
I am a computer architect specializing in system architectures that utilize the characteristics of applications, users, and materials in a holistic manner.

Computer Architecture serves as an interface between technology trends and marketplace demands. Traditionally, a computer system is usually represented as consisting of five abstraction levels: hardware, firmware, assembler, operating system and applications [1]. My research questions this fundamental definition. Instead of viewing computer architecture as an interface, I aim at introducing a holistic view on computing that involves users, applications, as well as materials. Specifically, during my PhD, I have worked on architectural optimizations that consider new layers lying at two extreme ends of the current set of abstraction levels: systems, users, and materials.

As we move into deeper sub-micron technologies, the complexity of pushing the performance of processors further faces important obstacles. During my PhD, I tried to invent methods that go beyond the traditional architectural approaches to increase performance. Most importantly, I argued that the collaboration of several layers (e.g., circuits and architectures) is likely to result in architectures that are not possible otherwise. For example, applications lie at the top of the whole spectrum for a computer architect. In other words, architects look into application characteristics and optimize the performance of their system accordingly. The ever-increasing need for improvement in system performance and power utilization led me to believe that we need to look beyond the application level to utilize the system resources intelligently and efficiently. For example, my recent work on user-aware power optimization shows that by considering the user satisfaction during system configuration can result in significant improvements [2]. On the other side of the spectrum, analysis of the materials that lie at the lowermost end of the abstraction level opens up a number of opportunities to optimize the system performance. In general, the layered approach is not limited to architects. Similarly, circuit designers also have to be conservative because of their own layered view on design, i.e., circuit designers typically consider the worst-case scenario to predict the default voltage properties of a processor chip. The hard constraint of reliability has created a gap between the default value and the threshold where a circuit can work flawlessly. In a recent work, I have shown that by customizing the system configuration to remedy the effects of process variations can improve the architecture performance [3, 4]. A final aspect of the holistic computing is the consideration of system properties during the architectural design process. I have shown that treating the correctness as an objective can improve the system performance with noted reductions in power consumption. Specifically, I have developed a scheme for networking systems that take advantage of the system-level resiliency [5]. If a system's inherent robustness can be used to correct faults introduced due to loosened reliability, much power and performance gain can be achieved. I have proposed novel schemes that utilize system attributes to ensure correctness of an application. It creates an opportunity to improve performance in lieu of loosened reliability. The results obtained from these research works have shown that the idea of the "*holistic computer architectures*" can be utilized by various processor architectures. My long-term goal is to establish a framework for the holistic architectures. So far, I have contributed to the following results towards this goal:

- (1) "User Driven Frequency Scaling (UDFS)" [2-4] that takes advantage of the observation that personal preferences vary greatly among users and adjusts system performance based on user's activity.
- (2) "Process Driven Voltage Scaling (PDVS)" [3, 4] that takes advantage of the effect of process variation and operating temperature in setting voltage levels in new generation processors. PDVS can be immediately applied to existing DVFS algorithms.
- (3) "Clumsy Processing" [5] that treats correctness of an application as an objective and not as a constraint.

I describe my contributions in each of these topics as well as my other research activities in the following sections.

User Aware Frequency Scaling

To explore the role played by the human factor in computer architectures, I have analyzed individual user's preferences over the system performance during execution of different applications. A double blinded user study reveals that personal preferences vary greatly among users (and that a user's preferences vary dynamically during application run-time) [3, 6]. Existing Dynamic Voltage and Frequency Scaling (DVFS) techniques in high-performance processors select an operating point (CPU frequency and voltage) based on the utilization of the processor. While this approach integrates OS-level control, such control is pessimistic about the user. Indeed, it ignores the user, assuming that CPU utilization is a sufficient proxy. A high CPU utilization leads to a high frequency and high voltage, regardless of the user's satisfaction or expectation of performance.

To remedy this limitation, I have developed User Driven Frequency Scaling (UDFS) that dynamically adapts CPU frequency based upon direct user feedback – as opposed to tracking CPU utilization, as is done by current methods. This dynamic power management scheme automatically adapts to different users and applications. UDFS effectively employs user feedback to customize processor frequency to the individual user. This typically leads to significant power savings compared to existing dynamic frequency schemes that rely only on CPU utilization as feedback. The amount of feedback from the user is reasonable, and declines quickly over time as an application or set of applications is used. Hence, it can reduce power consumption while still achieving high user satisfaction.

Process Driven Voltage Setting

Existing DVFS techniques are pessimistic about the CPU as well. They assume worst-case manufacturing process variation and operating temperature by basing their policies on loose worst-case bounds given by the processor manufacturer. However, as the manufacturing technologies are getting smaller, this conservative assumption becomes an important bottleneck. Therefore, the “one-size-fits-all” approach of current DVFS schemes is suboptimal in the presence of process variations.

