FPAA chips and tools as the center of an Design-Based Analog Systems Education

Paul Hasler, Craig Schlottmann, Scott Koziol, Shubha Ramakrishnan, Steven Brink, and Arindam Basu
Georgia Institute of Technology, Atlanta, GA 30328

***Abstract*— Abstract Field Programmable Analog Arrays (FPAA) are continuing to prove useful in educational environments. This paper describes the positive effects that improved FPAA hardware and software are having on classes. Generally, it has changed the pace and material that is covered in class and has provided a huge leap forward from the classical lab/projects environment when we first tried using the FPAA devices.**

The ever expanding research and commercialization activity around Large-Scale Field Programmable Analog Arrays (FPAA) asks the question whether we can broadly use these technologies in an educational environment. In fact, the research advances have been developed side by side with educational development, similar to what we saw for the early stages of digital VLSI design [1]. Previously, we demonstrated that FPAAs could be used in a classical classroom laboratory setting where students could perform lab experiments given to them, and they are not just a topic for the academic research laboratory [2]. Over the last four years, we have made progress in these devices, which has enabled wider use of them as well as enabling system / signal-processing level projects to be attempted. These improvements include revamping one graduate course, focused theoretical on analog VLSI / system design, into a design based analog systems course where the students can experimentally measure the results of their design.

We see these advances enabling classes that previously would have been only theoretical or would have required a laboratory setup, into classes that introduce design concepts into the approach. We enable these techniques using a highly portable, USB powered, self-contained chip / board / software , small enough to use with laptops and bring in class, flexible enough to use for multiple classes, and has a potential path to be the cost of a typical engineering textbook. These approaches might are possible for FPGA based design; our approach is not only considering analog design, but system-level analog design which is still an open field of research.

I. Overview of FPAA ICs

Fig. 1 shows the block diagram and resulting die photo of the FPAA we will use for these discussions [3]. The reconfigurable nature of the platforms allow rapid building and testing of different circuit configurations. Many other FPAA devices have been developed [4], [5], and many others are currently in review, but we focus on this FPAA device because it has been available for the last three years and has general resources for educational directions. Programmable floating-gate circuit technology enables the FPAAs to provide

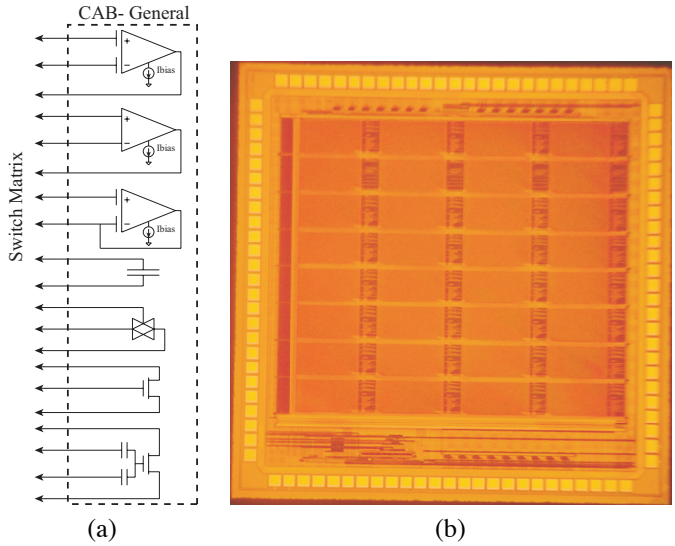

Fig. 1. The Large-Scale Field Programmable Analog Array IC used for the educational development. (a) Block diagram of the CAB components. The IC utilizes 32 CABs in a multi-level routing scheme. (b) Die Photo of the IC which consumes 3mm x 3mm area in 0.35μm CMOS process.

area-efficient, accurately programmable analog circuitry [7]. Further, because of the analog programmability, the switch elements have a dual role as computational elements [3], [6]. FPAA devices were invented and have been developed as part of the educational and research environment, similar to the original growth of the digital VLSI community, and as such, we want to continue technological innovation from both the research and educational communities; several other technologies (i.e. FPGAs) do not have such grounding.

