Statement of Research

Weiwei Chen

Embedded computer systems are ubiquitous in our modern society with a wide application domain, such as automobiles, medical devices, communication systems, mobile electronics, and so on. Embedded systems are the most pervasive technology ever with deeper penetration than electricity and safe drinking water [1].

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

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 era, *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 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].

My research so far has focused 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 for improving embedded system design.

Dissertation: **Out-of-order Parallel Simulation for Electronic System-Level Design**
My Ph.D. research entails three parts: 1) a new model of computation (MoC), namely *ConcurrenC*, that is aligned and consistent with current System-level Description Languages (SLDLs); 2) a parallel extension for the SLDL simulator to exploit simulation parallelism in the system-level models on multi-core computer hosts; 3) models and synthesis of standard embedded applications, including a H.264 video decoder, a JPEG image encoder, a video edge detector, and a DES cipher chip.

System design in general can only be successful if it is based on a suitable formal model of computation that can be well represented in a SLDL, such as SystemC and SpecC. While C-based languages are popular in system-level modeling and validation, current tool flows impose almost arbitrary restrictions on the synthesizable subset of the supported SLDL. I have proposed the *ConcurrenC* model of computation which aptly fits system model requirements and supports the essential features for system-level modeling, such as structural hierarchy, separate communication and computation, explicit parallelism, abstract communication, timing, and execution semantics.

In the context of system-level modeling, my research work further investigates the approaches for model validation. *Fast yet accurate simulation* is highly desirable for efficient and effective system design since system validation largely relies on simulation. Typically, C-based SLDLs define their execution semantics based on discrete event (DE) simulation. Most reference simulators adopt traditional discrete event simulation where the explicit parallelism in the design is expressed by using a cooperative multithreading model. This threading model greatly simplifies communication via shared events and variables; however, it only selects one thread to run at all times which makes it impossible to utilize any available parallelism in multi-core simulation hosts. Moreover, the execution semantics of discrete event simulation imposes a total order on event delivery and time advances for model simulation. The global simulation cycle barrier is a significant impediment to efficient multi-core utilization [4].

To address these limitations to efficient model validation, my research work extends the simulation kernel of the SpecC SLDL to support real parallelism during simulation [5]. The shared model resources and synchronizations are protected by automatic model instrumentation to ensure safe communications. My work enables the SLDL simulator to utilize the available computation resources in multi-core simulation hosts.

I have also developed a novel scheduling approach, namely *out-of-order parallel discrete event simulation* (OoO PDES), to address the obstacles to efficient multi-core utilization due to the discrete event execution semantics [6]. OoO PDES breaks the global simulation-cycle barrier of traditional DE simulation by localizing the simulation time into each thread, carefully delivering notified events, and handling a dynamic management of simulation sets. I have designed an advanced static model analyzer in the compiler to compute potential out-of-order scheduling conflicts caused by parallel accesses to shared events and variables [7]. As such, OoO PDES allows the simulator to leverage the parallel processing capabilities in the multi-core hosts while fully preserving the simulation semantics and timing accuracy of the model.

To demonstrate the viability of the ConcurrenC MoC and the efficiency of our parallel simulators, we have built ConcurrenC models for several standard embedded applications. Our experimental results show that the models can capture the necessary features of the system and can easily be synthesized to lower abstraction levels with more implementation details. Also, our advanced simulation approach achieves significant speedup close to the theoretical maximum with negligible compilation cost.

Research Plan

I strive to perform research work in the broader areas of system design in the future. I will describe my research plans in short and long term.

Short-term research goals

I intend 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.

Advanced parallel simulation

To further improve system-level parallel simulation, I plan to research several aspects. 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 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. Last but not least, hybrid multithreading 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.

Automatic debugging for parallel system models

A well-defined system-level model needs explicit parallelism for mapping and synthesis. However, most embedded applications are initially written in unstructured and sequential source code. Creating explicit parallelism in system models is a 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.

Long-term research goals

In the long-term, I am very excited about the prospect of applying my expertise in simulation, modeling and compilers into related areas to build fast and sophisticated embedded computer systems.

Efficient 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 [8, 9]. 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.

The parallel simulation approaches and predictive code analysis makes 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 languages in which the models are described.

System modeling approaches

While working on my dissertation research, I discovered that how the application is modeled can significantly affect the efficiency for simulation and the following synthesis steps. I plan to apply my understanding and experience on system modeling to provide the design guidelines to build system models that are efficient for validation and refinement.

Moreover, one major reason that prevents the parallel simulation from achieving high performance is because of the limited parallelism available in the model. State-of-the-art approaches for parallelism insertion in system-level models largely depend on the designer's intelligence to identify the blocks and manually construct the parallelism. While there are extensive works on automatic thread-level parallelization [10] and corresponding hardware-supported technologies [11, 12], research is needed for automatic parallelism extraction in the context of building system-level models. I wish to investigate the technologies in high performance computing and combine with my understanding on system-level design for possible automatic solutions to address the aforementioned problem. NSF "Exploiting Parallelism and Scalability" program is a potential source of funding for this research topic.

Research on next-generation embedded 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 smart buildings. CPS integrates computation and physical processes in one 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 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. Safety and reliability are imperative for computer systems to interact with the physical world. I plan to study some CPS applications in order to propose new modeling and validation approaches with the help of new abstract interface layers, enhanced programming language semantics, or novel system architectures. I am also looking forward to having interdisciplinary collaborations to combine my knowledge in embedded system design with biology, medicine, or mechanics to build real-world CPS applications. Possible funding support for this research can be sought from the NSF "Cyber-Physical Systems" program.

Research in computer science education

During my tenure as the Pedagogical Fellow in UC Irvine, I have had the chance to learn and practice different pedagogy skills and developed an avid interest in innovative educational practices for computer science education. I wish to explore the active learning strategies and educational applications to promote the learning processes in computer science and engineering. I also wish to carry out research on curriculum design and educational technologies to understand and improve our perception in the way of using computers.

References

[1] Chetan sharma: Technology & Strategy Consulting, "2012 Global Mobile Market Update".

[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] Rainer Dömer, Weiwei Chen, Xu Han, and Andreas Gerstlauer, "Multi-Core Parallel Simulation of System-Level Description Languages", in *Proceedings of 16th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2011.

[5] Weiwei Chen, Xu Han, and Rainer Dömer, "Multi-core Simulation of Transaction Level Models using the System-on-Chip Environment", *IEEE Design & Test of Computers*, vol. 28, no. 3, May-June 2011.

[6] Weiwei Chen, Xu Han, and Rainer Dömer, "Out-of-order Parallel Simulation for ESL Design", in *Proceedings of the Design, Automation and Test in Europe Conference (DATE)*, 2012.

[7] Weiwei Chen and Rainer Dömer, "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.

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

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

[10] Geoffrey Blake, Ronald G. Dreslinski, Trevor Mudge, and Krisztián Flautner, "Evolution of thread-level parallelism in desktop applications", in *the 37th annual international symposium on Computer architecture (ISCA)*, 2010.

[11] Josep Torrellas, "Thread-Level Speculation", *Encyclopedia of Parallel Computing*, Springer, 2011.

[12] Cedomir Segulja and Tarek S. Abdelrahman, "Architectural support for synchronization-free deterministic parallel programming", in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2012.