

Transforming Software to Silicon

C-based solutions are the future of EDA.

by Steve Kruse
Corporate Application Engineer
CebaTech, Inc.
kruse@cebatech.com

Sherry Hess
Vice President, Business Development
CebaTech, Inc.
hess@cebatech.com

Rumors of the possibility of translating the results of a high-level language such as C directly into RTL have abounded for the last few years. CebaTech, a C-based technology company, has turned rumors into reality with the introduction of an untimed C-to-RTL compiler, the central element in an advanced system-level design flow.

Operating in stealth mode for several years, the company developed its C-to-RTL compiler for use with IP development in TCP/IP networking. The nature of TCP/IP networking code, combined with CebaTech's desire to have a complete system-level design tool, dictated that the compiler handle both very large and very complex code bases. The CebaTech flow is a software-centric approach to FPGA and ASIC design and implementation, allowing hardware designers to describe system architectures and software engineers to implement and verify these architectures in a pure software environment with traditional software tools.

The challenge facing a team of system architects is to make the best design trade-off decisions so that the marketing requirements document (MRD) has been converted to the detailed specification for a balanced product. Only a balanced product is likely to find market acceptance, where careful consideration has been given to both innovative features and power-efficient usage of silicon. Until recently, a lack of suitable tools has hampered this all-important early decision-making.

The CebaTech flow allows architectural design at a high level of abstraction, namely untimed C. From this perspective, you can make design trade-offs in speed, area, and power of the end product up-front in C, where design issues are most easily addressed. The resulting “optimized” C-based design has a much higher likelihood of being competitive in the marketplace, thanks to the energy expended in honing it to an edge above and beyond the MRD.

A product with the correct cost/performance balance is what EDA strives to achieve. The eventual market price of a product is dictated by factors in the features and efficiency dimensions.

A Software-Oriented ESL Approach

Aside from striking the right balance between features and silicon efficiency, engineering teams must also grapple with verification of the chip design. Verification is by far the biggest component of product development costs because verification productivity has not grown at the same rate as chip capacity and performance. This issue has been acute for many years: the feasibility of integrating new features has not been matched by the designer’s/architect’s ability to guarantee correct implementation.

As represented by Figure 1, CebaTech aims to harness the convergence of three technological trends. First is the growing momentum of the open-source movement, wherein system-level specifications result from the collective intelligence of the global development community. Secondly, while these “golden” standards are freely available, successful implementation in silicon depends on domain-specific design

expertise. Finally, the cost of product development must be kept in check through increasing use of the latest system-level tools. Although the toolflow pioneered by

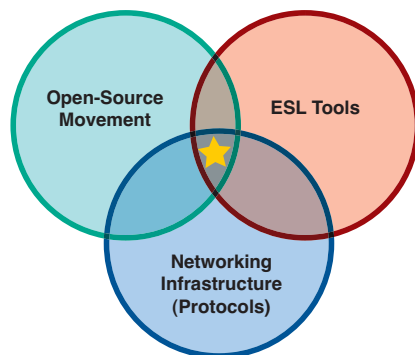


Figure 1 – The evolution and eventual convergence into CebaTech’s technology and methodology of C-based products that take “software to silicon” efficiently.

methodology has yielded ground to virtual system prototyping (VSP) platforms, assertion-based verification, and equivalence-checking methods. Nonetheless, the cost of design continues to grow.

CebaTech bypasses a bottom-up approach by using a direct top-down compilation from the original C source code base. The pervasiveness of C, especially for systems software, was a major consideration. C’s efficiency and intuitiveness for capturing hardware design benchmarks has achieved an overall improvement of at least 10x in manpower requirements as well as at least a 5x order-of-magnitude improvement in real-time verification.

The real benefit of compiling from the original C source code becomes apparent during the verification phase. CebaTech’s

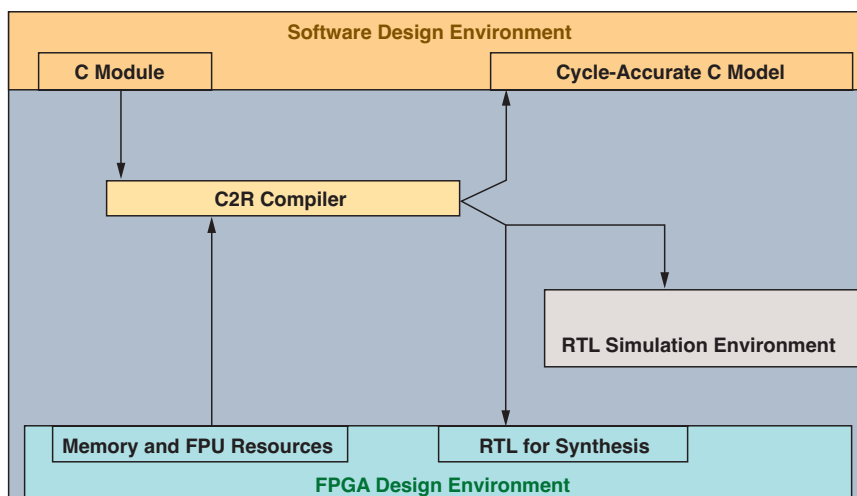


Figure 2 – The C2R compiler from CebaTech allows untimed ANSI C code bases to be rapidly implemented using an FPGA approach.