In response to this observation, I have developed a new power management technique, Process-Driven Voltage Scaling (PDVS). It adapts to process variation, permitting processors to operate at their lowest stable voltages. The UDFS and PDVS algorithms dramatically reduced typical operating frequencies and voltages while maintaining performance at a satisfactory level for each user. The techniques were evaluated independently and together through user studies conducted on a Pentium M laptop running Windows applications. Combining PDVS and the best UDFS scheme reduced measured system power by 49.9% (27.8% PDVS, 22.1% UDFS), averaged across all users and applications, compared to the Windows XP DVFS scheme. The average operating temperature of the CPU is decreased by 13.2°C.

Clumsy Processor

I have looked into the trade-off analysis between reliability of a processor and its performance [5, 7-9]. Traditionally, the circuit designers guarantee that the designed chip should work at the worst-case scenario. In my work, I have questioned this basic assumption about reliability. I have compromised the reliability of the system to gain in terms of performance. Please note that, while loosening the strict constraint on reliability, I have made sure that the system should not crash. Particularly, I looked into the networking applications that have an inherent robustness to detect and correct faults up to a certain threshold. I have shown that in such systems it is possible to gain in terms of power and performance by loosening the strict constraint on reliability. Note that, from the application’s perspective, correctness of execution is the primary objective. As long as the primary outputs of the application satisfy the correctness criteria, we can allow faults introduced in the system due to loosened reliability. I proposed simple, low cost schemes to detect and correct such faults subsequently. This type of processing is called “*Clumsy Processing*” and it results significant power and performance gain. Additionally, I proposed a generic parameter, the energy-delay-fallibility product metric, which can be used to measure the trade off between the energy, execution time, and the error probability. To perform a detailed analysis of the correctness-performance trade-off, I introduced a realistic model that determines the probability of a fault for a given cycle time of a cache and show that the delay of the cache and the energy consumed by the cache can be reduced significantly without incurring a large penalty due to faulty behavior. Using this model and cycle-accurate simulations, I investigated an optimal point for trading off the reliability for reducing cycle time of the data cache in a representative architecture. Moreover, I implemented a scheme to dynamically adjust the operation frequency of the data cache to achieve the desired objective (e.g., reduced energy).

Additional Interests

Holistic architecture being my primary interest, I have also contributed to research projects on the following topics.

Task allocation in CMP-based Network Processors based on statistical variation

I proposed a task allocation scheme that utilizes the probability distribution of the execution times of different modules in the networking applications [10]. The goal for the research is to minimize the effects of execution time variation. Note that, we can use a similar approach to solve the generic problem of task allocation in Chip Multiprocessors or any other scenario where tasks need to be distributed among a number of processing resources. Application domains exhibiting modular nature (e.g., data mining) may largely benefit from the proposed techniques.

Reliability-performance trade-offs for register files

Register files are in the critical path of most high-performance processors and their latency is one of the most important factors that limit their size. I developed error correction mechanisms at the architecture level [11]. I proposed novel techniques that utilize the fact that at a given instance many physical registers are not used in superscalar processors. These underutilized registers were used to store the values of active registers. The underutilization of register files can be efficiently used in superscalar embedded processors for reduction of power. The load instructions in representative embedded applications exhibited a large address locality. To take advantage of this observation, I devised a load elimination scheme [12], which tries to store the data values of load instructions in the register file.

Correlation based Cache Architecture

I introduced a new cache architecture that can be used to increase performance and reduce energy consumption in Network Processors. It was based on strong correlation between the source address of two consecutively executed load operations. This information was utilized by building a correlating cache architecture [13, 14].

Precision Analysis during High Level Synthesis

For my Masters thesis, I worked on the optimization for the bit-widths of fixed point variables in low power SystemC design environment [15, 16]. I proposed algorithms for optimal bit-width precision for two variables and a greedy heuristic which works for any number of variables. These schemes were used in the automation of converting floating point SystemC programs into ASIC synthesizable SystemC programs. The results showed that it is possible to trade-off the quantization error with the hardware resources used in the ASICs very effectively.

Future Directions

Ideally a computing system should be heterogeneous so that it can support the wide variety of platforms, networks and services. It should be capable of adapting to user preferences, ensure correctness in a variable environment and allow optimized performance in a reconfigurable environment. My holistic vision of system architecture would provide an efficient solution to this multifaceted requirement. In the short run, I plan to pursue my current philosophy of research by exploring and implementing microarchitectural enhancements for building the holistic architecture framework.

