CALYPTO®

Catapult C Synthesis Used by Hitachi, Ltd., Telecommunication & Network Systems Division to Tackle Complex Forward Error Correction (FEC) Design

The Hitachi, Ltd., Telecommunication & Network Systems Division planned to develop a Forward Error Correction (FEC) system to meet the needs of next generation communication systems. FEC devices correct errors due to imperfect transmission and as such, are very important for the reliability of data transfer over longer distances. Error detection and correction typically works by mixing and detecting the noise under transmission, a process which involves many complicated calculations and where algorithm efficiency directly translates to hardware complexity. In this context, Hitachi's project was considered to be a very challenging one. Moreover, the design team also faced considerable pressure because they had a short design cycle in which to get the work done.

Hitachi, Ltd., Telecommunication & Network Systems Division specializes in the leading edge telecommunication business area like next generation mobile and wide broad band. They provide base band stations for mobile phone, FTTH for high speed wide broadband network and IP network segments.

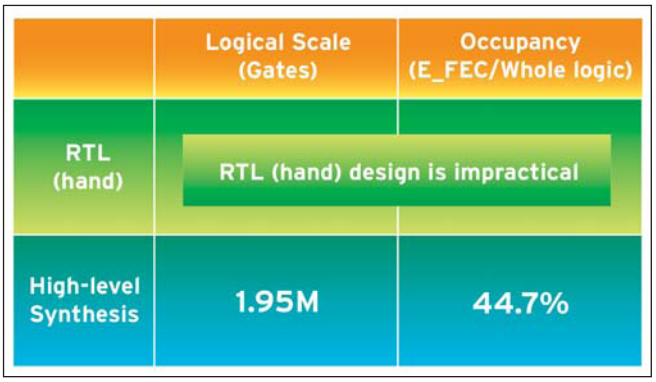
Given the circumstances, Hitachi looked at three options for designing and verifying their chips. The first option was to continue using a traditional hand-coded RTL flow. Sticking with this familiar option, the team was confident they could meet their timing constraints, but the concern was that this intense design effort would not leave enough time in the design cycle for verification.

The second option required them to purchase IP. This option would save design cycle time, but required the team to develop the interface circuit between the IP.

The third option was to adopt the use of a high-level synthesis flow. On the down side, trying a new methodology carried with it some risk, but on the positive side, if it was successful, they could expect to reduce design cycle time compared with traditional hand-coded RTL and allow for rapid verification.

"During the evaluation, we had a tricky corner-case bug that required 20k patterns of simulation data to trigger. While running those patterns took only an hour in the C simulation, and it would have taken almost 960 hours (or 40 days) in RTL simulation."

Hitachi, Ltd., Telecommunication and Network Systems Division



C synthesis was the only practical way to generate Hitachi's Enhanced FEC design.

Given that most of the algorithmic engineers were accustomed to using ANSI C++ to describe the algorithmic portions of the design, they decided to research and evaluate an ANSI C++ based high-level synthesis tool. Catapult® C Synthesis was chosen and the team performed a full trial, including running a traditional RTL flow in parallel during the project using the Catapult C Synthesis based flow.

Even after the decision was made to adopt a Catapult C Synthesis based flow, there was a fair amount of anxiety associated with trying a new methodology and tool. This was coupled with a high degree of uncertainly about the impact on the traditional design flow—especially design cycle time.

From an operational standpoint there were questions about ease of use, the limits of C-synthesis and the role of feedback and analysis in the new methodology. There were also concerns about the quality of the RTL results in terms of correct by construction logic. Could they get good result and meet the specification on clock frequency and area?

