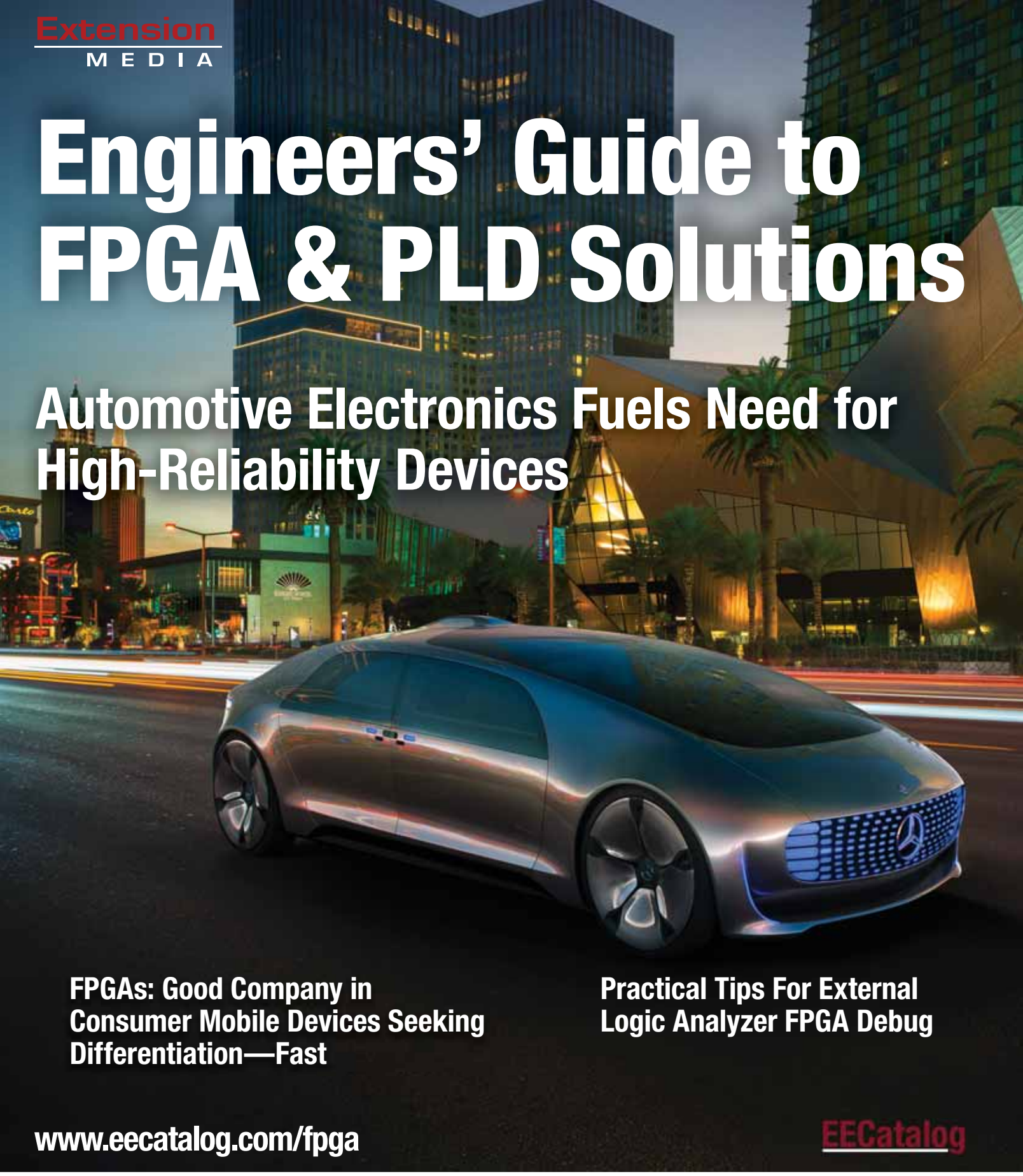
Extension MEDIA

Engineers' Guide to FPGA & PLD Solutions

Automotive Electronics Fuels Need for High-Reliability Devices



FPGAs: Good Company in Consumer Mobile Devices Seeking Differentiation—Fast

Practical Tips For External Logic Analyzer FPGA Debug

www.eecatalog.com/fpga

EECatalog

Market Leader Sponsor

XILINX ALL PROGRAMMABLE

Gold Sponsors

ELMA Your Solution Partner

SEALEVEL

Technologic Systems

A Few Questions on…FPGA-based Prototyping Software Tools

Did we say "a few"? Who's counting? We just know the topic range here spans everything from what designers ready to work on PCIe 4.0 projects should know to SoCs and design partitioning to whether FPGAs' reputation for being difficult to program is warranted.

