Incorporating Large-Scale FPAAs Into Analog Design and Test Courses

Christopher M. Twigg, *Member, IEEE*, and Paul E. Hasler, *Senior Member, IEEE*

***Abstract*—The development of large-scale field-programmable analog arrays (FPAAs) provides an excellent opportunity for expanding the capabilities of analog design courses and their laboratories. These devices allow the complete design and test cycle of an analog IC to be explored within the span of a single-term course without being limited to prefabricated integrated circuits. A brief discussion of the authors' experience in incorporating the reconfigurable analog signal processor (RASP) into courses and workshops is presented.**

***Index Terms*—Analog design, analog test, field-programmable analog array (FPAA), laboratory, reconfigurable analog.**

I. Reconfigurable Analog Devices in Education

THE real world is analog and thus requires engineers with the knowledge and experience to design and test analog integrated circuits (ICs) and systems. Although most systems revolve around digital design practices, their interfaces with real world signals will always require analog expertise. As more electronics are designed for portable applications, the power saving benefits of analog circuits have been tightly integrated with the flexibility of programmable digital devices to form complex system-on-a-chip (SoC) designs to save both power and area. All of this has increased the demand for students who are better educated in analog or mixed-signal design and test.

The course load of an electrical engineering student is usually very full, so there is little room for expanding the curriculum to include additional analog design and test courses. Most undergraduates interested in analog design are only able to take a single elective course in the subject. As such, this course should cover the entire analog integrated circuit development cycle from designing an IC to testing that IC [1]. As depicted in Fig. 1, the first stage of the analog IC development cycle entails an iterative loop around design and simulation, a process which is typically covered in most analog design courses. However, the next two stages in the cycle are IC fabrication and testing, which are handled in many different ways depending upon the particular school and course.

In some instances, ICs are designed in one course and tested in another to allow for the fabrication time of the ICs, which can be several months long. However, this schedule is problematic in that it requires a two-term course sequence, which is difficult given the already tight curriculum. The fabrication/testing iteration is also limited to a single attempt. If a student makes a mistake in the design or layout in the first term, they are unable to make corrections in the second term with enough time to test the revised IC.

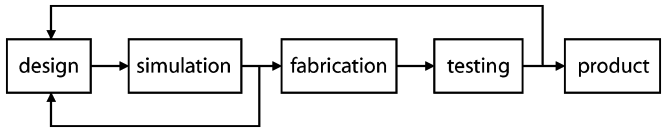

Fig. 1. The typical custom analog IC development cycle.

In single-term courses, IC testing is commonly performed on prefabricated ICs [1]–[3]. This approach introduces the various components of the complete development cycle and provides hands-on experience, which can also reinforce the material being taught concurrently in the classroom. However, the measurements that can be made during these laboratory exercises are limited to those devices and circuits that were predesigned and fabricated on the class IC.

Reconfigurable analog devices present an excellent opportunity for analog and mixed-signal electronics education. Like their digital counterparts, these devices can be used to synthesize hardware that can be tested immediately. By incorporating these devices into analog and mixed-signal course laboratories, a new level of flexibility in analog testing can be realized. In addition to the fixed circuit exercises performed using custom ICs, these reconfigurable devices allow the students to design, synthesize, and test their own circuits for more advanced laboratory projects. The devices can even be used to introduce the students to reconfigurable and programmable analog and mixed signal technologies, such as those being developed and produced by Anadigm [4] and Cypress Semiconductor [5].

II. A Reconfigurable Laboratory Station

To enable this level of reconfigurable analog hardware interaction in analog design and test courses at the Georgia Institute of Technology (Georgia Tech), Atlanta, the platform depicted in Fig. 2 was developed around the large-scale field-programmable analog array (FPAA) research efforts at Georgia Tech [6]. This FPAA setup was inspired by the educational boards and kits commonly used by field-programmable gate array (FPGA) manufacturers, such as Altera [7] and Xilinx [8], to promote the use of their devices within an educational setting. In addition to the hardware platform, a computer-aided design (CAD) tools flow for these FPAAs was also developed to facilitate quicker design synthesis, similar to that commonly expected of modern

Manuscript received June 30, 2007; revised November 26, 2007. First published June 17, 2008; last published August 6, 2008 (projected).

