

PANEL: Tools or Users: Which is the Bigger Bottleneck?

Chair: Andrew B. Kahng, Univ. of California at San Diego, La Jolla, CA

Organizers: Bob Dahlberg, ReShape, Inc., Mountain View, CA

Abstract

As chip design becomes ever more complex, fewer design teams are succeeding. Who's to blame? On one hand, tools are hard to use, buggy, *not interoperable*, and have missing functionality. On the other hand, there is a wide range of engineering skills within the user population, and tools can be abused within flawed methodologies. This panel will quantify and prioritize the key gaps, including interoperability, that must be addressed on both sides.

Panelist Statements

Ron Collett

Numetrics Management Systems, Santa Clara, CA

Tools, users and interoperability are all bottlenecks. This is well known. Each impacts design productivity and therefore cycle time and product development cost. Thus, it is not a question of which one of these is "the" bottleneck, but rather what is going to be the bottleneck on a particular project. The impact of each varies from project to project. The dominant bottleneck will vary depending on design complexity and time-to-market constraints. Let us assume that there will always be bottlenecks and thus a range of uncertainty in the product development process. The most important issue, then, is determining the impact of a particular set of tools, users and interoperability levels on cycle time for a particular chip design project. This must be done at the outset of the project and updated throughout the project, as design specifications and time-to-market constraints invariably change. Thus, the most important question is how to best manage inevitable bottlenecks and therefore the ever-present risk inherent in each development project.

Patrick Groeneveld

Eindhoven University of Technology / Magma

We must continue the drive toward complete IC design automation, to the point where most actions become almost push-button. The key is a carefully tuned RTL to GDSII design flow, in which hundreds of smaller steps encode a wealth of design methodology know-how and experience. Treating 'logic synthesis', 'placement' and 'routing' as big monolithic design steps is no longer compatible with achieving timing closure, managing on-chip parasitics and dealing with the many DSM peculiarities. Rather, harmonic orchestration of the many tooling steps in the flow that determines overall implementation success, and is more relevant than quality of individual point algorithms. A robust IC implementation flow simultaneously addresses many design constraints and DSM issues: timing closure, electromigration, crosstalk noise, multiple power supply regions with voltage drop constraints, etc. are just a few. The key is to find the proper equilibrium between the conflicting design goals and constraints. Building, debugging and tuning such a flow is the primary added value of a next generation EDA company.

An integrated data model is the enabling technology of a successfully automated RTL-to-GDSII design flow. The data model

supplies reliable data on parasitic coupling capacitances, wire delays, system timing, congestion/DRC etc., even in the earlier steps of the flow when little physical data is known. Running the steps directly on the data model ensures fast run times and consistency of design data. For instance, the effect of inserting an additional buffer on a wire results in a change of the net list and wiring that in turn triggers a fast incremental update of the timing and congestion data. Combined with Gain Based Synthesis technology and proper integration of design constraints, this can ensure timing closure without iteration. The ultimate result is a single integrated tool that maps RTL to a working chip.

Lavi Lev

Cadence Design Systems, Inc., San Jose, CA

The main cause of failure and bottlenecks in developing and deploying complex SOC chips into the market is the inability of tools and people to interoperate. If one looks at the development and deployment philosophy of the EDA industry, one can come to the conclusion that complex SOC's are conceived, designed, and fabricated by a single individual using a single EDA company's tool to develop the SOC. In reality, developing a SOC requires hundreds of people who are superb technical problem solvers, but who leave a lot to be desired in the areas of communication and teaming. As a result, a huge number of bugs and schedule delays have nothing to do with the technology; they are the direct result of poor coordination and synchronization between dispersed team and pieces of data.

The EDA industry does not focus on the areas of coordination and synchronization that beg for automation. This is because most EDA engineers have never experienced the joy of building a chip. Instead, they almost exclusively concentrate on what they know how to do best - solving technology problems. To make things even worse, no single EDA provider alone can provide all the tools needed to develop and deploy a SOC. As a result, these large teams are spending millions of dollars and endless frustrating hours trying to make all the tools work with each other. Two components are required to streamline and simplify the complex task of building large chips. First, we require an Open, Industry Standard database (e.g., OpenAccess) that will allow tools from various EDA providers to interoperate. Second, we require involvement of EDA companies in providing tools to assist with the task of managing and synchronizing all the elements that participate in large IC design projects. This requires the bringing in and blending of chip design talent into the EDA industry to enable creation of practical solutions for the market.

Nancy Nettleton

Sun Microsystems, Inc., Palo Alto, CA

Asking the question "Tools or Users (or Interoperability): Which is the Bottleneck?" points the finger at the most readily available culprits but leaves the larger questions unasked. How about project structures set up to handle small designs with less than 20 engineers

that collapse under the weight of 15M VDSM gates? How about design managers who haven't seen a transistor since they put down Weste & Eshraghian suddenly trying to handle both cell placement and signal integrity in disparate, uncommunicative tools that interpret timing and noise differently? How about business teams used to ASIC service who think that parts can be fabbed at UMC by design contracting services?

