

Panel:

(When) Will FPGAs Kill ASICs?

Chair: Rob A. Rutenbar, Carnegie Mellon University
Organizer: Rob A. Rutenbar

Abstract

There was a time – in the dim historical past – when foundries actually made ASICs with only 5000 to 50,000 logic gates. But FPGAs and CPLDs conquered those markets and pushed ASIC silicon toward opportunities with more logic, volume, and speed. Today's largest FPGAs approach the few-million-gate size of a typical ASIC design, and continue to sprout embedded cores, such as CPUs, memories, and interfaces. And given the risks of nonworking nanometer silicon, FPGA costs and time-to-market are looking awfully attractive. So, will FPGAs kill ASICs? ASIC technologists certainly think not. ASICs are themselves sprouting patches of programmable FPGA fabric, and pushing new realms of size and especially speed. New tools claim to have tamed the convergence problems of older ASIC flows. Is the future to be found in a market full of FPGAs with ASIC-like cores? ASICs with FPGA cores? Other exotic hybrids? Our panelists will share their disagreements on these prognostications.

Position Statements

Max Baron

Microprocessor Report,
Cahners Electronic Group, USA

Max Baron was formerly principal analyst at Cahners In-Stat Group, a Scottsdale, AZ-based research firm, and is now editor-in-chief of *Microprocessor Report*.

Thomas Daniel

LSI Logic, USA

Will FPGAs kill ASICs? Yes, of course. As surely as e-commerce was supposed to kill brick and mortar shops, Java eliminate C++ and ATM send IP to antique shops. FPGAs and ASICs address different market requirements and both will continue to exist and prosper. Just try to do a 10 million (real) gate design running at 400 MHz, with processor cores, analog cells and bunch of very high speed IOs in a FPGA. These are true SoC designs and they are purely and simply infeasible in an FPGA, not now and not any time soon. Each new religion is arrogant at its youth and FPGAs don't escape that trend. When ASICs reached 25,000 gates and we have put the first Sparc processor in a gate array, we also claimed that the end of full custom design is close. Today, we can put several Pentiums on a die, but we have grown out of such childish claims. The real battle will be about all the new applications that can be addressed by hybrid approaches. Here, ASIC vendors are by far better positioned.

Rajeev Jayaraman

Xilinx Inc., USA

Programmability and time-to-market, the centerpiece of the FPGA value proposition, are the key requirements for customers today. These are precisely the Achilles' Heel for the ASICs: they are neither programmable nor can they deliver a fast time-to-market. Cost used to be the trump card for ASICs over FPGAs. This is not the case anymore since Moore's law, the very law that made ASICs thrive in the past, has turned against the ASICs. Customers designing ASICs have to pay dearly for the cost of a mask set. And this cost is only getting higher as process geometries shrink and Moore's law takes effect.

To address programmability in ASICs, several vendors have embarked on adding programmability to ASICs. There are a couple of problems with this approach by the ASIC vendors. First, adding programmability is not just a matter of adding some programmable gates to silicon. One cannot underestimate the importance of the architecture and software for the programmable logic. Second, this approach only adds some programmable logic and this inherently limits the flexibility of applications.

Time-to-market is a serious problem with ASICs today and is getting significantly worse. This is due to the fact that the design cycle for ASICs is getting longer due to the deep sub-micron effects.

On the other hand, FPGAs have grown to become multi-million gate devices with system level features that can implement complete systems on a chip. More and more ASIC designers are using FPGAs while the number of designs moving from FPGAs to ASICs is reducing simultaneously. The design cycle for FPGAs is still fast and simple enough that the designer does not have to worry about the deep sub-micron effects, considerably reducing the time-to-market.

Considering these factors it won't be long before ASICs will be relegated to a niche market.

Zvi Or-Bach

eASIC Inc., USA

This question assumes that there will be no new innovation, such as application specific ICs with field programmability to bridge the best of both technologies. Clearly ASIC NRE and mask costs are spiraling out of control, with fewer than 1,000 out of the historic 10,000 annual design starts able to justify the design and manufacturing costs for 0.13 micron technologies. However, customization does not have to be an all or nothing proposition. The eASIC solution is to combine a regularized, programmable FPGA-like logic substrate with application-specific innovative single mask customization for the top four layers of 0.13 micron metal routing. More than just incorporation of FPGA cores on ASICs, or processor cores on FPGAs, the eASIC technology provides a continuum of possibilities to utilize the best of both technologies. The eASIC 0.13 micron products provide for 2

week TAT and modest mask costs while providing power, performance and density (20x) improvement over standard FPGAs.

