

**AC 2008-2890: NEW AND INNOVATIVE EDUCATIONAL MATERIAL FOR  
TEACHING MIXED DOMAIN EMBEDDED SYSTEM DESIGN TO  
UNDERGRADUATE/GRADUATE STUDENTS**

**Alex Doboli, State University of New York-Stony Brook**

**Eddie Currie, Resonance Publications, Inc.**

**Patrick Kane, Cypress Semiconductor Corporation**

**Dave Van Ess, Cypress Semiconductor Corporation**

# New and Innovative Educational Material for Teaching Mixed-Domain, Embedded Systems to Undergraduate/Graduate Students

## Abstract

Efforts related to development of innovative, mixed-signal system design, teaching materials and methodology are presented that focus on the co-design of performance-optimized modules for signal sensing, control, actuation, and communication in embedded systems. The methods developed to assess the quality and degree of assimilation by students of the key course concepts are also presented. Cypress Semiconductor's PSoC<sup>TM</sup> mixed-signal architecture was used to illustrate the concepts covered by the developed materials.

## Introduction

Markets continuously produce new demands and opportunities for creating innovative concepts and technologies in electronic design<sup>1,2,8</sup>. This includes traditional markets (e.g., telecommunications, consumer goods, military, and automotive industry) and other sectors, such as healthcare and education. Future generation electronic systems will increasingly require continuous information acquisition from distributed heterogeneous sources, real-time information processing, wireless access to large databases, efficient information retrieval/data mining, autonomous operation, and self-adaptation to new conditions, etc.<sup>8</sup>. Successfully meeting these challenges will provide not only interesting research problems but will allow the education curriculum to keep pace with these challenges and opportunities<sup>6,9,10</sup>.

Addressing the emerging needs of modern electronic systems presents several important characteristics that distinguish them from more traditional embedded systems. *Viz.*<sup>6</sup>:

- *Mixed-domain integrated electronic systems*: Many applications require integrated signal acquisition, data processing, control, and actuation<sup>2,8,10</sup>. The design process involves co-designing the analog circuits, digital blocks, firmware, and software. For example, devising the sensing front-end requires knowledge of analog and mixed-signal circuits, since the sensed signals are of different types (voltages, currents, charges, digital signals, etc.) and characteristics, e.g., offsets, noise, required driving capabilities, etc.. Moreover, efficient accessing of the sensed signal values in a high-level program language (e.g., C++) involves programming languages and techniques, firmware, compilers, and operating systems skills.
- *Reconfigurable systems*: Many modern execution platforms are reconfigurable, and their architecture can be customized (to various degrees) to address the application characteristics. Customization is a very powerful procedure for developing performance-optimized implementations (e.g., optimized for processing accuracy, speed, real-time constraints, energy/power consumption, cost, reliability, and so on). Presently, architectures, such as Cypress Semiconductor's PSoC family of mixed-signal array<sup>1</sup> devices, offer the possibility of customizing not only the digital blocks, but also the analog circuits, I/O ports, interconnect, supply voltages, and clock frequencies. While such customization is powerful, architectural reconfiguration can be a complicated

process involving (i) complex performance and cost trade-offs spanning the analog, digital and software domains, (ii) circuit nonidealities and nonlinearities (offsets, delays, noise, representation errors, sampling errors, etc.), and (iii) difficult system integration and testing.

- *Networked systems*: Many modern applications require decentralized signal acquisition, efficient data transfer among networked embedded systems, and distributed processing. Networking and data communication is achieved by using standardized protocols, such as SPI, UART, I2C, PCI, USB, etc..

Design of mixed-domain embedded systems involves a broad range of knowledge, some of which is currently provided by unrelated EE, CE, and CS disciplines<sup>5</sup>. Industry feedback suggests that the knowledge needed for designing real-life systems is currently acquired primarily as a result of industry experience occurring over a long period of time, since classroom education offers a disparate presentation of EE, CE and CS topics. This makes it difficult for students to understand the cross-disciplinary issues that occur in the development of modern embedded systems. Therefore, the curriculum must offer students the opportunity of studying jointly the related topics in analog and digital design as well as software with the following two goals:

