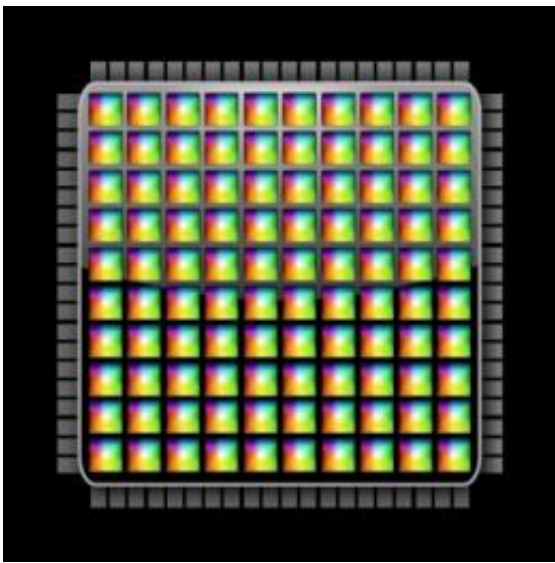# New software-simulation system promises much more accurate evaluation of multicore-chip designs

March 9 2012, by Larry Hardesty

For the last decade or so, computer chip manufacturers have been increasing the speed of their chips by giving them extra processing units, or "cores." Most major manufacturers now offer chips with eight, 10 or even 12 cores.

But if chips are to continue improving at the rate we've grown accustomed to — doubling in power roughly every 18 months — they'll soon require hundreds and even thousands of cores. Academic and

industry researchers are full of ideas for improving the performance of multicore chips, but there's always the possibility that an approach that seems to work well with 24 or 48 cores may introduce catastrophic problems when the core count gets higher. No chip manufacturer will take a chance on an innovative chip design without overwhelming evidence that it works as advertised.

As a research tool, an MIT group that specializes in computer architecture has developed a software simulator, dubbed Hornet, that models the performance of multicore chips much more accurately than its predecessors do. At the Fifth International Symposium on Networks-on-Chip in 2011, the group took the best-paper prize for work in which they used the simulator to analyze a promising and much-studied multicore-computing technique, finding a fatal flaw that other simulations had missed. And in a forthcoming issue of *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, the researchers present a new version of the simulator that factors in power consumption as well as patterns of communication between cores, the processing times of individual tasks, and memory-access patterns.

The flow of data through a chip with hundreds of cores is monstrously complex, and previous software simulators have sacrificed some accuracy for the sake of efficiency. For more accurate simulations, researchers have typically used hardware models — programmable chips that can be reconfigured to mimic the behavior of multicore chips. According to Myong Hyon Cho, a PhD student in the Department of Electrical Engineering and Computer Science (EECS) and one of Hornet's developers, Hornet is intended to complement, not compete with, these other two approaches. "We think that Hornet sits in the sweet spot between them," Cho says.

The various tasks performed by a chip's many components are

synchronized by a master clock; during each "clock cycle," each component performs one task. Hornet is significantly slower than its predecessors, but it can provide a "cycle-accurate" simulation of a chip with 1,000 cores. "'Cycle-accurate' means the results are precise to the level of a single cycle," Cho explains. "For example, [Hornet has] the ability to say, 'This task takes 1,223,392 cycles to finish.'"

Existing simulators are good at evaluating chips' general performance, but they can miss problems that arise only in rare, pathological cases. Hornet is much more likely to ferret those out, as it did in the case of the research presented at the Network-on-Chip Symposium. There, Cho, his adviser and EECS professor Srini Devadas, and their colleagues analyzed a promising multicore-computing technique in which the chip passes computational tasks to the cores storing the pertinent data rather than passing data to the cores performing the pertinent tasks. Hornet identified the risk of a problem called deadlock, which other simulators had missed. (Deadlock is a situation in which some number of cores are waiting for resources — communications channels or memory locations — in use by other cores. No core will abandon the resource it has until it's granted access to the one it needs, so clock cycles tick by endlessly without any of the cores doing anything.)

In addition to identifying the risk of deadlock, the researchers also proposed a way to avoid it — and demonstrated that their proposal worked with another Hornet simulation. That illustrates Hornet's advantage over hardware systems: the ease with which it can be reconfigured to test out alternative design proposals.

Building simulations that will run on hardware "is more tricky than just writing software," says Edward Suh, an assistant professor of electrical and computer engineering at Cornell University, whose group used an early version of Hornet that just modeled communication between cores. "It's hard to say whether it's inherently more difficult to write, but at

least right now, there's less of an infrastructure, and students do not know those languages as well as they do regular programming language. So as of right now, it's more work." Hornet, Suh says, could have advantages in situations where "you want to test out several ideas quickly, with good accuracy."

Suh points out, however, that because Hornet is slower than either hardware simulations or less-accurate software simulations, "you tend to simulate a short period of the application rather than trying to run the whole application." But, he adds, "That's definitely useful if you want to know if there are some abnormal behaviors." And furthermore, "there are techniques people use, like statistical sampling, or things like that, to say, 'these are representative portions of the application.'"

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology