependent on the proposal	Yes	No	No
This design project	Yes	Yes	Yes	Yes

focus is more on hardware/software co-design, and the underlying architecture is static. In contrast, the projects described in this paper allow students to generate their custom hardware design.

Edwards *et al.* [20] share their teaching experience on embedded systems using an FPGA as a teaching platform. This is a practical course where students are allowed to define their own project and implement it on hardware. Although the course does not necessarily require real-time systems, quite a few of the projects have real-time requirements (e.g., a real-time video effects processor). Another similar project, proposed by Ttofis *et al.* [16], uses a multiprocessor-based network-on-chip platform. They propose an FPGA-based teaching framework on this platform using several benchmark applications. While this also encompasses various aspects of designing embedded systems with multiprocessors, FPGAs, and network-on-chip communications, the projects described here simulate more industry-relevant scenarios by targeting a real application with real-time requirements.

For the previous real-time embedded systems design project at NUS [27], students defined a real-time application and implemented it on an embedded platform: a Motorola uCSimm module with an MC68EZ328 integrated processor and RTAI uClinux operating system. Due to the complexity of the platform and limited documentation, the students spent considerable amounts of project time debugging Linux and hardware issues instead of learning and implementing real-time concepts. In the design projects described here, a real-time soccer controller system is used as an application. This is a good example of a problem with real-time constraints to be met. The proposed hardware consists of multiple FPGA development boards, each configured as a multiprocessor system-on-chip.

B. Projects Teaching Embedded Hardware Design

Related work on embedded hardware design projects with a hardware accelerator as a co-processor is quite limited. Schaumont *et al.* propose a hardware-software co-design course to explore the design space of hardware accelerators for encryption. However, all custom hardware and software tradeoffs are only simulated [17], whereas in the projects described here, the accelerators are implemented on real hardware. Mitsui *et al.* [28] propose a design project for a DCT co-processor for JPEG application on an Altera NIOS2 development kit. The students are also expected to explore the design space of the hardware accelerator for the DCT. While they are similar to the projects described here, the goal of the NUS courses is to design the whole system for the complete application together with the co-processor design for encryption.

Hall *et al.* [18] describe a design project using system-on-a-programmable-chip for their student project. The students are expected to implement an application of their choice (e.g., a remotely controlled vehicle, robot control) on a system-on-a-programmable-chip consisting of a soft processor and custom hardware logic. A similar system-on-chip project is described by Bindal *et al.* [19], where a servo controller for a robotic arm is implemented on an Excalibur chip with an embedded ARM processor and programmable logic. In the proposed design projects, the computational-intensive components of two encryption algorithms are implemented as a co-processor to the main processor. Students are expected to explore multiple design space points (one software and several hardware implementations) and compare their tradeoffs. They must also implement this on real hardware and show the encoded and decoded images on the VGA display. This allows for a visually stimulating and interactive learning experience for the students.

III. GENERIC HARDWARE ARCHITECTURE

The NUS projects are carried out on a Xilinx Spartan-3E FPGA board from Digilent [9]. The board features a Xilinx Spartan-3 1600 FPGA with about 1.6 million reprogrammable gates and also offers a number of input/output options to interface with other peripherals, including two serial terminals and a VGA port. The serial port is used in the projects to communicate with other FPGA boards and/or PCs, while the VGA port is used to show output on an external screen. The generic architecture of FPGA configuration is shown in Fig. 1. The architecture consists of two Microblazes (MBs), a soft-processor from Xilinx, connected via a processor local bus (PLB). Using Microblaze allows the system behavior to be defined using C/C++ language. Furthermore, various enhancements to the processor architecture can be made through the Xilinx Embedded Development Kit/Xilinx Platform Studio (EDK/XPS) tool-suite, that is, the hardware multiplier and floating-point-unit. The EDK/XPS environment also allows easy customization of the system by adding/configuring various hardware IPs (intellectual property modules) provided in the toolkit.

For the two course projects, some relevant IPs, other than Microblazes, are the following.

