Statement of Research

Weiwei Chen

Embedded computer systems are ubiquitous and pervasive in our modern society with a wide application domain, such as automotive and avionic systems, electronic medical equipment, telecommunication devices, industrial automation, energy efficient smart facilities, and consumer electronics. Embedded system is one of the most popular computational systems in our current information era [1].

There are two trends for modern embedded systems in recent years. First, various types of components are integrated into one system, including sensors, analog-to-digital converters, general-purpose CPUs, digital-signal processors (DSPs), dedicated hardware accelerators implemented as application-specific integrated circuits (ASICs), application-specific instruction-set processors (ASIPs), and customized memory components [2]. Meanwhile, the fundamental physical bottleneck prevents single hardware units from increasing frequency at the speed Moore's Law predicts [26]. The shift towards *multi-core and many-core systems* is inevitable to improve the computing capabilities in the future [27]. However, the parallelism in the applications must be extracted to utilize the multiple hardware resources in the system. The *heterogeneity* of the system imposes both theoretical and engineering challenges on system modeling, validation, debugging, software synthesis, and design space exploration.

Second, the interaction with the physical world with strict real time requirements, massive amount of data processing and communication, and mixed signal interfaces brings traditional embedded systems into a new page, *cyber-physical systems* (CPS). In CPS, embedded computer systems are connected as a network working synergistically with physical processes. The cyber and the physical system are tightly coupled and affect each other via feedback loops. To handle such dynamics in CPS, research must address the areas of system abstraction, modeling, analysis, and design [3][4].

My research so far has been focusing on efficient embedded system modeling and validation. I have studied topics in the area of system-level description languages, models of computation, parallel simulation, and transaction level modeling. My work has utilized compiler technology, scheduling theory, and novel modeling approaches to improve embedded system design.

Dissertation Work

My dissertation research has been focusing on parallel simulation approaches for system-level description languages (SLDLs) in which most embedded system models are described. My work extends the simulation kernel of the *SpecC* SLDL for parallel discrete event simulation (PDES) so as to exploit the underlying parallel computational capability in today's multi-core simulation hosts [8]. Furthermore, I've proposed an advanced simulation approach called out-of-order PDES, which significantly improves model simulation speed by breaking the global simulation cycle barriers with the help of a corresponding conflict analyzer in the compiler [9, 11, 12].

Besides my dissertation work, I have also been working on different research projects in the area of embedded system design and computer aided tools. My experience includes designing the software architecture of a language front-end tool for the *SystemC* SLDL by using the *LLVM+Clang* compiler infrastructure; providing the compiler analysis support for an embedded system recoding tool [10]; proposing a new model of computation for system-level design [13]; building industrial-sized embedded system-level applications [14, 15, 16]; developing a symbolic analog circuit simulator based on a graph reduction algorithm [17,18]; optimizing algorithms on special-purposed hardware platforms; and building a heterogeneous multiprocessor instruction set simulator [19].

Research Plan

In the future, I strive to conduct research work in the broader areas of system design. I intend to apply my knowledge on simulation, modeling, and compilers to develop next generation embedded systems, and propose new programming paradigms for new architecture platforms.

Design and model the next generation embedded system - Cyber-Physical Systems

Cyber-physical systems (CPS) are getting tremendous attention in recent years due to their broad potential application domains, such as healthcare monitoring, traffic control, search and rescue, and energy efficient facility infrastructures. CPS has tightly coupled computational and physical processes in the whole system [3]. While there are many well-defined models for computations, new abstractions and design methodologies are needed for cyber-physical systems due to the qualitative differences between the physical and the computational world. For instance, the physical world is naturally continuous while the computation theories are fundamentally discrete. Physical processes are concurrent while concurrency is very hard to comprehend and describe in computer programs due to the underlying computer architecture and the intrinsic sequential semantics for most programming languages. In addition, computer systems are usually very predictive and reactive but unpredicted situations often happen in the physical world. Thus, safety and reliability are imperative for computer systems to interact with the physical world.

I am interested in research work on modeling approaches for cyber-physical systems. I wish to apply the philosophy behind the transaction-level modeling (TLM) [7] for electronic system-level (ESL) design onto cyber-physical systems to figure out what are the different abstraction levels for CPS, how to formalize the physical features in the context of CPS, how to define the interaction between the cyber and physical parts, and how to take essential system features such as security, reliability and timeliness into consideration while modeling the system.

Also, I am very interested in building real-world CPS applications, which are promising to make our future life more convenient and smarter. I am looking forward to collaborate with researchers in different disciplines, such as biological science, chemistry, medicine, and mechanics to transfer the theoretic research result into the application domain.

Advanced parallel simulation and automatic debugging

Along the line with my dissertation work, I wish to advance the parallel simulation approaches into a few unexplored areas, and utilize the compiler and simulation infrastructure to further optimize system modeling with parallelism.