By Anne Fisher, Managing Editor

Editor's note: Our thanks to Troy Scott, product marketing manager at Synopsys, who recently offered his insights on a number of questions. Scott is responsible for FPGA-based prototyping software tools at Synopsys. He has 20 years of experience in the EDA and semiconductor industries. His background includes HDL synthesis and simulation, SoC prototyping and IP evaluation and marketing.

EECatalog: What practices do you recommend for capitalizing on FPGAs' ability to address the challenges posed by concurrent hardware/software design?

Troy Scott, Synopsys: From an ASIC design perspective FPGAs are absolutely applied to help create a more parallel design process for hardware and software. High-performance FPGA-based prototypes make it feasible to boot an OS, develop drivers and run a software stack. And, perhaps more important, do so with a minimum number of high-capacity FPGAs to keep prototype cost as low as possible, which is a very important consideration if the prototype will be duplicated throughout the organization.

Best practices to maximize the prototype ROI typically result from collaboration between ASIC designers, the FPGA-based prototyping specialists and the software teams who use the prototype. "Over-the-wall" RTL drops are a recipe for failure and schedule delays. Design For Prototyping (DFP) RTL coding standards maximize prototype performance and speed the schedule at which it can be brought-up. At the deployment phase the most efficient prototyping teams will work closely with and treat the software team as the internal customer. Tracking software alongside RTL changes will minimize confusion as the system is integrated. Some software routines may need to account for slower operation of the prototype versus ASIC silicon. Internal probe points relevant to the software team may need to be designed in to help during the debug phase. Rapid reset cycles designed into the prototype help improve turnaround time. All of these examples require close collaboration between the prototyping team and the embedded software development team.

EECatalog: Your thoughts on some of the challenges to FPGA-based prototyping, beginning with design partitioning?

Scott, Synopsys: To achieve both rapid time to solution and highest performance you need super-fast partition software and a platform that can be tailored to the SoC needs. One example of this is what Synopsys calls the abstract partition flow with ProtoCompiler and HAPS-70. The combination of ProtoCompiler and HAPS-70 enables prototypers to quickly create an abstracted interconnect architecture representation, generate a partition solution, then incrementally customize the partition and the hardware based on the needs of the SoC.

A smart partition automation tool allows the prototyper to create an abstract representation of the interconnect between FPGAs. At this level of abstraction, there are no fixed traces between FPGAs nor exact connections, but rather a representation of possible I/O interconnections. From this vantage point, the prototyper can very quickly see the expected FPGA utilization and secondly, and most important, the signal-to-multiplexing ratio. A prototyping rule of thumb is: the higher the mux ratio the lower the system performance, and it's this performance that is the gating factor of overall performance in a prototype. Quick identification of bottlenecks and where to apply more physical I/O between FPGAs enables the prototyper to not only design an ideal partition scheme, but also accomplish the task quickly.

EECatalog: Long bring-up is another challenge.

Scott, Synopsys: The challenge of how to accelerate prototype bring-up remains a focus for commercial vendors of FPGA-based prototypes. FPGA logic synthesis tailored for the prototyping task, partitioning and sophisticated signal sharing schemes to maximize performance will help shorten project schedules. But even with strong

product roadmaps and innovations in prototyping EDA software, the most successful design teams have embraced Design for Prototyping (DFP) best practices throughout the ASIC development process. DFP adoption may require a culture change as profound to development teams as was the industry shift from an emphasis on ASIC design productivity to ASIC verification productivity. In the next two years those ASIC design teams that anticipate design best practices for both ASIC and FPGA targets will benefit most from FPGA-based prototyping methods.

EECatalog: What's the latest on avoiding debugging complications?

Scott, Synopsys: The reason debug of an FPGA-based prototype is complicated is due to at least two conditions prototyping specialists face. One, design modifications to some extent are required to fit into the architectural constraints of a multi-FPGA system where ASIC signal interconnect, reset and clocking, memory blocks, ASIC I/Os, DFT circuits, etc. may require careful replacement and budgeting by prototyping specialists. Functional equivalence checks either through a simulation regimen or formal methods help to confirm that these changes have not changed the logic of the initial RTL drop. Innovations in prototyping automation tools help accelerate this process with schemes to model these changes.