- PLB: This bus allows various peripherals to be instantiated and connected to it as slaves. Microblazes, as masters, are able to communicate with these peripherals arbitrated by PLB.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

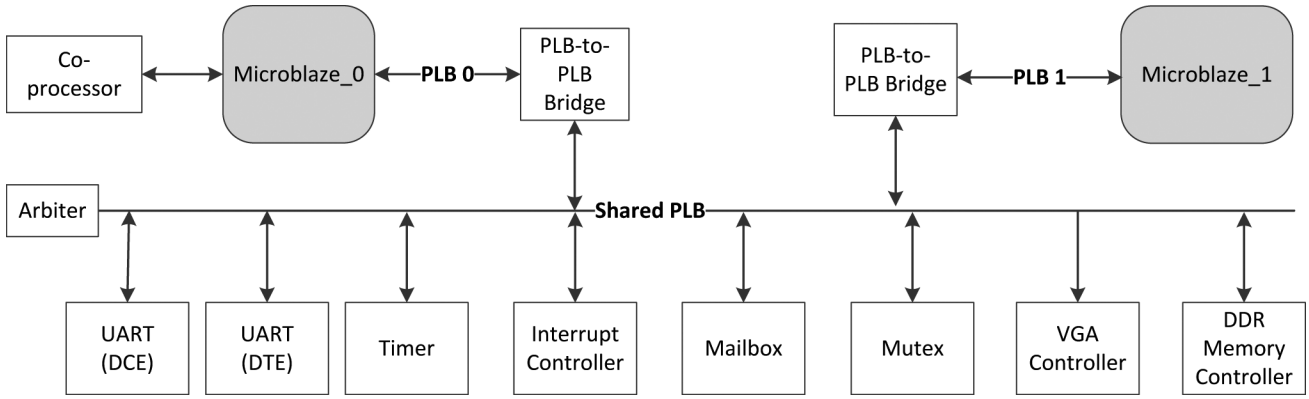

Fig. 1. Block diagram of the generic architecture for embedded systems projects.

- Timer: This IP allows Microblazes to keep track of time elapsed while executing various software/hardware routines. Furthermore, it is useful as a system-timer for operating system kernels.
- VGA controller: This controller delivers video frames to the VGA port from a video buffer. The on-board hardware supports maximum resolution of up to 640×480 pixels with 3-bit color depth at 25 Hz. This is taken into account while designing the project specifications.
- DDR SDRAM controller: The memory forms a very important part of the system, and the controller is used to arbitrate access to the DDR memory. The memory is used for storing the video buffer and program code for the processors and serves as data memory.
- UART controller: Serial ports are useful for both inter-board and PC-board communication.
- Co-processor: The use of custom co-processor allows hardware acceleration of some of the compute-intensive parts of software.
- Mutex and Mailbox: These are used for data communication between the Microblazes.
- Interrupt controller: This module is used to interrupt the processor to ensure real-time behavior of the software.

The generic architecture used in the labs allows a variety of projects to be developed that expose students to various issues in embedded and real-time systems while still stimulating the students with immediate visual output of their hard work. This platform allows both pure software and hardware-accelerated projects to be developed.

IV. EE4214 Real-Time Embedded Systems

The *EE4214* course starts with an overview of the importance of making embedded systems real-time, and an introduction to real-time systems concepts such as scheduling and handling shared resources. This is followed by an overview of the design methodology for real-time software. Other in-depth technical topics such as concurrent programming, deadlock management and synchronization mechanisms, as well as other aspects of an embedded computer system that affect real-time performance, are discussed. At the end of the course, students are expected to:

- be familiar with design methodologies for real-time embedded systems;
- understand the importance of analyzing timing behavior in embedded systems;
- understand the many factors affecting real-time performance in embedded systems;
- be able to apply these concepts to design embedded systems with real-time performance.

The objective of the project, inspired by the Soccer World Cup, is to develop a five-a-side soccer system using multiple FPGA boards and is carried out in groups of up to six students. The groups have to design the hardware architecture of the embedded system and the software for the strategy to control how to move the players in response to the position of all players and the ball. They also develop a server to communicate with the two teams and display the progress of the game on an attached VGA monitor.

A. EE4214 Lab Details

Each lab consists of two parts: 1) a tutorial where step-by-step guidance is given to implement real-time concepts on multiprocessor systems; 2) an assignment in which students apply the practical knowledge gained from the first part to solve a fairly simple design and implementation problem. A total of six lab assignments are provided to students as shown in Table II. Lab-1 introduces students to Xilinx tools by implementing a Microblaze-based system on a Spartan-3E board. In Lab-2, students learn to implement a dual-processor system with additional hardware such as mutexes and mailboxes, shown in Fig. 1. Students also learn to use a real-time OS called *xilkernel* and to implement round-robin and priority-based schedulers with multiple threads. Lab-3 allows students to experiment with both hardware and software mutexes and appreciate their relevance. In Lab-4 and Lab-5, students learn to implement interprocess communication and semaphores, respectively. In Lab-6, students implement solutions to problems caused by priority inversion.