C. M. Twigg is with the Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902-6000 USA (e-mail: ctwigg@binghamton.edu).

P. E. Hasler is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: phasler@ece.gatech.edu).

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.2008.916762

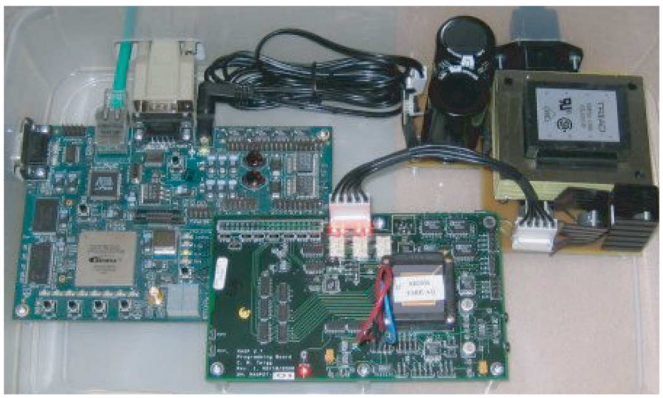

Fig. 2. Self-contained FPAA development platform within a portable box.

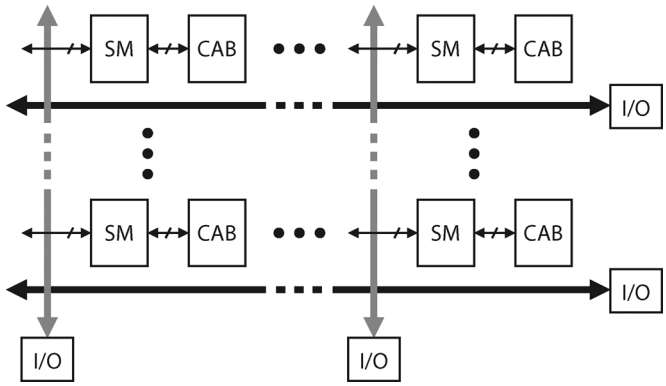

Fig. 3. RASP architecture.

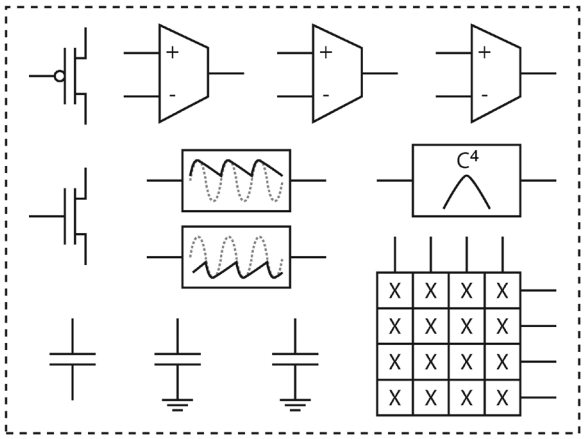

Fig. 4. Example CAB components.

FPGA software. A discussion of both aspects of this laboratory environment is presented in the following subsections.

A. FPAA Hardware Platform

The platform in Fig. 2 is composed of an FPAA interface board, a commercial Altera [7] FPGA development board, and a power module. The FPAA interface board provides the digital and analog control signals used to program the FPAA as well as perform basic measurement functionality. To reduce the dependence upon bench equipment, several measurement ICs were added to the board design, such as analog-to-digital converters (ADCs) and current-to-voltage converters. When combined with an onboard digital-to-analog converter (DAC), these devices allow basic voltage sweeps for measuring and characterizing the various circuits and systems that can be synthesized on the FPAA.

The FPGA board is used as a flexible digital interface between the FPAA board and a personal computer (PC), through a direct ethernet connection. An embedded Nios processor running on the FPGA board provides the digital control signals for the FPAA interface board components. Custom digital hardware synthesized on the FPGA provides parallel operation of the DACs and ADCs on the FPAA interface board, which allows for synthesized instrumentation on this platform. By synthesizing oscilloscope, ammeter, and other instrumentation functions, this portable hardware platform can be introduced into environments where these measurements have previously been infeasible, such as the classroom.

