



Embedded Digital Filtering

By Kendall Castor-Perry, Principal Architect, Cypress Semiconductor Corp.

The increasing availability of affordable microcontrollers with analog-to-digital front ends continues to change how embedded systems are being designed. Often termed ‘mixed-signal microcontrollers’, these devices consolidate both control and digital signal processing functionality onto a single device. Such programmable “system on chip” technology not only simplifies application architectures, it accelerates development by eliminating the need to communicate between chips, and by enabling engineers to write application code within a single development environment.

One of the primary advantages of mixed-signal microcontrollers is that they can do much more than simply convert external analog signals to the digital domain. Data acquisition is not an end in itself but rather is a first step before extracting the meaning behind acquired data, and deciding what to do with it. Many embedded developers are turning to newer mixed-signal microcontrollers because they make it possible to cost-effectively implement complex digital filtering in a wide range of new applications. More advanced filtering allows engineers to offer more capabilities – including higher precision, better efficiency and lower power consumption – with which to differentiate their products.

Bringing digital signal processing and control functionality onto the same device, however, does not come without its own challenges. When an application requires extensive signal processing, it can become a challenge for the firmware engineer to perform such processing in real-time when it is run on the same processor as all of the other housekeeping tasks the processor needs to support, including managing a screen, keyboard, storage, and so on. Additionally, the architecture must be flexible enough to support changing application specs, as well as be able to scale functionality to meet different market requirements.

Balancing Hardware and Software Resources

Flexibility in an embedded architecture is important. Many embedded applications do not have the volumes to encourage silicon manufacturers to spin application-specific microcontrollers. Even for high volume applications, application-specific chips appear on the market only after the functions they provide have become commoditized. In many cases, if developers want to implement advanced digital filtering, they have to rely upon controllers that are either general-purpose in nature or that are optimized for other applications. In addition, as with many embedded project developments, they go through frequent changes in scope, ambition, and architecture. Keeping up with the impact this has on a monolithic coding project on a single core is challenging, especially with today’s decentralized, multi-contributor design teams.

Some newer mixed-signal controllers provide the necessary flexibility through programmable or configurable logic, implemented as integrated coprocessors or hardware blocks that can be programmed to execute independently, in parallel to the main CPU. These devices can implement compute-intensive algorithms with a high degree of efficiency, and at minimal cost. In addition, the decoupling of signal processing from the main CPU allows the reuse of IP through an ecosystem of blocks that can be shared across a design community to reduce development time and lower application cost. This level of flexibility also allows developers to support entirely new applications with their own custom IP blocks in a cost-effective manner.

Being able to embed signal processing into functional components at the block level ensures that the project’s management can be ‘forked’ at the component design level. By adjusting the programmable hardware resources available to a particular function, developers can scale digital filtering algorithms independently of CPU software resources. This methodology ensures that signal processing load variations, which change as algorithms are modified during the design process, have no impact on other highly timing-critical tasks such as communications management.

Embedding digital filtering

This article illustrates some recent system designs employing a recently-introduced mixed-signal microcontroller with an embedded filter coprocessor. Many filtering topologies can be coded effectively onto this structure. Combining programmable hardware and data path blocks enables developers to marry the efficiencies of hardware with software in an optimal balance

for their specific application. Data and coefficients are stored in dedicated local memories and are shared between programmable hardware and software resources via a system bus. Both sets of resources have access to sources and sinks of digital data.

Tools are available for quickly configuring these systems-on-chip using a drag-and-drop interface, such as the PSoC Creator Integrated Development Environment from Cypress Semiconductor, which supports the newly-introduced PSoC3 and PSoC5 architectures. Developers have the option of using or modifying a wide range of existing library elements, or creating new blocks such as custom filters. These elements are then converted into the necessary hardware and software components as dictated by the programmable system-on-chip architecture in use.

Accurate electricity meters

Consider first the example of an electricity meter that needs to compensate for the interchannel timing offset of a single delta-sigma ADC, into which several phases of voltage and current are multiplexed. If this time difference isn't corrected, system accuracy will degrade rapidly for low power factor loads as well as for the estimation of the power in the higher harmonics of the line frequency.

In a recent implementation, we used embedded FIR structures to implement a polyphase interpolation filter with four channels of 20 taps each. This filter takes the multiplexed data stream from the single ADC, and 'unpacks' it into four new data streams through channels whose signal delays differ by the correct fractions of a sample time needed to 'realign' the data as if it had been captured by four simultaneously sampling ADCs. Figure 1 shows four data sets obtained by sequentially sampling the same (band-limited) signal with a four-input multiplexed converter.