B. EE4214 Project Details

The project, set up as shown in Fig. 2, requires three FPGA boards—one for the server that also referees the game, and two to run heuristics from each team. The teams send updates of player movements to the server periodically. The server board processes the updates from both teams and displays the players and the ball on the attached VGA screen. The positions of the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
LIST OF LAB ASSIGNMENTS FOR EE4214

Lab	Topic	Relation to Project
1	Familiarization with FPGA and EDK	Using VGA screen for the soccer game display.
2	Threads	Using different scheduling algorithms to control player tasks.
3	Software and Hardware Mutexes	Using shared resources like the serial communication channel between two boards in the project setup.
4	Message Queues and Mailboxes	Learning inter-process communication to pass data between threads within a core and between two cores.
5	Binary and Counting Semaphores	Controlling the number of active player threads on the soccer field.
6	Priority Inheritance Protocol/ Priority Ceiling Protocol	Implementation of dynamic priority for cases when the priority of a thread may need to be increased to limit the blocking time of higher priority threads.

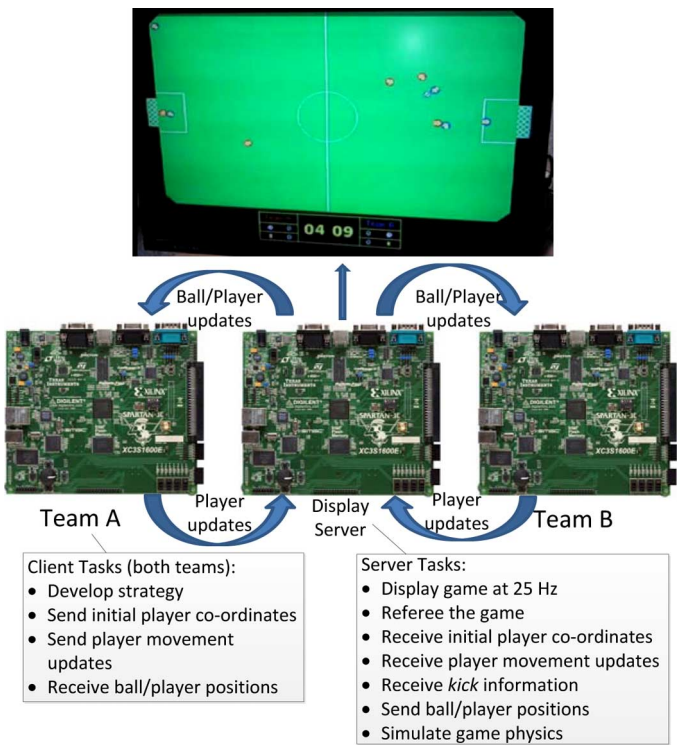

Fig. 2. Soccer project setup, with multiple FPGA boards.

ball and of both teams' players are sent back to them. Many added constraints simulate real-life scenarios, such as how fast the players can to run and the maximum speed of the ball; these need to be respected in the students' designs.

Fig. 2 also shows the tasks to be carried out by clients and the server. One of the main tasks is to send the player movement information to the server from the client. It is important to note that the client does not send actual positions of the players during the game, as this may result in some unrealistic movements; for instance, the client may instantly move a player from the center of the field to near the goal. The client, therefore, only sends the direction in which the player intends to move and the desired speed of that movement. The server, on the other hand, does send the absolute positions of the players back to the teams; the players' speed and direction are not sent by the server since that may reveal the other team's strategy. The server only sends the information that is generally available to the other team in a real game. Another major task of the server is to simulate the physics of the game. Physical laws, such as the conservation of momentum, have to be respected by the server when a collision occurs. When kick information is received by the server, it evaluates the distance between the ball and the player and only executes the kick if the two are sufficiently close. The server also referees the game. While it is not feasible to implement all of the rules of soccer in a virtual world with limited resources, some of them, such as the off-side rule, can be easily implemented.

Since each team individually develops their client and server designs, it is important to have a well-specified communication protocol to ensure that they can seamlessly communicate with the other teams. Such a well-defined protocol allows *plug-n-play* behavior where clients from any two teams can be taken with the server from a third one.

V. EE4218 EMBEDDED HARDWARE SYSTEMS DESIGN

The *EE4218* course starts with a description of the general embedded systems design methodology. Various implementation technologies (e.g., ASIC, FPGA) are introduced, and their tradeoffs are discussed. Students then learn the internal FPGA routing and details of its logic elements. The module also discusses the strategies for area and latency optimization, as well as designing finite state machines. The various steps involved in converting the RTL description to implementation are also covered. They learn about algorithms required for logic minimization, technology mapping, and physical synthesis. Special emphasis is laid on FPGA architecture. A revision to a hardware description language (VHDL, to be specific) covered in a first-year module is also done. At the end of the course, students are expected to:

- be familiar with design methodologies for embedded systems;
- be able to translate system specifications into a register-transfer level HDL that can be implemented on an FPGA;
- understand the performance–resources tradeoffs while designing embedded systems;
- appreciate the back-end algorithms used in electronic design automation.

