RESEARCH STATEMENT

Shantanu Gupta (shangupt@umich.edu)

My interests span the field of computer architecture and compiler technology with a focus on system reliability, performance and energy-efficiency. Within this scope, I have worked on numerous projects during the course of my doctoral research, industrial internships and collaboration within the University of Michigan. In the reliability domain, I have investigated hard-fault tolerance (both in processors and caches), soft-error tolerance, and concurrency bugs in parallel programs. In the performance domain, I have developed microarchitectural solutions for enabling dynamic multicores, which can cater to situations requiring single-thread performance, throughput computing and anything in between. And finally, in the energy-efficiency domain, I am exploring configurable compute engines that can save a large fraction of instruction and data supply energy.

Dissertation Research

With the increasing silicon integration, transistor today are cheaper and faster than ever before. This transistor scaling has long been a source of dramatic performance gains. However, at the same time, it has resulted in increasing levels of operating temperatures and power densities which can have serious repercussions on a chip's reliability, performance and computational efficiency. For instance, given that most silicon wearout mechanisms are highly dependent on chip temperatures and device sizes, significantly higher failure rates are projected for future technology generations. In modern multicore chips, this can jeopardize the objective of throughput sustainability over the lifetime of a chip. In terms of performance, multicore chips prevalent today (chosen as an alternative to complex monolithic designs) are effective for throughput computing, but they provide small gains for sequential applications. Even if a major transition towards parallel programming occurs in the future, Amdahl's law dictates that the sequential component of an application will present itself as a performance bottleneck. And lastly, going forward, chip-wide power and energy constraints will limit the number of cores / resources that can be kept active on a chip, motivating the need for highly energy-efficient computing.

My thesis is on design of adaptive architectures to deal with all of the issues discussed above. Further, the solutions proposed are complementary to each other, and when applied together, can effectively tackle reliability, performance and energy-efficiency demands expected in future microprocessors.

Hard Fault Tolerance (StageNet, 2007-10)

Traditionally, hard-faults in high-end servers and mission critical systems have been addressed by using mechanisms such as dual and triple-modular redundancy. However, such solutions incur high hardware overheads and can tolerate only a small number of defects. As a new direction in hard-fault tolerance paradigm, I proposed StageNet, a fine-grained redundancy solution for multicore chips. StageNet is a highly reconfigurable multicore architecture that is designed as a network of pipeline stages, rather than isolated cores. Its interconnection flexibility allows it to salvage health pipeline stages, by adaptively routing around defective ones in the multicore fabric. This fine-grained defect isolation enables StageNet to maintain a higher throughput over a system's lifetime compared to a conventional multicore chip. The primary challenge in this project was the design of a decoupled pipeline microarchitecture that allows pipeline stages from different cores to assemble together and form a logical processor. The original decoupled pipeline design appeared in CASES'08 [1] and the full system in MICRO'08 [2], TOC'10 [3]. A scalable and process variation tolerant version of StageNet also appeared in DSN'10 [4].

Hard Fault Detection (Adaptive Testing, 2008-09)

Given a scenario with increasing failure rates in commodity systems, processors would need to be equipped with fault tolerance mechanisms that can detect in-field silicon defects. In this project, I proposed an adaptive on-line testing framework to significantly reduce the overhead of in-field hard fault detection. The insight here was to leverage health monitoring sensors to guide the amount of testing applied to different components in a chip. Using this approach, a significant chunk of the periodic test time can be saved for the healthy components. This work appeared in ICCD'09 [5] and won the best paper award.

Unified Performance-Reliability Solution (CoreGenesis, 2009-10)

Single-thread performance, reliability and power-efficiency are critical design challenges of future multicore systems. In this project, our objective was to build upon the StageNet architecture (reliability solution), and construct an architectural platform that can tackle a variety of challenges seamlessly. Towards this end, we proposed CoreGenesis, a dynamically adaptive multiprocessor fabric that blurs out individual core boundaries, and encourages resource sharing across cores for performance and reliability. The CoreGenesis architectures relies on interconnection flexibility, microarchitectural innovations, and compiler directed instruction steering, to merge pipeline resources for high single-thread performance. The same flexibility enables it to route around broken components, achieving sub-core level defect isolation. Together, the resulting fabric consists of a pool of pipeline stage-level resources that can be fluidly allocated for accelerating single-thread performance, throughput computing, or tolerating failures. This work was accepted for publication in MICRO'10 [6].

