COVER FEATURE

Heterogeneous Chip Multiprocessors

Heterogeneous (or asymmetric) chip multiprocessors present unique opportunities for improving system throughput, reducing processor power, and mitigating Amdahl's law. On-chip heterogeneity allows the processor to better match execution resources to each application's needs and to address a much wider spectrum of system loads—from low to high thread parallelism—with high efficiency.

Rakesh Kumar
Dean M. Tullsen
University of California, San Diego

Norman P. Jouppi
Parthasarathy Ranganathan
HP Labs

With the announcement of multicore microprocessors from Intel, AMD, IBM, and Sun Microsystems, chip multiprocessors have recently expanded from an active area of research to a hot product area. If Moore's law continues to apply in the chip multiprocessor (CMP) era, we can expect to see a geometrically increasing number of cores with each advance in feature size.

A critical question in CMPs is the size and strength of the replicated core. Many server applications focus primarily on throughput per cost and power. Ideally, a CMP targeted for these applications would use a large number of small low-power cores. Much of the initial research in CMPs focused on these types of applications.[1,2] However, desktop users are more interested in the performance of a single application or a few applications at a given time. A CMP designed for desktop users would more likely be focused on a smaller number of larger, higher-power cores with better single-thread performance. How should designers choose between these conflicting requirements in core complexity?

In reality, application needs are not always so simply characterized, and many types of applications can benefit from either the speed of a large core or the efficiency of a small core at various points in their execution. Further, the best fit of application to processor can also be dependent on the system context—for example, whether a laptop is running off a battery or off wall power.

Thus, we believe the best choice in core complexity is "all of the above"— a heterogeneous chip microprocessor with both high- and low-complexity cores. Recent research in heterogeneous, or asymmetric, CMPs has identified significant advantages over homogeneous CMPs in terms of power and throughput and in addressing the effects of Amdahl's law on the performance of parallel applications.

HETEROGENEITY'S POTENTIAL

Table 1 shows the power dissipation and performance of four generations of Alpha microprocessor cores scaled to the same 0.10 μm feature size and assumed to run at the same clock frequency. Figure 1 shows the relative sizes of these cores. All cores put together comprise less than 15 percent more area than EV8—the largest core—by itself. This data is representative of the past 20 years of microprocessor evolution, and similar data exists for x86 processors.[3]

Although the number of transistors per microprocessor core has increased greatly—with attendant increases in area, power, and design complexity—this complexity has caused only a modest increase in application performance, as opposed to performance due to faster clock rates from technology scaling. Thus, while the more complex cores provide higher

Table 1. Power and relative performance of Alpha cores scaled to 0.10 μm. Performance is expressed normalized to EV4 performance.

Core	Peak power (Watts)	Average power (Watts)	Performance (norm. IPC)
EV4	4.97	3.73	1.00
EV5	9.83	6.88	1.30
EV6	17.8	10.68	1.87
EV8	92.88	46.44	2.14

single-thread performance, this comes with a loss of area and power efficiency.

Further, in addition to varying in their resource requirements, applications can have significantly different resource requirements during different phases of their execution. This is illustrated for the applu benchmark in Figure 2.

Some application phases might have a large amount of instruction-level parallelism (ILP), which can be exploited by a core that can issue many instructions per cycle, that is, a wide-issue superscalar CPU. The same core, however, might be very inefficient for an application phase with little ILP, consuming significantly more power (even after the application of gating- or voltage/frequency-scaling-based techniques) than a simpler core that is better matched to the application's characteristics. Therefore, in addition to changes in performance over time, significant changes occur in the relative performance of the candidate cores.

In Figure 2, sometimes the difference in performance between the biggest and smallest core is less than a factor of two, sometimes more than a factor of 10. Thus, the best core for executing an application phase can vary considerably during a program's execution. Fortunately, much of the benefit of heterogeneous execution can be obtained with relatively infrequent switching between cores, on the order of context-switch intervals. This greatly reduces the overhead of switching between cores to support heterogeneous execution.

