JULY 2006

Currents

IEEE Council on Electronic Design Automation

First IEEE Programming Challenge at IWLS sponsored by CEDA

Contributed by Christoph Albrecht, Cadence Berkeley Labs, calb@cadence.com

IEEE CEDA sponsored together with Synplicity and Cadence Design Systems the first programming challenge of the International Workshop on Logic and Synthesis (IWLS). The challenge was to implement logic optimization algorithms on the industrial EDA database OpenAccess. The programming challenge was organized and co-chaired by **Florian Krohm** (IBM) and **Christoph Albrecht** (Cadence Berkeley Labs). A committee consisting of **Robert Brayton** (UC Berkeley), **Valavan Manohararajah** (Altera) and the co-chairs judged the entries by how good the implementations make use of the OpenAccess database and the OA Gear infrastructure, how well the functionality is architected to allow maximum versatility, how the algorithms scale for large circuits, how good the unit and regression tests and the documentation are, and how well the OA Gear coding standard is followed. The goal of the programming challenge and OA Gear is to build and foster a new open source logic synthesis system which will provide the base for a complete RTL-to-layout implementation flow that also includes physical synthesis.

OA Gear is an open source project initiated by Cadence Berkeley Labs and developed mostly by students. It provides a collection of useful utilities to enable academic research with OpenAccess. Currently OA Gear consists of an RTL-Verilog reader and synthesis into a technology independent netlist (and-inverter graph), a simple mapper which directly maps the nodes of the and-inverter graph onto a specified set of three library elements (AND, NOT, FF), accurate timing analysis with slew propagation, and a simple equivalence checker which is based on the and-inverter graph representation. For more information about OA Gear visit the OA Gear project page at http://openedatools.si2.org/oagear/.

With the help of the funds from CEDA six participating students received a travel grant and could travel to the IWLS workshop held June 7 – 9, 2006 in Vail, Colorado. In addition to the travel grants the committee of the programming challenge awarded two entries of the challenge with a "Best Contribution Award". Each of the two awards came with a cash prize of $250.

The students **Kai-hui Chang** and **David A. Papa** from the University of Michigan received the award for their contribution "Simulation and Equivalence Checking". They developed a logic simulation engine which is 100 times faster than the simulator released previously with the OA Gear package. They use their simulator in combination with MiniSAT for combinational equivalence checking, and developed a new metric of circuit similarity that can be used in incremental verification and debugging. Kai-hui and David also integrated their simulator and equivalence checker into the OA Gear GUI Bazaar.

Qi Zhu and **Nathan Kitchen** from the University of California at Berkeley received the award for their submission entitled "SAT sweeping with Local Observability Don't Cares." They implemented a SAT sweeping algorithm which has applications in Boolean reasoning and functional verification. SAT sweeping finds nodes which are equivalent by simulating random vectors and then proving the equivalence by solving a SAT instance. The circuit is simplified by merging equivalent nodes. Qi and Nathan extended this technique by merging nodes which are not functionally equivalent but whose functional difference is not observable within paths of bounded length. Their implementation makes extensive use of the OA Gear and-inverter graph.

Both prize-winning submissions will become part of the next OA Gear release, and it is planned to launch the programming challenge again next year.

CEDA Currents is a publication of IEEE CEDA. Please send contributions to Kartikeya Mayaram, karti@eecs.oregonstate.edu or Preeti Ranjan Panda, panda@cse.iitd.ac.in.

ILWS Programming Challenge winners with the event organizers. Left to right on the back row: Robert Brayton, Valavan Manoharajah, David A. Papa, Kai-hui Chang, Florian Krohm, Petra Faerm, Donal Chai. Front row: Qi Zhu, Nathan Kitchen, and Christoph Albrecht.

IEEE Circuits and Systems Society Awards 2006 Announced

Education Award: **Wayne Wolf**
For outstanding education and leadership in VLSI systems and embedded computing.

Industrial Pioneer Award: **John A Darringer**
For the development of practical techniques and algorithms for automated logic synthesis, for their realization as usable tools, and for their successful application to high performance computing products.

