CECS eNEWS

Volume 3, Issue 3, July 2003

Center for Embedded Computer Systems, University of California, Irvine

Highlights:

Inside this issue:

CECS at DAC '03

The Center for Embedded Computer Systems (CECS) at the University of California, Irvine continues playing a prominent role at the 40th Design Automation Conference (DAC '03) held at Anaheim, CA on June 2-6, 2003. CECS faculty affiliates presented 7 technical papers, served as session moderators and organizers, and on various conference program committees.

The following technical presentations were made by CECS faculty affiliates and their graduate students at DAC '03; the technical papers can be found in the conference proceedings at the cited pages:

- *Scalable Modeling and Optimization of Mode Transitions Based on Decoupled Power Management Architecture*, Dexin Li, Qiang Xie, and Pai Chou, pp 119–124

Graduate Student Dexin Li with Opening Presentation Slide

- *Dynamic Hardware/Software Partitioning: A First Approach*, Greg Stitt, Roman Lysecky, and Frank Vahid, pp 250–255

DAC '03 at Anaheim Convention Center

- *Automatic Communication Refinement for System Level Design*, Samar Abdi, Dongwan Shin and Daniel Gajski, pp 300–305
- *On-Chip Logic Minimization*, Roman Lysecky and Frank Vahid, pp 334–337
- *Test Application Time and Volume Compression Through Seed Overlapping*, Weenjing Rao, Ismet Bayraktaroglu, and Alex Orailoglu, pp 732–737
- *Instruction Set Compiled Simulation: A Technique for Fast and Flexible Instruction Set Simulation*", Mehrdad Reshadi, Prabhat Mishra, and Nikil Dutt, pp 758–763
- "*Improved Indexing for Cache Miss Reduction in Embedded Systems*", Tony Givargis, pp 875–880

Professor Pai H. Chou served on the DAC '03 Technical Program Committee and as a session organizer. Professor Nikil D. Dutt served as Chair of Session 15: *Issues in Partitioning & Design Space Exploration for CoDesign* and Professor Rajesh K. Gupta served as Chair of Panel Session 42: *Formal Verification—Prove It or Pitch It*.

Graduate Student Greg Stitt with DAC Presentation Certificate

CECS is extremely proud of its technical influence at DAC '03 which reflects the level of technical relevance and international recognition of its collaborative research programs in embedded systems.

Hakhu Joins Board, Visits

The Center for Embedded Computer Systems (CECS) is pleased to announce that Dr. Jai K. Hakhu, Vice President, Technology Manufacturing Group and General Manager, Technology Manufacturing Engineering, Intel Corporation, Santa Clara, CA has joined the CECS Research Advisory Board. In conjunction with this appointment, Jai visited CECS on April 18, 2003 to tour our facility and receive an overview of our research programs from Professor Daniel D. Gajski, Director, CECS. During the visitation, he visited with Professor Nicolaos G. Alexopoulos, Dean, The Henry Samueli School of Engineering.

Professor Dan Gajski, Dean Nick Alexopoulos, Dr. Jai Hakhu

Dr. Hakhu was born in Srinagar, Kashmir, India and received a bachelor's degree in Physics, Chemistry, Mathematics, and English from Jammu and Kashmir University, India; a bachelor's degree in Electronics Communication Engineering from the Indian Institute of Science; a Master's degree in Solid State Electronics from the University of Roorkee, India; and a PhD in Electrical Engineering from the University of California, Irvine in 1979. Jai has served on the faculty of the University of Roorkee, India; University of California, Irvine; and California State University, Long Beach. In 2000, Jai received the Outstanding Alumnus Award from the UCI Henry Samueli School of Engineering.

Prior to joining Intel in 1997, Dr. Hakhu worked in the semiconductor industry for 18 years as a technical and business manager, most recently as Vice President of Rockwell International and Varian Associates.

Jai lives in Los Gatos, CA with his wife, Nalini, and daughter Nisha and son Navneet.