Heterogeneous multicore architectures have been around for a while, in the form of system-on-chip designs, for example. However, in such systems, each core performs a distinct task. More recently, researchers have proposed multi-ISA multicore architectures.[4] Such processors have cores that execute instructions belonging to different ISAs, and they typically address vector/data-level parallelism and instruction-level parallelism simultaneously. Not all cores can execute any given instruction. In contrast, in single-ISA heterogeneous CMPs, each core executes the same ISA, hence each application or application phase can be mapped to any of the cores. Single-ISA heterogeneous CMPs are an example of a multicore-aware processor architecture. The "Multicore-Aware Processor Architecture" sidebar provides additional information about this type of architecture.

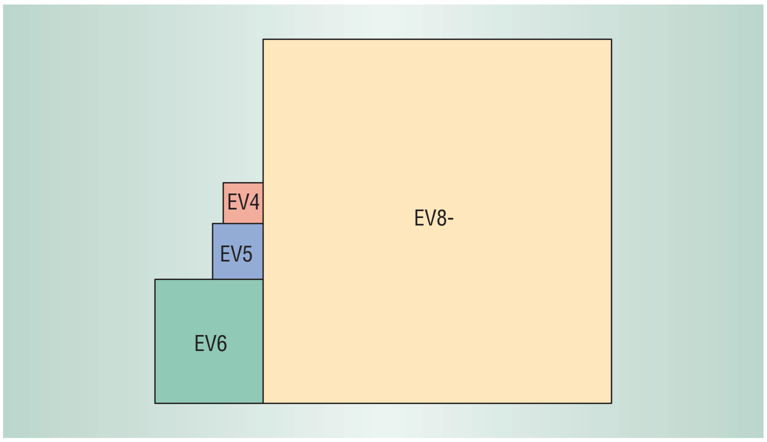

Figure 1. Relative sizes of the Alpha cores scaled to 0.10 μm. EV8 is 80 times bigger but provides only two to three times more single-threaded performance.