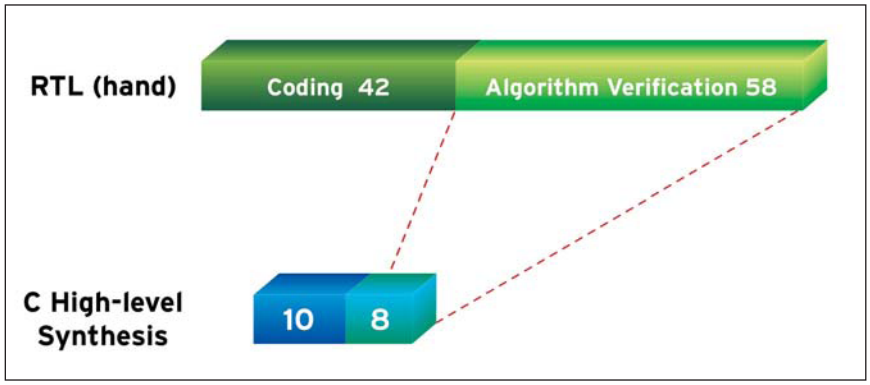
Catapult C Synthesis: Striving for Optimal Designs in Less Time

What Hitachi discovered during the evaluation relieved their fears. Once they were up to speed on Catapult C Synthesis, Hitachi was able to achieve an impressive 5x improvement on the overall project, starting from the original un-timed C++ source and ending with verified RTL. A detailed schedule comparison provided valuable insight on where and how Catapult paid-off. During the manual implementation of the design with their traditional approach, Hitachi measured that 42% of the development cycle was spent in coding and 58% in verification. In contrast, with the Catapult flow, the coding phase represented only 10% of the original schedule, translating to a sensible 4.2x improvement. Furthermore, verification accounted for only 8% of the reference schedule, or an even more impressive 7.25x gain.

Another significant finding was that Catapult C Synthesis could reduce area by one-third compared with hand-coded RTL.

Some choices were made early in the project to help the team learn a C-synthesis approach. Hitachi decided on a 2-step coding approach that produced the most efficient results. They began by creating a primary C code, which was used for rapid verification. Optimized for simulation speed, this primary C code was not suitable for high level synthesis. So after running rapid simulation with the primary C code, they refined it to a synthesis-efficient secondary C code. They could finish modifying the C code within a few days.

In addition, the Catapult C Synthesis flow allowed for early detection of bugs in the C code. In particular, C-code debug found bugs that were difficult, if not impossible to find in RTL. During the evaluation, a tricky corner-case bug required 20k patterns of simulation data to trigger. While running those



Difference in time required between RTL and high-level synthesis approach.

patterns it took only an hour in the C simulation, and it would have taken almost 960 hours (or 40 days!) in an RTL simulation, making the bug very unlikely to detect at this level.

Equally as important, Catapult C Synthesis gave the team the ability to tradeoff area and performance requirements via micro-architecture exploration, ultimately converging on an optimal implementation of the design. Catapult C Synthesis allows the design team to look at a variety of architectures from the C code. For example, it was quite simple to change the architecture from serial C and parallel RTL to parallel C and serial RTL. Also, the automatic construction of ROM/RAM control logic was straightforward and allowed for fast analysis of the most efficient approach. The ability to construct and revise architectures allowed the design team to evaluate different architectural options and choose the best one for the application.

Summary: 4.5M gates, from C to RTL

When the Hitachi team completed the FEC project, they were confident in Catapult C Synthesis and in their choice to adopt a high-level synthesis tool. Since this first experiment, the team has leveraged the benefits of the Catapult C Synthesis methodology on eight consecutive tape-outs (both ASIC and FPGA), the latest one being a 7.9M gate ASIC, 57% of which were synthesized from C with Catapult C.

"Initially, we created designs in both the traditional and the Catapult C Synthesis based flows to alleviate the fear involved in changing methodologies. After we proved the results with the Catapult C Synthesis flow were acceptable, especially from an ease of use standpoint for example Catapult can generate various RTL code by single C code, we began moving to a high-level synthesis strategy for complicated algorithmic designs."

Hitachi, Ltd.,
Telecommunication and
Network Systems Division