This issue features articles on the resurgence and many implications of big chips—from cache-core tradeoffs to power to on-chip networks to clock networks and more. Guest Editors Andrew Kahng and Vijayalakshmi Srinivasan have done an excellent job of pulling together key work related to big chips.

However, after a period of some consolidation in microarchitecture, big chips are but one of many flowers blooming in microarchitecture and processor design. For example, recent machines at #1 in the Top 500 list (http://www.top500.org) have featured the IBM Cell processor in Roadrunner, Nvidia GPUs in Tianhe-1A, and now the Japanese K computer with SPARC64 VIIIfx 8-core chips and new single-instruction, multiple-data (SIMD) extensions. The K computer achieves an impressive 800 Mflops per Watt, almost four times the mean 248 Mflops per Watt in the Top 500. However, yet another type of design, IBM’s prototype Bluegene/Q is most efficient in the Top 500 at more than 2,000 Mflops per Watt.

This flowering of microarchitectural designs is not limited to supercomputing. Google, Baidu, Facebook, Amazon, Microsoft, and others use ubiquitous x86 processors in their data centers. However, Marvell, Calxeda, ZT Systems, and others are trying to compete for this sort of business with power and volumetrically efficient ARM designs. Meanwhile, Intel’s Atom pushes into the low-power embedded space of cell phones and other devices, currently dominated by ARM. Others such as Tierra are trying to find a niche with tens and hundreds of general-purpose cores per chip, while Nethra tries to put large numbers of cores to use for specific applications such as image processing.

At the high end, frequency is not yet dead. IBM’s zEnterprise 196 (discussed in our March/April 2011 issue on Hot Chips) runs up to 5.2 GHz. The Power7 system behind IBM’s Watson (“Jeopardy”) system has 2,880 cores and consumes space and power that would make ENIAC proud.

Even more exotic approaches such as field-programmable gate array (FPGA) computing are in the offing. Although FPGAs are widely used for application-specific integrated circuit (ASIC) replacement, glue logic, and other system functions, 6 percent of the FPGA market now goes for more general-purpose computing.

All of this heterogeneity—not only in instruction sets, but in SIMD formats and programming models—has implications about the portability of software and even the languages that are used. There has been a trend toward languages perceived to be more portable, such as Java, JavaScript, PHP, and Ruby. However, two driving factors in creating the current heterogeneity are power and performance, which could tilt language use to traditional stalwarts such as C and Fortran. Whatever the resolution of these competing factors of portability and power performance, there will be strong influence to and from microarchitecture. That should make for interesting reading in IEEE Micro. Stay tuned!

Erik R. Altman
Editor in Chief
IEEE Micro