The second reason is that once the prototyping system is operational the very nature of an FPGA-based prototype may expose flaws that simulation and emulation will not expose during the RTL and IP verification phases. Because the prototype is running at multi-megahertz it makes software-driven test and real-world interface testing feasible. These tests are going to uncover problems, or perhaps better stated as "incompatibilities" that require driver and/or RTL changes. In 2015 debugging features tailored for prototyping systems provide high-capacity storage options to allow for long periods of evaluation with schemes to adjust instrumentation that minimizes disruption to the prototype implementation.

Success in prototyping is largely measured by how soon an operational prototype can be deployed. The faster the debug phase can be accomplished, the lower it will be on the prototyping community's list of priorities.

EECatalog: What are the top 5 things designers ready to work on the architecture of PCIe 4.0 projects should know?

Scott, Synopsys:

1. Keep your eye on the ball! Or in this case, the specification. The PCIe 4.0 specification draft 0.3 is out, draft 0.5 is expected around the end of 2014, and while most of the expected changes are electrical, there are some protocol changes too. Make sure your FPGA vendor's SERDES is going to be able to meet PCIe 4.0 electrical requirements—many are extremely configurable, but the devil may be in the details as the spec finalizes.

2. Go big or go home! In order to keep to FPGA-friendly clock frequencies, datapaths will get very wide. That means things like 128-bit SERDES interfaces, internal datapaths, etc. Make sure your FPGAs have the capacity to handle the increased routing resources, which come with such large internal busses.

3. Feed the beast! Probably obvious, but if you're going to feed PCIe 4.0's 16GT/s data rate, you'll need more bandwidth in whatever your applications are doing.

4. Bring friends! Plan for early interoperability testing with other implementers. Even though FPGAs make logic changes "easy" it will be important to make sure your application will work with upcoming chipsets and other PCIe 4.0 devices until PCI-SIG compliance testing becomes available. Keep an eye out for early opportunities to participate in PCI-SIG "FYI" testing.

5. Never walk alone! Work with an IP vendor who is committed to closely tracking the PCIe 4.0 specification, is involved in the development of both the specification itself and the associated compliance tests and has PCIe 4.0 code available.

EECatalog: Name 5 factors that have to be involved in winning business in a case where the customer had previously designed FPGA boards in-house for an ultra-low latency application, e.g., time-sensitive financial trades.

Scott, Synopsys:

1. Shorter lead time for the prototype availability

2. Superior prototype flexibility across validation scenarios

3. Superior quality compared to low-volume system builds

4. Software tools tailored for system with a deep feature set for bring-up automation and debug

5. Support for advanced prototyping scenarios via workstation connectivity

EECatalog: As FPGAs get used more frequently as co-processors, accelerators, or offload engines, what are the design challenges of writing optimized code to take advantage of this/these capability (ies)?

Scott, Synopsys: FPGAs are becoming more powerful and power efficient over time. This is enabling more widespread use in new applications. In addition FPGAs bring to the table a very high level of processing power that can significantly speed up algorithms. In the past FPGA designers would have needed to work at optimizing their code for a specific device. However, today, synthesis tools are on the market that enable techniques developers can utilize to gain a high level of optimization for area and performance in co-processing/acceleration applications

EECatalog: Please comment on OpenCL and other high-level languages applied to FPGA designs.

Scott, Synopsys: Over the years there have been many tools developed to help make the transition from higher-level languages to the lower-level RTL, but there still remains a need for some FPGA understanding. OpenCL has been a standard in the industry for a while, and recently both Altera and Xilinx have made announcements around OpenCL. The goal is to abstract away the traditional FPGA development flows and lower the barrier to entry for the masses. It is a difficult question to answer so broadly, but FPGAs are already being adopted more widely and in part due to the higher-level languages and abstraction they provide.

Anne Fisher is managing editor of EECatalog.com. Her experience includes being managing editor, Communications Group, at OpenSystems Media, where she had the opportunity to cover a wide range of embedded solutions in the PICMG ecosystem as well as other technologies. Anne enjoys bringing embedded designers and developers solutions to technology challenges as described by their peers as well as insight and analysis from industry leaders. She can be reached at afisher@extensionmedia.com