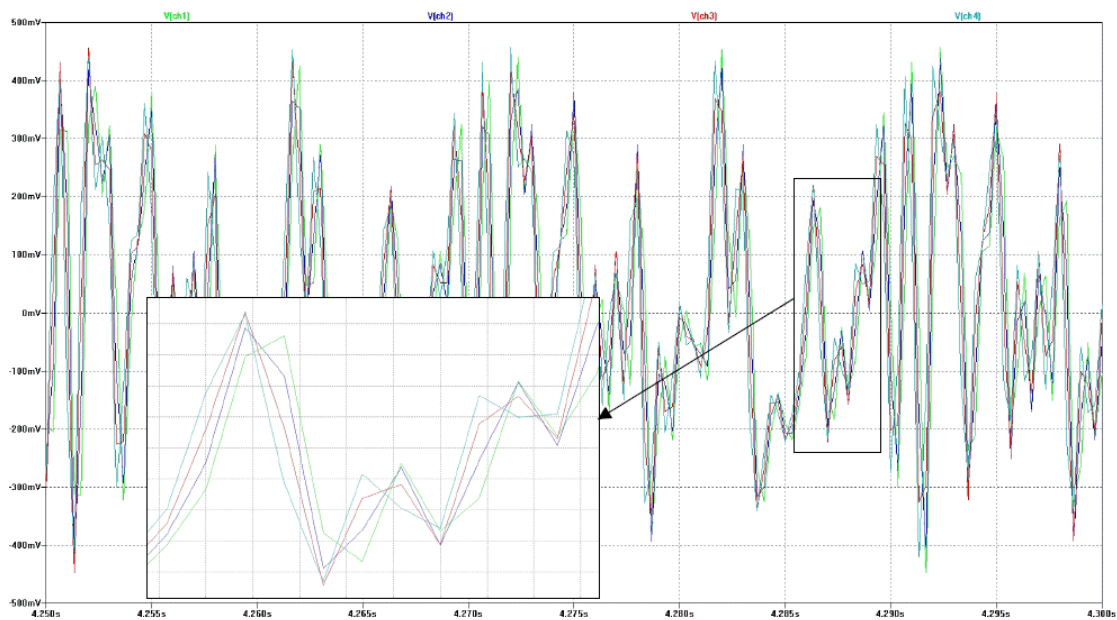


Figure 1: Four sequentially multiplexed inputs of an ADC are fed the same signal...

Figure 2 shows the four outputs from the interpolation filter system, showing that the underlying band-limited waveform has been correctly reconstructed in both shape and timing. This technique enables a single high quality ADC to service the very highest metering accuracy classes, at all relevant power factors and harmonic frequencies. It's also widely applicable in other applications requiring effectively simultaneous sampling.

