

# Case Study: Blu Wireless Boosts SystemC Design and Verification Productivity Using High-Level Synthesis Technology

By Ray McConnell, Blu Wireless, and Mike Meredith, Cadence Design Systems

As systems on chip (SoCs) continue to grow in size and complexity, the SystemC design language has emerged as an effective means of design verification. High-level synthesis (HLS), in turn, is proving to be an ideal methodology to increase SystemC design and verification productivity. This paper presents a case study of how Blu Wireless Technology ramped up quickly with a working prototype of its WiGig millimeter wave baseband technology via a streamlined design and verification process based on SystemC and HLS.

## Contents

Introduction.....	1
The Search for a Better Verification Methodology .....	1
Facilitating Design Parallelism with HLS.....	3
Verifying Designs in Two Stages ...	3
Summary .....	4
For Further Information.....	4

## Introduction

One of the advantages of SystemC is being able to use the language to create an architectural model that will be directly synthesized into the hardware architecture. Rather than wait for the register-transfer level (RTL) model, with the SystemC architectural model, you can more thoroughly, more easily, and more quickly verify that the architecture meets your design intent in terms of functionality and quality of results (QoR). HLS technology facilitates debugging of a large volume of RTL code, which can otherwise be laborious and time consuming.

Blu Wireless is a Bristol, UK-based silicon intellectual property (IP) company that has developed scalable, low-cost 60GHz WiGig technology. Getting high-quality working silicon from its IP into the hands of its prospective customers is a cornerstone of Blu Wireless's business philosophy. Let's take a look at how SystemC and HLS technology made it possible for Blu Wireless to accomplish this objective.

## The Search for a Better Verification Methodology

In a conventional flow, work is performed in a sequential, multi-phase manner: the algorithm team completes its part, then the architecture team completes its part, and then the RTL team completes its part. After all of these steps have been performed, the layout team comes in. Blu Wireless's technical team members have all had experience in previous companies developing design architectures in C and then handing off the code to an RTL team for implementation. In their experiences, they often found that RTL teams would diverge from the architectural modeling process because, in order to meet the schedules of a small team, RTL coding had to begin before all of the architectural components of a given design were completed and validated in a working system. As a result, it was quite common to encounter unintended compromises on architectural definitions at the system level, such as undesirable latencies or flow-based control.

In their experiences, the Blu Wireless team found that, typically, the skill sets needed to understand customers’ problems and to manage the design implementation didn’t overlap: simply put, RTL experts didn’t understand the algorithms and system compromises, while architecture experts didn’t understand what was needed to move the design into a form where it could be efficiently implemented. The resulting RTL would then diverge from the optimal scenario or the design intent.

When Blu Wireless was being formed, the founding team tapped into their collective experiences to develop a new design and verification methodology. The team wanted architects to have more control of the implementation, so they could make system and algorithm decisions in conjunction with hardware implementation decisions. With a traditional RTL-centric flow, architects would determine a potential target hardware architecture. Then, the RTL design team would implement the architecture and then verify that the two designs are consistent. This makes for a very time-consuming process, so generally only early architecture revisions can be fully implemented. That does not necessarily lead to the highest quality designs. Since the Blu Wireless team is fairly small, they wanted a methodology that would facilitate an efficient design and verification process and support a higher level of abstraction.

The Blu Wireless team opted for a methodology based on SystemC and HLS, which together deliver a higher level of abstraction along with a verification model that accurately reflects the targeted hardware architecture. The architects were able to directly refine their understanding of the parallel SystemC implementations, checking consistency with the abstract mathematical modeling tools. The RTL implementation is directly generated from the SystemC code via the HLS technology, which means the RTL is always in line with the architectural design intent. The generated RTL is used by the back-end team for synthesis and layout to double-check that the design is operating within the specifications that were outlined as realistic for the design. Architects are, therefore, more closely associated with QoR, empowered with the knowledge they need to make better architectural decisions much earlier in the design process rather than later during implementation. The ability to do a first-pass verification with the mathematical models—pulling together all the subsystems for an RTL simulation of the entire SoC—provided an early indication of whether the whole system would come together. This approach enabled the engineers to quickly discover system-related errors and performance issues. Figure 1 depicts Blu Wireless’ development flow.