The aim of the EE4218 project is to enable the students to appreciate the various steps involved in designing an embedded system using commercial tools. In this project, students (in groups of two) are required to implement an algorithm in both software and hardware and measure speedup, if any. Since the module assumes that students have only a basic understanding

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
LIST OF LAB ASSIGNMENTS FOR EE4218

Lab	Topic	Relation to Project
1	Familiarization with ISE and revision of VHDL	Design modules in hardware
2	Developing embedded systems using EDK/XPS	Write software description of modules
3	Designing and adding custom peripherals	Designing co-processor and communicating with Microblaze

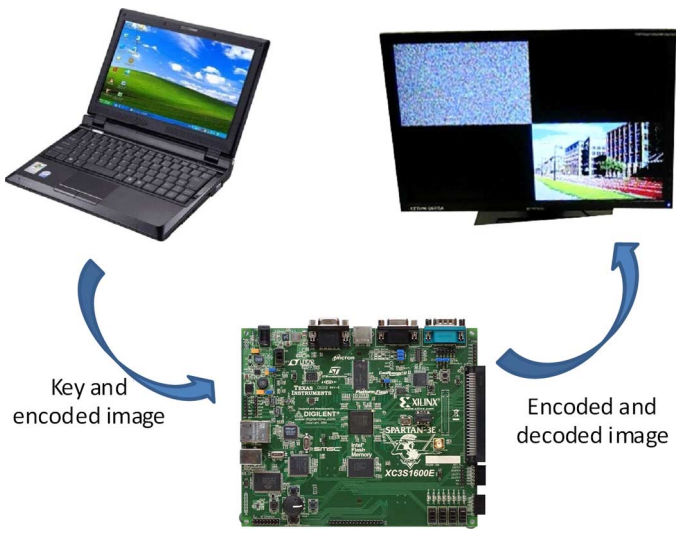

Fig. 3. Setup for EE4218 project.

of digital circuits, a series of labs is designed to help familiarize them with FPGA-based embedded systems development using soft-processors and hardware IPs.

A. EE4218 Lab Details

Table III shows the list of lab assignments that students undergo as a part of preparation for the main project. In the first lab, students familiarize themselves with Xilinx ISE tool-chain and Spartan-3E development board. They develop some simple combinational and sequential modules in VHDL and demonstrate the functionality on the FPGA board. The second lab introduces the students to embedded systems development. They learn how to design a processor-based system using Microblaze. This allows them to define system behavior using C-language. The EDK/XPS environment allows easy customization of the system by adding/configuring various hardware IPs and the processor provided in the toolkit. In the last lab, students learn how to design and add custom modules to the platform. These labs provide sufficient background knowledge for designing complete embedded system with hardware accelerators, thereby enabling hardware/software co-design and the study of the associated tradeoffs between various alternatives.

B. EE4218 Project Details

One of the major aims of the project is to emphasize the importance of hardware acceleration for compute-intensive tasks. As shown in Fig. 3, which gives an overview of the project setup, a screen is attached to the FPGA board. The encrypted image and the key used for encryption are sent over the serial port from the computer to the FPGA board. The decryption is carried out on the FPGA, and the encrypted and decrypted images are displayed on the attached screen. Since the supported screen size is 640×480 pixels, the image size is set to 320×240 pixels for convenient display of images. Since the VGA port on Spartan-3E FPGA board only supports 3-bit colors, a custom image format is defined to pack multiple pixels into bytes. This packing reduces the data to be transferred from the PC as well as the amount of data to be encrypted and decrypted. A desktop application is written to convert input images to 3-bit color and pack and encrypt them from commonly used formats to the required format. The application crops input images to the desired size and packs every five pixels into 2 B $(3 \times 5 + 1 = 16)$. A random bit is generated and stuffed at the end. The addition of a random bit adds noise, making the encryption stronger, and removes any visible data-dependent patterns in the image.