B. FPAA Architecture

At the center of the FPAA board is the reconfigurable analog signal processor (RASP) [6] developed at Georgia Tech. The internal architecture of the RASP is depicted in Fig. 3. This FPAA is composed of a two-dimensional array of computational analog blocks (CABs). Connected to each CAB is a local switch matrix (SM) that provides interconnectivity between the various analog components contained within the CAB. Global vertical and horizontal routing connects to the local switch matrices and provides access to the input/output (I/O) pins. The RASP 2.7 used during these courses at Georgia Tech contains 56 CABs and 26 I/O pins.

Each CAB contains a number of analog components, as depicted in Fig. 4. Unlike the case for FPGAs, there really is no core cell that can efficiently be used to synthesize any arbitrary function. Fine-grain components such as transistors and capacitors provide maximum flexibility in FPAAs. Almost anything can be constructed using these fine grain primitives. However, the complexity of synthesized circuits and systems can be limited by the available routing, and the parasitics resulting from the reconfigurable interconnectivity can reduce the performance of the compiled circuits. To mitigate these issues, medium-grain components, such as operational amplifiers, are included to provide increased performance while maintaining a high level of flexibility. The majority of CAB components fall within the medium-grain complexity, since they provide a good balance between performance and reconfigurability.

Course-grain components are also included within some of these CABs. Special-purpose devices, such as vector-matrix multipliers and programmable bandpass filters, provide complex functionality, high performance, and minimized area overhead. However, these devices are not as general purpose as medium-grain or fine-grain components, and therefore are not as abundant across the IC.

The core programmable element in these devices is the floating-gate transistor, which is used as both the switching element and the biasing element. These devices enable the FPAAs to be reconfigured many thousands or millions of times. Once programmed, floating-gate transistors maintain their stored charge even when powered off, making these configurations nonvolatile.

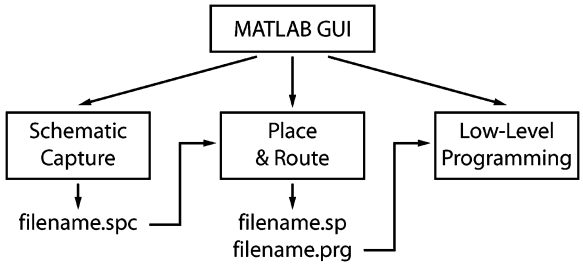

Fig. 5. RASP FPAA design and programming flow diagram.

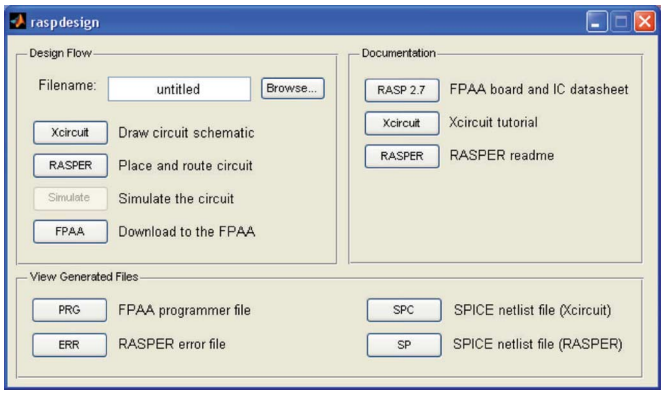

Fig. 6. MATLAB GUI integrating the various tools and software.

C. CAD Tools Design Flow

Students interface with the FPAA setup using a combination of custom, commercial, and open source software tools. As part of the research efforts into FPAAs, low-level programming routines were developed in MATLAB to control the various analog voltages required to program the RASP FPAA. Initial efforts to incorporate these FPAAs into the class laboratory relied upon these low-level routines. Students were given a list of MATLAB commands needed to program the various circuit configurations in the FPAA. However, this approach also required the students to know the detailed inner workings of the FPAA in order to use it. To improve the situation for the next course using the FPAA development platform, a new CAD tool design flow was created, as depicted in Fig. 5.