Energy-efficient Execution (Configurable Accelerator, 2010-present)

The unprecedented levels of technology scaling has introduced tight power constraints on manufactured parts. As a consequence, only a fraction of a chip can run at peak capacity at one time. In this scenario, there is a need to incorporate energy efficient processing resources that can enable more computation within the same energy budget. In this ongoing project, I am designing a configurable accelerator fabric that offers the flexibility to across application domains (including highly irregular integer benchmarks), and relies on recurring instruction sequences to significantly cut down on instruction fetch, decode, and register file access energy. This is a significant leap over prior efforts that focused either on application specific chips (ASICs) or loop accelerators that were only effective on media kernels and floating point benchmarks with tight innermost loops.

Industrial Research Projects

Data Race Detection (RaceTM, NEC Labs, 2007)

Widespread emergence of multicore processor is spurring the development of parallel applications. Along with its performance benefits, multi-threaded applications also introduce non deterministic and notoriously hard to reproduce synchronization bugs manifested through data races. Previous solutions to dynamic data race detection have required specialized hardware, at additional power, design and area costs. In this project, we proposed RaceTM, a novel approach to data race detection that exploits hardware that will likely be present in future multiprocessors, albeit for a different purpose. In particular, we show how emerging hardware support for transactional memory can be leveraged to aid data race detection. RaceTM introduces the concept of lightweight debug transactions that exploit the conflict detection mechanisms of transactional memory systems to perform data race detection. The details of the scheme, a proof-of-concept simulation prototype, and its evaluation on applications from the SPLASH-2 suite appear in SPAA'08 [7], IPDPS'09 [8].

Salvaging Broken Cores (Intel Hudson, 2008)

In this project, we made a case for architectural core salvaging to meet the challenge of hard-faults in future technology generations. The main observation here was that for a subset of faults, even a defective core can execute a large fraction of the ISA instructions correctly. And whenever a broken core encounters an instruction it cannot execute, we can migrate the thread to a fully functioning core. Given that modern chips are constituted of multiple cores, natural cross-core redundancy can thus be exploited to complement the functionality of broken cores. We showed that this hardware thread migration technique can effectively cover 20-30% of hard faults on an Intel-like core. Overall, the performance on a faulty die (an 8-core chip with one broken core) approaches that of a fault-free die. This work appeared in ISCA'09 [9].

Collaborative Projects at Michigan

In addition to my dissertation research and internship projects, I have actively contributed to research efforts lead by my colleagues at Michigan. All of these projects fall in the reliability domain, and are summarized below.

Hard Fault Tolerance (Wearout Detection, Thread Scheduling, Necromancer, ZerehCache)

Detecting the onset of a silicon defect is a challenging problem. In our research, we observed that most wearout mechanisms develop and intensify with age over the lifetime of a chip. Building upon this insight, we proposed a low-cost hardware structure that identifies increasing timing delay, which is symptomatic of many forms of wearout, to accurately forecast the failure within a core (MICRO'07 [10]). Further, we used these sensors to guide wearout-centric job scheduling in multicore chips, to balance the wearout across different cores [11]. Notably, detection and wearout-centric scheduling are only applicable *before* a failure happens. In a recent project, named Necromancer (ISCA'10 [12]), we looked at a technique to utilize a core *after* a failure happens. Necromancer is a heterogeneous core coupling scheme that exploits a functionally dead core to improve system throughput by supplying execution hints to a fully functional core. Apart from the processor logic, failures in large SRAM structures (like caches) are common due their high density and sensitivity to timing failures. In our ZerehCache (MICRO'09 [13]) project, we proposed a flexible and dynamically reconfigurable cache design that allows a large number of defects to be tolerated with modest hardware overhead.

Soft Error Tolerance (Register Caching, Shoestring)

In the class of reliability issues foreseen in future technology generations, soft errors are considered an immediate concern. We are quickly approaching a new era where resilience to soft errors is no longer a luxury that can be reserved for safety-critical systems. In light of this trend, we worked on two soft error tolerance solutions for commodity processors. First, we investigated the propagation of soft errors in the data path of an embedded core, and concluded that most errors end up corrupting the register file state. We used this insight to design register value cache (CASES'06 [14]), a small hardware structure that maintains a copy of most active register entries, and replaces them in case of a fault. Second, we developed a software-only technique, named Shoestring (ASPLOS'10 [15]), which is a minimally invasive compiler approach to detect soft errors. By leveraging intelligent analysis at compile time, and exploiting low-cost, symptom-based error detection, Shoestring is able to focus its efforts on protecting statistically-vulnerable portion of program code. For these vulnerable portions of the program, Shoestring uses instruction duplication to detect a mismatch in the event of a failure.