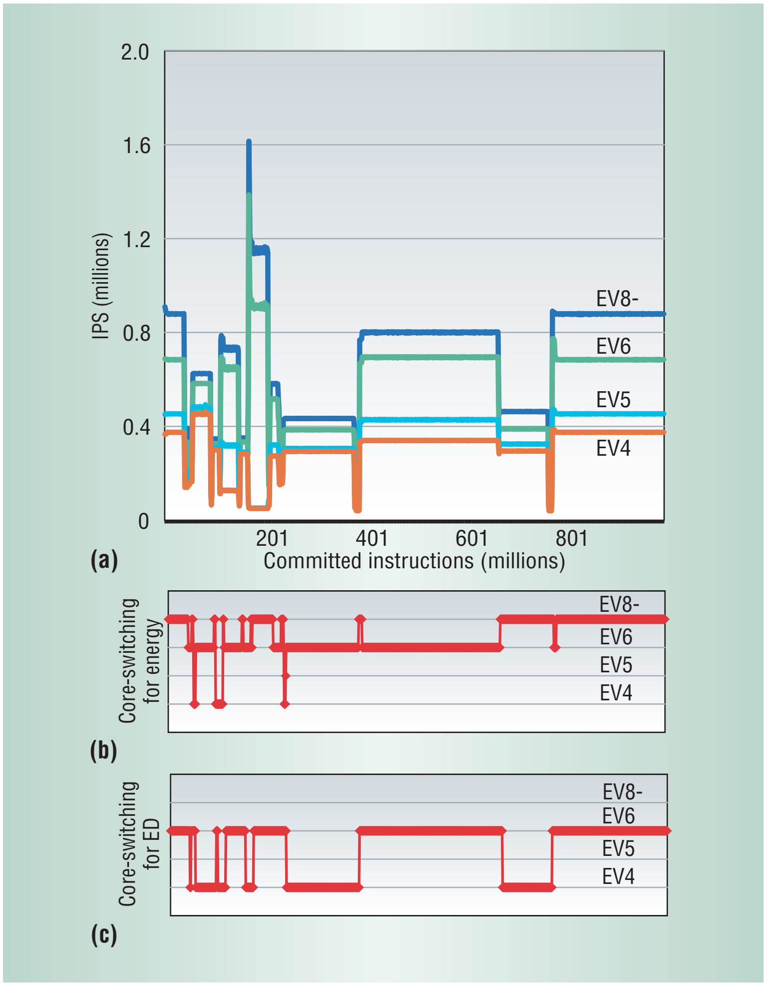

Figure 2. Applu benchmark resource requirements. (a) Performance of applu on the four cores; (b) Oracle switching for energy; and (c) Oracle switching for energy-delay product. Switchings are infrequent, hence total switching overhead is minimal.

POWER ADVANTAGES

Using single-ISA heterogeneous CMPs can significantly reduce processor power dissipation. As processors continue to increase in performance and speed, processor power consumption and heat dissipation have become key challenges in the design of future high-performance systems. Increased power consumption and heat dissipation typically lead to

Multicore-Aware Processor Architecture

As we move from a world dominated by uniprocessors to one likely to be dominated by multiprocessors on a chip, we have the opportunity to approach the architecture of these systems in new ways. Developers need to architect each piece of the system not to stand alone, but to be a part of the whole. In many cases, this requires very different thinking than what prevailed for uniprocessor architectures.

Heterogeneous CMP design is just one example of multicore-aware architecture. In the uniprocessor era, we designed one architecture for the universe of applications. Thus, for applications that demand high instruction throughput, applications with variable control flow and latencies, applications that access large data sets, or applications with heavy floating-point throughput, the best processor is superscalar and dynamically scheduled, with large caches and multiple floating-point units, and so on. However, few, if any, applications actually need all those resources. Thus, such an architecture is highly overprovisioned for any single application.

In a chip multiprocessor, however, no single core need be ideal for the universe of applications. Employing heterogeneity exploits this principle. Conversely, a homogeneous design actually exacerbates the overprovisioning problem by creating a single universal design, then replicating that overprovisioned design across the chip.

Heterogeneous CMP design, however, is not the only example of multicore-aware architecture: Two other examples are a conjoined core architecture and CMP/interconnect codesign.

Blurring the lines between cores

Some level of processor overprovisioning is necessary for market and other considerations, whether on homogeneous or heterogeneous CMPs, because it increases each core's flexibility. What we really want, though, is to have the same level of overprovisioning available to any single thread without multiplying the cost with the number of cores. In conjoined-core chip multiprocessing,[1] for example, adjacent cores share overprovisioned structures by requiring only minor modifications to the floor plan.

In the uniprocessor era, the lines between cores were always distinct, and the cores could share only very remote resources. With conjoined cores, those lines aren't necessary on a CMP.

Figure A shows the floorplan of two adjacent cores of a CMP sharing a floating-point unit and level-1 caches. Each core can access the shared structures either in turn during fixed allocated cycles—for example, one core gets access to the shared structure every odd cycle while the other core gets access every even cycle—or sharing can be based on certain dynamic conditions visible to both the cores. Sharing should occur without communication between cores, which is expensive.

Conjoining reduces the number of overprovisioned structures by half. In the Figure A example, conjoining results in having only four floating-point units on an eight-core CMP—one per conjoined-core pair. Each core gets full access to the floating-point unit unless the other core needs access to it at the same time. Applying intelligent, complexity-effective sharing mechanisms can minimize the performance impact of reduced bandwidth between the shared structure and a core.

The chief advantage of having conjoined cores is a significant reduction in per-core real estate with minimal impact on per-core performance, providing a higher computational capability per unit area. Conjoining can result in reducing the area devoted to cores by half, with no more than a 10 to 12 percent degradation in single-thread performance.[1] This can be used either to decrease the area of the entire die, increase the yield, or support more cores for a given fixed die size. Ancillary benefits include a reduction in leakage power from fewer transistors for a given computational capability.

