Pour your own

PROGRAMMABLE ANALOG

POURING ANALOG-CIRCUIT DESCRIPTIONS THROUGH AN INTERFACE SOUNDS LIKE A VERSATILE WAY TO FORM FUNCTIONAL BLOCKS. BUT BEFORE I HANG UP MY SOLDERING IRON, I WANTED TO KNOW HOW WELL THEY WORK IN PRACTICE.

MOST INDUSTRIES have their philosophical tugs of war, and if only for sheer entertainment value, perhaps it is fortunate that ours has more than its share. Throughout our industry's history and at nearly every turn, engineers and technologists with strong convictions have alternatively touted the superiority of tubes or transistors, discrete or integrated circuits, reduced or complex processor instruction sets, Harvard or von Neumann processor architectures, CMOS or bipolar circuits, systems on chips or functional segmentation, and certainly not least among these topics, though broad they may be, analog or digital circuits. Though the tensions have fueled many a lively conference panel discussion, taken by themselves, these little battles often suffer the same conceptual defect: They argue an implementation technology ahead of the goal.

Such discussions are not, however, mere displays of opposing chauvinisms entirely devoid of value. They help point out areas in which technologies overlap and in which engineers must choose between contending approaches. The choices are often tangled with trade-offs between competing criteria and technologies that enjoy at best orthogonal advantages. Prioritizing the selection

criteria, evaluating the relative merits of design alternatives, balancing the trade-offs, and optimizing a design for a set of goals are properly left as exercises each designer must work through in the context of an individual project, guarding all the while against confusing the most familiar approach with the best one.

And, as if that weren't enough, the design world is rarely homogeneous; more often than not it's a matter of one *and* the other, not one *or* the other. At minimum, for example, OEMs involved in analog design commonly use shards of control logic in the form of inexpensive microcontrollers or—thanks to mixed-signal integration—small cores or even homebrewed state-machines. More typically, analog circuits serve as the physical interfaces for predominantly digital systems (**Reference 1**).

Rather than representing opposing disciplines, as they are so often miscast, the crafts of analog and digital design have acted synergistically for decades; witness the great success of mixed-signal ASSPs (application-specific standard products). But despite significant strides in mixed-signal design, the analog discipline has not, by and large, evolved to include devices whose functions, like digital devices, are either statically or sequentially determined by programming, not by wiring.

Analog-IC manufacturers have used the term "programmable" in a variety of ways, mostly in place of "adjustable." In such cases, the term is often adjacent to a qualifier, such as "pin" or "resistor." Thus, programmable-gain amplifiers allow you to set the closed-loop gain with a fixed resistor, continuously variable resistor, or a network of fixed resistors and switches. A number of analog-IC manufacturers sell op amps that are bias-current-programmable with respect to gain-bandwidth product, slew rate, and power dissipation. Others offer digitally programmable potentiometers and other R-DAC-based cells whose behaviors vary by a digital word acting most often on a single parameter. These fixed functions, though indeed useful, do not offer the sort of functional programmability you think of when you consider digital constructs any more advanced than simple combinatorial logic.

Illustration by Rob Magiera

AT A GLANCE

▷ The good: You can be up and running in no time. The interface is suitably intuitive, and the development tool includes a rudimentary simulator that can speed circuit design. The parametric ranges for most cells are impressive, and reprogramming either individual parameters or the entire topology on the fly is quick and easy.

▷ The bad: There is no apparent way to simulate the product on or to export simulation data to a standard Spice tool. There is also no way to include components external to the FPAA in its simulation. This situation is similar to that which exists in many of the more sophisticated power ICs and ASSPs whose manufacturers provide limited but useful design-support software.

▷ The ugly: Not much, really. The antialiasing filter on the FPAA's input cells can increase distortion when set near the low-frequency extreme of its tuning range, particularly with large amplitude signals. Don't expect 16-bit noise performance from a 12- to 14-bit product. Be careful to read the evaluation board's manual *before* you apply power to the board.