Most of the parts are run on the Microblaze with a hardware module for image decryption. The decrypted data is then unpacked in software before being sent to the display. A dummy module is provided to students for the hardware co-processor connected to the MB with Fast Simplex Link (FSL) ports. While one software implementation is sufficient, at least a couple of hardware alternatives are expected. This offers sufficient opportunities to explore various hardware architectures and study the tradeoffs in area and latency, over and above the hardware versus software considerations. At the end of the project, students are expected to show the results in a Pareto curve, depicting the various architectures designed and evaluating them in terms of lookup tables (LUTs) required and the latency of the design.

In the last two years of running this project, AES (128-bit data and key) and PRESENT (64-bit data and 80-bit key) were chosen for the implementation [10], [11]. The general framework allows a different algorithm to be selected each year, so as to minimize plagiarism without major changes to the project.

VI. PROJECT EVALUATION

Labs/projects determine a significant portion of the grades for EE4214 and EE4218. At the end of the semester, all teams have to give a presentation describing their system design, key distinguishing features, and work division between the team members. All members of the team are quizzed on their understanding of the respective parts. The students have to justify the design choices they make in the projects and appreciate their importance. Individual students are separately required to show the functionality of their contributions. A short demonstration of the complete system is also expected to showcase the basic

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

functionality and extra features in the design. For the final grade, 30% is determined by their system-level design, 20% by the robustness of the design and the accuracy of performance analysis, 25% by the overall functionality of the design, 15% by the presentation, and the remaining 10% by the report. Various criteria are defined for each subsystem on which overall individual and team achievement is determined. For example, in EE4214, for system-level-design, the criteria used are the following.

1) Multiple processors are being used.
2) The tasks are well-balanced among multiple processors.
3) Robust mechanisms are used for interprocessor (e.g., mailbox/mutex) communication.
4) Appropriate and efficient mechanisms are used for intraprocessor synchronization primitives (e.g., semaphores).
5) An appropriate scheduling algorithm is used for processes on the same processor.
6) Interrupts have been used for receiving and sending Ethernet/UART data.

These items help to evaluate the students' understanding of the various concepts taught in the lecture, such as identifying the appropriate scheduling strategy depending on the tasks assigned to a processor and their computational requirements. The evaluation found that about 90% of the teams used multiprocessors for the server, while only 50% of the teams used them for client development. Almost all teams with multiprocessor architectures divided the tasks reasonably well between the processors. While most teams used robust interprocessor communication strategies within the FPGA board, some teams (about 20%) went even further and demonstrated robust communication between the FPGA boards; their system could even handle occasional disconnection of the serial cable. All teams used semaphores and/or mutexes for synchronizing intraprocessor processes. For scheduling, while about 60% of the teams used round-robin scheduling (which was not always the best choice, given the varying importance of various processes running on a processor), 30% of the teams prioritized the tasks correctly; the remaining 10% did not prioritize the tasks appropriately. About 60% of the teams used interrupts for handling communication.

The evaluation criteria are communicated to the students in advance. During the evaluation, detailed analysis is made of individual designs, and feedback is given to the teams on how they could improve it further. In summary, while not all teams managed to accomplish a perfect design for all criteria, they were at least exposed to the various aspects of embedded system design. In the future, it is intended to improve student performance by providing them with intermediate feedback. Additionally, some sample client and server designs (as black boxes) will be provided to students to make it easier for them to test and optimize one side without worrying about the other and to ensure protocol compliance.

A. Sample EE4214 Projects

This section mentions a few of the innovative features and ideas students came up with during the implementation of the project. While implementing the server, only a few teams were able to meet the constraint of refreshing the screen at the required refresh rate of 25 Hz, even after allocating a separate core exclusively to deal with the screen refresh task. This was because the Microblaze and the VGA controller vied for access to the video frame buffer; a part of the off-chip DDR2 memory was allocated to the VGA controller as video frame buffer. One team presented a very innovative idea that overcame this issue and enabled them to easily achieve the required refresh rate. Their solution was to have two separate video frame buffers, one for each alternate video frame. Their code swapped the pointers to the video frame buffers every alternate refresh cycle.

Another attractive feature integrated by one of the teams was a replay system. The positions of all the players and the ball were stored in the memory for the entire history of the game, and then replayed when triggered by on-board push-buttons. The team also allowed the game to be paused by using a push-button. Yet another interesting feature was to allow the run-time strategy to be changed depending on the progress of the game. The tournament provides an excellent platform for the groups to test the quality, robustness, and interfaceability of their design; a video clip of a tournament match conducted is available at [12].

B. Sample EE4218 Projects

A whole spectrum of hardware architectures is seen in the student project presentations. Some groups have a fully pipelined design capable of decrypting one block of data per clock cycle. Others decided to use multiple decryptor modules to speed up the image decryption. Few students explored hardware/software co-design by accelerating only specific functions in hardware, thereby providing good performance-resource tradeoffs.

