Multiprocessing Symphonies

We're always warned against mixing apples and oranges, but creative chefs have blended these diverse fruits in imaginative dishes that make the palette soar. Similarly, composers have taken advantage of counterpoint—the art of making dissimilar melodies work separately and yet together—to create beautifully intricate works, from compositions for a single instrument to symphonies, that delight both music lovers and casual listeners.

Perhaps taking a cue from the musical world, system designers building on digital signal processors by adding general-purpose processors, with technically salutary effects. In fact, the potential benefits derived from blending these two diverse processors is so appealing that Texas Instruments, for one, has developed the Digital Still Camera and OMAP platforms, which add a GPP to a DSP and much more on a single chip.

Such devices may set bells ringing in some engineering quarters. After all, the two kinds of processors were designed for very different tasks, operate almost completely differently from one another, and work in conjunction with two very different operating systems. Nevertheless, enterprising system engineers have been able to sidestep the differences and create harmonious works.

Noting the trend and the arrival of TI's DSP plus GPP technology, one enterprising company, RidgeRun, has stepped up to the software podium and conducted a stunning interpretation of the DSP plus GPP counterpoint.

In our cover story, RidgeRun's Gordon McNutt and Todd Fischer show how a new way of thinking and working allows the operating system running on the GPP to assist the processes on the DSP—just as if they were standard operating system processes. Likewise, DSP processes can make operating system calls that are processed by the GPP. Allowing standard process creation mechanisms to be used to load and launch DSP processes and allowing DSP processes to make standard operating system calls greatly simplifies software design for dual-core chips or systems with discrete DSPs and GPPs.

Harmonious arrangements of multiple-DSP systems, too, are music to our design ears. Synchronization has become particularly important because multiprocessing architectures are becoming pervasive. Indeed multiple-DSP arrangements can be found at the network edge and SoCs often include DSP cores to accelerate math-intensive computations.

As Ted Raineault of Electric Sand points out in our second article, DSP/BIOS provides a standard, efficient, robust API for single-processor applications; shared-memory multiple-DSP systems, which designers are encountering more frequently, on the other hand, call for clever interprocessor synchronization mechanisms and multiprocessor mutual exclusion. After reviewing mutual exclusion, Ted shows how to implement binary semaphores for interprocessor synchronization using DSP/BIOS, placing the semaphore's data structure in shared memory and using RTOS services on each processor to handle blocking. He offers a dual-processor implementation using Peterson's algorithm that can be extended to multiprocessor systems, but he warns about the need to attend to shared-variable coherence when processors have on-chip cache or systems use write posting.

On a completely different note, multiple DSPs are showing up at the center of specialized applications with a GPP, too, like test equipment aimed at ensuring the needed quality of service in telecommunications systems. In our last article, Gordon Wilkinson of the Motorola Computer Group shows how to design a generic two-port test system that employs a quintet of processors—four DSPs and a RISC processor—to stimulate a system under test as if it were connected to a network and verify that it follows the score faithfully.

—Stan Runyon
testman2@earthlink.net