CMP/interconnect codesign

In the uniprocessor era, high-performance processors connected by high-performance interconnects yielded high-performance systems. On a CMP, the design issues are more complex because the cores, caches, and interconnect all reside on the same chip and compete for the same area and power budget. Thus, the design choices for the cores, caches, and interconnects can interact to a much greater degree. For example, an aggressive

higher costs for thermal packaging, fans, electricity, and even air conditioning. Higher-power systems can also have a greater incidence of failures.

Industry currently uses two broad classes of techniques for power reduction: gating-based and voltage/frequency-scaling-based. Both of these techniques exploit program behavior for power reduction and are applied at a single-core level. However, any technique applied at this level suffers from limitations. For example, consider clock gating. Gating circuitry itself has power and area overhead, hence it typically cannot be applied at the lowest granularity. Thus, some dynamic power is still dissipated even for inactive blocks. In addition, data must still be transmitted long distances over unused portions of the chip that have been gated off, which consumes a substantial amount of power. Also, gating helps reduce only the dynamic power. Large unused portions of the chip still dissipate leakage power. Voltage/frequency scaling-based techniques suffer from similar limitations.

Given the ability to dynamically switch between cores and power down unused cores to eliminate leakage, recent work has shown reductions in processor energy delay product as high as 84 percent (a sixfold improvement) for individual applications and 63 percent overall.[5] Energy-delay^2—the product of energy and the square of the delay—reductions are as high as 75 percent (a fourfold improvement), 50 percent overall.

Ed Grochowski and colleagues[3] found that using asymmetric cores could easily improve energy per interconnect design consumes power and area resources that can then constrain the number, size, and design of cores and caches. Similarly, the number and type of cores, as well as on-chip memory, can also dictate requirements on the interconnect. Increasing the number of cores places conflicting demands on the interconnect, requiring higher bandwidth while decreasing available real estate.

A recent study shows how critical the interconnect can be for multicore design.[2] On an eight-core processor, for example, even under conservative assumptions, the interconnect can consume the power equivalent of one core, take the area equivalent of three cores, and add delay that accounts for over half the L2 access latency.

Such high overheads imply that it isn't possible to design a good interconnect in isolation from the design of the CPU cores and memory. Cores, caches, and interconnect should all be co-designed for the best performance or energy efficiency. A good example of the need for codesign is that decreasing interconnection bandwidth can sometimes improve performance due to the constrained window on total resources—for example, if it enables larger caches that decrease interconnect pressure. In the same way, excessively large caches can also decrease performance when they constrain the interconnect to too small an area.

Hence, designing a CMP architecture is a system-level optimization problem. Some common architectural beliefs do not hold when interconnection overheads are properly accounted for. For example, shared caches are not as desirable compared to private caches if the cost of the associated crossbar is carefully factored in.[2]

Any new CMP architecture proposal should consider interconnect as a first-class citizen, and all CMP research proposals should include careful interconnect modeling for correct and meaningful results and analysis.

References

1. R. Kumar, N.P. Jouppi, and D. Tullsen, "Conjoined-Core Chip Multiprocessing," *Proc. Int'l Symp. Microarchitecture*, IEEE CS Press, 2004, pp. 195-206.
2. R. Kumar, V. Zyuban, and D. Tullsen, "Interconnection in Multicore Architectures: Understanding Mechanisms, Overheads, and Scaling," *Proc. Int'l Symp. Computer Architecture*, IEEE CS Press, 2005, pp. 408-419.

