Out-of-Order Parallel Simulation of SystemC Models using Intel® MIC Architecture

G. Liu, T. Schmidt, R. Dömer (CECS)
A. Dingankar, D. Kirkpatrick (Intel Corp.)

Speaker: Rainer Dömer
doemer@uci.edu
Center for Embedded Computer Systems
University of California, Irvine

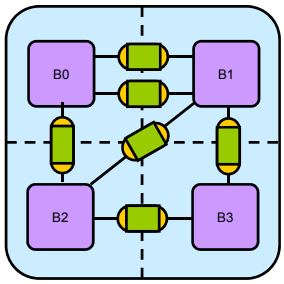
Outline

- Electronic System Level Design
  - Project context, goals, and overview
- Parallel SystemC Simulation
  - Traditional Discrete Event Simulation (DES)
  - Parallel Discrete Event Simulation (PDES)
  - Out-of-Order Parallel Discrete Event Simulation (OoO PDES)
- Project Realization
  - Ongoing Research and Development
- Promising Experimental Results
  - Parallel benchmarks
  - Highly parallel applications
- Concluding Remarks

Electronic System Level (ESL)

- Electronic System Level Models
  - Abstract description of a complete system
  - Hardware + Software
- Key Concepts in System Modeling
  - Explicit Structure
    - Block diagram structure
    - Connectivity through ports
  - Explicit Hierarchy
    - System composed of components
  - Explicit Concurrency
    - Potential for parallel execution
    - Potential for pipelined execution
  - Explicit Communication and Computation
    - Modules
    - Channels and Interfaces

System Model



Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 3

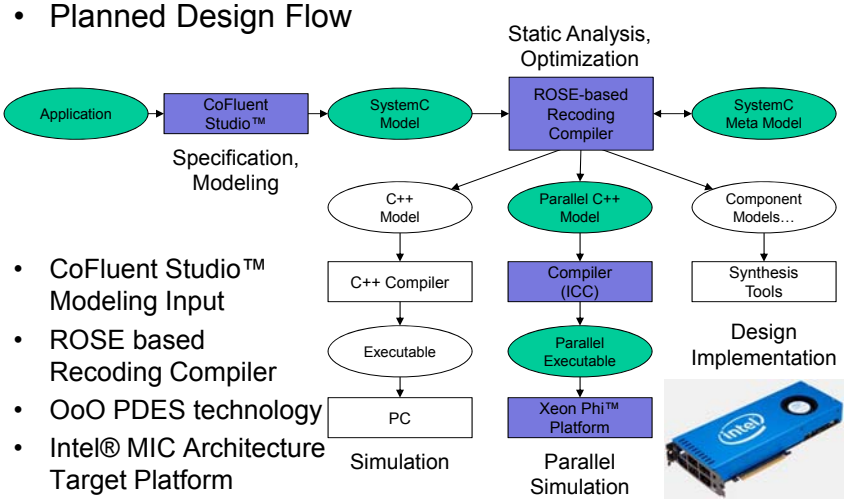
ESL Simulation

- Model Evaluation through Simulation
  - Efficient system-level simulation is critical
    - Fast *and* accurate!
  - Complexity of system models grows constantly
    - Need for speed!
- ➢ Parallel Simulation!
  - Parallelism is explicitly specified in model
    - SystemC: **`SC_THREAD, SC_METHOD`**
  - Parallel processing is available in standard PCs
    - Multi-core hosts readily available
    - Many-core technology is arriving
  - ➢ Target Simulation Platform
    - Intel® Many Integrated Core (MIC) Architecture

Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 4

Project Overview

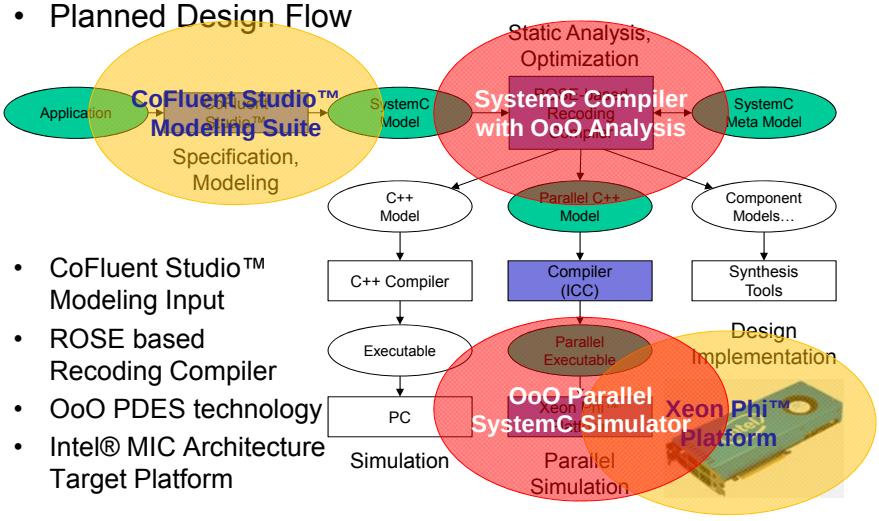
- Planned Design Flow



- CoFluent Studio™ Modeling Input
- ROSE based Recoding Compiler
- OoO PDES technology
- Intel® MIC Architecture Target Platform

Project Overview

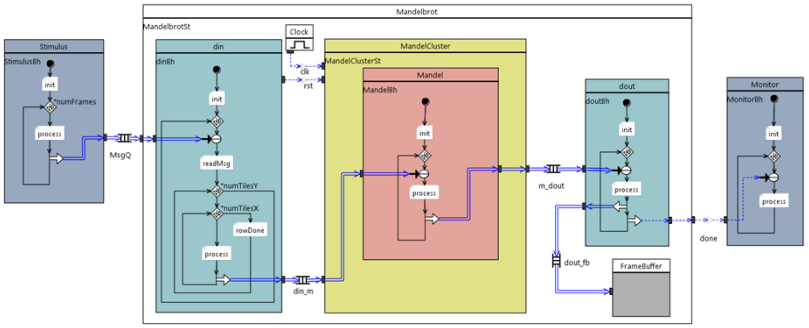
- Planned Design Flow



- CoFluent Studio™ Modeling Input
- ROSE based Recoding Compiler
- OoO PDES technology
- Intel® MIC Architecture Target Platform

SystemC with CoFluent Studio™

- Modeling and Simulation Tool Suite
  - Supports model-driven architecture (MDA)
  - Based on Eclipse modeling framework (EMF)



➢ CoFluent™ Modeling Concept ensures well-defined SystemC model as input

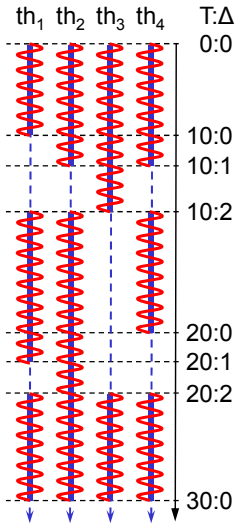
Parallel ESL Simulation

- Traditional Discrete Event Simulation (DES)
  - Reference simulators run *sequentially*, only one thread at a time (cooperative multi-threading model)
  - Cannot utilize the capabilities of multi- or many-core hosts
- Parallel Discrete Event Simulation (PDES)
  - Threads run in *parallel* (if at the same delta cycle and time)
  - Simulation-cycles are absolute barriers!

➢ Out-of-order Parallel DE simulation (OoO PDES)
  - Best technique known today, developed by CECS [DATE'12]
  - Threads run in *parallel and out-of-order* even in different delta and time cycles if there are no conflicts!
  - Aggressive, runs maximum number of threads in parallel, but *fully preserves DES semantics and model accuracy*!

Discrete Event Simulation (DES)

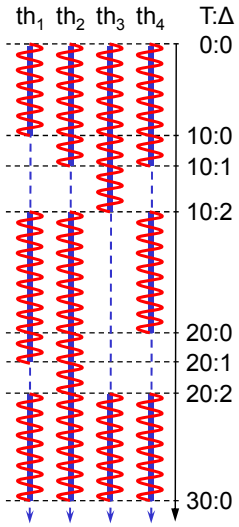
- Traditional DES
  - Concurrent threads of execution
  - Managed by a central scheduler
  - Driven by events and time advances
    - Delta-cycle
    - Time-cycle
  - ➢ Partial temporal order with barriers
- Reference Simulator
  - SystemC reference simulator uses cooperative multi-threading
  - ➢ A single thread is active at any time!
  - ➢ Cannot exploit parallelism
  - ➢ Cannot utilize multiple cores



Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 9

Parallel Discrete Event Simulation

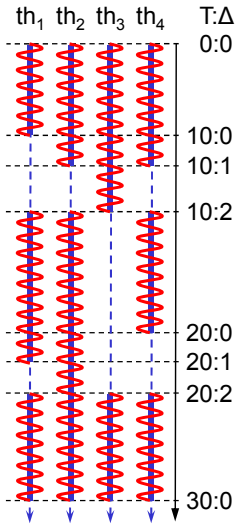
- Parallel DES
  - Threads execute in parallel *iff*
    - in the same delta cycle, *and*
    - in the same time cycle
  - ➢ Significant speed up!
  - *Synchronous* PDES: Cycle boundaries are *absolute barriers*!
- Aggressive Parallel DES
  - Conservative Approaches
    - Careful static analysis prevents conflicts
  - Optimistic Approaches
    - Conflicts are detected and addressed (*roll back*)



Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 10

Out-of-Order Parallel DES

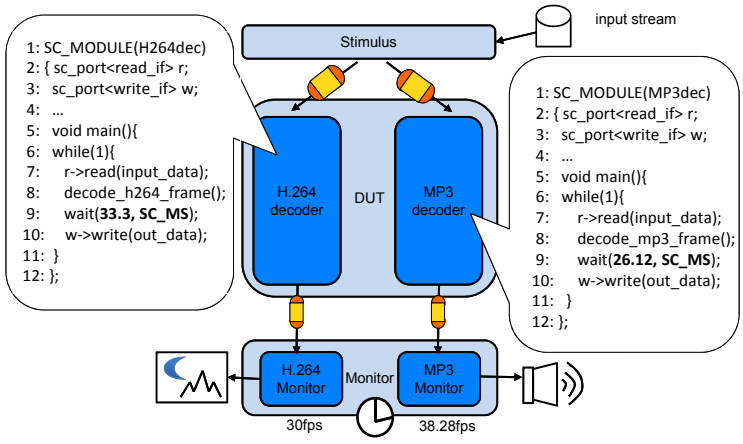
- Out-of-Order PDES
  - Threads execute in parallel *iff*
    - in the same delta cycle, *and*
    - in the same time cycle,
    - *OR* if there are no conflicts!
  - Can utilize advanced compiler for static data conflict analysis
  - ➢ Allows as many threads in parallel as possible
  - ➢ Significantly higher speedup!
    - Results at [DATE'12], [ASPDAC'12]
  - ➢ Fully preserves…
    - ➢ DES execution semantics
    - ➢ Accuracy in results and timing



Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 11

Synchronous vs. Out-of-Order PDES

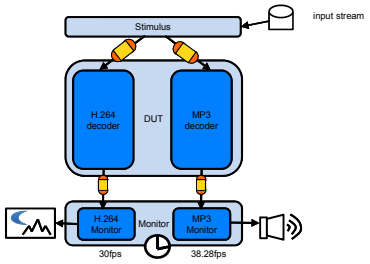
- Simple Example:
  - Parallel video and audio decoding with different frame rates



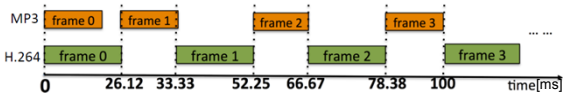
Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 12

Synchronous vs. Out-of-Order PDES

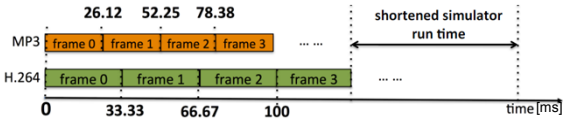
- Simple Example:
  - Parallel video and audio decoding with different frame rates
  - ➢ Synchronous PDES
    - Observes time and delta cycles
    - Global time
  - ➢ Out-of-Order PDES
    - Breaks cycle barrier
    - Local times (per thread)



- PDES:



- OoO PDES:



Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 13

Many-Core Target Platform

- Intel® Many Integrated Core Architecture
- ➢ Intel® Xeon Phi™ Coprocessor
  - Provides
    - 60 processor cores
    - 4 hyper-threads per core
    - ➢ 240 parallel hardware threads!

  - Hardware Features
    - Vector processing unit (VPU)
    - Extended Math Unit (EMU) for transcendental operations
    - Bidirectional ring interconnect
  - Peak performance
    - ➢ over 1 teraFLOPS (double-precision)
  - ➢ Uses familiar and standard programming models
    - ➢ Appears as a regular Linux machine with 240 cores!

Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 14

Project Realization

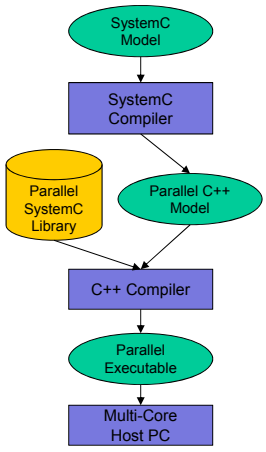
- Ongoing Research and Development

1. **SystemC Compiler** with Out-of-Order PDES Analysis
   - SystemC frontend for ROSE (lexer, parser, int. representation)
   - Segment Graph data structure for advanced conflict analysis
   - Code generator for parallel execution

2. **Parallel Simulator** with Out-of-Order Scheduler
   - Parallel scheduler with fast conflict table lookup
   - Target platform Intel® MIC architecture
   - Optimal thread-to-core task mapping
   - SystemC kernel extension
     - Protected communication
     - Mutually-exclusive access to shared resources

Project Realization

- SystemC Compiler
  - Build abstract syntax tree
  - Build SystemC internal representation
  - Build segment graph
  - Build variable access lists
  - Identify potential conflicts
  - Build segment tables
  - Instrument **`wait()`** calls
  - Protect user-defined channels
  - Generate parallel C++ model
- Parallel SystemC Library
  - POSIX multi-threading
  - Reentrant primitives
  - Protected central resources
  - Protected standard channels
  - Out-of-order parallel scheduler



Parallel Simulation

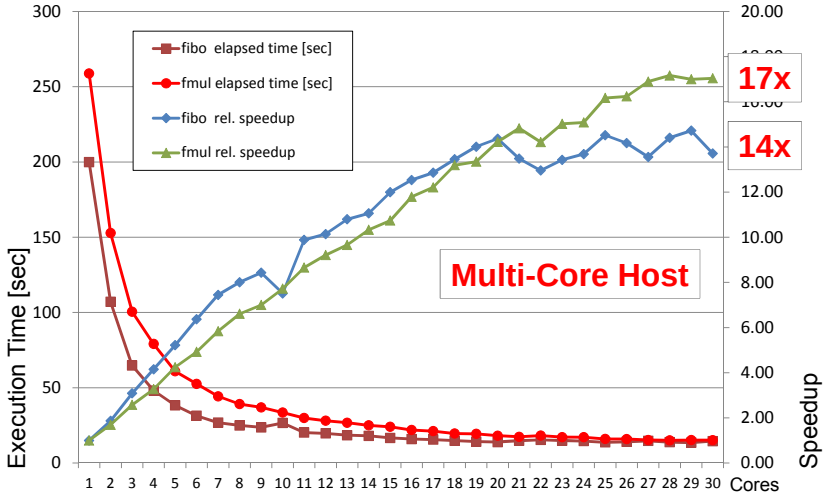
Promising Experimental Results

- What Speedup is achievable on Today's Multi-Core and Many-Core Host Platforms?
  - ➢ Early results using manually coded or SpecC-based examples
  - – Experimental Setup
    - SMP Host PC
      - – 2 Intel® Xeon® X5650 CPUs at 2.66 GHz
      - – 6 cores each, 2 hyper-threads per core
      - ➢ 24 parallel hardware threads available
    - Many Integrated Core (MIC) Platform
      - – 1 Intel® Xeon Phi™ Coprocessor 5110P at 1.053 GHz
      - – 60 cores on ring-bus, 4 hyper-threads per core
      - ➢ 240 parallel hardware threads available
  - – Highly parallel benchmarks
    - ➢ Parallel floating-point multiplications (**fmul**)
    - ➢ Parallel Fibonacci calculation (**fibo**)

Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 17

Parallel Benchmark Results

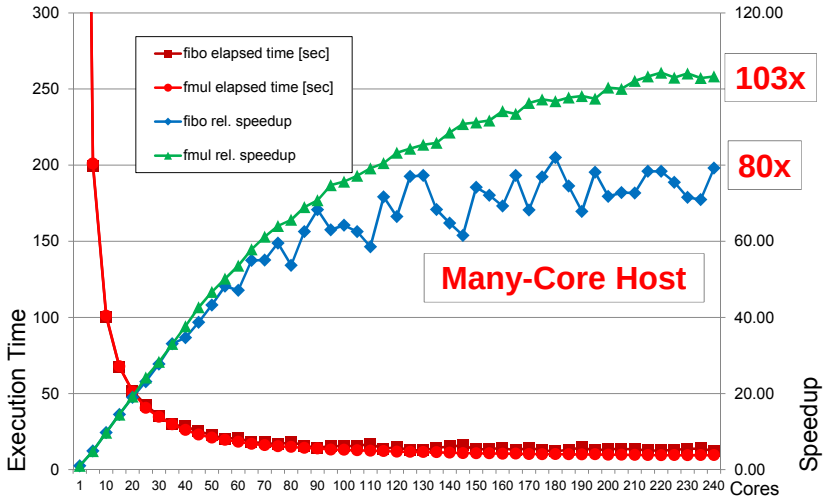
- Experimental Results (2 Intel® Xeon® X5650 CPUs, 2x6x2 cores)



Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 18

Parallel Benchmark Results

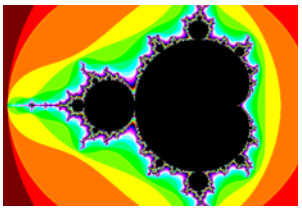
- Experimental Results (Intel® Xeon Phi™ coproc., 60x4 cores)



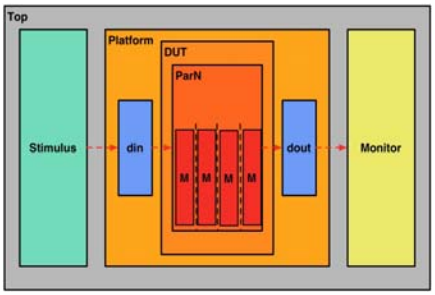
Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 19

GPU Pipeline Example

- Graphics Application: Mandelbrot Set
  - Mathematical set of points
    - Two-dimensional fractal shape
  - Complex computation
    - Recursive function
  - Extreme parallelism
    - Pixel level



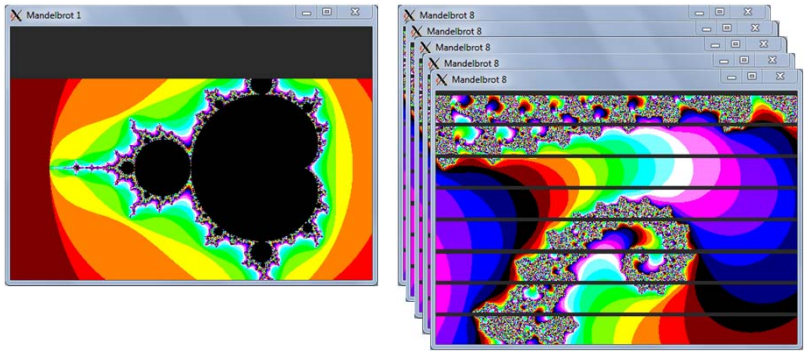
- SystemC Model
  - TLM abstraction
  - Parallel slices
  - Configurable
  - Executable



Out-of-Order Parallel SystemC Simulation using Intel MIC Architecture (c) 2014 R. Doemer, et.al. 20

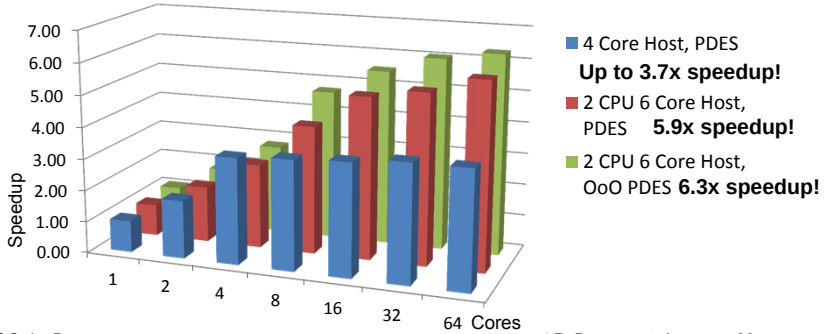
GPU Pipeline Example

- Graphics Application: Mandelbrot Set
  - When synthesized, real-time rendering is no problem
  - When simulated, regular DES is very slow
  - ➢ Parallel DES can significantly speed up simulation!