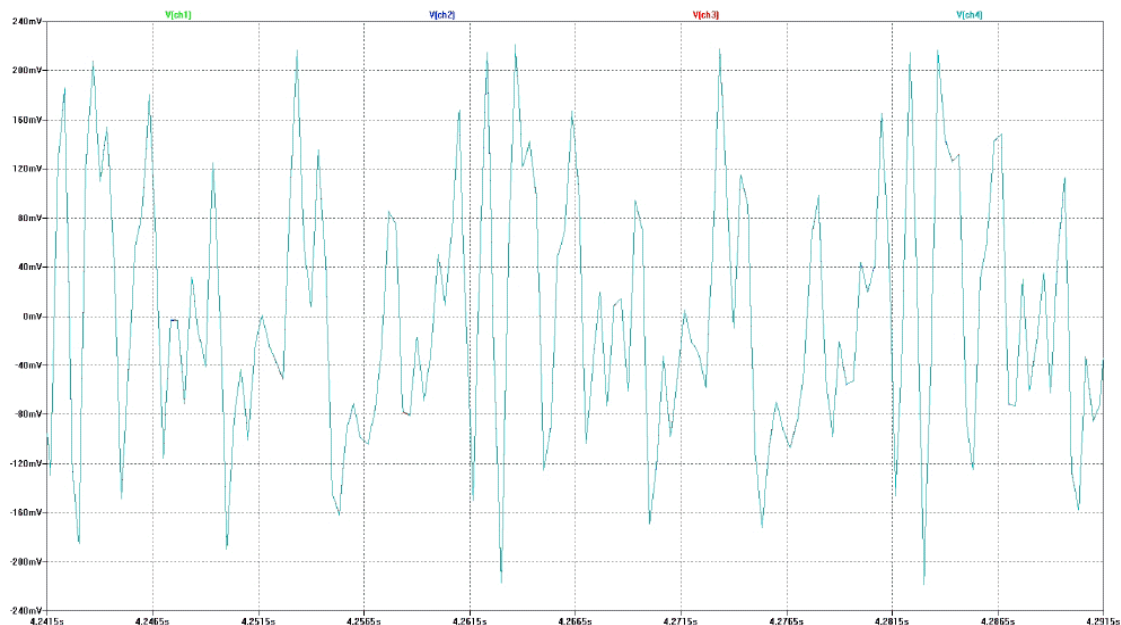
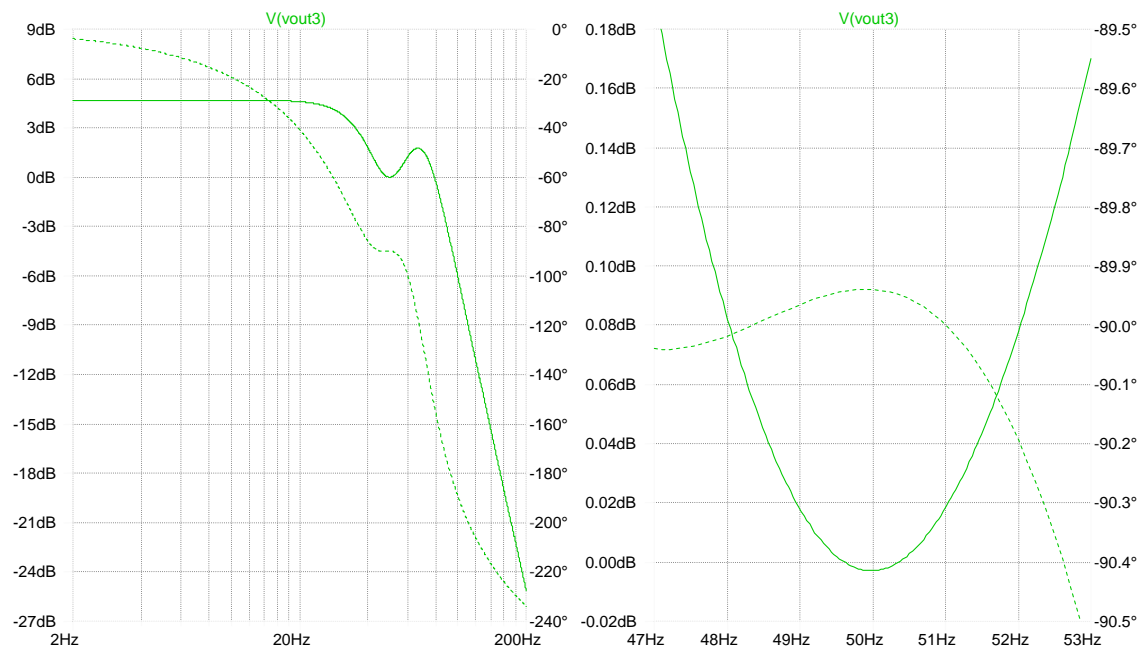


Figure 2: ...and the embedded polyphase interpolation filter eliminates the differences

To enable the accurate calculation of 'classical' fundamental reactive power, we then deployed a computer-optimized phase-shifting filter. Most commercial metering chips use either a time delay or an integrator to deliver the required 90 degree phase shift. The former method has flat amplitude response but is inaccurate in phase shift when the line frequency is not at the correct value. The latter method has the converse error; the phase is always accurate but the amplitude varies with frequency, in a way that fails all but the least stringent metering classes.

To solve this issue, we embedded a 6-pole IIR filter (figure 3) that beats the most stringent reactive power accuracy requirements over the entire line frequency range, without having to resort to the very wasteful Hilbert Transformer approach that would have consumed the entire system processing power. The IIR filter also had a lowpass characteristic that strongly attenuates the harmonics in the current waveform, leaving only the fundamental for reactive power estimation. The ability to implement this function on the programmable system-on-chip not only reduces system complexity and cost, it provides better performance.



Figures 3: a dedicated n=6 embedded IIR precision quadrature generator for 50Hz

The other critical frequency response shaping circuit in the modern electricity meter is the integrator needed to compensate the frequency response of a di/dt type of current sensor such as a Rogowski coil or Sentec 'Mobius'. The rising low frequency response of such a circuit exacerbates the low frequency analog noise inherent in the front end. For standard active power measurements this is not an issue. Increasingly, however, customers are demanding wider dynamic range for *current* measurements so that apparent power and the effective dissipation in the electricity infrastructure can be calculated accurately. At very low currents, the integrator noise component leads to an inaccurately-high current measurement.