An example of an extra feature implemented by some groups in the EE4218 project was to send encrypted images via an Ethernet port using lightweight Internet Protocol (lwIP) instead of a serial port. The Ethernet port allowed them to send images at a much faster rate than a serial port. Some of the groups went further and used this feature to decrypt the video on the fly. Video data was decoded on a PC into individual frames that were then encrypted on the PC and sent to the FPGA board for decrypting. Essentially, this was similar to Motion JPEG encoding, except that individual frames were encoded in the custom format. A couple of groups also extended the design to use an HDMI interface (using Atlys Spartan-6 board [9]) for displaying the video instead of the VGA interface, thus allowing a higher screen and color resolution. Many groups also implemented both encryption and decryption where the choice could be made via an on-board dip-switch. A video clip showing some of the features from a project can be seen online at [13].

C. Student Feedback

The university collects both quantitative and qualitative feedback for all courses taught via an online system. An incentive to submit such feedback is that students earn bidding points, which they can use to opt for their preferred courses.

Both projects were very well received by the students as indicated by both quantitative and qualitative feedback. Table IV shows the summary of the quantitative feedback received for the year before and the year after the introduction of the respective projects in both courses. The average teacher effectiveness score (out of 5.0) increased from 4.037 to 4.242 for EE4214 and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

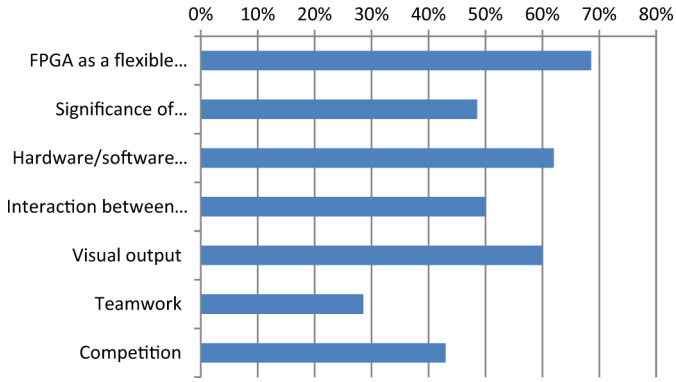

Fig. 4. Student rating of the various motivating factors for the projects.

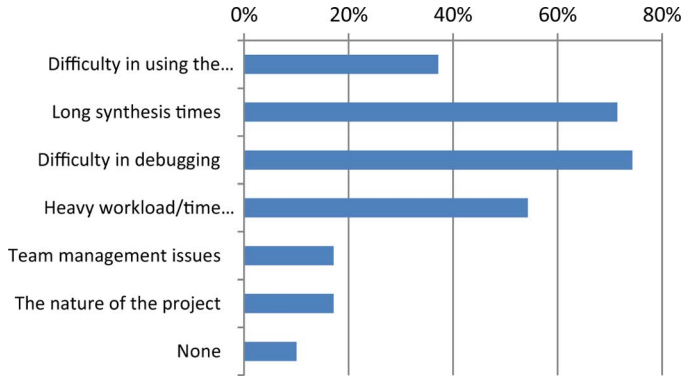

Fig. 5. Student rating of the various demotivating factors for the projects.

TABLE IV
QUANTITATIVE FEEDBACK COLLECTED FROM STUDENTS

Module	EE4214		EE4218	
Year	2009	2010	2010	2011
Number of students	76	83	93	73
Number of respondents	29	39	35	28
Percentage of respondents	38%	47%	38%	38%
Overall numerical score (out of 5)	4.037	4.242	3.886	4.214

TABLE V
PROJECT SPECIFIC SURVEY RESULTS

Survey questions	EE4214 (out of 5)	EE4218 (out of 5)
Practicality of the project	3.43	3.57
Difficulty level	4.36	3.48
Relevance of preparatory labs	3.71	4.05

from 3.886 to 4.214 for EE4218 in the year following the introduction of the projects. The qualitative feedback received from the students was also quite encouraging. Some selected comments follow.

"This course provides maximum practical exposure of the concepts learnt. Able to understand the course. The project in this course was time consuming, but gave an in-depth knowledge."

"This course is perfect. It teaches us a lot of stuff about real-time systems and the project is very fun to work on."

"This is a very interesting course because of the project."

"The projects were the best part of this course."

"The course gave good hands-on experience on FPGA implementation."

"We learn a lot about FPGA and VHDL programming."