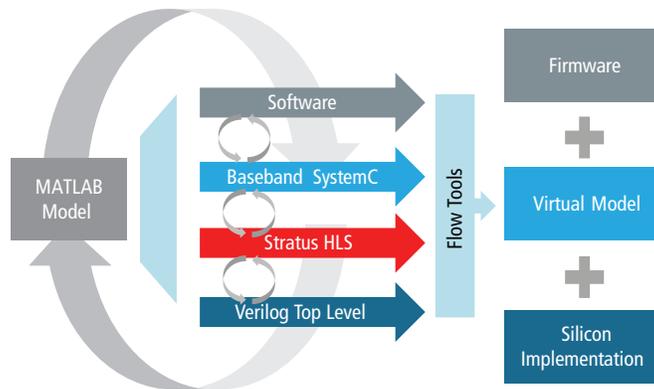


Fig 1: Blu Wireless Development Flow

The team also put into place an agile software development flow where software build tools would compile and run tests against the hardware model at daily intervals, letting engineers know very quickly whether they’ve introduced problems in the design that need to be resolved. Not only is the team operating much more quickly, it is also less costly for them to change the architecture of a given design. The Blu Wireless team calculates that its agile HLS- and SystemC-based methodology is saving at least two years of development time and provides superior results on a typical design.

## Facilitating Design Parallelism with HLS

The first Blu Wireless design is a WiGig millimeter wave baseband technology that operates at 8Gbps. A very high-speed datapath is connected to the analog/mixed-signal components of the design. To meet the performance requirements of this millimeter wave radio, much of the architecture has to compute in parallel. Using HLS made it a relatively fast and easy process for the team to partition data into multiple parallel flows in their design. Also, by applying the right expertise to the problem and working at the right level of abstraction, the team was able to properly define their parallel algorithms. This step called for the expertise of the design architect, generally more adept at C++ programming, to understand the parallel algorithms versus high-speed RTL expertise. HLS technology also facilitated some design experimentation—occasionally, the team would remap the algorithms with different parallelisms. Their HLS tool could implement each parallel design in three to four days, allowing them to experiment without a huge impact on the overall schedule.

Each parallel stream of their parallel system utilizes mature 32-bit processors in the control plane by connecting them to an extensive co-processor subsystem using custom instructions. The co-processor subsystem has a vector processor and a number of dedicated datapath-oriented computation blocks, such as fast Fourier transforms (FFTs) and error correction units tied together with proprietary mechanisms that move data among these blocks to keep it flowing efficiently. System programming is in C using the conventional compiler tools and debuggers, while the data flow processing is performed in the co-processor. Using some dedicated hardware to handle computations is a good way to increase performance while optimizing power.

To create its design under its agile design flow, Blu Wireless utilized HLS technology to generate the dedicated computation blocks, vector processor, and custom interconnect. The team had to be able to efficiently model and implement both highly control-dominated modules and highly datapath-dominated modules in SystemC using the same tool flow. To make this possible, they used the Cadence® Stratus™ High-Level Synthesis (HLS) platform. The Stratus HLS tool, as depicted in the diagram in Figure 2, provides an integrated design and verification flow with broad reuse of behavior IP cores.