I believe that satisfaction of the user is the prime objective of any kind of automation. Typically the surroundings of a user can vary from a resource-rich environment (working with workstations) to resource-constraint settings (using a smartphone). The generic applicability of holistic architecture can be implemented to this whole gamut of environment. I would like to observe its effect on different application domains – embedded systems, networking hardware, high performance computing, to name a few. I feel the introduction of such hybrid architecture can benefit the whole population of computing systems. The new generation of autonomic system needs to fulfill two major constraints – fault-tolerance and fidelity-awareness. A fault tolerant system needs to detect and recover from fault for to meet the correctness objective. On the other hand, a fidelity-aware computing ensures the system can perform optimally with variability in available resources (CPU performance, power, network bandwidth, memory space). I intend to investigate novel microarchitecture techniques that improve system performance through optimizations at every abstraction level (user, application, operating system, assembler, firmware, hardware and materials). I believe the philosophy of *holistic computing architecture* would be one of the most effective tools in the design of next generation computing system.

References

- [1]. Tanenbaum, A.S., *Structured Computer Organization*. 1979, Englewood Cliffs, New Jersey: Prentice-Hall.
- [2]. Mallik, A., B. Lin, G. Memik, P. Dinda, and R.P. Dick, *User-Driven Frequency Scaling*. IEEE Computer Architecture Letters (CAL), 2007.
- [3]. Mallik, A., B. Lin, P. Dinda, G. Memik, and R.P. Dick, *Process and User Driven Dynamic Voltage and Frequency Scaling*, in *Technical Report NWU-EECS-06-11*. 2006, Department of Electrical Engineering and Computer Science, Northwestern University.
- [4]. Lin, B., A. Mallik, G. Memik, P. Dinda, and R.P. Dick. *Power Reduction through Measurement and Modeling of Users and Cpus*. in *The International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS 2007)*. 2007. California, USA.
- [5]. Mallik, A. and G. Memik. *A Case for Clumsy Packet Processors*. in *International Symposium on Microarchitecture*. Dec. 2004. Portland, OR.
- [6]. Dinda, P., G. Memik, R. Dick, B. Lin, A. Mallik, A. Gupta, and S. Rossoff. *The User in Experimental Computer Systems Research*. in *Workshop on Experimental Computer Science in conjunction with The Federated Computer Research Conference (FCRC) (submitted for review)*. 2007.
- [7]. Mallik, A., M.C. Wildrick, and G. Memik, *Application-Level Error Measurements for Network Processors* Institute of Electronics, Information and Communication Engineers (IEICE) Transactions on Information and Systems, 2005. E88-D(8): p. 1870-1877.
- [8]. Mallik, A., M.C. Wildrick, and G. Memik. *Measuring Application Error Rates for Network Processors*. in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. July 2004. Hiroshima, Japan.
- [9]. Mallik, A. and G. Memik, *Analyzing Correctness-Performance Trade-Offs: Clumsy Packet Processors*. ACM Transactions on Architecture and Code Optimization (ACM-TACO) (awaiting revision).
- [10]. Mallik, A. and G. Memik. *Automated Task Distribution in Multicore Network Processors Using Statistical Analysis (Submitted for Review)*. in *The 20th ACM International Conference on Supercomputing 2007*.
- [11]. Memik, G., M.H. Chowdhury, A. Mallik, and Y.I. Ismail. *Engineering over-Clocking: Reliability-Performance Trade-Offs for High-Performance Register Files*. in *The International Conference on Dependable Systems and Network (DSN-05)*. 2005. Yokohama, Japan.
- [12]. Memik, G., M.T. Kandemir, and A. Mallik. *Load Elimination for Low-Power Embedded Processors*. in *Proceedings of the 15th ACM Great Lakes symposium on VLSI 2005*. Chicago, Illinois, USA.
- [13]. Mallik, A., M.C. Wildrick, and G. Memik. *Design and Implementation of Correlating Caches* in *Proceedings of the 2004 international symposium on Low power electronics and design*. 2004. Newport Beach, California, USA.
- [14]. Mallik, A. and G. Memik, *Low Power Correlating Caches for Network Processors* The Journal for Low Power Electronics (JOLPE), 2005. 1(2): p. 108-118.
- [15]. Mallik, A., D. Sinha, P. Banerjee, and H. Zhou. *Smart Bit-Width Allocation for Low Power Optimization in a Systemc Based Asic Design Environment* in *Proceedings of the conference on Design, automation and test in Europe: Proceedings 2006*. Munich, Germany
- [16]. Mallik, A., D. Sinha, P. Banerjee, and H. Zhou, *Low Power Optimization by Smart Bit-Width Allocation in a Systemc Based Asic Design Environment*. IEEE Transactions on Computer-aided Design of Integrated Circuits and System (IEEE-TCAD), 2007.