A simple graphical user interface (GUI), as depicted in Fig. 6, was generated in MATLAB to create a centralized location for accessing the various tools, since the underlying programming routines were already written in MATLAB. The entire design and programming flow is encapsulated by the action buttons along the left side of the GUI. The right bank of buttons provides access to the documentation for each tool in the design flow, and the bottom group of buttons display the various files generated by the tools during the design flow.

Students begin a new design by entering a file name and clicking on the first button, which launches the Xcircuit [9] open source schematic capture tool, as illustrated in Fig. 7. With this tool and a custom library containing the particular analog components included in the RASP 2.7, the students draw a schematic representation of the circuit to be synthesized and tested. Once the schematic is complete, Xcircuit is then used to generate a SPICE netlist for the circuit.

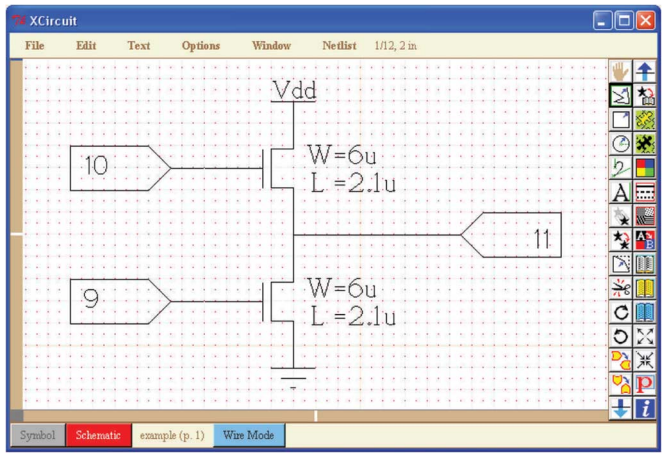

Fig. 7. Xcircuit open source schematic capture tool.

Clicking the second button on the GUI launches the RASPER place and route tool [10]. The RASPER routing tool uses the SPICE netlist generated by Xcircuit to map the drawn circuit to the analog components within the RASP. This process generates two additional output files, as seen in Fig. 5. The first file is a postsynthesis SPICE netlist, which includes the parasitics of the routing used to implement the circuit. A simulation tool for examining the effects of the routing on the circuit can be accessed by pressing the third button on the GUI. The second file is an ASCII configuration file used to program the FPAA. By pressing the fourth button on the GUI, the low-level MATLAB routines are invoked and convert the configuration file into the sequence of programming steps needed to synthesize the circuit in the FPAA hardware.

Once the FPAA is configured, the students write automated testing scripts using a handful of data acquisition commands that are given to them in the laboratory exercise documentation. These commands access the low-level MATLAB routines, which interface with the FPAA board through the FPGA board, to write values to the DACs and read values from the ADCs. Although these measurements are performed outside the GUI, the students have very little trouble using these MATLAB routines, since there are only a few, and they do not require detailed knowledge of the exact hardware being used to perform the measurements.

III. Courses at Georgia Tech

The FPAA development platform has currently been incorporated into two courses at Georgia Tech. The first is an undergraduate course in analog integrated circuit design and test, ECE4430 [11]. The second, ECE6435 [12], is a graduate course entitled “Neuromorphic Analog VLSI.” For both classes, the FPAA hardware was used to replace numerous prefabricated test ICs that required careful setup for each week’s laboratory exercise.

A. ECE4430

The first laboratory exercise in ECE4430 involved MOSFET characterization measurements and EKV model parameter extraction. For this and the next few laboratory exercises, the circuits to be tested were preconfigured in the FPAA by