- Students must understand the connections between analog, digital, and software domains. For example, apparently there is no correlation between analog signal sampling (studied in EE), and time-optimized vector multiplication (studied in CS). However, when selecting an Analog to Digital Converter (ADC) for an embedded application, it is imperative that they understand that the ADC sampling frequency not only sets the precision of the conversion but also defines a timing constraint for the speed of the related digital signal processing routines, some of which may involve vector multiplication. Therefore, in this simplified context, developing high precision ADCs involves not only analog design but also code optimization.
- Students must be familiar with the advantages and limitations of high-level design paradigms. High-level design flows (e.g., top-down design, successive refinement, etc.) are very effective in offering short time-to-market and low design costs<sup>3</sup>. However, they must also understand that often short time-to-market comes at the penalty of lower design performance, especially with respect to the analog and mixed-signal circuits as well as high-performance digital circuits. Therefore, students must know well (i) the nature of the performance trade-offs that are made, as well as (ii) the nonidealities that exist for electronic circuits, and which can invalidate the results of the high-level design.

Over the last 18 months, significant effort has been expended on the development of new teaching material and activities (courses, student design projects, and research) related to mixed-signal embedded system design<sup>11,12</sup>. The developed material includes the following:

- Complete course material for a one-semester course on embedded mixed-signal systems. The course is designed for senior undergraduate students and first-year graduate students in Electrical and Computer Engineering (ECE). This material is due to be published as a textbook and is available at [www.cypress.com](http://www.cypress.com). For an evaluation copy please contact [cuap@cypress.com](mailto:cuap@cypress.com).

- Related laboratory material was prepared by the engineering staff at Cypress Semiconductor Inc. This book is also available at [www.cypress.com](http://www.cypress.com) For an evaluation copy please contact [cuap@cypress.com](mailto:cuap@cypress.com).
- Several student design projects were defined and completed, including embedded controllers, encryption systems<sup>1</sup>, temperature log systems, monitoring systems, and telephone log systems.
- As a byproduct, this effort identified interesting research in the area of design methodologies for reconfigurable  $\Sigma\Delta$  ADC design. Papers have been published in peer-reviewed conference proceedings.

The main goal of the course material is to teach students the fundamental, theoretical concepts and practical skills required in designing and building embedded systems. The teaching material presents the concepts by stressing: (i) embedded design based on high-level specifications (including for the analog and mixed-signal interfaces), (ii) system performance and cost optimization by employing design trade-off analysis, and (iii) implementing embedded systems on reconfigurable platforms by following a module-based design paradigm. The four credit course includes both lectures and laboratory activities.

The presented high-level specifications express the abstract data flow in a system, including signal acquisition and conversion, control procedures, and actuation<sup>7</sup>. Examples use the C language, but the concepts are also valid for other specification formalisms<sup>3</sup>. The material provides a comprehensive treatment of design trade-offs, trade-off formulation and analysis. The principle trade-offs, viz., speed, cost, power consumption, and accuracy (precision), are illustrated by various examples, and different design solutions are presented for each targeted performance specification. Emphasis is placed on the importance of performance modeling, and a comprehensive set of modeling procedures for electronic circuits and software.

Section 2 details the material developed for teaching embedded mixed-signal systems based on the Cypress' PSoC, mixed-signal, embedded architecture. Section 3 presents the means devised for course assessment, including course projects and laboratory material. Section 4 offers data on student learning and feedback. Finally, conclusions are provided.

## **2. Developing New Teaching Material on Embedded Mixed-signal Systems**

The course material focuses on the development of mixed-signal embedded applications that utilize Systems On Chip (SoC) technology. The course discusses design issues, such as (i) implementing new functionality, (ii) developing new interfacing capabilities, and (iii) improving performance by programming the embedded microcontroller and customizing the reconfigurable analog and digital hardware of the SoC. Prerequisites for the course include introductory courses in computer programming, digital design and analog circuits, e.g., OpAmps.

---

<sup>1</sup> The case study of the encryption algorithm is presented in [The International Data Encryption Algorithm](#) by B. Schneier, Other Block Ciphers, Applied Cryptography, pp. 319-325, John Wiley, 1996. This algorithm is also presented in the developed course material.

Students successfully completing the course are expected to have sufficient theoretical and applied knowledge and design skills to:

- utilize a combination of analog and digital modules and develop software drivers for interfacing new devices,
  - develop system-level designs including specification, profiling, debugging, and trade-off exploration/optimization for mapping the system behavior to specific building blocks,
- and,
- customize the target architecture utilizing analog/digital, dynamic reconfiguration for implementing new functionality and obtaining better performance.