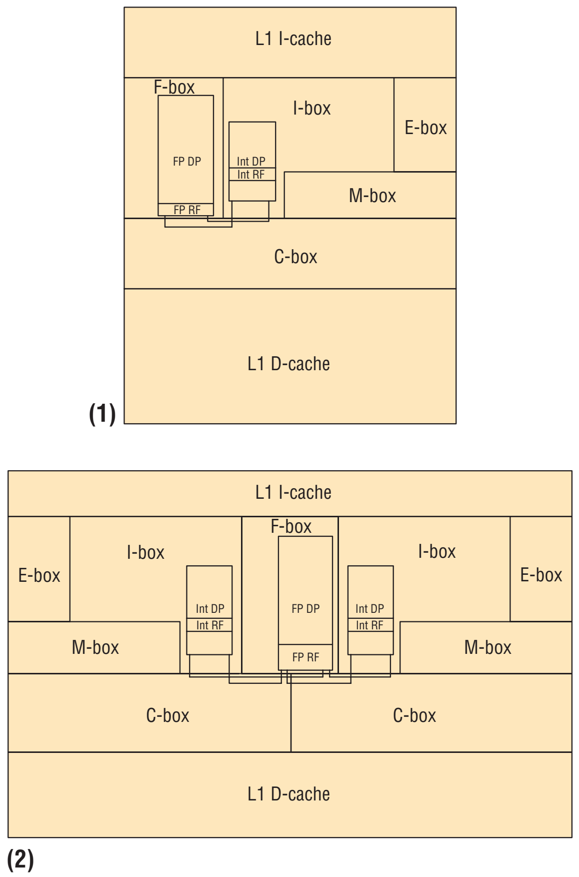

Figure A. CMP floorplan. (1) The original core and (2) a conjoined-core pair, both showing floating-point unit routing. Routing and register files are schematic and not drawn to scale.

instruction by four to six times. In comparison, given today's already low core voltages of around 1 volt, voltage scaling could provide at most a two to four times improvement in energy per instruction. Various types of gating could provide up to two times improvement in energy per instruction, while controlling speculation could reduce energy per instruction by up to 40 percent. These techniques are not mutually exclusive, and voltage scaling is largely orthogonal to heterogeneity. However, heterogeneity provided the single largest potential improvement in energy efficiency per instruction.

THROUGHPUT ADVANTAGES

For two reasons, given a fixed circuit area, using heterogeneous multicore architectures instead of homogeneous CMPs can provide significant performance advantages for a multiprogrammed workload.[6] First, a heterogeneous multicore architecture can match each application to the core best suited to meet its performance demands. Second, it can provide improved area-efficient coverage of the entire spectrum of workload demands seen in a real machine, from low thread-level parallelism that provides low latency for few applications on powerful cores to high thread-level parallelism in which simple cores can host a large number of applications simultaneously.

So, for example, in a homogeneous architecture with four large cores, it would be possible to replace one large core with five smaller cores, for a total of eight cores. In the best case, intelligently

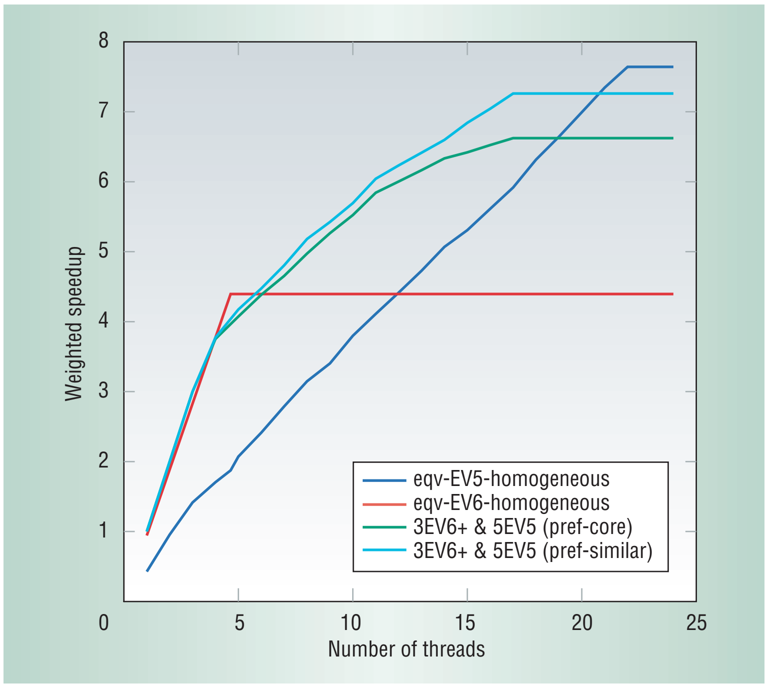

Figure 3. Performance of heuristics for equal-area heterogeneous architectures with multithreaded cores. EV6+ is an SMT variant of EV6. Pref-core biases sampling toward a new core over a new SMT context. Pref-similar biases sampling toward the last good configuration.

