Field–Programmable Analog Arrays Enable Mixed–Signal Prototyping of Embedded Systems

(Invited Paper)

Tyson Hall
School of Computing
Southern Adventist University
Collegedale, Tennessee 37315–0370
Email: tyson@southern.edu

Christopher Twigg
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: ctwigg@ece.gatech.edu

Abstract— **The field of embedded systems is an increasingly popular area of focus for industry and academia alike. In typical embedded systems, analog circuitry is often limited to the input and output sensors with analog–to–digital (A/D) converters and digital–to–analog (D/A) converters placed as close to the sensors as possible leaving the majority of the system completely digital. Although this is understandable given the reprogrammable nature of digital technologies and ease of design, it is unfortunate since many embedded systems are battery powered and could benefit from low–power analog circuitry. To increase the influence of analog circuitry into mainstream embedded system design, reprogrammable analog devices, dubbed field–programmable analog arrays (FPAAs), need to be integrated with reprogrammable digital hardware to form a mixed–signal rapid prototyping platform. Such a system will enable embedded system designers to develop and evaluate mixed–signal systems with greater ease and encourage the use of more analog computational hardware when it is deemed best for the overall system.**

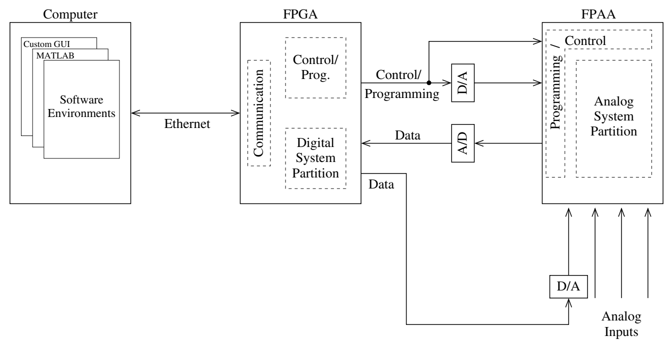

Fig. 1. This is a block diagram of the overall mixed–signal development system. During programming mode, the FPGA receives the high–level commands from the PC and implements the specific programming protocols to configure the FPAA. In run mode, the FPGA is configured with the digital partition of the embedded system and interfaces with the FPAA via the A/D and D/A converters to implement the entire mixed–signal system.

I. FPAAs in Embedded Systems Design

Field–programmable analog arrays (FPAAs) have been of interest for some time, but historically, these devices have had very few programmable elements and limited interconnect capabilities, making them limited in their usefulness and versatility. Currently available commercial and academic FPAAs are typically based on op–amp circuits with only relatively few op–amps per chip [1]–[7]—see [8] for a more exhaustive discussion of previous FPAA designs. However, to be of interest to embedded system designers, FPAAs need to be larger, more flexible, and easier to use; they need to be more analogous to today's high–density field–programmable gate arrays (FPGAs). A large–scale FPAA and modern FPGA can be combined to form a very useful rapid prototyping system. Commercial FPGAs are clearly capable of meeting the demands of prototyping embedded systems. Therefore, the focus of this paper will be on the FPAA side of a mixed–signal prototyping platform for embedded systems.

Work on a class of new, large–scale FPAAs based on analog floating–gate technologies has made good progress towards a modern analog prototyping platform [9]–[11]. The most recent experimental data from these FPAAs illustrates their usability and flexibility [12]. With a working architecture documented, a discussion of these FPAAs can now focus on their potential functionality in various design spaces and the architectural customizations necesssary for targeting these FPAAs to different design modalities such as audio, RF, bioengineering, neuromorphic VLSI, embedded systems, etc. For this paper, the area of embedded systems has been selected. In the following sections, a mixed–signal rapid prototyping platform for embedded systems will be introduced, an FPAA archetecture targeted at embedded systems applications will be discussed, and a variety of example systems will be explored in terms of their prototyping potential on such a system.