Professor Dan Gajski, Quent Cassen, Bob Larsen, Dr. Jai Hakhu

CECS is delighted to have Dr. Hakhu associated with our Research Advisory Board as we strive to envision the future of embedded systems technologies. Jai's technical and managerial experience will be most valuable to our graduate students and faculty affiliates as we develop future research paradigms and platforms for the advancement of embedded computer systems. It is most gratifying that an outstanding UCI alumnus has chosen to become involved with shaping the future CECS research program. Welcome Jai to the CECS team!

Shin Delivers CECS Distinguished Lecture

On May 13, 2003 Professor Kang G. Shin, Kevin and Nancy O'Connor Professor of Computer Science, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan, delivered a CECS Distinguished Lecture titled *Real-Time Dynamic Voltage Scaling for Low Power Embedded Operating Systems* at the UCI McDonnell Douglas Auditorium. Professor Nikil D. Dutt, CECS, introduced Professor Shin and served as his host for his visitation to CECS. This lecture presented the algorithmic aspects of Dynamic Voltage Scaling (DVS) as a technique in exploiting the hardware characteristics of processors to reduce energy dissipation by lowering the supply voltage and operating frequency. Professor Shin discussed a class of novel algorithms, called real-time DVS (RT-DVS), that modify the OS's real-time scheduler and task management service to provide significant energy savings while maintaining real-time deadline guarantees.

Professor Shin is the Founding Director of the Real Time Computing Laboratory at the University of Michigan. His current research focuses on QoS-sensitive networking and computing as well as on embedded real-time OS, middleware and applications, all with emphasis on timeliness and dependability. He has supervised the completion of 42 PhD theses, and authored/coauthored over 500 technical papers and numerous book chapters in the areas of distributed real-time computing and control, computer networking, fault-tolerant computing, and intelligent manufacturing. He has coauthored (with C. M. Krishna) a textbook *Real-Time Systems*, McGraw Hill, 1997. He received the BS degree in Electronics Engineering from Seoul National University, Seoul, Korea in 1970, and the MS and PhD degrees in Electrical Engineering from Cornell University, Ithaca, New York in 1976 and 1978, respectively. From 1978 to 1982 he served on the faculty of Rensselaer Polytechnic Institute, Troy, New York.

Professor Shin is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM), and a member of the Korean Academy of Engineering. He received the Outstanding IEEE Transactions on Automatic Control Paper Award in 1987, Research Excellence Award in 1989, Outstanding Achievement Award in 1999, Service Excellence Award in 2000, and Distinguished Faculty Achievement Award in 2001 from the University of Michigan. He also received a Distinguished Alumni Award from the College of Engineering, Seoul National University in 2002, and the IEEE Communications Society William R. Bennett Prize Paper Award in 2003. Professor Shin has served on numerous conference program committees and as editor of several technical journals.

CECS was pleased to present this CECS Distinguished Lecture and host this outstanding professor and researcher.

CECS DAC Open House

The Center for Embedded Computer Systems (CECS) hosted an Open House, in conjunction with the 2003 Design Automation Conference, on June 6, 2003 from 2:00 PM to 5:00 PM, Over 125 attendees had the opportunity to visit with our graduate students and faculty affiliates, review the research project posters, and enjoy refreshments. Of these attendees it is interesting to note the foreign countries represented: 5 from Brazil, 1 from China, 1 from France, 3 from Germany, 1 from India, 6 from Japan, 1 from Portugal, 3 from Sweden, 8 from South Korea, and 1 from Taiwan.

During the Open House Professor Gajski conducted a raffle of a CECS baseball cap which was won by Atsushi Takahashi, Japan, and a CECS polo shirt which was won by D. C. Chen, Taiwan.

All of us at CECS were delighted with the turn out and hope everyone attending had a very enjoyable experience in learning more about the CECS research program in embedded systems.