Recent dramatic increases in chip design complexity have coincided rather unfortunately with fundamental shifts in supporting business models. This collision of increased complexity with changing business models has spread and exacerbated today's design bottlenecks. Three examples are as follows. (1) Wide availability of VDSM silicon process technologies has put microprocessor-like complexity within reach of more design teams who do not have the microprocessor margins that support extensive CAD organizations. Chip integration and design size issues that would have affected a handful of microprocessor design teams seven years ago are impacting hundreds of design teams today. (2) Tight-pitched VDSM metal enables us to pack millions more transistors onto a die, but engineers are waking up to the bad dream of post-route signal integrity iterations every bit as painful as the old timing closure loop. (3) System signaling strategies cause major upticks in volume and complexity of analog design, just as workhorses of system analog design (ASIC suppliers) are sapped by increasing cost of VDSM fabs. System teams who push performance envelopes with tricky signaling strategies based on, complex analog and package design must deal with multiple players, instead of one party who understands all of the technologies involved (device, package, test). Instead of focusing on system logic, these design teams must shop for or design their own analog IP, then try to mate it with separate fab and package houses. This increases risks of missed specs or market windows.

So, tools, users, interoperability: bottlenecks, or symptoms of industry changes that have not yet been absorbed? If we focus on addressing the fundamental project and technology changes within the emerging business models, then solutions to these three "bottlenecks" will naturally fall out.

Paul Rodman
ReShape, Inc., Mountain View, CA

" 'Tis a poor craftsman that blames his tools". This old saw was coined in a world where craftsmen of the pre-industrial age had great understanding of the materials and the tools with which they practiced their trade. Unlike the craftsmen of old, today's chip designers rarely have an opportunity to gain in-depth familiarity with a particular set of tools or materials. Thanks to Moore's Law, IC design tools and materials are not static; each becomes more and more complicated to use. The tools used on any chip design are new (even mature tools have releases every six to 12 months), which means a new UI to learn, and new bugs and instabilities to contend with. And vendors always sing their siren's song of new features to fix last year's problems. It is no wonder that people blame their tools at the end of projects.

Will tool suppliers change? Not likely! And who knows more about how to build chips: Tool developers or Tool users? Clearly, it is users who build chips, and when EDA vendors have tried to tell users how to do their job, they've always failed. Will we see a Consumer Reports for tools and processes? www.deepchip.com has tried, but there is just so much time a passionate moderator and kindhearted users can devote to sharing what they know.

Ultimately, today's chip-designer craftsman will likely fall victim to the industrial age. Just as standard part innovation enabled the industrial age, a standard part foundation is now being laid in our industry to make automation possible. We see the rise of standard silicon intellectual property building elements. EDA tools have undergone de facto standardization in RTL simulation, logic synthesis, place and route, and DRC/LVS. Process technology may be going this way too, e.g., standardizing on a single, open 0.10-micron semiconductor manufacturing process. With such standardized tools and design elements in place, best in class methods can be codified to automate the building of chips. Yes, automation comes at a cost: craftsmen's flexibility, and the ability to optimize at the leaf level, are lost. But automation of methods will enable fast throughput, and allow engineers to do many "what if" experiments to globally optimize the design, predictably yielding smaller and faster designs.

Lambert van den Hoven
Philips, Eindhoven, The Netherlands

Tools, Users and Interoperability are all causing productivity problems. That designs are successfully made is a combination of experience (do's and don'ts), adoption of less-ideal design methodologies (e.g. combination of flat and hierarchical methodologies), guardbanding of portions of a design (e.g., for timing), trade-offs between available time and risk (e.g., test coverage), a strong drive for re-use at higher levels, a patchwork for tool deficiencies, and sometimes luck. It is no surprise that almost every design requires a redesign to meet original specs (cf. Numerics). Though the trend line is for fewer redesigns per gate count, rapid increase in complexity, impact of advanced process technologies, and prototyping costs represent growing concerns.

Tools have become more complex due to increased functionality, demand for higher capacity of individual tools, tighter connections/interfaces to the tool environment, and the need for improved user interfaces. A big challenge lies in the area of test and QA scenarios that are often in direct conflict with market pressure to release. The focus is still too much on single-tool testing, while the end user is only helped by a combination of tools suited for specific subflows. Our own experience is that most problems found in subflow release (qualification) are due to weak interfaces or lack of appropriate standards.

Interoperability has been a discussion topic for many years. Standardization far beyond the level of direct interfaces has been attempted, but without much success. Recently, new initiatives have again started to tackle this problem. New elements (e.g. open source based standards) may lead to some successes, but expectations are sometimes optimistic. A minimum requirement is to develop interoperability rules and standards for technology parameters, libraries, and IP blocks (both in hardware and software, based on various re-use levels (e.g. hard/soft macros)).

Users can be grouped into various categories, ranging from individual point-tool end user to design manager/supervisor. While the end user is more focused on functionality and stability, overall management attention is on throughput time and cost. Since truly intuitive solutions are often not available, proper (= efficient) use of point tools requires extensive training, augmented by on-the-job experience (i.e., to comprehend do's and don'ts).