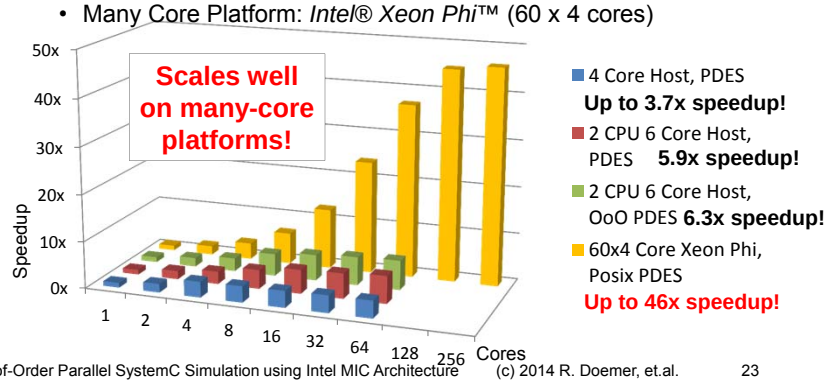
GPU Pipeline Example

- Graphics Application: Mandelbrot Set
  - Parallel DES can significantly speed up simulation!
  - Experimental Results
    - Sequence of 100 Mandelbrot images (640x448, depth 4096)
    - SpecC models with increasing number of parallel blocks
    - Hosts: *Intel® Core™ 2 Quad* (4 cores), and *Dual Xeon®* (12 cores)



GPU Pipeline Example

- Graphics Application: Mandelbrot Set
  - Parallel DES can significantly speed up simulation!
  - Experimental Results
    - Sequence of 100 Mandelbrot images (640x448, depth 4096)
    - Simplified PDES model (Posix based, manually created)
    - Many Core Platform: *Intel® Xeon Phi™* (60 x 4 cores)



Concluding Remarks

- ESL design needs fast and accurate SystemC simulation
- Traditional DES and PDES are insufficient
- Out-of-order PDES
  - Novel, aggressive, fast
  - Maximum parallelism
  - Fully semantics compliant and accurate
  - ➢ Promise of near-linear speedup on highly parallel platforms
- Parallel SystemC Compiler and Simulator
  - SystemC Compiler with Out-of-Order PDES Analysis
  - Parallel Simulator with Out-of-Order Scheduler
- Ongoing and Future Work
  - Completion of implementation, further evaluation
  - Collaboration with Accellera SystemC LWG

References (1)

- [DATE'12] W. Chen, X. Han, R. Dömer: *"Out-of-Order Parallel Simulation for ESL Design"*, Proceedings of DATE, Dresden, Germany, March 2012.
- [ASPDAC'12] R. Dömer, W. Chen, X. Han: *"Parallel Discrete Event Simulation of Transaction Level Models"*, Proceedings of ASPDAC, Sydney, Australia, February 2012.
- [ASPDAC'12] W. Chen, R. Dömer: *"An Optimizing Compiler for Out-of-Order Parallel ESL Simulation Exploiting Instance Isolation"*, Proceedings of ASPDAC, Sydney, Australia, February 2012.
- [IEEE D&T'11] W. Chen, X. Han, R. Dömer: *"Multicore Simulation of Transaction-Level Models Using the SoC Environment"*, IEEE Design & Test of Computers, vol. 28, no. 3, pp. 20-31, May-June 2011.
- [ASPDAC'11] R. Dömer, W. Chen, X. Han, A. Gerstlauer: *"Multi-Core Parallel Simulation of System-Level Description Languages"*, Proceedings of ASPDAC, Yokohama, Japan, January 2011.
- [HLDVT'10] W. Chen, X. Han, R. Dömer: *"ESL Design and Multi-Core Validation using the System-on-Chip Environment"*, Proceedings of HLDVT, Anaheim, California, June 2010.

References (2)

- [DATE'14] W. Chen, X. Han, R. Dömer: *"May-Happen-in-Parallel Analysis based on Segment Graphs for Safe ESL Models"*, Accepted for publication at DATE, Dresden, Germany, March 2014.
  - **Best Paper Award!**
- [DATE'13] W. Chen, R. Dömer: *"Optimized Out-of-Order Parallel Discrete Event Simulation Using Predictions"*, Proceedings of DATE, Grenoble, France, March 2013.
- [IEEE D&T'13] W. Chen, X. Han, C. Chang, R. Dömer: *"Advances in Parallel Discrete Event Simulation for Electronic System-Level Design"*, IEEE Design & Test of Computers, vol. 30, no. 1, pp. 45-54, Jan.-Feb. 2013.
- [HLDVT'12] W. Chen, C. Chang, X. Han, R. Dömer: *"Eliminating Race Conditions in System-Level Models by using Parallel Simulation Infrastructure"*, Proceedings of HLDVT 2012, Huntington Beach, California, November 2012.