Only a few companies, most notably Zetex, Lattice Semiconductor, and Anadigm, offer topological programmability within analog ICs (**references 2** and **3**). These devices provide programmable connectivity as well as parametric control of analog-signal-processing resources, and their manufacturers often compare them with FPGAs; indeed, Anadigm refers to its device as an FPAA (field-programmable analog array).

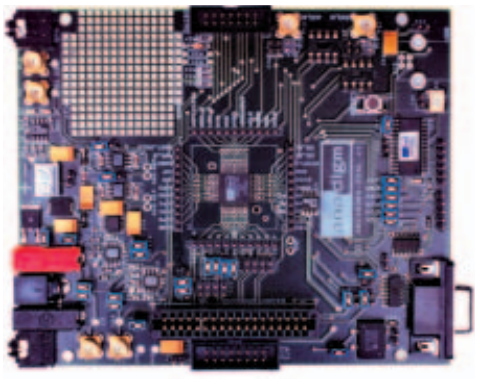

Figure 1 The AN220D04, which measures 4.5×5.375 in., includes a 1.9-sq-in. breadboard area and a host of jumpers, connectors, and pin headers, forming a flexible development environment that's easy to use once you've spent a few minutes with the map.

JUMP(ER)ING IN

Though topologically programmable analog ICs are relatively immature from a market-acceptance perspective, their manufacturers are demonstrating an increasingly sophisticated understanding of the market needs. The evaluation/development package from Anadigm that I recently testdrove is a case in point. The package, which supports Anadigm's AN220E04 FPAA, comes with an evaluation board, AnadigmDesigner2 software, a serial cable, and documentation. Among the documentation is a hard-copy version of the 24-pg AN220D04 evaluation-board user manual, a must-read that also appears with the AN220E04 user manual and the AnadigmDesigner2 user manual in electronic form on the CD-ROM. The AnadigmDesigner2 software itself contains much information about the parts it supports and presents that information through a context-sensitive help facility and help index similar to those that familiar Windows applications provide. Notably, to Anadigm's credit, I was able to find answers to a greater proportion of my questions through the company's various software-based help facilities and in less time that I've experienced with "office"-oriented software packages. Though you could argue that Anadigm's application enjoys a more narrowly focused scope and is therefore easier to support, I found the experience no less refreshing and, from a productivity perspective, no less important.

The AN220D04 evaluation board is festooned with jumpers, connectors, and pin headers—a testament to its versatility (**Figure 1**). The board facilitates numerous signal, configuration control, clock, and power options. It includes a daughterboard connector for applications requiring two and three FPAAs. Two chip applications route their analog signals through the daughterboard only, bypassing the FPAA position on the main board. The evaluation board also provides an extension connector for cascading multiple AN220D04 boards.

You can set up the board's analog inputs and outputs independently for either balanced or unbalanced signals. A differential input attaches either through a pair of SMA connectors or through a set of headers that provide direct access to all of the AN220E04's pins. Two single-ended signals can connect through the SMAs or through a standard $^1/_8$-in. TRS jack, such as you commonly find on sound cards, laptops, and many consumer-audio products. The AN220E04 FPAA's inputs are differential structures, so the evaluation board provides a pair of Analog Devices AD8132 differential amplifiers to convert single-ended input signals. The input path also includes 50Ω terminations that you can connect or disconnect with jumpers J23 and J25. Be sure to check these jumpers as part of your initial setup.

Similarly, the FPAA provides differential outputs that you can use directly, again either through header pins or SMA connectors or through a pair of Analog Devices AD8130 differential receivers, which provide single-ended signals to the SMAs or another $^1/_8$-in. TRS jack. The SMA connectors also provide the signal paths for cascaded boards.

The AN220D04's versatility extends to the evaluation board's control-logic options as well. In its default mode, the board takes configuration data by way of an RS-232 serial link from your PC through the board's DB-9 connector. The onboard 16-MHz oscillator provides the FPAA's analog clock. A freestanding mode makes use of either an SPI EEPROM, such as Atmel's AT25080 8-kbit device, or an FPGA EEPROM, such as Atmel's AT17C65 65-kbit chip. In this mode, the onboard clock drives the configuration process during the turn-on sequence and provides the timebase for the analog clock. If yours is an embedded application development, you can directly drive the digital signals through the DIO (digital-I/O) ribbon connector. Though the current software version does not support USB, the board's USB-capable microcontroller, a Microchip PIC16C745, is wired to an unpopulated connector site. Should Anadigm choose to include USB support in a future software revision, you will need only to solder to a connector and plug in.