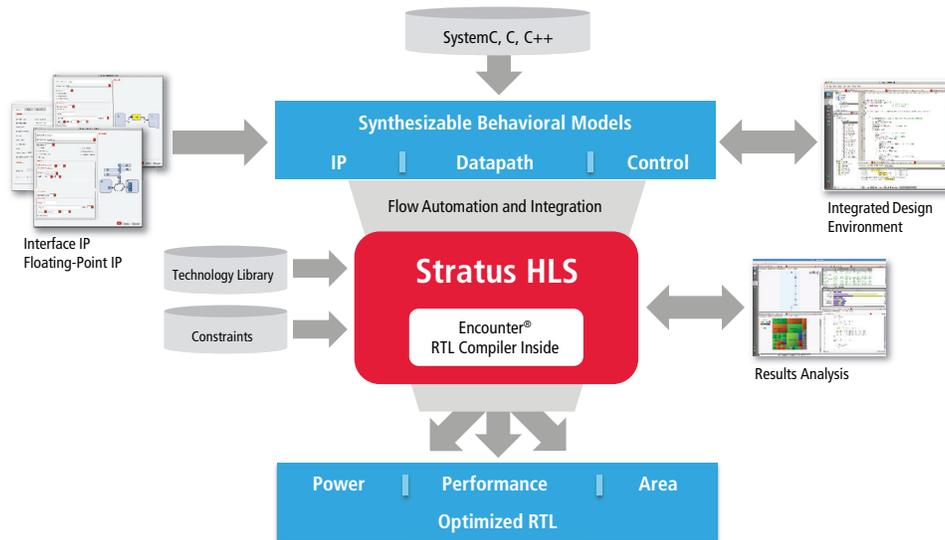


Figure 2: Stratus HLS Synthesis Flow

## Verifying Designs in Two Stages

Blu Wireless used two stages of design verification. First, there is the architectural validation of the system architecture with C++ and the MATLAB high-level modeling language. SystemC views are implemented at the transaction level for use with the Universal Verification Methodology (UVM) and with pin-level interfaces used with HLS. At this stage, corner-case verification is less important because it is more critical to ensure that the system architecture itself is sound. With their approach, Blu Wireless is generating a good first pass of how a whole architectural system fits together and also testing the system-level algorithms.

Second, there's a more detailed validation at the subsystem level with the UVM models, where a more exhaustive functional verification is completed. This second stage also includes a system-level validation with simulation of the full SoC with CPU models, software, and the hardware implementation.

Blu Wireless used Verilog to stitch together the top-level netlist of their SoC with SystemC at the lower level, so that system-level simulations could be performed at the RTL level. With this approach, the team could use pre-built pieces of Verilog IP for the buses and other system-level components of the SoC. They needed to be able to perform RTL system-level simulations so they could quickly turn algorithm and architecture decisions into fully functional RTL. The Stratus HLS platform facilitated this entire process.

## Summary

As a result of their SystemC/HLS design flow, the Blu Wireless team was able to have a complete multi-gigabit baseband technology available at the time of integration and with a complete system/software validation. Without this design flow, the team would not have been able to deliver a working prototype chip in this timeframe with their small team. For Blu Wireless, having an early working prototype provides a competitive advantage.

Also advantageous is having the ability to deliver on a diverse product roadmap. By using HLS technology, Blu Wireless can flex their technology to change some of the system- and block-level components to adapt their design for specific applications. Their millimeter wave baseband technology currently operates at 8Gbps, but they can scale it to 100Gbps by replicating and flexing the individual blocks for more parallelism as needed. They can also tune individual blocks and the system to add new features. Because the underlying blocks in their design are implemented at a higher level of abstraction, they're able to easily make changes. Single team members are working on blocks in months—something that, without the HLS flow, would take a team a year or more of development time.

In summary, by supporting a higher level of abstraction, HLS technology brings greater efficiency to a SystemC design and verification flow. The resulting flow can lead to fewer iterations and faster prototyping. For Blu Wireless, using SystemC with Cadence's Stratus HLS platform provided an effective, efficient way for architects to quickly move through the design process while validating design intent at checkpoints along the way.

## For Further Information

Learn more about the Cadence Stratus HLS platform at [www.cadence.com/products/sd/stratus/pages/default.aspx](http://www.cadence.com/products/sd/stratus/pages/default.aspx).

Learn more about Blu Wireless at [www.bluwirelesstechnology.com](http://www.bluwirelesstechnology.com).



Cadence Design Systems enables global electronic design innovation and plays an essential role in the creation of today's electronics. Customers use Cadence software, hardware, IP, and expertise to design and verify today's mobile, cloud, and connectivity applications. [www.cadence.com](http://www.cadence.com)