We realized the strong need to vastly increase the number FPAA devices in multiple classes over multiple years. We utilized industrial methods and practice of designing custom mask sets, as opposed to typical MEP or MPW type fabrication runs, which allows for 1000's of chips from a single batch of wafers, as well as a cost effective approach to get more devices when needed. In typical IC processes, the masks used for the fabrication are the largest part of the cost to make a single chip; once we have the masks, wafer fabrication, slicing, and plastic packaging of parts is only a small fraction of the cost. This FPAA chip was the first of many devices designed on a wafer fabrication through a 350nm IC CMOS process. This capability enables moving from initial educational experiments to wider deployment of FPAA technology. Further, these

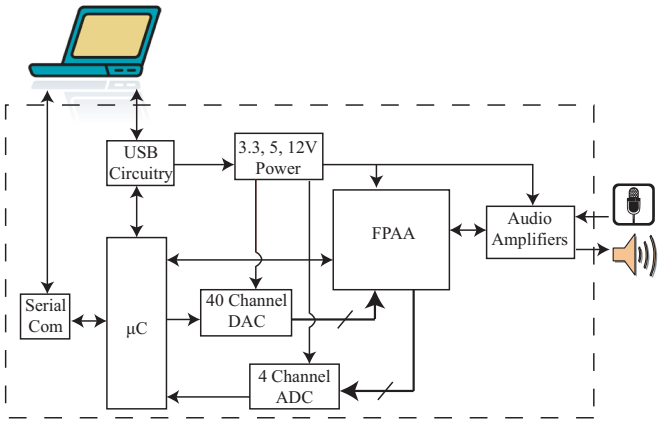

Fig. 2. Overview of the FPAA board hardware used in the educational classes. The board consists of a USB connection, used for both communication, data, and power, a 40 pin DIP microcontroller module, as well as other components used for educational tests as well as controlling the FPAA board. The FPAA Board (25.76 square inches) uses a 100 pin ZIF socket for inserting the FPAA ICs; many 2x4 pin headers connected to FPAA I/O, DAC outputs, ADC inputs, FPAA control pins, and power/ground; 4 SMA for FPAA I/O interface; and stereo audio jacks (on the lower right). The Adaptor Board (4.16 square inches) is a custom PCB for on a quad flat pack (QFP) packaged FPAA IC on one side and pins on the other side. The pins plug into the 100 pin ZIF socket on the programming and control board. The adaptor board concept allows a range of FPAA chips to be used by a single setup.

approaches show the technical viability for a commercial entity; having a commercial source of FPAA chips would further accelerate the adoption both in educational, research, and commercial settings.

II. FPAA Hardware Infrastructure

The platform enables easy portability of the setup, resulting in a self-contained programming and testing system only requiring a laptop (or other computer) running MATLAB and an FPAA board. It is ideal for educational environments, and low-cost laboratory experiences. The FPAA board design can be decreased in size and cost and that would enable wider use of these approaches in the classroom; the board has many headers to allow an extremely flexible design platform, although most of that flexibility has not been not utilized in educational projects to date. Figure 2 shows a block diagram of the board ($\approx 26in^2$) which includes the FPAA IC, discrete ADC, DAC and amplifier ICs, and a a 32-Bit ARM7 based microcontroller (Atmel AT91SAM7S) for programming, interfacing, measuring, and computing using the FPAA IC [8]. The board hardware communicates and receives power through its USB port. The audio port and amplifiers enable additional testing inputs as well as allowing student projects to connect to MP3 players and headphones. The previous hardware platform fit into an enclosure of a large shoe box and communicated with the computer using ethernet; the development platform included a commercial FPGA development board, a custom FPAA board, and a AC-DC power module [2]. That platform was useful for traditional laboratory type settings, with students coming to a fixed lab at fixed scheduled times with experts who could monitor the students using the setups.

III. FPAA Software Infrastructure

What drives the FPAA devices and hardware infrastructure is a software control system and FPAA development tool environment that enables using a wide range of FPAA chips with the correct corresponding configuration files. The on-board software controls the on-board data converters and supplies, programming code for the floating-gate devices, and low-level testing functions for characterization and computation. The baseline computer (MATLAB) software tools includes the high-level control system to program the floating-gate elements either as switches or specific currents [7], a SPICE to FPAA compiler (GRASPER) [9], and a targeted FPAA visualization tool (RAT). The tool flow is illustrated in Fig. 3. All of these tools are wrapped with a self-documenting GUI system used for creating, targeting, programming, and characterizing an FPAA design.

The high level tool infrastructure is built around a set of Simulink tools for system development. This tool enables the drawing / design of an FPAA circuit or system design, the high-level simulation of this design, and an automation tool which converts Simulink models to a Spice netlist. This allows signal processing / system students to have a fast method of implementing low-power analog solutions without requiring, in some cases, depth in circuit design [10]. Fig. 4 shows the results from design flow for implementing a lowpass filter on an FPAA, including showing the results from compiling the two OTA circuit, result showing the simulation from the compiled spice deck, and result showing the experimental measured data from the FPAA chip.