TCAD Donald O. Pederson Best Paper Award: **Mark Kassab, Janusz Rajski, Jerzy Tyszer, Nilanjan Mukherjee**
Embedded Deterministic Test, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 23, no. 5, pp. 776-792, May 2004.

TVLSI Best Paper Award: **Bipul C Paul, Animesh Datta, Kaushik Roy, Amit Agarwal, Hamid Mahmoodi**
Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technologies, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 1, pp. 27-38, January 2005.

CEDA Distinguished Speaker Reception

The *Council's* Distinguished Speaker Series features detailed presentations of the most significant research results in EDA over the past year, as demonstrated by awards at our top conferences and journals. The second presentation in this series will feature **Janusz Rajski, J. Tyszer, M. Kassab,** and **N. Mukherjee**, the authors of this year's IEEE Transactions on Computer Aided Design Donald O. Pederson Best Paper Award.

The *Council* invites you to the Distinguished Speaker Reception at **5:30 PM on Monday, 24 July 2006** during DAC at the **Moscone Center Room 124,** San Francisco, CA. The reception will be followed by an introduction to the Council and the above presentation.

Top Articles in CEDA Publications for 2005

The *Council* publications consist of jointly sponsored *IEEE Transactions on CAD* (jointly with IEEE Circuits and Systems Society) and *IEEE Design and Test of Computers* (jointly with IEEE Computer Society and IEEE Circuits and Systems Society). Here are the top 20 articles as ranked by the number of PDF downloads from IEEE Xplore during the year 2005.

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, *vol(number), downloads*

1. BSIM plus: an advanced SPICE model for submicron MOS VLSI circuits, 13(9), 9,374
2. A unified approach to reduce SOC test data volume, scan power and testing time, 22(3), 3,989
3. LPRAM: a novel low-power high-performance RAM design with testability and scalability, 3(5), 971
4. Local watermarks: methodology and application to behavioral synthesis, 22(9), 755
5. VHDL-AMS and Verilog-AMS as alternative hardware description languages for efficient modeling of multidiscipline systems, 24(2), 740
6. Digital ground bounce reduction by supply current shaping and clock frequency Modulation, 24(1), 601
7. Interconnect-aware low-power high-level synthesis, 24(3), 596
8. An RLC interconnect model based on Fourier analysis, 24(2), 576
9. Asymptotic waveform evaluation for timing analysis, 9(4), 557
10. Accurate estimation of total leakage in nanometer-scale bulk CMOS circuits based on device geometry and doping profile, 24(3), 536
11. C-based SoC design flow and EDA tools: an ASIC and system vendor perspective, 19(12), 506
12. Implementation of a UMTS turbo decoder on a dynamically reconfigurable platform, 24(1), 490
13. Indirect test architecture for SoC testing, 23(7), 486
14. A CAD methodology for optimizing transistor current and sizing in analog CMOS design, 22(2), 478
15. Analysis of power dissipation in embedded systems using real-time operating systems, 22(5), 478
16. Compact reduced-order modeling of weakly nonlinear analog and RF circuits, 24(2), 472
17. Automated bus generation for multiprocessor SoC design, 23(11), 444
18. Capacitive coupling noise in high-speed VLSI circuits, 24(3), 442
19. A general hierarchical circuit modeling and simulation algorithm, 24(3), 437
20. Behavioral modeling for high-level synthesis of analog and mixed-signal systems from VHDL-AMS, 22(11), 422
21. MOS table models for circuit simulation, 24(3), 422.