Though most of us are in the habit of hooking up a new piece of gear while the documentation is still in the shrink wrap, this board offers so many options that I recommend perusing the user manual and familiarizing yourself with the jumper's functions, locations, and initial states before powering the product. Anadigm provides a full set of board schematics at the end of the book that helped confirm my understanding of the board's architecture. Unfortunately, those schematics are small, and you may find yourself tempted, as I was, to reach for a magnifying glass or low-power microscope. Thanks to Adobe Acrobat's zoom and image-rotation capabilities, the .PDF version of the manual on the CD-ROM rectifies the problem. Thus, with the evaluation board on the lab bench, the schematics on my laptop's display, and the hard-copy manual in my hand open to the introductory narrative, I was able to familiarize myself with the board's functions and settings in just a few minutes.

Most critical to review are the power-supply settings and turn-on behavior. You can provide power from either regulated 5V or unregulated 9V supplies. The 9V power port is a 2.1- mm jack that mates with common wall-wart supplies. You can use an unregulated supply as long as its output voltage is bounded by the board's absolute ratings of 8 and 12V, minimum and maximum, respectively. The 9V power port feeds a 5V linear regulator that drives the rest of the board through a jumper, J28, which you must remove if

you are going to provide regulated 5V directly to either the banana jacks or the screw terminals. Because neither the screw terminals nor the banana jacks provide a mechanically polarized connection, these power inputs connect to the rest of the board through a Schottky diode to prevent reverse-polarity damage. If your application cannot accommodate the 200-mV diode drop, you can add a jumper that shunts the Schottky device.

The three power paths—the output of the linear regulator, the banana jacks, and the screw terminals—join at the Schottky diode's cathode. It is important, therefore, that you connect no more than one port at a time. Also note that, although the documentation states that the 5V regulated input is the default configuration, the board I received had jumper J28 fitted. Had I simply hooked up a 5V supply and flipped the switch, I would have applied power to the output of the linear regulator while its input was at ground. As an Anadigm field-application engineer confirms, this condition could damage the 5V regulator, so be sure that the jumpers are set for your power connection.

When the AN220D04 operates without a daughtercard, its maximum supply current is 350 mA, except at turn-on, when a 1.2A peak current is typical. This turn-on surge is not, as the board's data table suggests, an inrush current. The narrative portion of the manual warns that if the power-supply current limit prevents the supply output from reaching 5V while satisfying the turn-on current demand, the board may power up in a "high current state." With a regulated bench supply, the board reliably achieved this "high current state" every time I energized the supply, as indicated by a sustainable 1.1A on the power supply's current monitor and a comparator that detects the undervoltage condition and drives a board-mounted, red-LED indicator. The recommended way to deal with this ungraceful start-up behavior is to pull the analog power jumper (J5) before turning on the power supply and to fit the jumper back after the supply reaches regulation.

Although this workaround might solve the problem on the bench, it raises power-supply-management questions for your application.

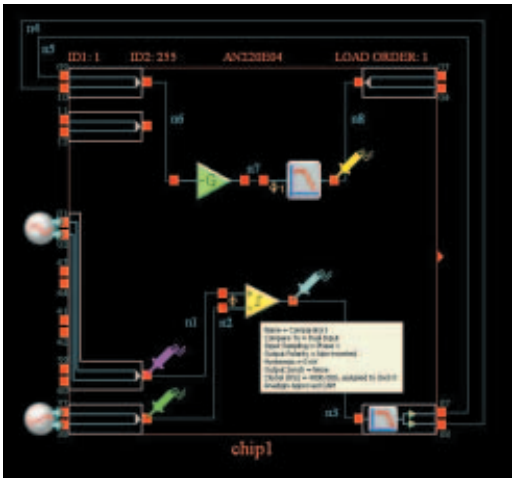

Figure 2 The AnadigmDesigner2 graphical development environment provides a basic drawing tool for configuring the evaluation board's FPAAs. It also provides access to a simulator and support information on each of the analog array's programmable-function blocks.

Unfortunately, the current version of the FPAA's data sheet is a preliminary draft lacking the spec tables, so any insight you might hope to gain into the IC's start-up behavior outside the context of the evaluation board will have to wait until Anadigm releases a complete data sheet. Meanwhile, you need to manage the digital and analog supply sequencing within your application. Also, software revisions notwithstanding, until Anadigm more elegantly addresses this issue than having users pull and fit jumpers, I have doubts about seeing a USB version of the evaluation board, given USB's 500-mA load limit.

UP AND RUNNING

Despite the number of jumpers and connectors you need to consider while setting up the evaluation board for the first time, the process takes remarkably little time or fuss. It took less than an hour to load the software; read through the evaluation board's manual; locate and set the jumpers for my bench setup; connect the control, power, and signal leads; load a basic configuration; and confirm the signal path. Although conceptually none of these tasks is a brain buster, it's always gratifying when a sophisticated product works the first time without a hitch.

The development environment gives a block-diagram view of your circuit. By moving the cursor over any functional block, you can confirm its identity and parametric settings (**Figure 2**). Double-clicking on the block opens a dialogue box wherein you can interactively modify the block's parametric behavior. In addition to providing a graphical programming environment for the FPAA, the development software allows you to connect virtual signal sources and probes to drive the built-in simulation tool (**Figure 3**). The drawing and simulation tools are simple to operate and arranged to make intuitive sense to anyone who has used schematic-capture and Spice software. Keep in mind, however, that, for the entry price of a couple of hundred dollars, you are getting a basic tool without many of the bells, whistles, and functions that come with the five-figures-per-seat EDA suites. So, for example, Anadigm's simulator gives you some insight into the part's behavior in the time domain, but for almost everything else, plan on learning about the part's behavior by examining real signals running through the evaluation board. No apparent hooks allow you to export simulations or behavioral models to your own Spice environment, and no means exist for the native tool to incorporate any external components. This limitation is similar to that which you find with power ICs and ASSPs whose manufacturers provide limited support software but no means to connect to the rest of your EDA environment.

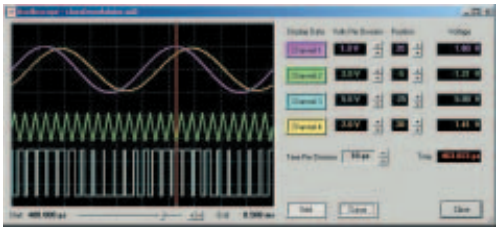

Figure 3 The development environment's simulator provides virtual signal sources and probes and displays the FPAA's time-domain behavior, as this simulation of a rudimentary class-D modulator from Figure 2 shows.

BUILDING BLOCKS

FPAAs will take analog designers some getting used to. Rather than combining diodes, transistors, and passives with functional blocks, such as op amps and

comparators, the FPAA environment is entirely populated by more highly abstracted functional blocks that Anadigm refers to as CAMs (configurable analog modules). (See **Table 1** on the Web version of this article at www.edn.com.) Consider a comparator: The off-the-shelf part you're familiar with is an amplifier whose topology and parametric behavior are appropriate for a nonlinear comparison response rather than linear amplification. However, your application circuit must provide the additional elements to set the reference potential and loop behavior, including hysteresis.

