Updated October 2015

What is an RTL tool doing next to ARM embedded software?

by *Don Dingee*
(published at the Embedded Software Store - www.embeddedsoftwarestore.com, July 2013)

In a statement that probably changed the direction of my career, one of my colleagues made the following observation a few years ago: "Embedded is not about designing chips; it is about designing *with* chips." That view was quite accurate for 30 years starting with the inception of the microprocessor, but things are changing rapidly.

RTL, once the exclusive domain of those EDA gurus, is making headway into embedded design for two big reasons. The first is the emergence of ARM architecture and a fabless semiconductor ecosystem, enabling both traditional vendors and in-house OEM design teams to design ARM cores into many more parts. The second is the proliferation of FPGA technology and corresponding growth in IP, a key element of more and more embedded designs.

The collision between the forces of change and the newer embedded technologies is evident in not one, but four parts, each carrying the "SoC" label but providing various combinations of ARM cores and programmable gate technology. Below is a quick overview of these parts:

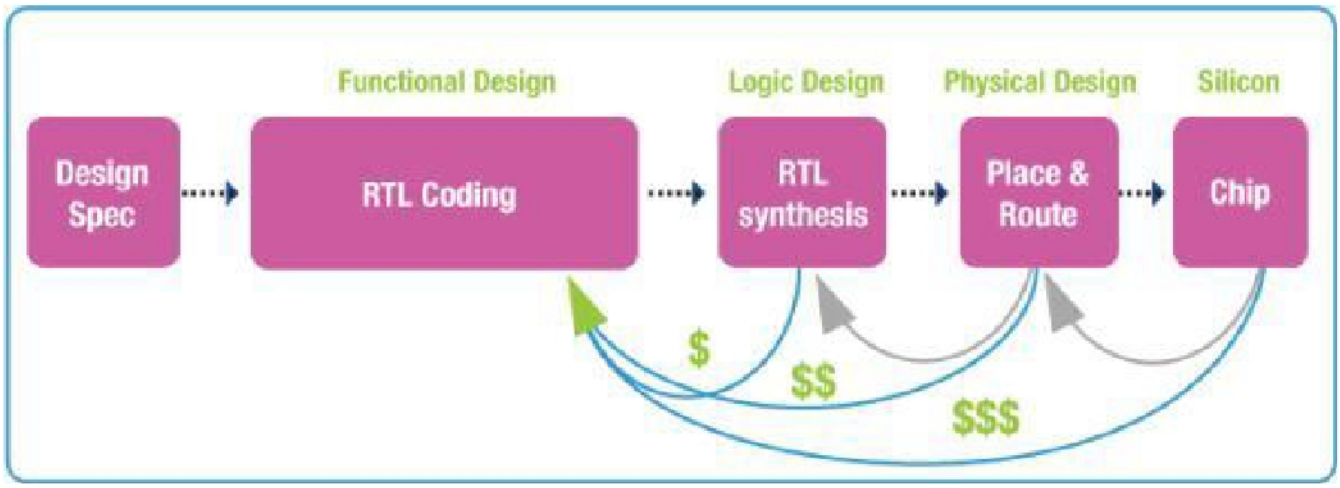
Device	*Processor core*	*Logic gates*	*Selected features*
Altera **Cyclone V**	Dual 800 MHz ARM Cortex-A9	110K LE 250 MHz clock	Hardened PCIe and memory controllers, 6.144 Gbps transceivers
Cypress Semi **PSoC 5LP**	Single 67 MHz ARM Cortex-M3	24 PLD blocks 62 MHz clock	20-bit D-S ADC, two 12-bit 1Ms ADCs CapSense touch control capability
Microsemi **SmartFusion 2**	Single 166 MHz ARM Cortex M3	150K LE 200 MHz clock	SEU immunity Physically unclonable function keys
Xilinx **Zynq-7000**	Dual 1.09 GHz ARM Cortex-A9	350K LE 250 MHz clock	2 GigE, 2 USB, 2 CAN RAS/AES/SHA-256b for secure boot 12.5 Gbpstranceivers

On the roadmap is even more advanced technology, strengthening the case for hybrid SoCs. Xilinx has just announced theirUltraScale architecture, setting the stage for multi-terabit throughput, improved clocking, and reduced system bottlenecks through parallelism - improvements that go beyond what can be achieved with process scaling alone. Altera has announced some specifics for their next-generationArria 10, a dual 1.5 GHz ARM Cortex-A9 tied to a 660k LE logic array running at 500 MHz, with 17.4 Gbps transceivers and DDR4 memory support at 1333 MHz.

These are not just FPGAs bolted on to an ARM processor; these are highly complex devices that can be designed to accelerate and process a wide range of interfaces, with signal processing capability well beyond what a general purpose processor alone could accomplish. Several of these vendors refuse to use the term FPGA describing these parts, even though the technology and tools are similar. Their reasoning is these designs can boot the ARM core complex without the programmable logic, ensuring ARM-compliant software runs.

In order to design with and optimize these SoCs, fully leveraging the programmable gates, one has to understand both RTL and the fabric architecture inside these parts. ARM software and FPGA design tools will start appearing side-by-side as we see more and more of these types of devices. One of the indispensable tools in FPGA space today is the design checker, such as Analyze RTL from Blue Pearl Software.

Similar to static analysis techniques for C code,RTL can be "linted" using rules and constraints, This can be very important when bringing in RTL from outside sources (including other design teams in your company), which may be functionally correct but can pose problems at FPGA integration. The key to success is stopping readily detected errors early and quickly.



Timing closure is a great example of a problem RTL linting can tackle. Functional logic can be mapped to gates, but doing so may create issues including false paths, multi-cycle paths, negative slack and other problems that would hinder FPGA performance or even prevent a place & route tool from completing a solution. Finding these problems manually is tedious, especially if the RTL is from a third-party and unfamiliar at a detailed level, and it gets worse as designs get larger with a wider variety of IP involved.

RTL linting tools use a pre-defined rule set, built from experience over thousands of designs, to quickly and automatically find problems. Instead of waiting for bad results from place & route that could take hours on a large FPGA design, RTL linting can spot issues in minutes. With the ability to extend and customize the rules, in-house or vendor best practices can be added to the checking. For example, Blue Pearl recently released a version of Analyze RTL with the Xilinx design methodology checklist for the Zynq-7000, automating checks beyond the native reporting in the Vivado® Design Suite.

Rules and constraints are just the beginning. Blue Pearl Analyze RTL combines straightforward linting with more advanced symbolic analysis techniques, digging through design properties to find more complex issues like bus contention, register conflicts, race conditions, and clock domain crossings - which can present serious problems in these SoCs with numerous domains.
There are also checks for scan-path integrity, a big aid in ensuring that JTAG provides the intended visibility.

The difference between designing chips and designing with chips for embedded applications is headed for zero. Expect to see more of these hybrid SoCs, and the software and tools to deal with them, in the near future.

Don Dingee is bringing ideas and information from across the social computing community to unite developers, vendors, and anyone interested in connected device technology and software development. His journey has taken him to places like the University of Southern California, General Dynamics, and Motorola, where he built a foundation in embedded engineering and product marketing. Don has taken to storytelling in recent years as an author, speaker, and consultant.