What's Next For Programmable Devices?

Microprocessor and programmable logic design are converging, though they still require a common language.

By Dave Van Ess, Applications Engineering Member of Technical Staff, Cypress Semiconductor Corp.

After one of my engineers told me that she had found a typo in her hardware, I thought just how far design has come in the last 30 years and more. Back then, the analog section of a system consisted of discrete components and op amps. The digital section was implemented with either small-scale integrated (SSI) ICs or custom logic. I showed this engineer the little green logic template I used back then. She thought it was cute.

A Time To Break Away

Engineers seldom have to design with discrete components anymore. Integrated analog-to-digital converters (ADCs), digital-to-analog converters (DACs), high-speed amplifiers, and switch capacitor ICs are now readily available. The most truly amazing advances, however, have been in digital design. The shortage of logic chip designers and the expense of design has led to development along two different avenues: microprocessors and programmable logic.

Microprocessors led to the need for quality and documentation metrics for design and implementation. The addition of digital peripherals resulted in the microcontroller. Armed with timers, counters, UARTs, and other features, many complete digital systems could be implemented with a single chip. Adding a specialized peripheral containing a multiply-accumulator (MAC) and the logic needed to load and unload it yielded the digital signal processor (DSP).

Analog components such as amplifiers, comparators, ADCs, and DACs have been integrated into microcontrollers to help build a more complete system. These, however, have been fixed in performance. To fine-tune selection, different chips provide different accuracy, resolution, and sample speed components. The combination of different digital and analog peripherals has caused some microcontroller manufacturers to provide thousands of different “flavors” of their parts. And when additional functions were required, they would have to be implemented with external hardware.

Engineers have risen to the occasion with programmable logic, not only through innovative chip topologies but also the tools to use them. Early programming was done by selecting the fuses from a printed fuse map. PALASM, the first language to program logic devices, was used to express Boolean equations. Learning the benefits of a common language from software, developers created Verilog and VHDL.

Designers now had a common language to develop across different platforms, and hardware could be developed using a logical language. Also, chip manufacturers could go back to developing chips instead of tools. All these tools not only can define a design but test and verify its performance as well. This has led to digital design becoming a very disciplined art. It is also quite common for digital designers to split up their work, one doing the design on one chip while the other creates the test vectors.

A Time To Bring It All Together

Combining the advantages of microcontrollers, programmable digital logic, and analog peripherals, a truly programmable reconfigurable system-on-a-chip (SoC) would have programmable logic to allow the design of specific digital peripherals; a programmable language to support design and verification; the reprogrammability of FPGAs and their volatile configuration registers; permanent-configuration complex programmable-logic device registers for immediate hardware configuration at startup; a CPU with common digital and analog peripherals; and analog circuitry that can be reconfigured to design analog and mixed-signal peripherals.

There have been several attempts to add programmable logic to microcontrollers. As transistor geometries get smaller and digital gates become closer to free, this should become more common in SoC architectures.

In the future, the dilemma of choosing between permanent and volatile registers will be resolved by loading volatile registers immediately with a permanent ROM base register at startup, permitting hardware reconfiguration while allowing quick operation at startup. This becomes feasible as transistor geometries get smaller. Some examples already can be seen.

Published in Electronic Design (http://electronicdesign.com/Articles/ArticleID/21305/21305.html) June 2009

As more configurable analog is added to SoC architectures, it will have to function well within CMOS design processes. Designs will have as little analog as possible, boasting post-digital processing. This is the very definition of mixed-signal design. High-level components can then be implemented for specific resolution and sample speed with commutating amplifiers removing the dc offset and high 1/f noise (wander) inherent with CMOS transistors.

Such SoCs will start with a CPU and specific hardware peripherals. But as transistors become free, they will be implemented with programmable logic. Already, “softcore CPUs” are implemented with large FPGAs.

And, there will need to be a design language. Analog design is still very graphically oriented, and a way is needed to be able to define an analog system in a logical written matter. This language also must take from the digital community and have some way of testing and verifying the design’s performance. I have no idea what this language will look like. However, at the time I had no idea what “C,” Verilog, or VHDL would look like either.

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