scheduling jobs on the smaller cores that would have seen no significant benefit from the larger core would yield the performance of eight large cores in the space of four.

Overall, a representative heterogeneous processor using two core types achieves as much as a 63 percent performance improvement over an equivalent-area homogeneous processor. Over a range of moderate load levels—five to eight threads, for example—the heterogeneous CMP has an average gain of 29 percent. For an open system with random job arrivals, the heterogeneous architecture has a much lower average response time and remains stable for arrival rates 43 percent higher than for a homogeneous architecture. Dynamic thread-to-core assignment policies have also been demonstrated that realize most of the potential performance gain. One simple assignment policy outperformed naive core assignment by 31 percent.[6]

Heterogeneity also can be beneficial in systems with multithreaded cores. Despite the additional scheduling complexity that simultaneous multithreading cores pose due to an explosion in the possible assignment permutations, effective assignment policies can be formulated that do derive significant benefit from heterogeneity.[6]

Figure 3 shows the performance of several heuristics for heterogeneous systems with multithreaded cores. These heuristics prune the assignment space by making assumptions regarding the relative benefits of running on a simpler core versus running on a simultaneous multithreaded core. Various sampling policies are used to choose good assignments from the reduced assignment space.

We observed that learning policies that assume the current configuration has merit and that the next configuration will perform particularly well.[6] The graph in Figure 3 also demonstrates the primary benefit of heterogeneity. Using four big cores yields good few-threads performance, and using many small cores (more than 20) yields high peak throughput. Only the heterogeneous architecture provides high performance across all levels of thread parallelism.

With these policies, this architecture provides even better coverage of a spectrum of load levels. It provides the low latency of powerful processors at low threading levels, but is also comparable to a larger array of small processors at high-thread occupancy.

These thread assignment heuristics can be quite useful even for homogeneous CMPs in which each core is multithreaded. Such CMPs face many of the same problems regarding explosion of assignment space. In some sense, such CMPs can be thought of as heterogeneous CMPs for scheduling purposes where the heterogeneity stems from different marginal performance and power characteristics for each SMT context.

As computing objectives keep switching back and forth between single-thread performance and throughput, we believe that single-ISA heterogeneous multicore architectures provide a convenient and seamless way to address both concerns simultaneously.

MITIGATING AMDAHL'S LAW

Amdahl's law[7] states that the speedup of a parallel application is limited by the fraction of the application that is serial. In modern CMPs, the overall power dissipation is an important limit. Murali Annavaram and coauthors[8] point out a useful application for heterogeneous CMPs in power-constrained environments.

During serial portions of execution, the chip's power budget is applied toward using a single large core to allow the serial portion to execute as quickly as possible. During the parallel portions, the chip's power budget is used more efficiently by running the parallel portion on a large number of small area- and power-efficient cores. Thus, executing serial portions of an application on a fast but relatively inefficient core and executing parallel portions of an algorithm on many small cores can maximize the ratio of performance to power dissipation.

Using a simple prototype built from a discrete four-way multiprocessor, Annavaram and colleagues show a 38 percent wall clock speedup for a parallel application given a fixed power budget. Single-chip heterogeneous multiprocessors with larger numbers of processors should be able to obtain even larger improvements in speed/power product on parallel applications.

WHAT HETEROGENEITY MEANS FOR SOFTWARE

To take full advantage of heterogeneous CMPs, the system software must use the execution characteristics of each application to predict its future processing needs and then schedule it to a core that matches those needs if one is available. The predictions can minimize the performance loss to the system as a whole rather than that of a single application.