The FPAA's comparator CAM includes these elements. The CAM provides three programmable options for the inverting input: You can refer the comparator's inverting input to any of your analog signals within the FPAA. Alternatively, you can program the CAM to refer the inverting input to ground or to a dc potential whose amplitude you specify as a CAM parameter. For cases in which you would apply the threshold voltage to the comparator's noninverting input, you can invert the output in lieu of reversing the input connections. Programmable hysteresis levels include 0, 10, 20, and 40 mV. The hysteresis function is available when you drive or ground the inverting input, but not when you pro-

FINDING THE BASELINE

Just looking at the antenna farm that surrounds the FPAA, you can tell that the Anadigm AN220D04 evaluation board is built to maximize its flexibility, not to minimize clock noise. It's a reasonable trade. Low-noise layouts are peculiar to their applications and tend to make many nodes difficult to probe. By contrast, as a general-purpose tool, the evaluation board must support a host of external connections for application circuit topologies limited only by your imagination—a layout goal that is not generally sonorous with low-noise optimization. But what overall noise performance you can expect from the FPAA outside the context of the evaluation board is a question that remains unanswered by the FPAA's preliminary data sheet for anyone seriously considering the part.

A bit of accounting suggests that you directly can observe the FPAA's baseline noise. The board's noise terms group into four categories: Clock artifacts—the noise elements most dependent on the pc-board layout—couple capacitively, by radiation, or by conduction through the power and ground systems. The resulting signal disturbances should appear as spectral lines on a noise plot at the clock rates—16, 8, 4, and 0.25 MHz—and at their harmonics. Because the board uses a 16-MHz timebase from which the FPAA derives the other rates, it would be reasonable to expect that the corresponding spectral line would dominate the clock-related noise. The spectrum of interest for this study, however, extended to only about 40 kHz, so the noise plot does not contain clock-related spurs (**Figure A**). The differential amplifiers at the FPAA's inputs and the differential receivers at the FPAA's outputs contribute flatband noise, but their contribution—calculable from their data sheets—is minuscule compared with the noise plot's amplitude. Likewise, resistors on the signal path contribute Johnson noise:

$$e_n = \sqrt{4kTR\Delta_f},$$

where e_n is the resistor's noise voltage, k is Boltzmann's constant, T is the temperature in Kelvin, R is the resistance, and Δ_f is the measurement bandwidth. But, in this case, the resistances are so small that their contribution is smaller than that of the amplifiers.

The fourth item is the FPAA noise, which comprises the active devices' flatband noise, the active devices' 1/f noise, and kT/C noise from the switched capacitors. CMOS processes tend to exhibit 1/f corners near 10 kHz. As the noise plot shows, there is no sign of increasing noise with decreasing frequency, suggesting that the 1/f component is comparatively small or well-suppressed by the discrete-time signal-processing process.

The CMOS-flatband and kT/C terms are left. Though I could not numerically separate the two terms within the scope of this brief study, comparisons with other CMOS products, both discrete- and continuous-time, suggest that the kT/C noise likely dominates. As the name makes clear, this noise source is linearly related to the reciprocal of the capacitance. For a given process, the capacitance is linearly related to area, as area is to cost. The gains, center frequencies, and Q settings are managed by switching capacitor ratios, so the CAMs tend to call upon a few elements from comparatively large capacitor arrays, naturally limiting the area usage for a given set of operating parameters.

The purpose of this exercise, however, is not to make conjectures on the noise sources for sport but to try to come to an understanding about the range of applications that FPAAs as a category of component suit. Historically, high-resolution applications, such as professional and high-end consumer audio, have implemented complex signal-processing functions either in continuous-time analog circuits or in DSPs, due to the kT/C noise limit on discrete-time switched-capacitor analog designs. Semiconductor-process developments, such as high-k dielectrics, could further extend switched-cap techniques. For the time being, it appears that the technology suits a host of 14-bit applications, such as sensor-signal conditioning, process control, and medical applications.