CECS Merchandise

During the CECS DAC Open House, several attendees inquired about purchasing a CECS polo shirt and/or CECS baseball cap. Because of this interest, we are please to announce that we will offer, to our friends, the CECS polo shirt (small, medium or large sizes) for $25.00 and the CECS baseball cap (adjustable size) for $15.00. This CECS merchandise is being offer at our cost—we are making absolutely no profit on these sales. If you are interested in making a purchase, please email Juancho Banaag at *banaag@cecs.uci.edu* and request the items desired and payment arrangements. We are very pleased with the interest shown by our friends in this CECS merchandise. We hope you enjoy wearing these distinctive items!

Easy SoC Design

This presentation is a condensed version of a paper titled ***System-Level Design: The Easy Way*** *by Daniel D. Gajski and Justin E. Harlow III published in SRC's Cavin's Corner dated April 10, 2003.*

Current System-on-Chip (SoC) complexities are forcing everyone to rethink their design strategies and methodologies. It is no longer possible to design millions of gates using traditional methods and tools. Everyone agrees that one obvious solution to the management of complexity problem is to elevate the level of abstraction or, in other words, use bigger building blocks. However, there are not good models, tools and methodologies commercially available to support system-level design. Thus, the same questions remain: *How do we effectively manage system design complexity and continue generating productivity gains?*

Before we examine system-level abstraction we should notice that system-level design requires a different *state of mind* than designing at the logic level. In the past (circa 1960-1980) we used a ***Design-Simulate*** methodology in which the logic design was completely finished at the gate level before we validated it by simulation. By the end of the century (circa 1980-2000) we replaced the ***Design-Simulate*** methodology with a ***Describe-Synthesize*** methodology, enabled by the introduction of automated logic synthesis. With logic synthesis, designers first describe a design in terms of logic equations or even as a timed, register-level architectural description, while the synthesis tool produces the netlist of gates and flip-flops. This was a significant productivity improvement over the previous approach, but by the 2000's, the complexity of SoC designs had increased to millions of gates, and neither of the previous methodologies was economically feasible for the largest designs. Synthesis brought an enormous productivity improvement over manual logic design, but we are again facing a productivity crisis, for the same reason: The gap between the SoC specification and the gate netlist is too great. Designers cannot afford to execute a costly design process, only to find out that the final gate netlist does not completely satisfy the specification. To overcome this current productivity crisis, we must divide the gap from specification to gate netlist into several design phases and recursively apply a ***Specify-Explore-Refine*** (SER) methodology. In each design phase, we explore alternatives, select the most optimal one, and refine the specification to reflect the design decisions. The new refined design model becomes the starting specification for the next design phase. The SER methodology can be iterated as many times as necessary until the optimal gate netlist is derived. This SER methodology promises significant productivity improvements owing to the high efficiency of automatic model refinement.

Automatic model refinement is only possible if the starting and ending models are well defined, if there is a set of well-defined design decisions to be made and for each design decision there is a corresponding well-defined model modification. In order to explain ***automatic model refinement*** we will look at the system-level design flow. We will assume that the system-level design starts with large amounts of C code which serves as the system specification. We assume that proper algorithms and data structures have already been selected. First, we have to convert this C code into a technology-independent, behavioral model by encapsulating C code into behaviors and channels of proper granularity, exposing concurrency and introducing hierarchy. Second, we want to generate a system-level architectural model through a series of design decisions and model refinements including allocating components, partitioning the specification to allocated components, inserting ports, protocols, and timing. Third, we want to generate a communications model by replacing channels and busses, creating arbiters, interrupt controllers, and interface translators, creating tasks and inserting RTOS models, synthesizing application software, device drivers, and other software routines for standard processors in the design.

Automatic model refinement produces huge productivity gains! Application experiments conducted at CECS have achieve productivity gains of 2000X. With such enormous productivity gains, it is easy to understand why industry is welcoming this innovative SoC methodology.

This leads us to the development of a generalized SoC design methodology. There are basically two approaches in generating system-level design methodologies: ***syntax-first*** and ***semantics-first***. The ***syntax-first*** approach strives to define first the syntax of an extensible system-level language and allows everyone to experiment and extend it for their own application or purpose. This leads to chaos, since everyone develops slightly variant models that are not interoperable. Although many of the models are very similar or almost the same, they are not portable outside the inventing organization. This is the primary reason that IP exchange and trade is not blossoming today as expected.