II. Rapid Prototyping of Embedded Systems

Embedded system designs typically involve software, digital hardware, and analog hardware. The analog hardware is typically limited to analog sensors on the input and/or output. Embedded systems most often use a microprocessor—either fixed–core or in FPGA logic—as their primary digital platform. Senor inputs and outputs are interfaced to the main controller with A/D and D/A converters. The current trend in sensor development is to integrate a converter into the sensor module itself and provide a digital signal at the output of the sensor. While this integration is popular among digital designers, it eliminates the possible advantages of performing analog computations before digitizing the signal. For example, experimental data from analog signal processing systems has shown that power requirements can be decreased up to five orders of magnitude over typical DSP microprocessor implementations [10], [13], [14]. With many embedded systems operating on battery power, any amount of power savings (even far less than five orders of magnitude) is significant.

To provide a prototyping environment for embedded systems, reconfigurable analog and digital hardware must be incorporated into a single platform as shown in Fig. 1. Modern FPGAs are a natural fit for synthesizing the digital logic and even soft–core microprocessors. The large–scale FPAAs proposed in this paper will complete the platform by providing reconfigurable analog prototyping capabilities.

III. Adapting the FPAA Architecture

Fundamentally, FPAAs include two functions: routing and computation. The routing elements are typically networks of switches connected together by signal lines with the network architecture and switch types varying dramatically across different FPAAs. The switch networks connect to the computational elements, which are grouped together to form a computational analog block (CAB) that is analagous to the computational logic blocks found in FPGAs. For the FPAAs discussed here, the switches are floating–gate transistors that can be programmed to an "on" state, "off" state, and variable resistance state [9]. The CABs in these FPAAs are comprised of a collection of analog computational elements that range from fine–grained components (transistors, fixed–value capacitors, etc.) to medium–grained components (operational amplifiers) and coarse–grain components (bandpass filter blocks, vector–matrix multipliers, peak detectors, etc.).

FPAAs targeted at prototyping the analog computational components of embedded systems will have several important characteristics including special–purpose analog I/O blocks, digital interfaces, and specialized computational elements. A top–level block diagram of the proposed embedded systems FPAA is shown in Fig. 2 with all of these functions illustrated.

a) Analog Interfaces: In many embedded system applications, the FPAA will implement the front–end and back–end of the mixed–signal system. In this role, the FPAA needs to have a range of I/O modules to ease the attachment of common devices. In particular, standard line–in and line–out audio interfaces will allow audio sources and amplified speakers to be attached directly to the FPAA. Output driver buffers will improve routing of analog signals between multiple FPAAs (for larger systems) and to measurement instrumentation. On the input side, input buffers with programmable gains will allow a wider range of input signals to be supported by the FPAA.

b) Digital Interfaces: Since FPAAs will be used as a part of a larger mixed–signal system, they need to easily integrate with digital systems such as microprocessors or FPGAs. This integration requires on–chip A/D and D/A converters. To support a wide range of application, multiple converter flavors will need to be available. A high–speed, moderate resolution (16 bits) converter is needed in many

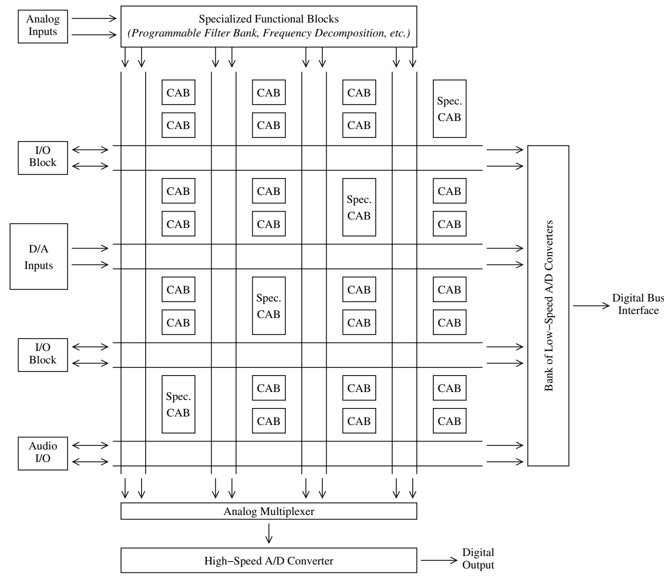

Fig. 2. This is a top–level block diagram of an FPAA architecture targeted at prototyping embedded systems applications. This proposed FPAA is characterized by special I/O blocks for accommodating various analog signal interfaces, data converters for easing integration with digital systems, and specialized CABs for optimizing common functions.

high–bandwidth applications. Alternatively, a large number of compact, low–speed converters are needed for passing large amounts of parallel data from the FPAA to a digital system (high–throughput applications). Providing a bank of compact, integrating A/D converters with approximately eight bits of resolution and operating in the hundreds of kilohertz range should be sufficient for many embedded systems applications.