Besides the generic survey conducted by the university, a more specific survey was made of students' perception of the use of projects for learning embedded systems. The survey included questions about the practicality of the projects, the difficulty level, and the relevance of preparatory labs. It also captured the factors students found most motivating/demotivating while doing the project. The results are summarized in Table V. The list of motivating and demotivating factors for the two modules are combined and presented in Figs. 4 and 5, respectively. Please note that students could select multiple options. Most students appreciate the flexibility of using FPGAs for design projects. The hardware/software co-design approach of the project was also well received by the students. The visual output was another strong motivating factor when working on these projects. Long synthesis times and the difficulty in debugging the system were the two biggest demotivating factors.

VII. CONCLUSION AND DISCUSSION

Teaching embedded systems can be quite challenging since it spans multiple disciplines. Hands-on experiments are essential to convey the many design principles of such systems. In addition, projects are needed to give students a sense of achievement while reinforcing the concepts taught in class. The real-time embedded systems project and an embedded hardware design project described here used a generic architecture and exposed students to embedded systems design concepts while still making it fun for them. Sufficient foundation is provided to accelerate the learning curve for students.

One of the disadvantages of the EE4214 project is that a significant number of FPGA boards are necessary since the project requires extensive use of hardware. Three FPGA boards are given to each team so that they can easily build and test the entire system in their group. For large classes, significant initial investment may be needed. On the other hand, the same board can be used for multiple courses—for example, digital fundamentals, microprocessor design, or computer architecture.

Other universities have shown interest in adopting the projects/lab material in their own relevant courses. It is sincerely hoped that this work will assist faculty members elsewhere to develop projects to help students better appreciate the constraints imposed by embedded platforms and gain experience in working with them.

ACKNOWLEDGMENT

The authors would like to thank the Xilinx University Program for their support. They donated licenses for their ISE Embedded Edition, sponsored the prizes for the top three teams and sponsored a trip to Beijing for the top team to participate in an open hardware contest [24].

REFERENCES

[1] DuPont, Wilmington, DE, USA, "Facts and figures for trends in embedded systems," 2012 [Online]. Available: http://www.dupont.com/mcm/techinfo.html