IEEE Design and Test of Computers

1. Dynamic power management in wireless sensor networks, 18(2), 1,072

2. Evaluation of MEMS capacitive accelerometers, 16(4), 653
3. FPGA and CPLD architectures: a tutorial, 13(2). 611
4. Driving the $5 Billion Innovation Engine at Intel: An Interview with Patrick P. Gelsinger, Digital Enterprise Group Senior Vice President and General Manager, Intel, 22(2), 551
5. A methodology for architectural design of multimedia multiprocessor SoCs, 22(1), 539
6. Jitter measurements of high-speed serial links, 21(6), 499
7. Behavioral simulation of fractional-N frequency synthesizers and other PLL circuits, 19(4), 492
8. Software development for high-performance, reconfigurable, embedded multimedia systems, 22(1), 464
9. Architecture exploration for a reconfigurable architecture template, 22(2), 458
10. Design, synthesis, and test of networks on chips, 22(5), 431
11. The truth about outsourcing, 22(1), 420
12. Platform-based design and software design methodology for embedded systems, 18 (6), 405
13. Soft errors in advanced computer systems, 22(3), 401
14. Seamless hardware-software integration in reconfigurable computing systems, 22(2), 398
15. System-on-chip or system on package?, 16(2), 393
16. Jitter models for the design and test of Gbps-speed serial interconnects, 21(4), 385
17. BEE2: a high-end reconfigurable computing system, 22(2), 382
18. FPGA-enabled computing architectures, 22(2), 370
19. New test paradigms for yield and manufacturability, 22(3), 344
20. AEthereal network on chip: concepts, architectures, and implementations, 22(5), 331

Upcoming CEDA Events*/Dick Smith, dsmith@topher.net*	
CODES+ISSS	www.esweek.org
DAC	www.dac.com
DATE	www.date-conference.com
FMCAD	www.fmcad.org
ICCAD	www.iccad.com
MEMOCODE	memocode.irisa.fr
MPSoC	tima.imag.fr/mpsoc
PATMOS	www.patmos-conf.org
VLSI-SOC	tima.imag.fr/conferences/VLSI-SoC06
Nano-Net	www.nanonets.org

Opinion: Who Verifies Your Third-Party Design IP?

Contributed by Vigyan Singhal and Harry D. Foster

Many forces at play contribute to a gap between what we can manufacture (silicon capacity) and what we have time to design. These forces contribute also to the gap between what we can design and what we have time to verify. Third-party IP offers the promise of filling these gaps by increasing both design and verification productivity. For instance, to improve design and verification productivity, a startup developing a WiFi chipset might rely on an IP vendor to deliver a fully functional PCI Express interface block—thus allowing them to focus engineering efforts on their wireless datapath. However, the startup needs assurances that the third-party IP is functionally defect-free and will not delay their production schedule.

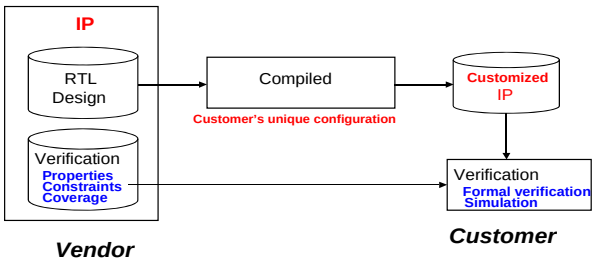
Who is responsible for the functional correctness of third-party IP—particularly soft RTL cores? IP customers expect that the IP they purchase will be sound and will not require re-verification. On the other hand, it is not hard to empathize with the IP vendor who would like to thoroughly verify the IP, but is challenged with the difficult task of verifying the IP under multiple compile-time configurations. The IP vendor is also pressured by a competitive marketplace with extremely thin margins. This environment has vendors racing to complete verification and find all the bugs before their early customers find them (which doesn't always happen). In addition, some customers tinker with the RTL, jeopardizing the verification confidence the vendor originally achieved. Can verification tool vendors be responsible? They are too far removed from the application problem, but given a suitable solution, they would happily introduce products to this ecosystem. Perhaps the solution lies with IP standards organizations, such as PCI-SIG and HyperTransport Consortium, to define executable verification requirements for IP compliance and interoperability.

Given the disproportionate amount of resources spent in verification activities, it seems right to demand that IP be redefined as "IP = Design + Verification." The IP vendor's verification environment should be designed for reuse. Hence, the IP customer would reuse the verification for evaluation, custom configuration during integration, and design-specific modifications (See Figure).

