# Startling Tales of Nonfiction

If you're amazed by the ongoing advances in DSP silicon and development tools, you ought to be nothing less than astounded by the commensurate reach of DSP applications. Not only are the applications startlingly complex and more powerful, but they can be written and debugged in reasonable amounts of time—without a mainframe computer and swarms of computer scientists.

Take the cover story, "Efficient Multirate Signal Processing with the C55x." Here, Michael Tsiroulnikov, of MIKET DSP Solutions, shows how to take advantage of the copious on-chip features of the TMS320C55x generation of DSPs—a dual multiplier-accumulator, multiple buses, and lots of RAM—for multirate operations in the complex domain. In particular, he shows how to shape decimating filter banks, a key component of multirate signal processing systems.

Working in the complex domain helps sidestep the limitations of real-domain bandpass sampling, increases performance, and simplifies later signal processing. At the same time, the large store of RAM satisfies the often unquenchable thirst of multirate signal processing systems for fast RAM in which to store filter bank coefficients. In addition, the C55x's speed, boosted in large part by its dual MAC, enables you to compensate for burgeoning program code by running more instances of an algorithm in a given amount of time. All told, you should realize about an order-of-magnitude cumulative improvement in performance over previous generations of DSPs when you implement signal detectors, decimating filter banks, and other functions that filter input data through an array of same-length complex filters.

Increasingly, such powerhouse signal processors are finding themselves working alongside general-purpose processors—on one chip, no less. This teamwork is great for distributed heterogeneous applications, but it poses a challenge if you're seeking to wring the best efficiency out of the on-chip resources.

For instance, transformational algorithms are best performed by digital signal processors, whereas control-intensive and graphical user interface functions are best served by a general-purpose microprocessor. Accordingly, applications that are both control-intensive and algorithmically complex typically call for multiple-processor architectures.

Take heart. A new methodology, termed coordination-centric design, uncouples the architecture from the platform and takes the pain out of developing distributed heterogeneous applications. As David McCooey, Ken Hines, and Ross Ortega of Consystant Design Technologies relate in our second article, in developing distributed software applications software architects must consider multiple operating systems and think about how applications will map to specific processors. Specialists could be required to deal with obscure programming paradigms or tight performance requirements. Such implementation details can obscure the high-level software architecture.

Coordination-centric design lets designers cleanly separate the two and automatically produces efficient source code from high-level software models. Thus it supports heterogeneous distributed architectures and allows you to specify low-level implementation details without reference to the software design.

Eliminating trouble is the theme of our third article, this time the troublesome signaling tone distortion and other problems that can pop up in speech-compressed packet data networks. The solution: Use a tone relay to pass tones in voice-over-packet systems.

Adaptive Digital Technologies' Scott Kurtz delineates the problems caused by low-rate speech compression algorithms and delves into the design and nuances of tone relays and their accompanying algorithms. For instance, although a tone passer seems straightforward at first glance, its implementation details can make or break performance. Kurtz's thorough discussion should help clear up any distortion.

—Stan Runyon
testman2@earthlink.net

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265