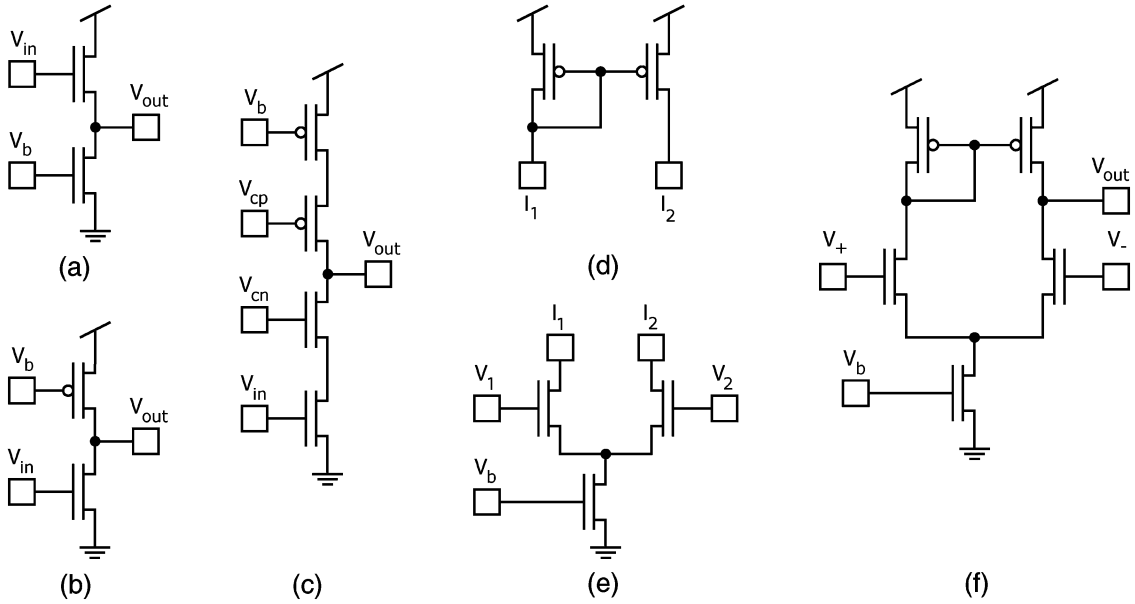

Fig. 8. Progression of circuits synthesized and tested using the FPAA platform. (a) Source follower . (b) High-gain amplifier. (c) Cascoded high-gain amplifier. (d) Current mirror. (e) Differential pair. (f) Operational transconductance amplifier.

the teaching assistants. The students were given exercise instructions that included pin numbers relating to the compiled MOSFET configurations, and a diagram illustrating the various measurement ICs on the FPAA board. Using wire jumpers, the students were required to connect the various MOSFET terminals to the DAC and ammeter pins needed to perform the characterization sweeps. From these sweeps, the students extracted the parameters necessary for a very basic EKV model, which they used to simulate the same characterization sweeps. The simulation results were then compared to the measured sweeps and adjustments to the models were made as necessary.

The next set of initial laboratory exercises in ECE4430 followed the progression illustrated in Fig. 8, beginning with simple amplifier structures. Several single transistor amplifier structures, such as Fig. 8(a) and (b), were presynthesized and programmed into the FPAA for the students before the laboratory session. Again, the laboratory handouts included the FPAA pin numbers corresponding to the various circuit terminals, which the students had to connect to the appropriate stimulus or measurement pins on the FPAA board. Using the onboard DACs and ADCs, the students biased the circuits based upon their MOSFET characterization data and performed dc measurement sweeps. For step response measurements, the students again used a DAC on the FPAA board as the stimulus, but measured the output waveform using an oscilloscope. Although this platform has the capability to perform such measurements on the FPAA board, the oscilloscopes were used to reinforce the students' experience using standard instrumentation equipment.

The students then progress through various other circuit topologies, such as cascodes, current mirrors, and differential pairs, as depicted in Fig. 8(c)–(e). At this point, the students were introduced to the concept of reconfigurable and programmable analog devices, such as the FPAA. A laboratory exercise utilizing the FPAA CAD tools was then completed by the students to familiarize them with the design flow. Being the core programmable component in the RASP FPAA, a laboratory exercise involving floating gate transistors was also inserted at this point to expose the students to more advanced semiconductor phenomena, such as hot carrier transport and Fowler–Nordheim tunneling. Using these two mechanisms, the students programmed the amount of charge stored on the floating gate transistors.