IV. Analog VLSI Course: 2008-10

We implemented these configurable analog approaches in the analog VLSI circuits course over the last three Fall semesters (2008, 2009, and 2010). The assignments are performed in groups of usually two students, and are submitted using powerpoint slides and a class oral presentation. We use the oral presentation as a method for all students to see various solutions to the recent assignment, where assignments for each group might be significantly different and enable greater breadth of real examples for the students to learn. A typical course flow, that has been used in all three of these courses (with some modification each year) is shown below.

Weeks	Topic / Design Aspect
1 4	Analog / FPAA lab assignments
5, 6	Design of Low- and High-Pass Spatial filters
7, 8	Design of Dynamic Circuits (e.g. IC neuron models)
9, 10	Design of Auditory Front-End Circuits
11,12	Design of Low-Power Classifiers
13-16	Final Design Project

The course has typically had a focus connecting to neuro-inspired circuits, so some of the topics are more slanted to that perspective. This could easily be adapted to different focus

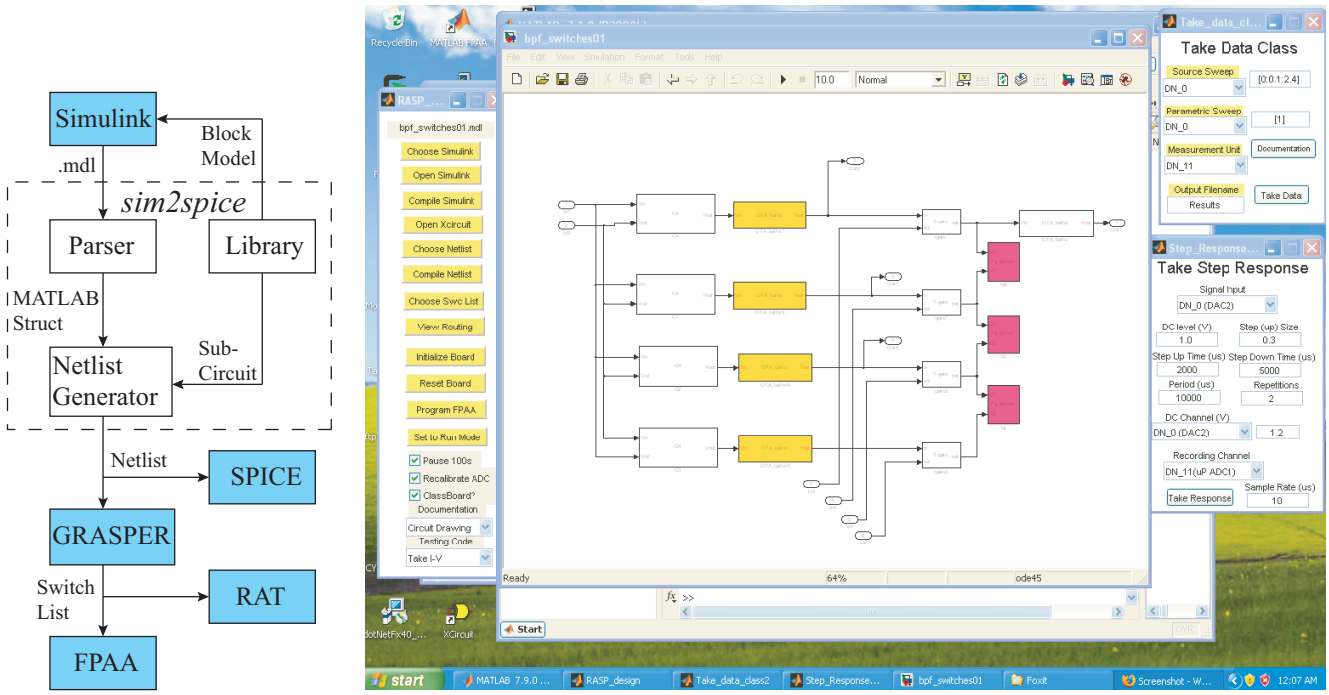

Fig. 3. Tool / software infrastructure for the FPAA setup. (a) Software flow for designing systems on the FPAA. Top level designs are done in Simulink. *Sim2Spice* converts it to a Spice netlist, which can then be compiled into an FPAA switch list. (b) Screen shot of a typical FPAA educational demonstration setup. The setup includes Matlab GUI for selecting software tools and basic measurement tools. Code is available for the students to write their own routines for more complex tasks.