Jonathan Rose

***Altera Toronto Technology Centre and
The University of Toronto***

ASICs are already dead; they just don't know it yet. Sure, there will be a need for a few super-high volume ultra-low-cost devices that will enable some ASIC chip design houses to tolerate the ever-increasing agony required to successfully design, fabricate and test one-off chips. Everyone else will give in to the warm, comfortable environment provided by programmable logic vendors: no fab, no NRE, much faster time-to-market, no fab, no test, no inventory problems, field upgrades, no fab, and all-encompassing software environments. OK, so everything isn't quite this perfect yet, but it will be, as the forces leading us there are just too strong. Whenever a potentially successful hard ASIC application comes along, the programmable logic vendors will simply incorporate it into their offerings. As the percentage of the die given over to hard cores increases, the speed and area penalties of the programmable off-the-shelf solution decreases.

Will programmable logic cores inside ASICs suffice? No, because the end user still has to fab the damn chip. Leave that up to the vendors. Also, the notion of a programmable core within a custom ASIC is all wrong: you have to figure out in advance what it is going to be connected to and therefore how it is going to be used. The right way, when a hard core is embedded into a programmable fabric you can always do whatever you need to do. AND, you don't have to make the chip!

Do you want an interesting job working on processors, network cores, wireless hardware, or whatever the next big thing is? Join us! We're going to be the only ones doing it for real; the rest of the chip world just doesn't know it yet.

Carl Sechen

The University of Washington, USA

For those who believe that ASIC design will maintain its current percentage of design starts five to ten years from now, there are some truly serious storm clouds on the horizon. In fact, the number of ASIC starts per year is already on the decline. The severe process variations and signal integrity issues encountered as we move forward to the 100 nm, 70 nm and 50 nm process regimes will rock ASIC design to its core. The design rules and design complexity, given current design methodologies, for future

sub-100 nm fabrication technologies actually have the potential of simply overpowering existing ASIC design tools and design methodologies. The challenge at 130 nm is already daunting. What I am seeing today is that layout tools are being developed, with great difficulty I might add, that merely attempt to handle a reasonably large subset of the constraints and objectives, all the while foregoing any hope of achieving anything remotely close to optimality.

Since we can no longer afford to and/or are no longer able to develop layout tools that are optimal or near optimal with respect to area, delay, etc., then why don't we gravitate toward different ASIC design methodologies and design styles where design efficiency can be exploited? Because ASIC design is being driven by proponents of old design styles, fab-controlled design methodologies, as well as circuit designers and EDA vendors that kow-tow to the fabs. Superior design styles and methodologies can be developed, and fabs must be told what makes sense from an algorithmic and efficiency point of view with respect to layout and verification tools. If this is not done soon, ASICs will be killed off permanently. This will devastate EDA layout and verification companies operating in the ASIC world.

Curiously, the number of EDA vendors in the ASIC layout domain is growing, even while you have to ask the question: Is there a long-term future for ASIC layout tool vendors? Their problems are getting ever more difficult to solve. They have to raise prices considerably to achieve or maintain profitability for two reasons: 1) the challenge of keeping up with the latest design rules and outmoded design styles demands huge development, maintenance, and application engineer costs, and 2) the number of potential customers is decreasing continually. The ASIC layout vendors should be shaking in their boots!

If you want to design an ASIC, you are handed a document the size of a Ph.D. dissertation covering the design rules, you are confronted with the equivalent of boxes of documents on how to use the EDA tools, and still you need a small army of application engineers to nurse you through a year-long design and verification process. And all when you could have done it yourself in two days if an FPGA had been used!

Sure, there are performance issues with FPGAs and density issues. The former can be addressed with new logic styles, and the latter can be addressed by introducing a small degree of mask programmability. Can ASICs compete? Surely not if business as usual is conducted. Will someone be bold enough to invent a new, efficient ASIC design style and methodology so that it can survive? That's the question.