The students then start configuring the FPAA themselves to put the circuits characterized earlier in the term together to form simple operational amplifiers, such as the five transistor operational transconductance amplifier (OTA) of Fig. 8(f). For the remainder of the scheduled laboratory exercises, the students investigate various amplifier topologies using a combination of the FPAA and simulation. As a final team project, the students were given a set of amplifier specifications for which they were to design, simulate, and generate layout for the amplifier topology of their choice. At the end of the course, the students gave formal presentations of their results in the form of a slideshow.

B. ECE6435

The ECE6435 course performed the same initial set of laboratory measurements as the ECE4430 course, including the programming of floating gate transistors. However, this course diverged after the introduction of the OTA to pursue biologically inspired circuits and systems. Course topics included continuous-time low-pass and bandpass $\mathrm{G}_m - \mathrm{C}$ filters, reference circuits, diffusor networks, winner-take-all (WTA) circuits, and integrate-and-fire neurons, which were all synthesized using the OTAs, transistors, and capacitors within the FPAA. For example, the integrate-and-fire neuron circuit is depicted in Fig. 9 and utilizes a combination of medium- and fine-grain CAB components.

The students of this course were exposed to system-level design through a final group project. Each group was given a

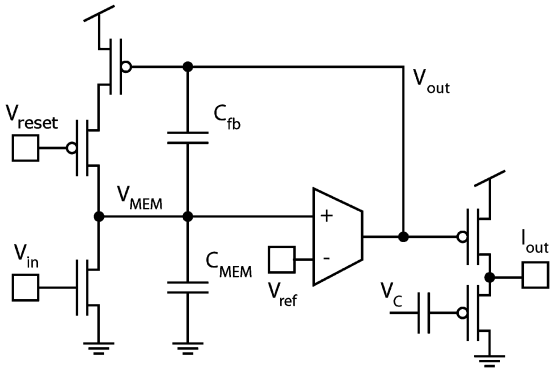

Fig. 9. Integrate-and-fire neuron with synapse output stage.

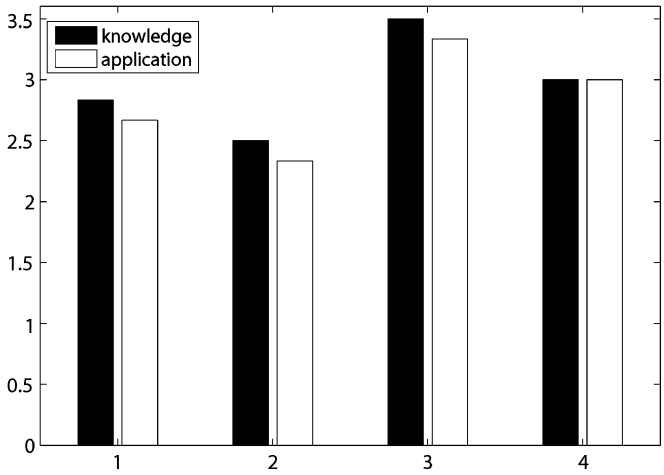

Fig. 10. Postcourse survey data for ECE6435.

unique signal processing task for which to design and implement an analog solution using the FPAA development platform. Projects included basic imaging using CMOS photodetectors synthesized within the FPAA, dendritic processing and classification using diffusors and winner-take-all circuits, and cochlear modeling using logarithmically-spaced band-pass filters and peak detectors. At the completion of these projects, the students presented their designs and experimental results as part of an oral presentation.

IV. Workshops

In addition to traditional analog design classes, several accelerated short courses have been taught using this FPAA development platform. These workshops ranged from just a few days to nearly three weeks. Two of these short courses were held at the three week Telluride Neuromorphic Engineering Workshop during the summers of 2005 and 2006. The first course focused on basic analog VLSI, similar to the material covered in ECE4430 at Georgia Tech, but at a very accelerated pace. The second course covered programmable analog floating-gate transistors in much greater detail than was discussed in the courses at Georgia Tech. Each course included lectures, laboratory exercises, and independent projects. Participants included graduate students and instructors from various disciplines including engineering and biology. By the end of the workshop, the participants were actively implementing their own circuits and systems based upon their own research interests.