The more sensible alternative is the ***semantics-first*** approach, which strives to define the semantics of models and refinements first, and the language to cover it last. Furthermore, if refinements are provably correct, then the designers do not even have to learn the system-level language. Even further, if the original specification model is given in an extension of C which can be captured graphically, then the issue concerning underlying system-level language becomes irrelevant. The C language is necessary because standard processors have C compilers and not VHDL/Verilog compilers for software generation. Hardware can also be expressed in a styled C language. These requisite C extensions must support processing concurrency and synchronization for modeling SoC architectures.

The main question still remains: *How many models are required to model an SoC*? In general, fewer models and refinements are required than the syntax-first approach. The consensus of the design community suggests that the minimal modeling requirements at the system-level are: (a) a behavioral model or executable specification that defines system behavior without implementation for checking system functionality, (b) an architectural model consisting of selected components and connections for checking concurrency and synchronization, and (c) a communication or bus-functional model to evaluate data transfers and protocols. Now each component or behavior can be modeled on a functional or cycle-accurate level requiring only semantics for two levels, three models, and a few dozen transformations. Thus, the primary challenge consists largely of formalizing the semantics of known models. The time has arrived to standardize semantics for SoC design, since progress is not possible without it!

Interested— you're invited to become proactive, encouraged to make technical contributions, and join the crusade of advocates for easy SoC design!

Visitor Profile

CECS is pleased to profile Visiting Professor Chun-Myoung Park, Professor of Computer Engineering, School of Electrical, Electronic and Information Engineering at Chungju National University, Chungju, South Korea who is spending his sabbatical year with CECS. He was born in Inchon, South Korea and received the BSc degree in 1983, the MSc degree in 1986 and the PhD degree in 1994, all from Inha University, Inchon, South Korea. Since 1995, he has been a professor at Chungju National University.

Professor Park organized and currently serves as Director of the Global Media Research Institute (GMRI) at Chungju National University. GMRI's research programs are focused on embedded systems, multimedia systems, and information security.

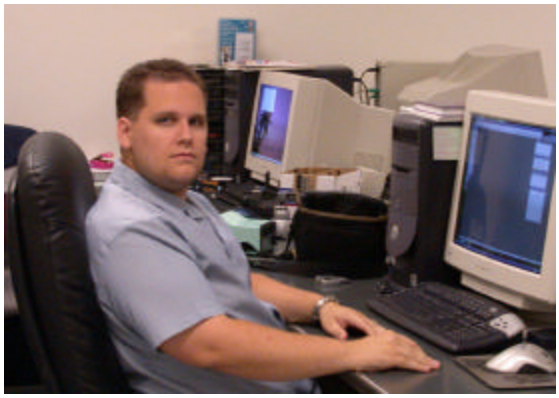
Professor Park is currently serving as Editing Chair of the Korea Multimedia Society (KMMS) and is a member of the Institute of Electronics Engineers of Korea (IEEK) and the Digital Contents Society of Korea (DCSK). In 2001, Professor Park served as Conference Chair of KMMS.

Professor Park's research interests are in embedded systems and multimedia systems. He has authored 75 technical publications and the following are three selected publications:

- *The Construction of Universal Multiple Processing Unit Based on De Bruijn Graph*, Chun-Myoung Park and Hong-bok Song, ITC-CSCC 2002
- *A Study on Constructing Highly Adder/Multiplier Systems Over Galois Fields*, Chun-Myoung Park, ITC-CSCC 2000
- *A Study on Constructing Multimedia Systems Based on Embedded Systems*, Chun-Myoung Park, Chungju National University, 2002

Graduate Student Profile

CECS has selected Roman Lysecky as an outstanding graduate student researcher. He received a B.S. in Computer Science in 1999 from the University of California, Riverside (UCR). As a UCR undergraduate student, he received awards for Outstanding Student in Computer Science and IAE College of Fellows Outstanding Student in Engineering. After receiving his MS in Computer Science from UCR in 2000, he worked for a year at Conexant Systems, Inc. in Newport Beach, CA. In 2001, he returned to UCR and is currently a PhD candidate in his third year.