Formal property verification, enabled by recent assertion language standardization efforts (PSL and SVA), plays a key role in reusable verification. Though such verification can be successfully deployed on designer-sized blocks, it still needs significant user test planning and effort to be effective. Using powerful but complex verification strategies such as abstractions, reduction, and compositional verification can enable end-to-end verification of previously intractable designs. Even though the effort in accomplishing this task is non-trivial, it has significant return in the context of IP delivery. The up-front cost in implementing these strategies can yield reusable proof scripts—enabling customers to validate the correctness of the IP for their specific compiled configuration.

Deliverable verification for IP is not limited to formal property verification. Coverage goals and performance modeling are other examples of verification components that could be delivered to enable a smoother transition of the IP from the vendor to the customer. By defining third-party IP as "IP = Design + Verification," while crafting the IP vendor's verification environment for

reuse, the IP promise of increasing both design and verification productivity will be achieved.



Reusing IP vendor's verification environment on customers configuration

Response: IP vendors can not leave verification to the user

by Grant Martin, Sumit Gupta and Dhanendra Jani

IP providers must verify their deliverables. Their very existence depends on it, which is a strong motivator. The verification problem is compounded for complex configurable IP, such as a configurable and extensible processor. Besides architectural and micro-architectural verification, the IP provider must verify the process of generating an instantiation of the IP based on configuration and extension parameters. Finally, each execution of the configuration and generation process needs to incorporate verification steps, before delivery. This demands an automated verification environment.

A wide variety of methods need to be used for all three levels. With configurable IP, with a large design space, every new advance in verification tools and methods might increase the quality and confidence of the provider's verification processes. From assertion-base design and verification, through property checking, to new formal methods, it is important to monitor and incorporate them when proven to add value. Specific IP domains, such as processor design, may also generate domain-specific verification tools. High quality IP providers are constantly improving their verification processes.

The quality of IP has always been governed by IP = Design + Verification. If an IP provider is not putting at least twice as much effort into verifying their IP as in designing it, and leaves the verification to their users, then the productivity gains provided through IP will disappear. Of course, the IP user must verify the IP within the SoC context to check for correct interconnection and use.

There is no doubt that it takes a few successful customers with successful working silicon to truly validate the IP provided by a IP vendor. IP users are thus well advised to choose IP providers who demonstrate through their track record that they deliver quality pre-verified IP.

Response: Formal Verification Specification for IPs Benefits IP Creators as well as Consumers

by Pranav Ashar, CTO, Real Intent, Inc.

It is time that IP creators and consumers both begin viewing formal functional specification of IPs using assertion languages like PSL and SVA as a fundamental advance that lubricates the third-party IP methodology and marketplace.

From the perspective of IP creators: Assertion based verification tools have reached a level of maturity such that using assertions with simulation and formal tools tangibly improves verification efficiency. Writing PSL/SVA assertions and constraints today is becoming a means to shorten project schedules rather than a cause for project delays. More importantly, these assertions also can be a self-documenting test plan for the design. An IP creator with such a test plan can easily leverage it as a competitive advantage. Given that an assertion-based functional spec for IPs pays for itself in these multiple ways, it is expected that the "IP = Design + Verification" view will gain more ground soon.

From the perspective of IP consumers: An IP accompanied by an assertion-based spec is certain to be better verified and better documented than one without. Beyond that, the IP consumer can leverage the fact that a PSL/SVA based spec is executable. The IP assertions, when added to the assertion database of the entire SOC, provide analysis tools with important short cuts that make the overall verification a lot more efficient. They also enable an automatic or manual compositional verification methodology that is sometimes critical for making the SOC verification succeed. Finally, the executable assertion spec can act as a proxy for the IP's RTL in many cases, providing the consumer visibility into the IP without revealing its implementation details. Given these multiple advantages, it is only a matter of time when more IP customers begin demanding it.

Well, that is what the experts think. Do you have an opinion that you would like to share with our readers? If so, please drop us a line to karti@eecs.oregonstate.edu or panda@cse.iitd.ac.in.