Recent work has shown that effective schedulers for heterogeneous architectures can be implemented and integrated with current commercial operating systems.[9] An experimental platform running Gentoo Linux with a 2.6.7 kernel modified to support heterogeneity-aware scheduling resulted in a 40 percent power savings, for a performance loss of less than 1 percent for memory-bound applications. Less than a 3.5 percent performance degradation was observed even for CPU-intensive applications.

To achieve the best performance, it might be necessary to compile programs for heterogeneous CMPs slightly differently. Compiling single-threaded applications might involve either compiling for the lowest common denominator or compiling for the simplest core. For example, for heterogeneous CMPs in which one core is statically scheduled and one is dynamically scheduled, the compiler should schedule the code for the statically scheduled core because it is more sensitive to the exact instruction order than is a dynamically scheduled core. Having multiple statically scheduled cores with different levels of resources would present a more interesting problem.

Programming or compiling parallel applications might require more awareness of the heterogeneity. Application developers typically assume that computational cores provide equal performance; heterogeneity breaks this assumption. As a result, shared-memory workloads that are compiled assuming symmetric cores might have less predictable performance on heterogeneous CMPs.[10] For such workloads, either the operating system kernel or the application must be heterogeneity-aware.

Mechanisms for communicating the complete processor information to software and the design of software to tolerate heterogeneity need more investigation. As future systems include support for greater virtualization, similar issues must be addressed at the virtual machine layer as well.

These software changes will likely enhance the already demonstrated advantages of heterogeneous CMPs.

Effective schedulers for heterogeneous architectures can be implemented and integrated with current commercial operating systems.

FURTHER RESEARCH QUESTIONS

Many areas of future research remain for heterogeneous CMPs. For example, research to date has been performed using off-the-shelf cores or models of existing off-the-shelf cores. If designers are given the flexibility to design asymmetric CMP cores from scratch instead of selecting from predetermined cores, how should they design them to complement each other best in a heterogeneous CMP? How do the benefits of heterogeneity vary with the number of core types as a function of the available die area and the total power budget?

In previous studies, simple cores were a strict subset of the more complex cores.[5,6] What benefits are possible if all the resources in the cores are not monotonically increasing? Further, as workloads in enterprise environments evolve toward a model that consolidates multiple services on the same hardware infrastructure,[11] heterogeneous architectures offer the potential to match core diversity to the diversity in the varying service-level agreements for the different services. What implications does this have on the choice and design of the individual cores?

These are just some of the open research questions. However, we believe answering these and similar questions could show that heterogeneity is even more advantageous than has already been demonstrated.

Future work should also address the impact of heterogeneity on the cost of design and verification. Processor design and verification costs are already high. Designing and integrating more than one type of core on the die would aggravate this problem. However, the magnitude of the power savings and the throughput advantages that heterogeneous CMPs provide might justify these costs, at least for limited on-chip heterogeneity.

What is the sensitivity of heterogeneous CMP performance to the number of distinct core types? Are two types enough? How do these costs compare with the cost for other contemporary approaches such as different voltage levels? The answers to these questions might ultimately determine both the feasibility and the extent of on-chip diversity for heterogeneous CMPs.

Increasing transistor counts constrained by power limits point to the fact that many of the current processor directions are inadequate. Monolithic processors consume too much power, but they do not provide enough marginal performance. Replicating existing processors results in a linear increase in power, but only a sublinear increase in performance. In addition to suffering from the same limitations, replicating smaller processors cannot handle high-demand and high-priority applications.

Single-ISA heterogeneous (or asymmetric) multicore architectures address all these concerns, resulting in significant power and performance benefits. The potential benefits from heterogeneity have already been shown to be greater than the potential benefits from the individual techniques of further voltage scaling, clock gating, or speculation control.