With the class of FPAAs introduced here, one can imagine implementing programmable filters, frequency decomposition, smoothing (and other signal conditioning), thesholding, peak detection, and more in the FPAA. With the addition of on–chip ADCs, the FPAA can be thought of as a "smart A/D." This term implies the FPAA's extensive computational capability in addition to traditional data conversion functionality. There is another subtlety at play in a smart A/D converter. By adding computational effort in the analog domain, the amount of data that needs to be converted may be drastically reduced, thus requiring a simpler, smaller, slower, lower power A/D converter in the system.

c) Specialized CABs: In addition to the general computational fabric of FPAA CABs [12], specialized CABs with coarse–grained blocks are needed to optimize common tasks such as frequency decomposition, programmable filter banks, and arbitrary waveform generators. By providing coarse–grained elements such as these, common tasks can be performed with improved performance that is more predictable (an important characteristic for the development of CAD tools such as synthesis, place and route, and simulation software). The development of specialized CABs that are distributed across the chip within the the general computational fabric closely parallels the design path that FPGAs have followed.

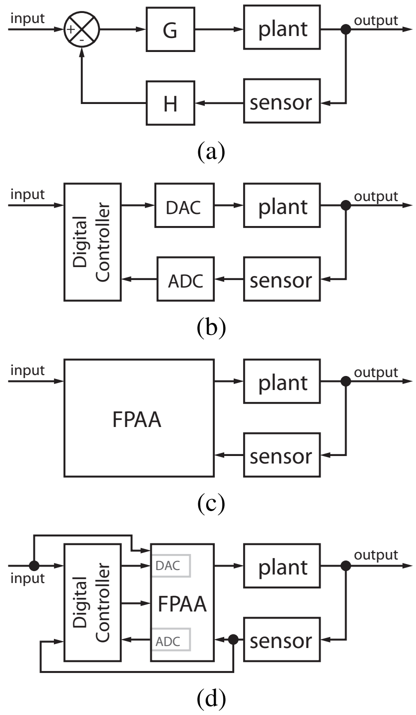

Fig. 3. (a) A typical control system generally involves an error signal driving some system plant through a transfer function, G. The output of the plant is observed by some sensor and processed with another transfer function, H. The error signal is then constructed by subtracting the feedback signal generated by the sensor from the control input. (b) Modern control systems are generally built using digital controllers due to their flexibility and ease of use. Since real world signals are analog, data converters are required. (c) FPAAs provide a more flexible and easier to use analog controller that may appeal to engineers desiring low–power or continuous–time controls. (d) A mixed–signal approach provides an even greater level of flexibility in control systems by enabling low–power, continous–time transfer functions with digital supervisors capable of reconfiguring the analog device as the condition necessitates.

Modern FPGAs have special computational blocks for implementing common functions such as memory, multiplication, and even full processors.

IV. Implementing Mixed–Signal Embedded Systems

One potential application for mixed-signal programmable and reconfigurable devices is control systems. Simple control problems are generally characterized by transfer functions that transform signals going into and coming out of a system plant, Fig. 3a. In these feedback systems, an error signal is fed through one transfer function to drive the plant. A feedback signal is generated by passing sensor signals through another transfer function, and the error signal is constructed by simply subtracting this feedback signal from a control input. Most of the processing involved in these systems is now done in digital controllers, Fig. 3b, such as DSP processors, FPGAs, microcontrollers, etc. Since real–world signals are analog, most of these digital controllers require data converters, which can consume a significant amount of power. However, control systems were not always designed using digital systems.