In terms of ABET's (a)-(k) criteria<sup>4</sup>, the following are the primary outcomes set for the course:

- (a) Ability to apply knowledge of mathematics, science and engineering.
- (b) Ability to design and conduct experiments, as well as, analyze and interpret data.
- (c) Ability to design a system, component, or process, to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health, and safety manufacturability, and sustainability.
- (d) Ability to function on multidisciplinary teams.
- (e) Ability to identify, formulate and solve engineering problems.
- (g) Ability to communicate effectively.
- (k) Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

A significant challenge was the identification and definition of the proper “interface” between the EE, CE and CS topics involved in embedded mixed-signal system design. This is important because it serves as common ground for students with different backgrounds, and as a liaison for topics that are covered in more traditional courses.

In the course material, embedded mixed-signal architectures are presented at the following levels: (i) analog subsystems and circuits are introduced as macromodels with a rich set of nonidealities and nonlinearities<sup>7</sup>, (ii) digital circuits are treated at the FSM (Finite State Machine) and basic block level (registers, adders, etc.), the microcontroller at the instruction set architecture level, and (iii) software is based on a three-layer structure beginning with assembly code and ending with high level API routines. This description style has the advantage of serving both as a theoretical presentation, useful for high-level design flows, and the introduction of the nonidealities specific to electronic circuits. Students study theoretical signal representation, sampling, quantization, accuracy errors, feedback, model composition, modularity concepts and hierarchy. In addition, they are also introduced to the important “real” electronic circuits issues such as various speed-precision-cost-power consumption trade-offs, and the impact of circuit nonidealities/nonlinearities on system operation and performance, e.g., saturation, stability, hysteresis, delay, bandwidth, linearity, and noise.

Using this description style for embedded mixed-signal systems, the teaching material focuses on the three concepts:

- *Comprehensive presentation of the data flow involving sensing, control, actuation and communication:* Students study and understand a complete signal processing flow that

consists of analog signals being present at the I/O ports, analog signal conditioning, filtering, ADC, interrupt service routines, digital signal processing, firmware, software routines for the control application, and actuation by PWM's and power transistors. Emphasis is placed on the importance of observing and understanding the interactions between the analog, digital, and software domains.

- *Identification, formulation and analysis of inter-domain performance trade-offs:* Students are taught about different facets of the cost-accuracy-speed-energy/power consumption trade-offs that manifest in mixed-signal design. Some of the discussed trade-offs include: (i) signal bandwidth – conversion accuracy – software latency, (ii) cost – speed (and hardware-software partitioning for speed optimization), and (iii) flexibility – speed (and the impact of the used addressing modes and data mapping to memory).
- *Application-specific customization of the reconfigurable architecture:* Students learn how trade-offs guide the selection of performance-optimized implementation alternatives, which are then developed and tested using the reconfigurable platform. This is very important in acquiring hands-on experience with the implementation of embedded mixed-signal systems.

The teaching material relies on the PSoC embedded mixed-signal architecture from Cypress Semiconductor Inc. to illustrate these three concepts. PSoC is a family of Mixed Signal Arrays with an on-chip microcontroller, memory, and highly configurable analog and digital peripherals.

**Course Lectures.** The lecture material covers the following topics:

1. Introduction (1 week).

- a. Types of embedded applications, including a simple mixed-signal embedded design as an illustrative example.
- b. Importance of performance requirements for design.
- c. Summary of mixed-signal embedded architectures.
- d. Top-down design flow (Design refinement. Performance modeling. Testing).

2. Mixed-signal embedded SoC architectures (3 weeks)

- a. Mixed-signal SoC architectures.
- b. Microcontroller core. Instruction set.
- c. Memory system.
- d. Interrupts.
- e. I/Os. ISR and drivers.

3. Digital subsystem (3 weeks)

- a. Principles of RTL (Register Transfer Level) design.
- b. Basic digital building blocks (timers, counters, CRC generator).
- c. Dynamic reconfiguration.
- d. Developing simple application specific co-processors using dynamic reconfiguration.

4. Analog building blocks (1 week)

- a. Basics of switched capacitor analog circuits.

- b. Presentation of basic building blocks (ideal op amps, comparators, gain stages, integrators).