Roman's PhD research is focused on developing WARP, which executes standard software binaries but transparently and dynamically partitions the critical regions on the executing application to a hardware implementation using on-chip configurable logic. This research is being conducted under the supervision of Professor Frank Vahid. His research interests include dynamic hardware/software partitioning, development of lean synthesis, technology mapping, and place and route tools targeted to execute on-chip within WARP processors, and designing a simple, yet robust, configurable logic fabric to enable the design of such lean tools.

Roman has an impressive publication record, and at the Design Automation and Test in Europe Conference in 2000, he received the Best Paper Award for *Techniques for Reducing Read Latency for Core Bus Wrappers* coauthored with Frank Vahid and Tony Givargis.

Some of Roman's selected publications are:

- *A CoDesigned On-Chip Logic Minimizer*, Roman Lysecky and Frank Vahid, to appear in Proceedings of the CODES + ISSS, October 1-3, 2003
- *Dynamic Hardware/Software Partitioning: A First Approach*, Greg Stitt, Roman Lysecky and Frank Vahid, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 250–255
- *On-Chip Logic Minimization*, Roman Lysecky and Frank Vahid, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 334–337
- *A Fast On-Chip Profiler Memory*, Roman Lysecky, Susan Cotterell and Frank Vahid, Proceedings of the 39th Design Automation Conference, June 10-14, 2002, pp 28-33

The following were published by CECS faculty affiliates during the period of April 1, 2003 to June 30, 2003:

Focus	Title, Authors, Publication
Mode Transitions	*Scalable Modeling and Optimization of Mode Transitions Based on Decoupled Power Management Architecture*, Dexin Li, Qiang Xie, and Pai Chou, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 119—124
H/S Partitioning	*Dynamic Hardware/Software Partitioning: A First Approach*, Greg Stitt, Roman Lysecky and Frank Vahid, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 250—255
Communications Refinement	*Automatic Communication Refinement for System Level Design*, Samar Abdi, Dongwan Shin and Daniel Gajski, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 300—305
Logic Minimization	*On-Chip Logic Minimization*, Roman Lysecky and Frank Vahid, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, 2002, pp 334—337
Test Time/Volume Compression	*Test Application Time and Volume Compression Through Overlapping*, Weenjing Rao, Ismet Bayraktaroglu, and Alex Orailoglu, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 732—737
Instruction Set Simulation	*Instruction Set Compiled Simulation: A Technique for Fast and Flexible Instruction Set Simulation*, Mehrdad Reshadi, Prabhat Mishra and Nikil Dutt, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 758—763
Cache Indexing	*Improving Indexing for Cache Miss Reduction in Embedded Systems*, Tony Givargis, Proceedings of the 40th Design Automation Conference, June 2-6, 2003, pp 875—880
Cache Recinfiguration	*Adaptive Online Cache Reconfiguration for Low Power Systems*, Andre Nacul and Tony Givargis, UCI CECS Technical Report 03-01, April 23, 2003
Formal Verification	*Formal Verification of Specification Partitioning*, Samar Abdi and Daniel Gajski, UCI CECS Technical Report 03-06, April 23, 2003
Transaction Modeling	*Transaction Level Modeling in System Level Design*, Lukai Cai and Daniel Gajski, UCI CECS Technical Report 03-10, April 2003
C -> SpecC	*C to **SpecC** Conversion Style*, Kiran Ramineni and Daniel Gajski, UCI CECS Technical Report 03-13, April 4, 2003
Loop Shifting and Compaction	*Loop Shifting and Compaction for the High-Level Synthesis of Designs with Complex Control Flow*, Sumit Gupta, Rajesh Gupta, Nikil Dutt, and Alex Nicolau, UCI CECS Technical Report 03-14, April 2003
Energy Efficient Communications	*Energy Efficient Communication for Quality and Reliability Aware Sensor Networks*, Cristiano Pereira, Sumit Gupta, Koushik Niyogi, Iosif Lazaridis, Sharad Mehrotra, and Rajesh Gupta, UCI CECS Technical Report 03-15, April 2003
Algorithms for Synthesis	*Greedy and Heuristic-Based Algorithms for Synthesis of Complex Instructions in Heterogeneous-Connectivity-Based DSP's,* Partha Biswas and Nikil Dutt, UCI CECS Technical Report 03-16, April 29, 2003
Exploration Framework	*A Framework for GII-Driven Design Space Exploration of a MIPS4K-Like Processor*, Sudeep Pasricha. Partha Biswas, Prabhat Mishra, Aviral Shrivastava, Atri Mandal, Nikil Dutt and Alex Nicolau, UCI CECS Technical Report 03-17, April 29, 2003
SpecC : SystemC	*Comparison of SpecC and SystemC Languages for System Design*, Lukai Cai, Shireesh Verma, and Daniel Gajski, UCI CECS Technical Report 03-11, May 15, 2003
Power Management	*Integrated Power Management for Video Streaming to Mobil Handheld Devices*, Radu Cornea, Shivajit Mohapatra, Nikil Dutt, Alex Nicolau and nalini Venkatasubramanian, UCI CECS Technical Report 03-19, May 28, 2003
Interface Synthesis	*Interface Synthesis Using Memory Mapping for an FPGA Platform*, Manev Luthra, Sumit Gupta, Nikil