Because the rising of the gain can't be allowed to continue indefinitely – or else the gain would be infinite at DC – the integrator is 'damped' to a lower limit frequency in conventional devices. This introduces a phase error that becomes noticeable at the highest accuracy classes. To support the use of di/dt sensors, we designed in another 6-pole IIR filter with a specific restricted low frequency response. This gives us 9-15 dB better integrated noise performance, depending on the front end. In addition, it still delivers the amplitude and phase response of an ideal integrator within the working bandwidth to an accuracy better than the 'standard' metering chip we used as a reference (the green traces in figure 4).

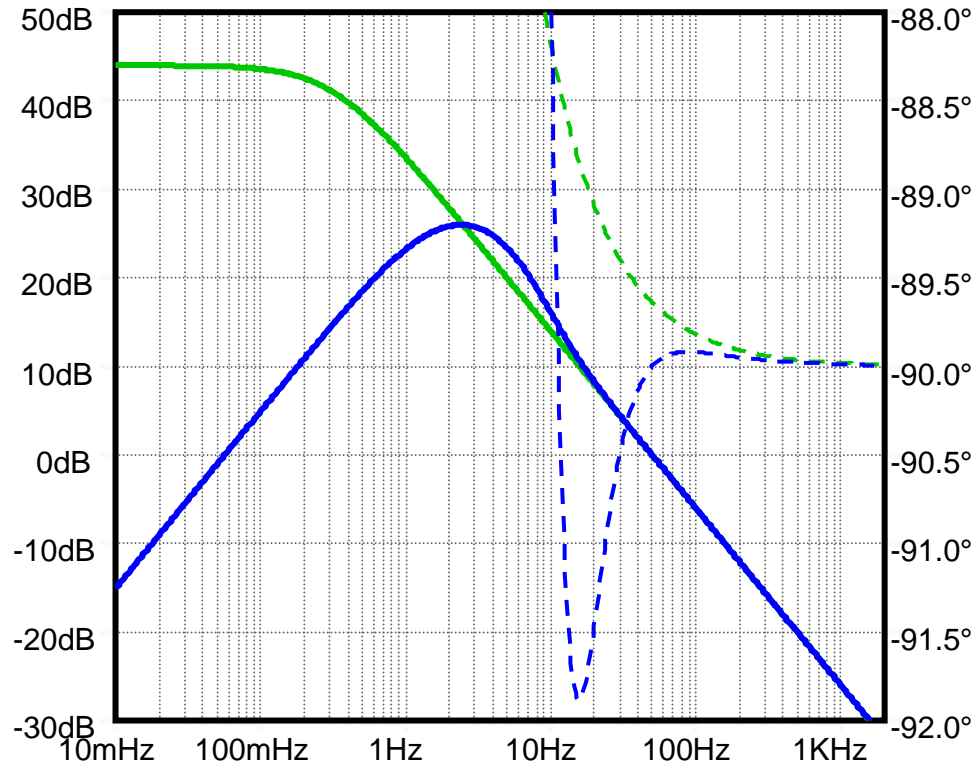
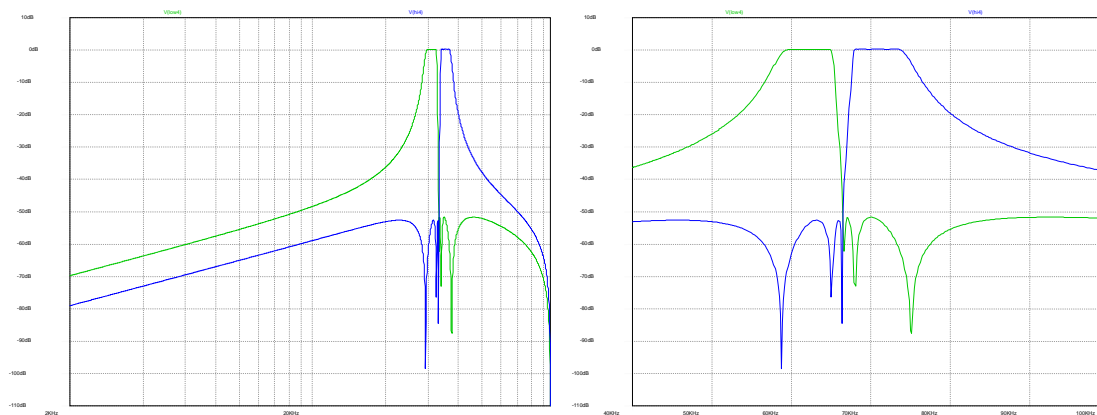