Future Directions

Going forward, with the ever changing technology landscape, I wish to engage in exciting research for developing robust and efficient computing systems. Further, I feel confident that my experience from working on a diverse set of research problems, and infrastructures that range from circuit level simulations all the way to OS job scheduling, have provided me with sufficient background to tackle complex cross-disciplinary research problems. Some of the near term ideas I am interested in investigating are:

1. *Hardware and software solutions for energy-efficient computing*
2. *Microprocessor timing error tolerance using techniques at compiler and architecture level*
3. *Configurable chip architectures for performance and reliability*

References

[1] S. Gupta, S. Feng, A. Ansari, J. A. Blome, and S. Mahlke, "Stagenetslice: A reconfigurable microarchitecture building block for resilient cmp systems," in *Proc. of the 2008 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 1–10, 2008.

[2] S. Gupta, S. Feng, A. Ansari, J. A. Blome, and S. Mahlke, "The stagenet fabric for constructing resilient multicore systems," in *Proc. of the 41st Annual International Symposium on Microarchitecture*, pp. 141–151, 2008.

[3] S. Gupta, S. Feng, A. Ansari, and S. Mahlke, "Stagenet: A reconfigurable fabric for constructing dependable cmps," *IEEE Transactions on Computers*, vol. 60, no. 1, 2011.

[4] S. Gupta, A. Ansari, S. Feng, and S. Mahlke, "Stageweb: Interweaving pipeline stages into a wearout and variation tolerant cmp fabric," in *Proc. of the 2010 International Conference on Dependable Systems and Networks*, June 2010.

[5] S. Gupta, A. Ansari, S. Feng, and S. Mahlke, "Adaptive online testing for efficient hard fault detection," in *Proc. of the 2009 International Conference on Computer Design*, 2009.

[6] S. Gupta, S. Feng, A. Ansari, and S. Mahlke, "Erasing core boundaries for robust and configurable performance," in *Proc. of the 43rd Annual International Symposium on Microarchitecture*, 2010.

[7] S. Gupta, F. Sultan, S. Cadambi, F. Ivancic, and M. Roetteler, "Racetm: Detecting data races using transactional memory," in *SPAA '08: 20th Annual ACM Symposium on Parallel Algorithms and Architectures*, 2008.

[8] S. Gupta, F. Sultan, S. Cadambi, F. Ivancic, and M. Roetteler, "Using hardware transactional memory for data race detection," in *2009 IEEE International Symposium on Parallel and Distributed Processing*, 2009.

[9] M. D. Powell, A. Biswas, S. Gupta, and S. S. Mukherjee, "Architectural core salvaging in a multi-core processor for hard-error tolerance," in *Proc. of the 36th Annual International Symposium on Computer Architecture*, June 2009.

[10] J. A. Blome, S. Feng, S. Gupta, and S. Mahlke, "Self-calibrating online wearout detection," in *Proc. of the 40th Annual International Symposium on Microarchitecture*, pp. 109–120, 2007.

[11] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Maestro: Orchestrating lifetime reliability in chip multiprocessors," in *Proc. of the 2010 International Conference on High Performance Embedded Architectures and Compilers*, pp. 186–200, Jan. 2010.

[12] A. Ansari, S. Feng, S. Gupta, and S. A. Mahlke, "Necromancer: enhancing system throughput by animating dead cores," in *Proc. of the 37th Annual International Symposium on Computer Architecture*, pp. 473–484, 2010.

[13] A. Ansari, S. Gupta, S. Feng, and S. Mahlke, "Zerehcache: Armoring cache architectures in high defect density technologies," in *Proc. of the 42nd Annual International Symposium on Microarchitecture*, pp. 100–110, 2009.

[14] J. A. Blome, S. Gupta, S. Feng, S. Mahlke, and D. Bradley, "Cost-efficient soft error protection for embedded microprocessors," in *Proc. of the 2006 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 421–431, 2006.

[15] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Shoestring: Probabilistic soft-error reliability on the cheap," in *18th International Conference on Architectural Support for Programming Languages and Operating Systems*, Mar. 2010.