First, the code analyzer maintains a complete picture of the control flow of the models. This information can be used to get the future simulation status at run time so that the scheduler can look ahead and make smarter decisions to increase simulation speed. Second, the order to issue parallel threads with different characteristics, such as workload and dependencies, can significantly affect the simulation efficiency. Scheduling strategies can be evaluated for better simulation performance based on different thread characteristics. Third, hybrid multi-threading models, where user-level threads are running on top of multiple OS-kernel threads, are also promising to reduce the heavy multithreading overhead that is caused by OS-kernel threading models.

On the other hand, for system-level modeling, a well-defined system-level model with explicit parallelism is the starting point for proper mapping and synthesis. However, most embedded applications are initially written in unstructured and sequential source code. Creating explicit parallelism in system models is a critical prerequisite for building cost-effective MPSoCs and for fast model validation. Writing a correct parallel system model is difficult as it involves the task of inserting parallelism. I plan to apply my knowledge on static code analysis and parallel simulation to automate the debugging process for system modeling by providing meaningful information including risky shared variables and nondeterministic behaviors.

Efficient full-system validation

The system validation phase usually expands through the whole design life cycle of a product as it includes system-level, RTL, and post-silicon stages [5, 6]. Validation speed is very crucial to the products' time-to-market. For system-level validation, simulation performance of the models at lower abstraction levels is a critical bottleneck in today's system design flows. The simulation can take several days or weeks to run for

industrial applications. Efficient full system simulation is highly desirable for fast and low cost system validations.

My out-of-order parallel simulation approach and predictive code analysis make it promising to improve full system validation with components described at different abstraction levels on today's multi-core simulation hosts. I wish to push the parallel simulation to lower abstraction levels, such as pin- and cycle-accurate level and instruction-set simulation level. Here, we need to understand the lower-level synchronization mechanisms and explore possible approaches to reduce the overheads caused by parallelization. Also, we need strong compiler infrastructure for the description languages in which the models are described.

Parallel programming approaches for new hardware architectures

During my dissertation research, I found out that one major reason that prevents the parallel simulation from achieving high performance is because of the limited parallelism available in the model. Automatic parallelizing techniques and tools are highly desirable for software engineers to exploit the computational capabilities on today's multi-core / many-core hardware platforms by running their traditionally designed programs. However, automatic parallelization techniques are usually highly dependent on the compiler supports, and cannot be effective and successful without taking the characteristics of the underlying hardware platform into consideration [20, 21, 22, 23]. While innovative computers architectures, such as Intel's many core integrated (MIC) architecture [24], general-purpose graphics processing units (GPGPUs) [25], and multi-core supercomputers with distributed memories, are rapidly emerging these days, we need to well study the architecture features so as to propose efficient programming paradigms and library supports for the programmers who wish to use these new platforms. Also, while working on my dissertation research, I discovered that how the application is modeled could significantly affect the efficiency for simulation and the following synthesis steps. Different implementations with the same functionality can lead to very different execution performance on multi-core simulation hosts. Therefore, good practices for parallel programming are also worthwhile studying for the upcoming multi-/many-core era.

I wish to apply my understanding and experience on computer systems and parallel programming to propose new programming paradigms, build system libraries, improve compiler supports, and provide design guidelines on applications development to facilitate the utilization of the upcoming high performance computer systems.

Research in computer science education

During my graduate study in UC Irvine, I am very fortunate to be selected as the Pedagogical Fellow by the Teaching, Learning and Technology Center (TLTC). I have received an extensive training on pedagogy skills and developed an avid interest in innovative practices for computer science education. I wish to explore the active learning strategies and design educational applications to promote the learning processes in computer science and engineering.

There are also some challenges for computer science education that I would like to address. First, I believe that our students, who will be the future computer engineers and scientists, should be trained to take the new features of the future systems into consideration at the very first stage of their development work. Otherwise, it will be very challenging to adjust legacy designs to new systems. For example, the fundamental reason why it is difficult to exploit the parallelism in the existing software applications is because that most of the software engineers are trained to think sequentially by using the traditional structural programming languages. The disparity between the traditional programming philosophy and new hardware architecture is the very source of the performance barriers for software development. To eventually address this issue, a paradigm shift on educating the next generation programmers is imperative. While there are many languages and libraries supporting parallel programming, we need to understand how people abstract parallelism and develop intuitive tools to facilitate the education process.

Second, embedded system is an area that is usually underrepresented in traditional computer science courses [1]. It requires the knowledge from many areas such as software engineering and computer hardware systems, and tearing down the walls between disciplines of computer science, electrical engineering and physics. Since embedded / cyber-physical systems will very likely be dominant in the future, our students need to be educated accordingly to understand the fundamental features and design methodologies for these innovative systems. I wish to contribute on designing new undergraduate-level course curriculum for embedded systems to promote educating our future designers.

References

[1] Peter Marwedel. *Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems Series*. Springer Publishing Company, Incorporated.

[2] Wayne Wolf, Ahmed Amine Jerraya, and Grant Martin, "Multiprocessor System-on-Chip (MPSoC) Technology", *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems (TCAD),* vol. 27, no. 10, October 2008.