Figure 4: high accuracy low-noise integrator by embedded IIR filtering (blue traces)

Communications filters and detectors

The IEC 61334-5 SFSK powerline communication standard, popular in electricity metering applications, uses SFSK (Spread FSK), a variant of FSK in which the mark and space frequencies are much more widely separated than you'd expect from the data rate. If the incoming signal is split by a pair of sharp bandpass filters that only pick out the mark or the space frequency component, the data modulation can be extracted independently from either channel. This confers exceptional interference resistance, since a single-tone interferer can't prevent the demodulation of both channels at once, if the filters' frequency responses don't overlap. Conventional correlator-based FSK demodulators can't deliver this interference resistance.

Figure 5 shows the frequency response of a pair of filters designed for the Cypress PSoC3 for a commonly used pair of mark/space frequencies. The filters can easily be reconfigured for different frequencies and bandwidths at any time. In a practical realization, the filter is fed from one of the main ADCs, preceded by an AGC circuit built around the analog PGA (programmable gain amplifier).



Figures 5: Frequency response for embedded splitting filters (60/73kHz SFSK; 2x n=8 IIR at 384ksps) designed for the Cypress PSoC 3 DFB for a common pair of mark/space frequencies.

To extract the data from the filtered signals, they are rectified by taking the absolute value of each signal, available trivially just by setting the appropriate control register bit. The rectified signals are then passed through lowpass filters also running on the embedded filter engine, and compared against a threshold value that tracks the signal level.

In this implementation, which meets all of the interference rejection requirements of the spec, all of the signal processing is executed autonomously in programmable hardware on the relevant multiplexed signals from the high quality delta-sigma modulator, without intervention from the processor. Likewise, the SNR of each channel is estimated in hardware and the data passed onto a standard UART.

Multi-channel filtering

The number of channels that can be filtered concurrently is dependent upon the microcontroller's architecture. In a software implementation, the available processing resources must be time-sliced, creating a complex array of independent threads that must be serviced consistently to meet real-time constraints. Using interrupts to handle non-real-time tasks, such as refreshing a screen or polling a keyboard, can complicate handling of real-time interrupts significantly.

In programmable hardware architecture, hardware resources can be allocated to each channel. These resources execute in parallel with each other and with the CPU, thus eliminating many of the issues relating to managing real-time interrupts. Consider a consumer audio application utilizing a stereo ten-band graphic equalizer whose filter coefficients are calculated by the CPU on the fly. A stereo audio codec is interfaced to the microcontroller over a standard I2S port clocked from an embedded frequency synthesis system that can produce all standard audio master clock frequencies from a single local crystal. If required, this synthesis system can also be synchronized to the framing patterns of common digital interface formats such as USB. All this is implemented in programmable hardware.

At a 44.1 kHz sample rate, this equalizer uses only a fraction of the available resource and filtering power. The response-fitting and coefficient calculation routines can take update information dynamically from local controls directly managed by the microcontroller and also from control protocols arriving over an (embedded) interface from a mobile device used as a signal source and control unit.

This recently-implemented system still has enough resources left to implement a multi-band crossover filter bank. The outputs can be delivered to external DACs or digital amplifiers through multiple I2S interfaces. The super-precise control of frequency response obtained eases the acoustic design of 'difficult' loudspeaker enclosure designs, such as the driver subsystems designed for in-vehicle and public address applications. It also allows consistent high quality results from compact multiway acoustic designs in docking units, micro-stereos and flat-panel TVs. High channel-count distributed sound reinforcement and messaging systems benefit from the simplified frequency response tuning, all achieved on the same processor that is managing the user interface, communications and power supply.



Summing up

Introducing a powerful digital filtering engine into embedded applications extends the value developers can deliver to their customers, and it reduces system cost, complexity, and time-to-market. By making use of mixed-signal microcontrollers with programmable hardware signal processing resources, such as the PSoC3 device in these examples, it becomes possible to address the evolving requirements for complex signal filtering as they change during the design process. With them, developers can cost-effectively introduce a wide range of capabilities to enhance their products, from adding 'stereo enhancement' functions, to decimation filters for digital microphones, and even advanced control algorithms for industrial sensor conditioning and medical applications.

About the author

Kendall Castor-Perry is a Principal Architect at Cypress Semiconductor Corp. Based in the UK, he is known for his passion for filters and audio, and is striving to make the electronic design world a better place, one dB at a time, with PSoC.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
<http://www.cypress.com>

© Cypress Semiconductor Corporation, 2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™, Programmable System-on-Chip™, and PSoC Express™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and/or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.