CECS—Solving Tomorrow's Problems!

Center for Embedded Computer Systems, University of California, Irvine

***CECS** Mission Statement:*

To conduct leading-edge interdisciplinary research in embedded systems, emphasizing automotive, communications, and medical applications, and to promote technology and knowledge transfer for the benefit of the individual and society.

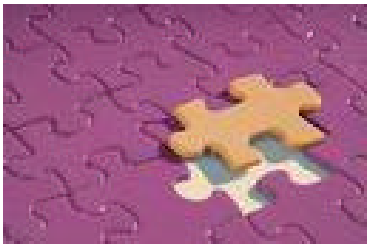
Primary Contact:
Robert P. Larsen
Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-3425
Phone: 949-824-9638
Fax: 949-824-8919
Email: larsen@cecs.uci.edu

SMILE—you're using an embedded system!

All commercial products today consist of embedded microprocessors; general-purpose or application-specific. Today, system designers are afforded the integration luxury of simply plugging in the microprocessor into the mosaic architecture of the system. Thus creating a plug and play style of design. This cell oriented modularity forms the basis of most embedded systems today. These embedded systems range from personal digital assistants to wireless phones to robotic toys.

Just what is an embedded system? I have struggled with succinctly answering this question for sometime. I am proposing the following simple definition: *"An embedded computer system is a heterogeneous system containing at least one computational component."*

The simplest visualization of an embedded computer system is the following picture.



Here the entire puzzle represents the end product platform—the heterogeneous system. The orange puzzle piece represents the embedded microprocessor which is inserted in a plug and play fashion. This hardware integration style is suitable for general purpose and application-specific modules. All the system customization is accomplished in software; thus, the product value-added is primarily provided through the application-specific software.

In today's society, every one, young or old, is being touched by an embedded computer system every minute of every day. The exciting application domains for embedded computer systems includes automotive, communications, entertainment, educational, industrial, medical, and military . These wide ranging application domains illustrates the multidimensional challenges confronting the design of innovative embedded computer systems.

Wow! Embedded systems are pervasive and ubiquitous. So the next time you push the start button on your microwave oven—smile! The next time you answer your wireless phone–smile! The next time you listen to your digital radio or apply the brakes in your car–smile! Now, embedded computer systems are responsible for a happier society. SMILE!

Bob Larsen