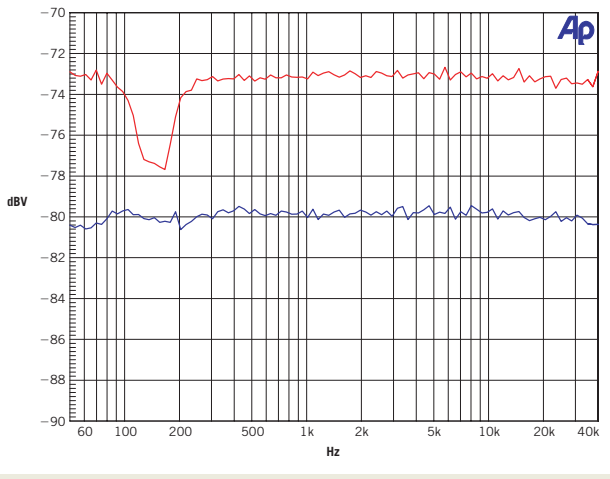

Figure A **The low-frequency noise plot (blue) suggests that kT/C noise dominates. It also indicates that noise can infiltrate the development environment through the serial port when connected to a computer with a noisy power supply as is common with laptops (red). Solution: Run the laptop from batteries or disconnect the serial port when not configuring the FPAA.**

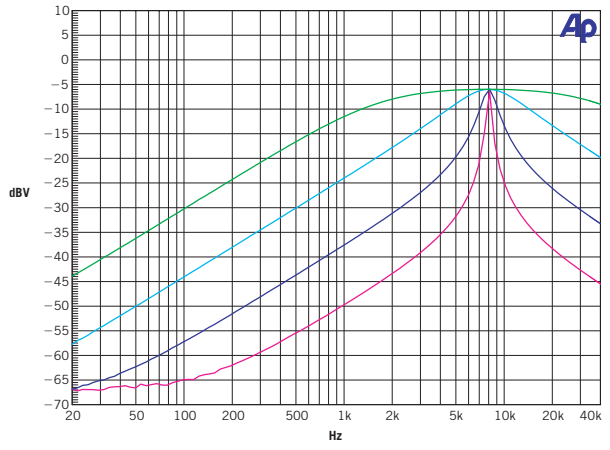

Figure 4 The biquadratic filter CAM allows you to program the corner frequency, gain, and Q. The bandpass-filter plots correspond to an 8-kHz corner frequency at Qs of 0.2 (green), 1 (cyan), 5 (blue), and 20 (magenta).

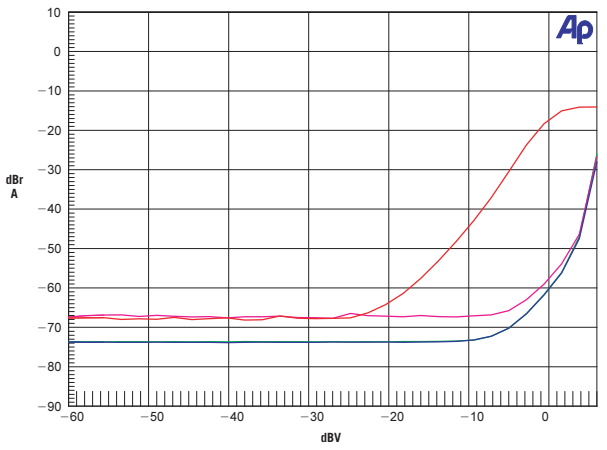

Figure 5 THD+N plots suggest that the input cell's antialiasing filters can limit the useful dynamic range for a given THD+N criterion, particularly if the signal bandwidth approaches the filter corner. Here, the 1-kHz (green) and 20-kHz (blue) plots taken with the antialiasing filters off are congruent. The 1-kHz (magenta) and 20-kHz (red) plots taken with the filter corner set to 34 kHz show degradation, particularly with signals approaching the corner frequency.