A short four-day, 20-hour workshop was also held at Georgia Tech to investigate teaching higher-level reconfigurable and programmable circuits concepts. Participants in this workshop were from various areas of engineering, including industry marketing people. Given several example circuit topologies and a basic understanding of how the biases for these circuits can be adjusted in the CAD tools, the participants were able to design and synthesize first- and second-order filters without an in-depth understanding of the underlying programmable analog technology. Thus, this workshop demonstrated that an FPAA platform like that described here can be used by individuals at a higher level, even if only in limited ways. With a CAD library of signal processing components, users can synthesize low-power analog circuits and systems for certain applications.

V. Assessment and Observations

A postcourse survey was administered to assess the students' perceived effectiveness of the laboratory exercises using the FPAA platform in ECE6435. The students were asked to evaluate how much the laboratory experience has contributed to their knowledge and their ability to apply that knowledge to the following items on a scale from 0 (none) to 4 (high):

1) custom analog IC design;
2) custom analog IC testing;
3) analog design with a reconfigurable IC;
4) testing a reconfigurable analog design.

The averaged results of this survey are shown in Fig. 10 for the students who returned the survey out of a class of 20 students. The low number of responses has been attributed to waiting too long before distributing the survey, since most of the students were away on break at that time. As seen in Fig. 10, the students seem to believe that the laboratory experience has significantly contributed to their understanding of analog design and test. The slightly higher value for reconfigurable analog devices also seems reasonable, since this is likely the first time that these students had been exposed to the idea of reconfigurable analog.

Midcourse surveys were also administered to the fall 2007 ECE4823 students. This course focuses on nonlinear dynamics in analog circuits and has used the FPAA platform for both laboratory exercises and in-class live demonstrations. The students were asked to evaluate what they perceived to be the effectiveness of the FPAA platform as it pertained to several aspects of the course. When asked if the in-class demonstrations improved their understanding of the course content, 67% believed it did, while 17% were neutral, and 16% believed it did not. The same distribution was observed when the students were asked if the in-class demonstrations helped them to perform their laboratory exercises. When asked if the laboratory exercises improved their understanding, 83% of the course responded affirmatively, while 17% were neutral. The final question attempted to ascertain the potential of the FPAA platform to improve student understanding by asking the students if they believed the ability to synthesize and measure any arbitrary circuit example would be helpful, to which the response was a 100% "yes."

The incorporation of the FPAA development platform enabled the students to synthesize and test hardware for their final projects, instead of just performing design and simulation. For the first time students were able to complete their designs and simulations while simultaneously testing their results in physical hardware. The postcourse survey also asked the students if they believed the ability to synthesize and test hardware for their final project was beneficial. The nearly unanimous answer was "maybe." The instructors expected a higher number of students to find the hands-on final project more beneficial than just simulation, but it is believed that some of the technical difficulties experienced with the development platform may have clouded their perceptions on this issue. The student response to the more hands-on final project is expected to improve as these difficulties are resolved, which seems to be indicated by the improved response on the more recent midcourse survey in ECE4823.

The benefits of incorporating the FPAA platform was much more noticeable for the instructors. The new laboratory station consolidated the setup requirements for multiple courses. The generality of the reconfigurable IC eliminated the need for multiple ICs for different courses or different aspects of a single course. Additionally, the time required to set up and test each week's laboratory exercise was significantly decreased and will continue to decrease as the technology is refined.

VI. Conclusion and Future Plans

The incorporation of the FPAA platform into an educational setting proved to be relatively successful for both students and instructors. For most laboratory exercises, the student results of using the reconfigurable IC were not significantly different from using custom ICs. However, the platform did provide the students with an opportunity to synthesize and test their final projects in the course, which had not been possible previously. Student perceptions indicate that they do see a benefit in the hands-on experience gained through the laboratory exercises. However, the benefit of using the FPAA platform over custom ICs was not conclusive based upon student feedback. The FPAA platform was shown to be beneficial to the course instructors in that it consolidated the laboratory setup for multiple courses and reduced the amount of time needed to prepare each week's exercise.