#### 5. Analog filters and converters (3 weeks)

- a. Filter characteristics.
- b. Filter types.
- c. Circuit non-idealities and their impact on filter performance.
- d. Mapping filters to building blocks).
- e. Analog to digital converters (ADC characteristics.  $\Delta\Sigma$  ADC). Digital to analog converters.

#### 6. System level trade-off analysis (2 weeks).

- a. System performance modeling.
- b. Trade-off curves.
- c. Trade-off analysis (cost – speed – power/energy consumption – number of pins).
- d. System optimization.

### **Laboratory Material.**

The laboratory is scheduled as 12 lab sessions of 3 hours each. Lab activities are based on the PSoC development board (Eval 1) from Cypress Semiconductor. (PSoC is a mixed-signal SoC with an integral 8-bit microcontroller, on-chip flash/RAM memory, reconfigurable analog/digital array, and a variety of other blocks commonly used in embedded applications<sup>1</sup>.)

Each of the 12 lab sessions focuses on a particular concept, but all are tied together with the underlining theme of constructing a temperature-compensated, fan controller.

The lab sessions topics are:

Lab 1: CPU and General Purpose I/O

Lab 2: Interrupts

Lab 3: Pulse Width Modulation and Global Outputs

Lab 4: Three Wire Fans, Tachometers and Global Inputs

Lab 5: Integrating Speed Controller

Lab 6: I2C Serial Interface

Lab 7: Analog Grounds and DACs

Lab 8: Comparators

Lab 9: Delta Sigma Modulation

Lab 10: Measuring Temperature

Lab 11: Filters

Lab 12: A Temperature Compensated Fan Controller with I2C Interface.

These lab activities were chosen because they:

- represent typical examples of the types of projects an engineer will design when employed in industry,
- incorporate analog, digital and software design,
- introduce some basic control theory concepts,

- require the use to several different types of digital serial communication protocols,
- illustrate all of the basic concepts that are commonly encountered in the development of an embedded mixed-signal system

Section 3 presents the means devised for course assessment, including course projects and laboratory material.

### 3. Assessment Methods

Course assessment is based on three course projects and the 12 lab sessions. The lab sessions develop the students' practical skills required to build a system and the course projects expose the students to system design and trade-off analyses between analog, digital and software domains. Given the complexity of system design, it is difficult to devise exam questions or any other short term testing form that would accurately assess the degree of the students' assimilation of the course material. However, successful students' performance of the labs and course projects do require that the students mastery of the core concepts presented and can be used to gauge the students mastery of the material presented.

The three course projects involve increasingly difficult design concepts and system complexity (size). The first project requires a software-only implementation of an embedded design with the goal of familiarizing students with the PSoC architecture, its assembly language programming and I/Os. The second project involves co-designing the digital circuits and software required to optimize the system's speed. The third project requires the co-design of the analog/digital circuits and software for optimizing speed, precision and communication bandwidth. In all three projects, students must apply both theoretical and design knowledge to successfully complete the project assignments.

The following table lists the desired course outcomes, the performance criteria through which each is measured, and the precise assessment procedure to measure each criterion.

<b>Outcomes</b>	<b>Performance Criteria (PCs)</b>	<b>Assessment Procedure</b>
(a) Ability to apply knowledge of mathematics, science and engineering.	<ul style="list-style-type: none"> <li>• Building signal (data)-flow descriptions for complex systems.</li> <li>• Developing large-scale FSMs for reactive behavior.</li> <li>• Designing logic equations optimized for speed and cost.</li> </ul>	<ul style="list-style-type: none"> <li>• Three course projects that focus on the development of understanding of embedded systems, from (1) software only implementation, (2) digital-software implementations, and (3) mixed analog-digital-software implementation</li> <li>• Each project addresses the PCs</li> </ul>
(b) Ability to design and conduct experiments as well as analyze and interpret data.	<ul style="list-style-type: none"> <li>• Testing digital, reconfigurable hardware.</li> <li>• Testing analog, reconfigurable hardware.</li> <li>• Testing reconfigurable I/Os.</li> </ul>	<ul style="list-style-type: none"> <li>• Lab activities require testing of the designs.</li> <li>• Course projects require testing and trouble shooting the designs.</li> <li>• Instructor-developed set of test cases that must be passed by each project.</li> </ul>
(c) Ability to design a system, component, or process to meet desired needs within realistic	<ul style="list-style-type: none"> <li>• Understanding performance trade-offs across mixed-domain modules.</li> <li>• Designing modules incorporating digital hardware, drivers in assembly language,</li> </ul>	<ul style="list-style-type: none"> <li>• Develop a module for reactive control with 20+ states.</li> <li>• Develop a module that requires extensive data processing and</li> </ul>