The earliest control systems were analog, but they were eventually replaced by digital circuits that were more flexible and significantly easier to use. These digital circuits provided a means for rapidly prototyping the control algorithms in hardware in mere minutes. Traditional analog could not compete, since discrete analog parts required physical changes to the setup and integrated analog circuits take months to fabricate. However, FPAAs provide a means for rapidly prototyping analog hardware in a fashion similar to that of FPGAs. If the control input and the feedback mechanism are both analog, then an FPAA can provide continous–time low–power signal processing [12], such as the subtraction and transfer functions required to implement these control systems, Fig. 3c. Without data converters and the power associated with switching elements in digital systems, this analog solution becomes very attractive for battery powered applications including embedded controls.

A truly flexible solution can be achieved by combining both programmable digital and analog technologies, Fig. 3d. With such a device, any control system of this type can be synthesized. A single device can interface with any number of sensors, system plants, or control inputs either digital or analog. With this flexibility, a controls designer can move the analog/digital boundary arbitrarily between an all–digital and all–analog solution. For example, a design can start as a completely digital circuit and be incrementally replaced by analog components as the design matures to lower power consumption. Since the FPAA can handle much of the signal processing, the data converters can potentially be smaller than in the all–digital case, thus further reducing the power requirements. However, a mixed–signal solution trades power consumption for greater flexibility. During times of inactivity or when using an all–analog implementation on this device, the digital circuitry and data converters could be powered down to conserve power.

For this class of mixed–signal controller to be truly useful, it must have some clear applications. One such application is robotics. These flexible mixed–signal controls chips can be integrated onto a robot platform to generate intelligent behaviors through connections between the robot's sensors and motors [15]. Fig. 4 illustrates an example of such a robot. The photodetectors provide an analog signal of light intensity and are angled slightly towards the sides of the robot. Each motor takes an analog input voltage that determines the direction and velocity of the wheel attached to it based upon polarity and magnitude, respectively. By creating positive or negative links between the motors and the photodetectors in the FPAA/FPGA, the robot can be configured to be attracted to or repelled by light sources. For example, if the left photodetector is positively connected to the right wheel, and the right photodetector is negatively connected to the left wheel, the robot will avoid light sources to the right of it and head towards light sources on its left. Under the right conditions, the robot may even orbit a light source while driving counter–clockwise due to the attraction of the left photodetector and the repulsion by the right photodetector.

Another example system using a mixed–signal prototyping platform is feature extraction for speech recognition systems.

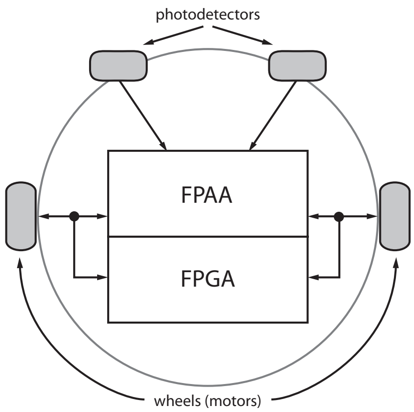

Fig. 4. A typical "Braitenberg machine" is a simple robot with positive and negative connections made between the robot's sensor inputs and its motors. Using a mixed–signal reconfigurable device allows for great flexibility in making these interconnections.

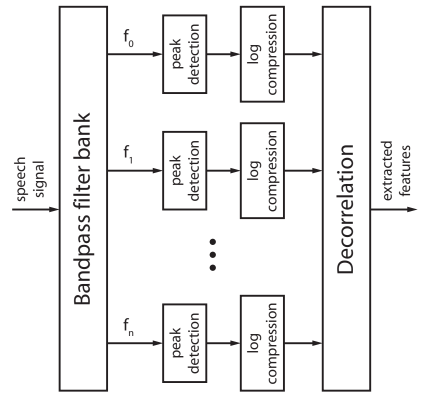

Fig. 5. The analog algorithm depicted here is based on a model for the human cochlea. It begins with a set of bandpass filters (filter bank) acting in parallel to separate the incoming audio signal into its composite frequencies, f_0 - f_n. By passing each of these decomposed frequencies through a peak detector, a power spectrum of the audio signal is generated similar in nature to the output calculated by an FFT. These signals are then log compressed and decorrelated using a DCT / KLT. The features generated by this block can be further processed by the digital system for speech recognition.

Although digital algorithms exist, they tend to be computationally intensive. For mobile devices that use voice commands, this can mean a significant reduction in battery life. A mixed–signal version of a feature extraction algorithm is shown in Fig. 5. Much of the signal processing is moved into the analog domain where the power consumption is significantly lower and the operations can occur in parallel [16]. The remainder of the algorithm is computed digitally. By shifting the analog/digital boundary, longer battery life and faster speech recognition times can be realized.