In the upcoming academic term, the FPAA platform will be incorporated into another graduate level course at Georgia Tech and will continue to be used for the courses described in this paper. In addition, the number of institutions adopting this technology will also be expanded as two new locations will be using these devices in course laboratories in spring 2008. As part of this expansion, the assessment of this technology's affect on the course will also be improved by including precourse, midcourse, and postcourse surveys to track the students' perceptions as these evolve throughout the term.

References

[1] R. H. Caverly, "A laboratory-oriented paradigm for undergraduate analog CMOS microsystems design and education," in *Proc. University, Government, and Industry Microelectronics Symp.*, Research Triangle Park, NC, May 1993, pp. 193–196.

[2] T. A. Hudson, E. R. Doering, and G. Lee-Thomas, "Teaching mixed—Signal integrated circuit design to undergraduates," *IEEE Trans. Educ.*, vol. 49, no. 1, pp. 39–48, Feb. 2006.

[3] A. F. Mondragon-Torres, T. Mayhugh, J. P. de Gyvez, J. Silva-Martinez, and E. Sanchez-Sinencio, "An analog integrated circuit design laboratory," in *Proc. IEEE Int. Conf. Microelectronics Systems Education*, Anaheim, CA, Jun. 2003, pp. 91–92.

[4] Anadigm Product Range Anadigm [Online]. Available: http://www.anadigm.com/products.asp

[5] PSoC Mixed-Signal Arrays Cypress Semiconductor Corporation [Online]. Available: http://www.cypress.com/index.html

[6] C. M. Twigg and P. Hasler, "A large-scale reconfigurable analog signal processor (RASP) IC," in *Proc. IEEE Custom Integrated Circuits Conf.*, San Jose, CA, 2006, pp. 5–8.

[7] Development Kits Altera Corporation [Online]. Available: http://www.altera.com/products/devkits/kit-dev_platforms.jsp

[8] Development Boards and Kits Xilinx Inc. [Online]. Available: http://www.xilinx.com/products/devboards

[9] XCircuit Tim Edwards [Online]. Available: http://opencircuitdesign.com/xcircuit

[10] F. Baskaya, S. Reddy, S. K. Lim, and D. V. Anderson, "Placement for large-scale floating-gate field programable analog arrays," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 8, pp. 906–910, Aug. 2006.

[11] Analog Integrated Circuit Design (ECE4430), Georgia Institute of Technology, Atlanta, GA [Online]. Available: http://users.ece.gatech.edu/phasler/Courses/ECE4430

[12] Neuromorphic Analog VLSI (ECE6435), Georgia Institute of Technology, Atlanta, GA [Online]. Available: http://users.ece.gatech.edu/phasler/Courses/AVLSI

Christopher M. Twigg (S'97–M'05) received the B.S. degree from West Virginia University, Morgantown, and the M.S. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, all in electrical and computer engineering, in 2000, 2002, and 2006, respectively.

He is currently an Assistant Professor at Binghamton University, Binghampton, NY. Previously, he was a postdoctoral Researcher at the Georgia Institute of Technology. His research interests include reconfigurable and programmable mixed-signal devices, embedded systems, cooperative analog/digital signal processing, floating-gate circuits, and neuromorphic applications of reconfigurable circuits.

Paul E. Hasler (S'87–M'95–SM'04) received the M.S. and B.S.E. degrees in electrical engineering from Arizona State University, Tempe, and the Ph.D. degree from the California Institute of Technology, Pasadena, in computation and neural systems in 1991 and 1997, respectively.

He is currently an Associate Professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His research interests include low power electronics, mixed-signal system ICs, floating-gate MOS transistors, adaptive information processing systems, "smart" interfaces for sensors, cooperative analog-digital signal processing, device physics related to submicron devices or floating-gate devices, and analog VLSI models of on-chip learning and sensory processing in neurobiology.

Dr. Hasler received the 2001 NSF Career Award, the 2002 ONR YIP Award, the Paul Raphorst Best Paper Award, the 2005 IEEE CICC Best Paper Award, and the 2005 IEEE ISCAS Sensors Best Paper Award.