Much research remains to be done on the best types and degrees of heterogeneity. However, the advantages of heterogeneous CMPs for both throughput and power have been demonstrated conclusively. We believe that once homogeneous CMPs reach a total of four cores, the benefits of heterogeneity will outweigh the benefits of additional homogeneous cores in many applications. ■

References

1. K. Olukotun et al., "The Case for a Single-Chip Multiprocessor," *Proc. 7th Int'l Conf. Architectural Support for Programming Languages and Operating Systems (*ASPLOS VII), ACM Press, 1996, pp. 2-11.
2. L. Barroso et al., "Piranha: A Scalable Architecture Based on Single-Chip Multiprocessing," *Proc. 27th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, 2000, pp. 282-293.
3. E. Grochowski et al., "Best of Both Latency and Throughput," *Proc. Int'l Conf. Computer Design*, IEEE CS Press, 2004, pp. 236-243.
4. D. Pham et al., "The Design and Implementation of a First-Generation Cell Processor," *Proc. Int'l Symp. Solid-State Circuits and Systems*, IEEE CS Press, 2005, pp. 184-186.
5. R. Kumar et al., "Single-ISA Heterogeneous Multicore Architectures: The Potential for Processor Power Reduction," *Proc. Int'l Symp. Microarchitecture*, IEEE CS Press, 2003, pp. 81-92.
6. R. Kumar et al., "Single-ISA Heterogeneous Multicore Architectures for Multithreaded Workload Performance," *Proc. Int'l Symp. Computer Architecture*, IEEE CS Press, 2004, pp. 64-75.
7. G. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," *Readings in Computer Architecture*, M.D. Hill, N.P. Jouppi, and G.S. Sohi, eds., Morgan Kaufmann, 2000, pp. 79-81.
8. M. Annavaram, E. Grochowski, and J. Shen, "Mitigating Amdahl's Law through EPI Throttling," *Proc. Int'l Symp. Computer Architecture*, IEEE CS Press, 2005, pp. 298-309.
9. S. Ghiasi, T. Keller, and F. Rawson, "Scheduling for Heterogeneous Processors in Server Systems," *Proc. Computing Frontiers*, ACM Press, 2005, pp. 199-210.
10. S. Balakrishnan et al., "The Impact of Performance Asymmetry in Emerging Multicore Architectures," *Proc. Int'l Symp. Computer Architecture*, IEEE CS Press, 2005, pp. 506-517.
11. P. Ranganathan and N.P. Jouppi, "Enterprise IT Trends and Implications on System Architecture Research," *Proc. Int'l Conf. High-Performance Computer Architecture*, IEEE CS Press, 2005, pp. 253-256.

***Rakesh Kumar** is a PhD student in the Department of Computer Science and Engineering at the University of California, San Diego. His research interests include multicore and multithreaded architectures, low-power architectures, and on-chip interconnects. Kumar received a BS in computer science and engineering from the Indian Institute of Technology, Kharagpur. He is a member of the ACM. Contact him at rakumar@cs.ucsd.edu.*

***Dean M. Tullsen** is an associate professor in the Department of Computer Science and Engineering at the University of California, San Diego. His research interests include instruction- and thread-level parallelism and multithreaded and multicore architectures. Tullsen received a PhD in computer science and engineering from the University of Washington. He is a member of the IEEE and the ACM. Contact him at tullsen@cs.ucsd.edu.*

***Norman P. Jouppi** is a Fellow at HP Labs in Palo Alto, Calif. His research interests include multicore architectures, memory systems, and cluster interconnects. Jouppi received a PhD in electrical engineering from Stanford University. He is a Fellow of the IEEE and currently serves as chair of ACM SIGARCH. Contact him at norm.jouppi@hp.com.*

***Parthasarathy Ranganathan** is a principal research scientist at HP Labs in Palo Alto, Calif. His research interests include power-efficient design, computer architectures, and system evaluation. Ranganathan received a PhD in electrical and computer engineering from Rice University. He is a member of the IEEE and the ACM. Contact him at partha.ranganathan@hp.com.*