constraints such as economic, environmental, social, political, ethical, health, and safety manufacturability, and sustainability.	<ul style="list-style-type: none"> <li>and interfacing procedures in C language.</li> <li>• Designing modules incorporating analog hardware, drivers in assembly language, and interfacing procedures in C language.</li> <li>• Developing efficient I/O interfaces and I/O data communication modules.</li> <li>• Identify the main modules for the system performance and optimize the modules.</li> </ul>	<ul style="list-style-type: none"> <li>optimizing the speed of the module.</li> <li>• Develop three alternative design solutions and characterizing their performance and cost.</li> </ul>
(d) Ability to function on multidisciplinary teams.	<ul style="list-style-type: none"> <li>• The experience of working as part of a team</li> <li>• The experience of leading a team</li> </ul>	<ul style="list-style-type: none"> <li>• All lab activities are done in groups of two students.</li> <li>• All project activities are done in groups of two students.</li> </ul>
(e) Ability to identify, formulate and solve engineering problems.	<ul style="list-style-type: none"> <li>• Ability to understand the importance of trade-offs in fine-tuning the system functionality and performance.</li> </ul>	<ul style="list-style-type: none"> <li>• Identify and formulate the trade-offs of the course projects.</li> <li>• Understand opportunities for using available hardware to expand the functional capabilities of the system.</li> </ul>
(g) Ability to communicate effectively.	<ul style="list-style-type: none"> <li>• Improving technical writing skills</li> <li>• Improving presentation skills.</li> </ul>	<ul style="list-style-type: none"> <li>• Writing project reports for each project.</li> <li>• Presenting the projects to the class.</li> <li>• Writing lab reports for each lab.</li> <li>• Extra credit if a Application Note is submitted and accepted by Cypress.</li> </ul>
(k) Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.	<ul style="list-style-type: none"> <li>• Learning the use of GUI-based tools for SoC layout design.</li> <li>• Learning the use of debugging tools for embedded systems.</li> </ul>	<ul style="list-style-type: none"> <li>• All lab activities require using PSoC Designer for assembly programming, C programming, layout design and debugging.</li> <li>• Logic optimization tools (like ESPRESSO) might be used.</li> <li>• Extra credit is given for using SystemC for high-level system specification.</li> </ul>

Each group must demonstrate the practical implementation, which is evaluated by the instructor using a predefined set of test cases (unknown to the students). The grade for each project is computed based on the formula:

$$0.5 \times [10\% \text{ for C specification} + 10\% \text{ for testing} + 20\% \text{ for profiling} + 20\% \text{ for developing implementation alternatives} + 20\% \text{ or system implementation on PSoC} + 20\% \text{ for project report}] + 0.5 \text{ grade for demo}$$

The next part lists three sample projects used in the class.

**Course Project 1:** The project goal is to develop the routines in PSoC assembly language for the newspaper vending machine controller. The vending machine application has been a popular project topic for several ECE design courses (such as digital design and VLSI design) at many universities.

The specification states that a newspaper costs \$x. The vending machine accepts nickels, dimes, and quarters. If exact change has been entered then the output signal Unlatch is generated, so that the customer can get the newspaper. If the deposited amount exceeds the price of a newspaper,

then change is given. If there is not enough change in the repository then the coins are refunded by issuing the output signal Refund. Change is given to the customer by generating pulse signals for the repository: the signal NR releases one nickel, and the signal DR releases one dime. One coin is released at a time.

Each group of two students must hand in a written report and the complete implementation describing the design, debugging and testing procedures, and the related experiments, including the following:

- The overall algorithm, the structure of the routines, and the selected data structures.
- Motivation for choice of algorithms, routine and data structures used in the design.
- Discussion of the interfacing signals used.
- Estimation of the hardware resources that are used in the design, including I/O pins, flash memory, and RAM memory.
- Estimation of the number of clock cycles required to execute the routines, and the computational complexity of the routines and their required program/data memory.
- Presentation of the testing and debugging procedure to verify the correctness of the implementation.
- Development of a procedure that guarantees that the design is correct for as many situations as possible.
- Providing sufficient waveform plots to justify the correctness of the design.
- Discussion of possibilities to improve the performance and cost of the design.