A number of embedded system examples can be implemented using the mixed–signal prototyping plaform proposed here. In addition to the feature extraction and controls systems mentioned in detail above, one can imagine mixed–signal implementations of noise suppression, speech enhancement and recognition, digitally controlled adaptive analog filters, communication protocol implementations, and more.

V. Conclusion

Large–scale FPAAs based on floating–gate technology provide a viable platform for designing a rapid prototyping environment for embedded systems. This paper has proposed an FPAA architecture based on previously demonstrated designs that is optimized for typcial embedded system applications. Additionally, a number of example systems have been explored that can be implemented on a mixed–signal prototyping platform comprised of the proposed FPAA and a commercial FPGA.

References

[1] M. A. Sivilotti, "Wiring considerations in analog VLSI systems, with application to field-programmable networks (VLSI)," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 1991.

[2] K. Lee and P. Gulak, "A transconductor-based field-programmable analog array," in *IEEE International Solid–State Conference Digest of Technical Papers*, Feb. 1995, pp. 198–199.

[3] D. Anderson, C. Marcjan, D. Bersch, H. Anderson, P. Hu, O. Palusinski, D. Gettman, I. Macbeth, and A. Bratt, "A field programmable analog array and its application," in *Proc. IEEE Custom Integrated Circuits Conference*, May 1997.

[4] *ispPAC Overview*, Lattice Semiconductor Corporation, Hillsboro, OR, Sept. 1999.

[5] X. Quan, S. Embabi, and E. Sanchez-Sinencio, "A current-mode based field programmable analog array architecture for signal processing applications," in *IEEE 1998 Custom Integrated Circuits Conference*, Santa Clara, CA, May 1998, pp. 277–280.

[6] *Totally re-configurable analog circuit–TRAC*, Fast Analog Solutions, Ltd., Oldham, UK, Mar. 1999, issue 2.

[7] C. A. Looby and C. Lyden, "A CMOS continuous–time field programmable analog array," in *Proc. 5th International ACM Symposium on Field–Programmable Gate Arrays*. ACM Press, 1997, pp. 137–141.

[8] T. S. Hall, C. M. Twigg, P. Hasler, and D. V. Anderson, "Developing large–scale field–programmable analog arrays for rapid prototyping," *International Journal of Embedded Systems*, 2004.

[9] T. S. Hall, P. Hasler, and D. V. Anderson, "Field–programmable analog arrays: A floating–gate approach," in *Proc. 12th International Conference on Field Programmable Logic and Applications*, Montpellier, France, Sept. 2002, pp. 424–433.

[10] T. S. Hall, C. M. Twigg, P. Hasler, and D. V. Anderson, "Application performance of elements in a floating–gate FPAA," in *Proceedings of the 2004 IEEE International Symposium on Circuits and Systems*, May 2004, pp. II.589–II.592.

[11] ——, "Developing large–scale field–programmable analog arrays," in *Proc. 18th International Parallel and Distributed Processing Symposium*, Santa Fe, New Mexico, April 2004.

[12] T. S. Hall, C. M. Twigg, J. D. Gray, P. Hasler, and D. V. Anderson, "Large–scale field–programmable analog arrays for analog signal processing," *IEEE Transactions on Circuits and Systems*, 2005, accepted, to be published.

[13] R. Ellis, H. Yoo, D. Graham, P. Hasler, and D. Anderson, "A continuous–time speech enhancement front–end for microphone inputs," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 2, Phoenix, AZ, May 2002, pp. II.728–II.731.

[14] P. D. Smith, M. Kucic, R. Ellis, P. Hasler, and D. V. Anderson, "Mel–frequency cepstrum encoding in analog floating–gate circuitry," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 4, 2002, pp. IV–671 – IV–674.

[15] V. Braitenberg, Ed., *Vehicles: Experiments in Synthetic Psychology*. The MIT Press, 1986.

[16] S. Ravindran, D. Anderson, and M. Slaney, "Low-power audio classification for ubiquitous sensor networks," in *IEEE International Conference on Accoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.