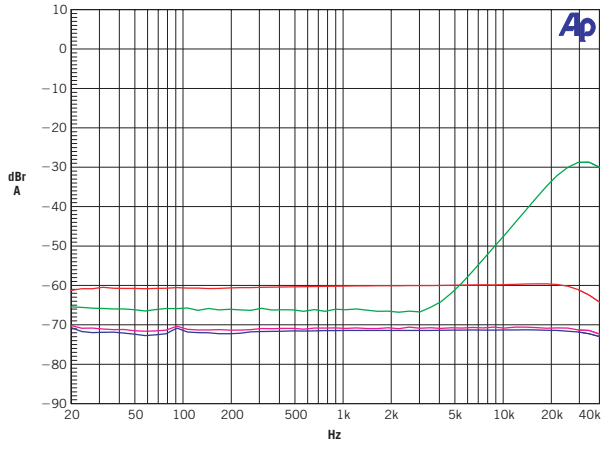

Figure 6 Fixed-amplitude spectral sweeps help fill in Figure 5's message: Here, the blue plot shows a 0-dBV sweep without the antialiasing filter. The red plot shows the same input sweep with the antialiasing filter set to 400 kHz. Reducing the input amplitude to −6 dBV and noting that the plot's amplitudes are relative to the input amplitude, the magenta plot suggests that the filter's THD+N degrades across the spectrum with inputs exceeding several hundred millivolts. With the filter's corner tuned down to 34 kHz, the −6-dBV sweep shows a sharply increasing THD+N trend in the passband's last decade.

gram the threshold voltage.

Like most FPAA CAMs, the comparator is a synchronous discrete-time block. You can program the comparator to sample its inputs on either the first or the second clock phase—a concept that simply doesn't apply to continuous-time comparators. The comparator makes its decision on the following clock phase, but the decision may not cause an immediate change in the comparator's output state. You can program the output to either follow the decision as soon as it is available or force it to synchronize the state transition with either the first or the second clock phase.

Comparators and amplifiers are the simplest CAMs, with close analogies in the world of discrete ICs. Also common in the CAM list are functions that offer a higher level of abstraction, moving you somewhat further away from the implementation details. One example, the biquadratic filter is tunable from 500 Hz to 400 kHz in three ranges corresponding to your selection from three available clock frequencies. As you would expect of a biquad, you can select lowpass, highpass, bandpass, and bandstop transfer functions. But unlike the biquads you make yourself from several op amps and a fistful of resistors, the biquad CAM provides only one of the four transfer functions at a time. The tuning and Q controls are noninteractive, but, not surprisingly, your choice of Q from 0.15 to 70 (!) does limit the gain range (**Figure 4**).

THE PLOTS THICKEN

Some of the less common functions might hold the true power of an FPAA, such as the multiplier or arbitrary periodic waveform-generator CAMs. However, to get a sense of how well the device works as a signal-chain element, I examined less exotic CAMs, such as inverting amplifiers, biquadratic filter blocks, and basic I/O cells. Most of this investigation employed an Audio Precision System Two Cascade dual domain analyzer for frequency response, noise, and THD plots.

The evaluation board is clearly designed to maximize your access to the various facilities it offers. It is, therefore, not optimized for minimum clock noise (see **sidebar** "Finding the baseline"). Though some care may be necessary when developing applications that make full use of the FPAA's bandwidth, which extends to about 8 MHz, the board should serve as a good development environment with little practical interference from clock artifacts for lower-bandwidth circuits.

The FPAA's data-sheet limit on differential input signal swings is 3.8V. Be careful, however, if you are going to run your signals at or near the swing limit. For ex-

ample, the input cell provides an antialiasing filter with a programmable corner frequency of 34 to 470 kHz. THD+N measurements show congruent 1- and 20-kHz plot lines when the antialiasing filter is disengaged, (the blue and green plots in **Figure 5**). At 1 kHz, the THD+N penalty is small for a 34-kHz input filter (magenta), but as signal components approach the corner frequency, the distortion products rise with signal amplitudes as small as −20 dBV (red). Fixed amplitude spectral sweeps fill in the picture (**Figure 6**). Compared with the baseline measurement at 0 dBV with no antialiasing (blue), the THD+N penalty is about 10 dB for adding a 400-kHz filter on the input (red). Dropping the input to −6 dB reduces the THD+N with the 400-kHz filter to the same relative level that the 0-dBV, nonfiltered case gives (magenta). Note that the vertical scale is decibels relative to input level. With the input held at −6 dBV and the filter corner tuned down to 34 kHz, the THD+N level increases another 5 dB or so and exhibits a rising characteristic as the signal approaches the corner frequency. Considering these traits, you should operate the CAM cells at the fastest clock rates you can to push out the Nyquist frequency and reduce the need for close-in filter corners. Also, use the input filter only after paying due consideration to your expected signal levels and bandwidths, and consider a simple external filter if you need to accommodate high amplitudes with a low frequency corner.