in other classes. The first three weeks are focused on more traditional homework type assignments to get understanding of the FPAA device, as well as understand / review basics of analog IC design. These would include transistor physics (particularly sub V_T), two and three transistor amplifiers, transconductance / voltage amplifiers, floating-gate circuits, and first-order G_m-C filters. We have seen that having at least 4 weeks is essential for the initial approaches; although more weeks would be preferred, the time constraints for a single semester course keeps this aspect to 4 weeks. During these initial weeks, we heavily integrate the lecture time with the lab time, often lecturing for 30 minutes, having the students set up an experiment for 15 minutes using their laptop and boards we brought into class and measure data, and then continue the discussion.

Differences in the available tools had significant differences in the pace and material covered in the class; The heavy use of oral presentations in the class makes the instructors very aware of areas not learned, as well as involvement in lab sessions. For the 2008 course, the hardware development boards had been working for four months, we could reliably program the FPAA boards from a given switch list, but compilation from SPICE was not reliable until the very end of the semester. These constraints kept most of the projects to sizes no more than 5-8 FPAA IC components, and very low utilization of switch fabric elements. For the 2009 course, in addition to another 8 months of technical use and maturity in the tools, compilation from SPICE became very reliable, but the Simulink tools, integrated GUI elements or documentation were not available. When they became available, the resulting improvement allowed for designs using between 10-30 FPAA IC components, and improved switch fabric utilization.

These approaches are a huge leap forward from the classical lab projects / environment when we first tried using FPAA devices. A project that was considered too hard for the entire class can now be replicated easily, enabling discussions of system level applications. Even reliable programming of the FPAA devices was not certain until the 2008 class. For the 2010 course, we had the full set of FPAA hardware and software tools working, based on the approach we demonstrated earlier in the year [8], The resulting improvements allowed for designs only limited in FPAA IC component sizes, and focused the entire effort on effectively communicating the technical material. The inherent self-documenting feature of the simulink tool allowed students and instructors to quickly and easily understand how circuits were configured and debug where necessary. Students using GUI approaches created a lower barrier for students getting measurements from the board, and also gave them the ability to write custom code when necessary. We still had some issues related to mismatches caused by indirect floating-gate programming, which has been recently solved in an automatic way in the research environment. When collaborators have taken the course / workshop at a particular level of complexity, they almost

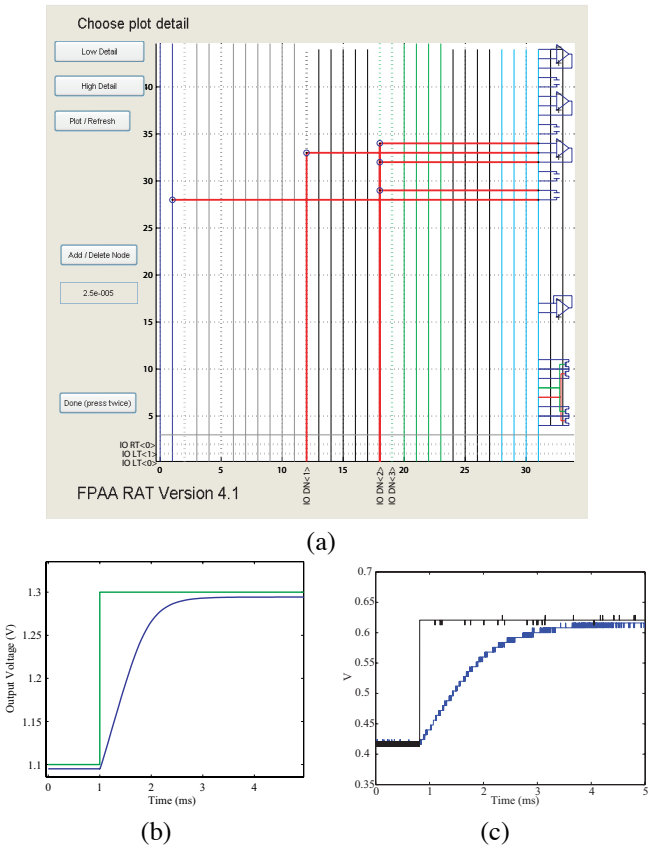

Fig. 4. Design Flow for a low pass filter. (a) Simulink level diagram and resulting compiled circuit. We view the resulting compiled circuit using the RAT viewing tool. (b) Spice level simulation of the resulting circuit. Part of the compilation process is generating a SPICE netlist that is valid for simulation. (c) Measured Results from FPAA IC, including the the input signal step and the lowpass filterd output.

always stay using that level of tools, even when more advanced tools are possible. We have listed some of the final projects that were successfully built in the final project period in the table below.