**Course Project 2.** The goal of the second project is to develop and implement on PSoC an embedded application optimized for execution speed including:

- Understanding the functionality, I/Os, performance requirements, and design constraints of the application.
- Developing the specification of the application in C language.
- Testing and simulating the C program to guarantee its correct execution.
- Profiling (counting the number of clock cycles, and number of memory bytes) the sub-parts of the program.
- Developing alternative implementations based on PSoC's digital hardware blocks for the profiled sub-parts of the program.
- Identifying the implementations of the sub-parts that lead to the system implementation with optimized performance, i.e., the least number of clock cycles.
- Developing the PSoC implementation for the performance-optimized design.

Testing and simulation of the specification are extremely important to determine the correctness of the design. Thus, the goal is not only the specification of the functionality of the system, but also the design of a test bench for confirming the correctness of the description. The report must include sufficient simulation evidence to establish that the specification is correct.

Each group must hand in the implementation of the design and a report with the following sections:

- Introduction.
- Description of the application.
- Organization of the C specification, data structures, and algorithms.
- Description of the test bench for debugging and profiling.
- Simulation and profiling results and discussion of the results.
- Development of alternative implementation solutions.
- PSoC implementation and experiments using the implementation.
- Conclusions.
- References.

**Course Project 3:** The goal is to develop and implement the design of a temperature logging system that has the following functionality:

- The system monitors and records the temperature measured using a temperature sensor. The system records the temperature value and the corresponding time when of the reading was recorded.
- The system computes the overall average temperature, its standard deviation and the average temperature and standard deviation for the readings over the last  $x$  minutes.
- The recorded temperature values and the computed statistics are transmitted through UART to a computer and stored in a file.
- Determine the computational resources required for temperature sampling, data processing, and data communication. How do these values differ for different values of  $x$ ?
- If the temperature rises above temperature  $T_1$  then the system has to shut down immediately. How fast is “immediately” in the design?
- Estimate how often the system can read and store new values. Obviously, this depends on the speed of the sampling sub-system.
- Provide a design solution that maximizes the number of samples that are read and processed per second.
- If the temperature rises above temperature  $T_2$  ( $T_2 < T_1$ ), then the system displays a warning message and shuts down in  $y$  seconds, unless the user decides otherwise.

Each group must submit the design implementation, and a report, describing the solutions to all of the requested design issues, design details, motivations for the solutions, discussion of the experimental data, and suggestion of ways to improve the performance of the implementation.

#### **4. Student Learning and Feedback**

The course has been offered twice: once in the Fall of 2006 and again in the Spring of 2007. Post-course evaluations indicate that students found the course strengths to be the comprehensive presentation of embedded system topics and the associated practical skills. The course projects proved to be an efficient means for learning design concepts, including hardware circuits and assembly code programming.

At the end of the course, the majority of the students developed the necessary theoretical knowledge and practical skills to allow them to create medium-size embedded applications based on Finite State Machines programmed in assembly code, C language, and a mixture of both. Students also understood I/O interfacing procedures, developing firmware routines of intermediate complexity, and utilizing standardized interfacing modules, such as SPI, UART and I2C. These general observations are also supported by the fact that the following semester, some students decided to use PSoC for various course and research projects, including senior design. They were capable of easily interfacing the chip to new sensing and display modules, such as sonar, GPS, color display, sound cards, and flash memory.

Furthermore, the course helped the students to understand the importance of design tradeoffs, and learn techniques for analyzing them. The course material for parts 3 and 6 (see Section 2) and Project 2 showed students that significant improvements in execution time are possible by using customized hardware modules, such as MAC, timers and counters as opposed to pure software implementations. Based on Project 2, students learned to profile embedded programs, analyze and identify the computationally intensive parts of their programs, and subsequently develop customized hardware for these parts. Many students successfully completed the second project, and developed alternative design implementations, analyzed their performance, and suggested a promising solution. However, we learned that understanding the importance of design tradeoffs involves a steep learning curve and more complex design examples are required to illustrate possible design alternatives.