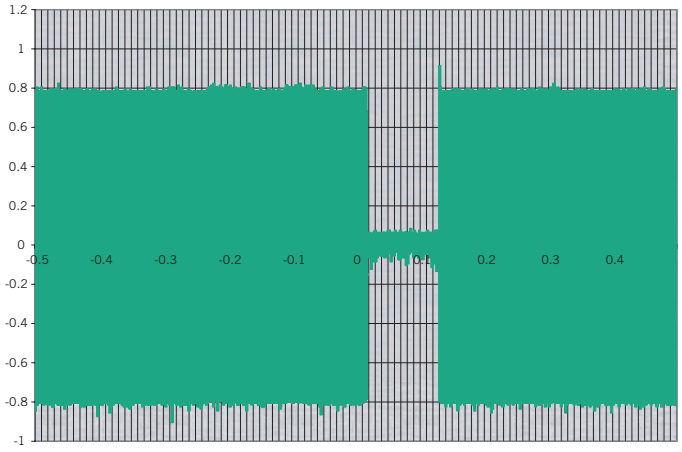

Figure 7 Configuring the FPAA from a laptop computer through the development environment imposes a signal-processing interruption of about 110 msec.

The output cells also come with lowpass filters associated with their voltage-output mode. The cell cascades a pair of similarly tuned single-pole sections and is intended as a reconstruction filter to remove switching artifacts. Tuned to a relatively open 400 kHz, they performed better following a simple inverting amplifier than did the raw output cell. An inverting amplifier with a gain of −4 processing a −18-dBV swept sine wave produces essentially flat THD+N plots over the 20-Hz to 40-kHz band. Operating the output cell as a voltage output with a 400-kHz lowpass corner resulted in a −62-dBr THD+N. With the output cell in its raw configuration, the performance dropped to −52 dBr. Further investigation of the output structure is certainly warranted should you choose to design with FPAAs to help identify the best operating conditions for your application.

The FPAA's configuration memory includes a shadow RAM that helps minimize configuration time and thus the disturbance to a signal chain that includes FPAAs. Configuration changes may constitute a complete change of the analog array's internal resource allocation or simply cause a change in a single parameter. Operating the board from a laptop computer's serial port, a null configuration change—one in which the existing circuit and parameter set is simply reloaded to the part—imposed an interruption of about 110 msec (**Figure 7**). Applications that load configuration data from either an embedded processor or a co-resident PROM can optimize the configuration process to further reduce the load time.□

You can reach Technical Editor Joshua Israelsohn at 1-617-558-4427, fax 1-617-558-4470, e-mail jisraelsohn@edn.com.

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, go to www.edn.com/info and enter the reader-service number. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

Anadigm
1-408-879-6677
www.anadigm.com
Enter No. 348

Analog Devices
1-781-329-4700
www.analog.com
Enter No. 349

Atmel
1-408-441-0311
www.atmel.com
Enter No. 350

Audio Precision
1-800-231-7350
www.audioprecision.com
Enter No. 351

Lattice Semiconductor
1-503-268-8000
www.latticesemi.com
Enter No. 352

Microchip
www.microchip.com
Enter No. 353

Zetex Semiconductors
1-631-360-2222
www.zetex.com
Enter No. 354

SUPER INFO NUMBER
For more information on the products available from all of the vendors listed in this box, enter no. 355 at www.edn.com/info.

REFERENCES:

1. Israelsohn, Joshua, "Click picks," *EDN*, May 1, 2003, pg 67.
2. Schweber, Bill, "Programmable analog ICs: designer's delight or dilemma?" *EDN*, April 23, 2000, pg 72.
3. Marsh, David, "Programmable analogue ICs challenge spice-and-breadboard designs," *EDN Europe*, October 2001, pg 30.

ACKNOWLEDGMENTS

Thanks to Anadigm field-application engineer Troy Giles for clarifications. Thanks to Bob Adams of Analog Devices for the generous use of lab facilities, including the AP system 2. Thanks to Audio Precision for making that rare combination—a first-rate analyzer that's a genuine pleasure to drive.