CebaTech has already proven its merit in IP networking, it is applicable to other emerging market sectors

CebaTech’s approach is a logical step in the evolution of the EDA industry, especially given the current bottleneck in verification. For the most part, complex system-on-a-chip (SoC) products are still developed in the traditional bottom-up way. Developers learned to use hardware description languages and write test benches, and eventually came to rely on FPGA-based prototyping systems and development environments for embedded cores. In recent years, traditional simulation

belief is that the entire SoC ought to be coded in C and run as software in a native C environment, where running tool-generated cycle-accurate C will precisely represent the behavior of the generated RTL running in an HDL simulator. The architecture of the compiled RTL is congruent to the C source and you can achieve extensive functional verification in the C environment.

As shown in Figure 2, the CebaTech compiler generates a cycle-accurate C model in addition to synthesizable RTL. The functional correctness of the compiler-generated RTL can be shown using formal tools that prove equivalency with the cycle-

accurate C, which in turn has been validated in the pure C software environment. The compiler is able to map arrays and other C data structures onto on-chip or external memory resources. It has the structure of a typical software compiler, complete with linker. Thus, it can quickly compile designs ranging from small test cases to truly enormous software systems, making it feasible for a small team of engineers to implement full hardware offload for a system as complex as the OpenBSD TCP/IP stack. There are enormous time-to-market and die-size advantages to this methodology.

A Real-World FPGA Example

The CebaTech flow is unbounded by design size. It can address designs from small (several hundred lines of C code) to immense (45,000 lines or more, as is the case with TCP/IP). To give you an overview of an implementation with the CebaTech compiler, let's look at an FPGA conversion of a C-based compression algorithm. Although quite modest in the amount of code, this example demonstrates the power and uniqueness of the CebaTech flow for architecture exploration/design trade-offs with respect to area and performance. It highlights the narrowing of the verification gap and provides an excellent vehicle with which to demonstrate CebaTech's software-to-silicon methodology.

To illustrate the extent to which the CebaTech flow utilizes pure C source language constructs to describe system architecture, we developed and compiled six incremental versions of the compression code. All versions function identically when compiled and run natively on a computer system; however, each successive version improves the performance and synthesis results of the generated RTL, using a small number of source code changes for each step. (Further refinement beyond the architectures explored in this article is also possible.)

Architecture Revision Details	Xilinx XC4V – Resource Utilization (%)					Fmax (MHz)		
	Slices	Slice Flops	LUTs	SelectRAM	# ROM	Clocks	Area	Speed
1 - Baseline Architecture					1	146		
2 - Eliminate Redundant Functions	89	24	68	18	1	56	12	58
3 - Define Parallel Processes	44	14	33	18	1	33	40	69
4 - Optimize Any Shared Functions	20	8	17	18	1	33	35	57
5 - ROM and FIFO Improvements	20	9	18	18	3	12	36	52
6 - Utilize Block SelectRAM	20	9	18	10	3	12	36	52

Table 1 – The results of exploring six versions of compression source code

We can summarize the six versions of the compression source code as:

- Original restructuring of the source code. The bulk of the program is put into a “process” that can be directly compiled into an RTL module. This initial restructuring does not try for any optimization, and is indicated as Revision 1 in Table 1.
- A first cut at optimization where certain critical functions are placed into separate modules. This prevents them from in-line expansion and yields a baseline architecture, shown as Revision 2 in Table 1. Regrouping the functions into two parallel processes is shown as Revision 3. The area-optimized implementation yields a tripling of clock frequency and the number of clock cycles decreases from 56 to 33. Revision 4 involves functions that are common to different callers.
- Flagging these functions as shared results in a quick area reduction. Various other common optimizations – notably loop unrolling – occur during Revision 5. A process-splitting operation allowed three simultaneous accesses to what was previously a single ROM. This step brought the clock cycles from 33 down to 12.
- The final iteration in this example, shown as Revision 6, is an example of a vendor-specific optimization. The Xilinx® Virtex™-II family makes both

distributed and block on-chip RAM available. Using the latter implementation allows storage utilization to drop by almost 50%.

For all of these C-based design instances, we synthesized the resulting Verilog RTL netlist using Xilinx ISE™ software v8.1 against a Virtex-4 XC4VLX25-12 FPGA.

For this C-based design flow, we expended less than one week to understand the compression code base, create the test benches used for RTL simulation, perform the six design explorations, and synthesize and verify the results. You can see that the CebaTech approach enables much speedier product development and delivery with the best price/performance balance.

Conclusion

CebaTech has been using a compiler-driven ESL flow to achieve rapid prototyping of leading-edge networking products for several years. The compiler was forged in the crucible of high-performance SoC development, but is now available to other forward-looking development teams who need to quickly develop and deliver novel SoCs that strike the right balance between price and performance.

To learn more about CebaTech's ESL flow, visit www.cebatech.com and register to download our compiler white paper. Alternatively, you can call (732) 440-1280 to learn how to apply our flow to your current SoC design and verification challenges. 🌟