Moreover, students learned techniques for modular design of analog and digital blocks. They acquired fundamental knowledge with respect to the hierarchical organization of IP modules, the importance of circuit macromodels in predicting performance, and developing and organizing firmware and API methods. The majority of students successfully completed a modular design for Project 3. However, many found it hard to explore the open-ended aspects of their project description, as well as the correlations between the precision – speed – and memory requirements of the subsystems.

Student suggestions for improvement included an overview presentation of popular embedded architectures (including FPGAs) and summaries of the strength and weaknesses of each. Students also indicated that the work requires very good coursework planning so that the lab work and course project work do not become overwhelming.

Another aspect that we plan to further detail is the impact of constraints such as power and energy consumption on reconfigurable embedded design. We are considering the presentation of models for power and energy consumption in analog, digital hardware and software, and then utilizing these models in design. Currently, we are analyzing the idea of modifying certain laboratory descriptions and Project 2 to include frequency scheduling and voltage scaling as means for reducing power and energy consumption. Finally, we plan to strengthen the emphasis on using formal descriptions for design, such as expressing mixed-signal functionality using data and signal flow graphs, and trade-off analysis methods based on the formal models in Projects 2 and 3.

## 5. Conclusions

This paper presents efforts at State University of New York, at Stony Brook, related to developing new teaching material/methodologies with respect to embedded, mixed-signal, system design. The material focuses on the co-design of performance-optimized modules for signal sensing, control, actuation, and communication in embedded systems. The methods used for assessing the core concepts have been also presented in the paper. The PSoC mixed-signal architecture was used to illustrate the concepts.

The course material teaches students the fundamental theoretical concepts and practical skills required in designing and building embedded systems. The material stresses (i) embedded design based on high-level specifications (including for the analog and mixed-signal interfaces), (ii) optimizing the system performance and cost through design trade-off analysis, and (iii) implementing embedded systems on reconfigurable platforms by following a module-based design paradigm. The high-level specification notations presented express the abstract data flow in a system, including signal acquisition and conversion, control procedures, and actuation. Examples are based on the C language. The material offers a comprehensive discussion of design trade-offs, trade-off formulation and analysis. The main trade-offs, viz., speed, cost, power consumption, and accuracy (precision), are illustrated by various examples. Different design solutions are presented for each targeted performance specification.

## Bibliography

1. "PSoC Mixed Signal Array", Technical Reference Manual, Document No. PSoC TRM 1.21, Cypress Semiconductor Corporation, 2005.
2. R. Ohba, "Intelligent Sensor Technology", J. Wiley, 1992.
3. F. Vahid, T. Givargis, "Embedded System Design. A Unified Hardware/Software Introduction", John Wiley, 2002.
4. "Accreditation Policy and procedure Manual", <http://www.abet.org>.
5. "IEEE CS/ACM Computing Curricula & Computer Engineering", <http://www.eng.auburn.edu/ece/CCCE/WoodenManReport.pdf>, 2003.
6. H. De Man, "System-on-chip Design: Impact on Education and Research", IEEE Design & Test of Computers, July-September 1999.
7. A. Daboli, R. Vemuri, "Behavioral Modeling for High-Level Synthesis of Analog and Mixed-Signal Systems from VHDL-AMS", IEEE Transactions on CADICS, Vol. 22, No. 11, November 2003.
8. D. Estrin, D. Culler, K. Pister, G. Sukhatme, "Connecting the Physical World with Pervasive Networks", IEEE Pervasive Computing, Vol. 1, No. 1, pp. 59-69, January-March, 2002.
9. A. Laffely, W. Burleson, "Using Systems-on-a-Chip as a Vehicle for VLSI Design Education", Proc. Of the 2002 Microelectronics Systems Education Conference, 2002.
10. J. Nestor, D. Rich, "Adding Analog and Mixed Signal Concepts to a Digital VLSI Course", Proc. Of the 2002 American Society for Engineering Education Annual Conference & Exposition, 2002.
11. A. Daboli, E. Currie, "Introduction to Mixed-Signal Embedded Design", textbook, to appear, 2008.
12. ESE 366: Design using Programmable Mixed-Signal Systems-on-Chip, <http://www.ece.sunysb.edu/~adaboli/ESE366.html>.