[2] L. Brownsword, D. Carney, and T. Oberndorf, "The opportunities and complexities of applying commercial-of-the-shelf components," *Crosstalk*, vol. 11, pp. 4–6, Apr. 1998.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[3] J. Bruce, J. Harden, and R. Reese, "Cooperative and progressive design experience for embedded systems," *IEEE Trans. Educ.*, vol. 47, no. 1, pp. 83–92, Feb. 2004.
[4] P. Koopman, H. Choset, R. Gandhi, B. Krogh, D. Marculescu, P. Narasimhan, J. M. Paul, R. Rajkumar, D. Siewiorek, A. Smailagic, P. Steenkiste, D. E. Thomas, and C. Wang, "Undergraduate embedded system education at Carnegie Mellon," *Trans. Embed. Comput. Syst.*, vol. 4, pp. 500–528, Aug. 2005.
[5] M. Grimheden and M. Törngren, "What is embedded systems and how should it be taught?—Results from a didactic analysis," *Trans. Embed. Comput. Syst.*, vol. 4, pp. 633–651, Aug. 2005.
[6] J. K. Muppala, "Bringing embedded software closer to computer science students," *ACM SIGBED Rev.*, vol. 4, pp. 11–16, Jan. 2007.
[7] D. Jackson and P. Caspi, "Embedded systems education: future directions, initiatives, and cooperation," *ACM SIGBED Rev.*, vol. 2, pp. 1–4, 2005.
[8] A. Kumar, S. Fernando, and M. Manoharan, "Bringing soccer to the field of real-time embedded systems education," in *Proc. 6th ACM WESE*, New York, NY, USA, 2011, pp. 46–52.
[9] Digilent, Inc., Pullman, WA, USA, "Spartan 3E-1600 development board," 2012 [Online]. Available: http://www.digilentinc.com/Products/Detail.cfm? NavPath=2,400,793&Prod=S3E1600
[10] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-the Advanced Encryption Standard*. New York, NY, USA: Springer-Verlag, 2002.
[11] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in *Proc. CHES*, 2007, pp. 450–466.
[12] National University of Singapore, Singapore, "EE4214 project Webpage," 2012 [Online]. Available: http://wiki.nus.edu.sg/display/ee4214
[13] National University of Singapore, Singapore, "EE4218 project Webpage," 2012 [Online]. Available: http://wiki.nus.edu.sg/display/ee4218
[14] W. Wolf and J. Madsen, "Embedded systems education for the future," *Proc. IEEE*, vol. 88, no. 1, pp. 23–30, Jan. 2000.
[15] E. Lee, "What's ahead for embedded software?," *Computer*, vol. 33, no. 9, pp. 18–26, Sep. 2000.
[16] C. Ttofis, T. Theocharides, and M. Michael, "FPGA-based laboratory assignments for NoC-based many core systems," *IEEE Trans. Educ.*, vol. 55, no. 2, pp. 180–189, May 2012.
[17] P. Schaumont, "A senior-level course in hardware software codesign," *IEEE Trans. Educ.*, vol. 51, no. 3, pp. 306–311, Aug. 2008.
[18] T. Hall and J. Hamblen, "System-on-a-programmable-chip development platforms in the classroom," *IEEE Trans. Educ.*, vol. 47, no. 4, pp. 502–507, Nov. 2004.
[19] A. Bindal, S. Mann, B. Ahmed, and L. Raimundo, "An undergraduate system-on-chip (SoC) course for computer engineering students," *IEEE Trans. Educ.*, vol. 48, no. 2, pp. 279–289, May 2005.
[20] S. A. Edwards, "Experiences teaching an FPGA-based embedded systems class," *ACM SIGBED Rev.*, vol. 2, pp. 56–62, Oct. 2005.
[21] A. Hansson, B. Akesson, and J. v. Meerbergen, "Multi-processor programming in the embedded system curriculum," *ACM SIGBED Rev.*, vol. 6, pp. 9:1–9:9, Jan. 2009.
[22] "Microsoft Imagine Cup," 2012 [Online]. Available: http://www.imaginecup.com/
[23] Shanghai Jiao Tong University, Shanghai, China, "Intel Cup Undergraduate Embedded Design Contest," 2012 [Online]. Available: http://nuedc.sjtu.edu.cn/
[24] "Xilinx OpenHW Design Contest," 2012 [Online]. Available: http://www.openhw.org/
[25] J. McCormick, "We've been working on the railroad: a laboratory for real-time embedded systems," in *Proc. 36th ACM SIGCSE Tech. Symp. Comp. Sci. Educ.*, 2005, pp. 530–534.
[26] M. Amirijoo, A. Te"anoviC, and S. Nadjm-Tehrani, "Raising motivation in real-time laboratories: the soccer scenario," in *Proc. 35th ACM SIGCSE*, New York, NY, USA, 2004, pp. 265–269.
[27] National University of Singapore, Singapore, "Old EE414 project Webpage," 2012 [Online]. Available: http://cnl-ece.nus.edu.sg/courses/ee4214/
[28] H. Mitsui, H. Kambe, and H. Koizumi, "Use of student experiments for teaching embedded software development including HW/SW co-design," *IEEE Trans. Educ.*, vol. 52, no. 3, pp. 436–443, Aug. 2009.

Akash Kumar (S'05–M'09) received the B.Eng. degree in computer engineering from the National University of Singapore (NUS), Singapore, in 2002, the joint Master of Technological Design degree in embedded systems from NUS and the Eindhoven University of Technology (TUe), Eindhoven, The Netherlands, in 2004, and the joint Ph.D. degree in electrical engineering in the area of embedded systems from TUe and NUS in 2009.

In 2004, he was with Philips Research Labs, Eindhoven, The Netherlands, where he worked on Reed Solomon codes as a Research Intern. From 2005 to 2009, he was a Researcher with TUe working on his Ph.D. Since 2009, he has been with the Department of Electrical and Computer Engineering, NUS, where he is currently working as an Assistant Professor. He has published over 40 papers in leading international electronic design automation journals, and conferences. His current research interests include analysis, design methodologies, and resource management of fault-tolerant embedded multiprocessor systems.

Shakith Fernando received the B.Eng. degree in computer engineering from the National University of Singapore (NUS), Singapore, in 2005.

He has worked as a Research Engineer in several leading research institutes including NUS and the A*Star Data Storage Institute, Singapore. Currently, he is a Researcher with the Eindhoven University of Technology, Eindhoven, Netherlands. He has published over 10 papers in leading international journals and conferences. His current research interests are analysis, synthesis, and exploration of heterogeneous multiprocessor system-on-chip on FPGA platforms.

Rajesh C. Panicker (S'09–M'13) received the B.Tech. degree in electronics and communication engineering from the University of Kerala, Trivandrum, India, in 2005, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore (NUS), Singapore, in 2012.

He was a full-time Teaching Assistant with NUS from 2010 to 2012. Currently, he is a Research Fellow with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. His interests are in brain–computer interfaces (BCIs), embedded systems, signal processing, and pattern recognition.