[3] Edward A. Lee and Sanjit A. Seshia, *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*, 2011, ISBN 978-0-557-70857-4, http://LeeSeshia.org.

[4] "Strategic R&D Opportunities for 21st Century Cyber-Physical Systems", Report of the Steering Committee for Foundations in Innovation for Cyber-Physical Systems, NIST Foundations for Innovation in Cyber-Physical Systems Workshop, 2013.

[5] "Validation, Verification and Certification of Embedded Systems", *NATO Research and Technology organization*, 2005.

[6] Frank Schirrmeister, System-Level Market Trends, *Synopsys Interoperability Forum*, 2010.

[7] L. Cai, D. D. Gajski, "Transaction level modeling: an overview", in Proceedings of the IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis (CODES+ISSS), 2003

[8] W. Chen, X. Han, R. Doemer, "Multi-core Simulation of Transaction Level Models using the System-on-Chip Environment", *IEEE Design & Test of Computers*, vol.28, no.3, pp.20-31, May-June 2011

[9] W. Chen, R. Doemer, "Optimized Out-of-Order Parallel Discrete Event Simulation Using Predictions", in *Proceedings of the Design, Automation and Test in Europe (DATE) Conference,* 2013

[10] W. Chen, C. Chang, X. Han, R. Doemer, "Eliminating Race Conditions in System-Level Models by using Parallel Simulation Infrastructure", in *Proceedings of the IEEE International High Level Design Validation and Test Workshop (HLDVT),* 2012

[11] W. Chen, X. Han, R. Doemer, "Out-of-order Parallel Simulation for ESL Design", in *Proceedings of the Design, Automation and Test in Europe (DATE) Conference*, 2012

[12] W. Chen, R. Doemer, "An Optimizing Compiler for Out-of-Order Parallel ESL Simulation Exploiting Instance Isolation", in *Proceedings of the 17th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2012

[13] W. Chen, R. Doemer, "ConcurrenC: A New Approach towards Effective Abstraction of C-based SLDLs", *Analysis, Architectures and Modeling of Embedded Systems* (ed. A. Rettberg, M. Zanella, M. Amann, M. Keckeisen, F. Rammig), Springer, 2009, ISBN 978-3-642-04283-6

[14] X, Han, W. Chen, R. Dömer, "A Parallel Transaction-Level Model of H.264 Video Decoder", *TR-11-03*, Center for Embedded Computer Systems, UC Irvine, 2011

[15] W. Chen, S. Sun, B. Zhang, R. Dömer, "System Level Modeling of a H.264 Decoder", *TR-08-10*, Center of Embedded Computer System, UC Irvine, 2008

[16] W. Chen, R. Doeme, "System Specification of a DES Cipher Chip", *TR-08-01*, Center of Embedded Computer System, UC Irvine, 2008

[17] W. Chen, G. Shi, "Implementation of a Symbolic Circuit Simulator for Topological Network Analysis", in *Proceedings of IEEE Asia Pacific Conference on Circuit and System (APCCAS),* 2006

[18] G. Shi, W. Chen, C.-J. R. Shi, "A Graph Reduction Approach to Symbolic Circuit Analysis", in *Proceedings of the 12th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2007

[19] R. Zhong, Y. Zhu, W. Chen, M. Lin, W-F. Wong, “An Inter-core Communication Enabled Multi-core Simulator Based on SimpleScalar”, in *Proceedings of the 4th IEEE International Symposium on Embedded Computing (SEC),* 2007

[20] G. Blake, R. G. Dreslinski, T. Mudge, and K. Flautner, “Evolution of thread-level parallelism in desktop applications”, in *Proceedings of the 37th annual international symposium on Computer architecture (ISCA),* 2010

[21] M. Feng, R. Gupta, and Y. Hu, "SpiceC: Scalable Parallelism via implicit copying and explicit Commit", in *Proceedings of the 16th ACM SIGPLAN Symposium on Principles and Practices of Parallel Programming (PLDI),* 2011

[22] J. Torrellas, “Thread-Level Speculation”, *Encyclopedia of Parallel Computing*, Springer, 2011.

[23] C. Segulja and T. S. Abdelrahman, “Architectural support for synchronization-free deterministic parallel programming”, in *IEEE International Symposium on High Performance Computer Architecture (HPCA),* 2012.

[24] "Intel® Many Integrated Core Architecture (Intel® MIC Architecture) – Advanced", http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html

[25] J. D. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. Lefohn, T. J. Purcell, “A Survey of General-Purpose Computation on Graphics Hardware”, *Computer Graphics Forum*, 2007

[26] D. Woo and H. S. Lee, "Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era", *Computer*, Volume 41 Issue 12, 2008

[27] J. L. Manferdelli, N. K. Govindaraju, and C. Crall, "Challenges and Opportunities in Many-Core Computing", in *Proceedings of the IEEE*, Vol. 96, No. 5, 2008