Final Project Title
Subbanding auditory processor for sound classification
Localization of auditory sounds using binural models
Wordspotting classification using cortical models
Spiking Winner Take All (WTA) Network built from neuron elements

Currently, we have collaborators at over 10 institutions that have FPAA boards (all built at GT) who are working to introduce them into their courses or research efforts. We are looking to using the results of the FPAA based approach in this graduate course towards implementing the approach for senior / junior level courses. Given the Simulink and related tool set, we see these techniques having applicability in circuits, signal processing, and embedded systems courses.

The biggest bottleneck of the 2010 class was understanding analog signal processing concepts at a level that they can get working experimental systems. This issue is not surprising, if the class works well, because the framework for analog signal processing is still a research topic. In fact, most universities teach analog design entirely as a bottom-up approach, only reaching system concepts in graduate level courses. Much like the early days of digital design, this approach is necessary to build the first systems, but analog design remains at this conceptual level; digital approaches made the transition to top-down approaches in the VLSI revolution in the mid 1970s.

TABLE I
SUMMARY OF FALL 2010 SUMMARY QUESTIONS. SCORES OUT OF 10 POSSIBLE POINTS, 10 BEING MOST FAVORABLE RATING

Question	Ave
In-class demo improve subject understanding?	9.2
Lab time improve subject understanding?	9.0
In-class Demo help complete lab?	7.8
Documentation Helpful?	6.8
Having FPAA test board useful?	9.2
Personally Purchase FPAA board?	9.8
How confident on Analog system design before?	5.4
How confident on Analog system design after?	8.0
Was synthesize / test final project beneficial?	9.0

Table II shows the table of questions we asked the students at the end of the class related to our use of FPAA chips, boards, and tools for the course in 2010, as well as evaluating how they saw improvement from the start of the course.

The largest logistic question going forward to widening these configurable ICs and systems for courses is making these boards easily available to students. Ideally, we would want a board that could be purchased by the students, and would cost the students at a level similar to a high-end textbook (e.g. $150 level). From the data in Table 1, it is clear that the students in 2010 class would overwhelmingly purchase such a board. Given current results, such a class board seems possible.

REFERENCES

[1] C. Mead and L. Conway, *VLSI Design*, Addison Wesley, 1980.

[2] C. Twigg and P. Hasler, "Incorporating Large-Scale FPAAs Into Analog Design and Test Courses," *IEEE Transactions on Education*, Vol. 51, No. 3, pp. 319-324, August 2008.

[3] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. M. Twigg, and P. Hasler, "Floating-gate based Field Programmable Analog Array," *IEEE Journal of Solid State Circuits*, September 2010.

[4] A. Basu, S. Ramakrishnan, C. Petre, S. Koziol, S. Brink and P. E. Hasler, "Neural Dynamics in Reconfigurable Silicon," *IEEE transactions on Biological Circuits and Systems*, September 2010.

[5] C. Schottmann, D. Abramson, and P. Hasler, "A MITE-based translinear FPAA," *IEEE Transactions on VLSI*, December 2010.

[6] C. Twigg, J. Gray, and P. Hasler, "Programmable oating-gate FPAA switches are not dead weight," *International Symposium on Circuits and Systems*, May 2007, pp. 169-72.

[7] Arindam Basu and Paul E. Hasler, A Fully Integrated Architecture for Fast and Accurate Programming of Floating Gates over Six decades of Current, *IEEE Transactions on VLSI*, June 2010.

[8] S. Koziol, C. Schlottmann, A. Basu, S. Brink, C. Petre, S. Ramakrishnan, P. Hasler, "Hardware and Software infrastructure for a Family of Floating-Gate FPAAs," IEEE International Symposium on Circuits and Systems 2010. Winner of the best demonstration paper award.

[9] F. Baskaya, D.V. Anderson, P. Hasler, S. K. Lim, A generic reconfigurable array specication and programming environment (GRASPER), European Conference on Circuit Theory and Design, August 2009. pp. 619 - 622.

[10] C. R. Schlottmann, C. Petre, and P. E. Hasler, "A high-level Simulink-based tool for FPAA configuration," *IEEE Transactions on VLSI*, December 2010.