Introduction

The Catapult High-Level Synthesis (HLS) Platform empowers designers to use industry-standard C++ and/or SystemC to describe functional intent, enabling them to move up to a more productive abstraction level for both design and verification of ASICs and FPGAs. For designs and IP where time-to-market is critical, power/performance information is needed early and specifications are frequently changing (Automotive Vision, Image Processing, Deep Learning, Video CODEC, 5G/IoT Communications, etc.), Catapult HLS provides the only effective way to meet these pressures without compromising quality and functionality.

To achieve the maximum productivity gain from a C++/System HLS methodology, it is necessary to have the performance and capacity to handle today’s large designs coupled with a comprehensive flow through verification and implementation. Catapult has been proven in production with 1,000s of designs and the resulting RTL adheres to the strictest corporate design guidelines and ECO flows. In addition to Catapult Synthesis, only Catapult has integrated High-Level Verification (HLV) tools and methodologies that enable designers to complete their verification signoff at the C++ level with fast closure for RTL.

FEATURES AND BENEFITS:

- Easier to design functionality in standard C++/SystemC
  - Create C++/SystemC executable specification
  - Write 80% less code for easier development and debug
- Complete platform for power, performance, area optimization
  - 10X productivity over hand-coded RTL with equivalent QoR
  - Explore microarchitecture alternatives
  - Deep-Sequential analysis with PowerPro “under-the-hood”
- Reduces verification cost by 80%
  - Simulate functionality 100-500x faster
  - Catapult DesignCheck to find bugs fast before synthesis
  - Catapult Coverage for HLS-aware coverage metrics
  - Automatic generation of RTL-to-C verification environment
  - Fast path to automated RTL coverage closure
- Easy to use HLS debug and visualization
  - Visualize the HLS transformation process
  - Analysis tool based on the Visualizer Debug Environment
- Production proven flows with thousands of designs
  - Full support for datapath and control logic synthesis
  - Capacity for multi-million gate ASIC and FPGA designs
  - Top-down and bottom-up design management
  - Complete ECO flow
Catapult Enables Faster and Easier Design in C++/SystemC

Using Catapult HLS simplifies the traditional design flow by automating the RTL generation based on a higher level functional description and architectural constraints. Using C++/SystemC, compared to RTL, reduces the number of lines of code up to 80%, making HLS code significantly easier to write and debug. Incorporating last minute specification changes and even retargeting to a different technology is possible because of the separation of the design functionality and the implementation details. The RTL can simply be regenerated based on the modified HLS model and new constraints.

Catapult Supports Multiple Abstractions for Faster Simulation and Modeling

With Catapult, the designer can choose from an RTL-like coding style in SystemC to a fully untimed high-performance model in C++/SystemC, or a combination of both. During the synthesis process, Catapult transforms the designer’s algorithmic/behavioral description and applies micro-architectural constraints through user-specified directives. Catapult’s patented interface synthesis technology allows timing, protocol, and bandwidth to be defined, adding the necessary RTL hardware during the synthesis process. Catapult allows the designer to specify parallelism, design throughput, and memory vs. register implementation using tool options instead of hardcoding them in the RTL. This results in an easy to reuse design and an optimized implementation.

Catapult is a Complete Platform for Power, Performance, Area Optimization

Catapult automatically applies general performance and power optimizations during design transformation. In addition, Catapult is the industry’s first HLS tool that targets power as an optimization goal and uses PowerPro® “under-the-hood” for power analysis and to apply detailed optimizations. This provides a fast and accurate flow for tradeoffs of power, performance, and area. The generated RTL is able to match or surpass quality of results (QoR) of hand-coded RTL.

Catapult Provides Easier Debug and Optimization Control

Catapult provides built-in graphical analysis tools, such as a Gantt Chart Viewer, Resource Viewer, and a Schematic Viewer, to enable full visibility of the HLS results and to allow the designer to exert control over design decisions. The integrated cross-probing support between different analysis views, including the C++/SystemC source code, enables the designer to rapidly focus on the problematic areas, to add or change directives, and to converge interactively on the optimal solution. The Visualizer-based analysis enables easier and deeper debugging where designers are able to utilize advanced source code navigation, control flow analysis tools, and focused debug capabilities such as failed schedule analysis.
HLS Verification (HLV)

The benefits for verification in an HLS design flow are numerous. HLS synthesizable C++ code is one fifth the number of lines of code compared to RTL, making it easier to write and debug. The simulation speed is typically between 100-500X faster than RTL, allowing much more verification and consuming far less compute resources. The HLS design flow also enables the verification team to be involved very early in the design process before any RTL is ready. All of this translates to large productivity gains and significant reductions in verification time and cost.

Catapult C++/SystemC Design Checking to Catch Bugs with no Testbench

C++ Simulation often misses critical bugs such as uninitialized memory reads, out-of-bound array-accesses, and overflow/underflow problems, that can lead to hard-to-debug failures during both C++ and RTL verification. Furthermore, the C++ and SystemC languages have gray areas that may give unexpected simulation or synthesis results depending on the platform they execute. The advantage of Catapult HLS Design Checking is that bugs can be found automatically without a testbench, saving valuable debug time later in the design cycle.

Catapult C++/SystemC Coverage to Provide Quality Coverage Metrics

Similar to RTL, HLS designers need a way to capture quality metrics while testing their design. Catapult Coverage is HLS-aware coverage for C++/SystemC that understands concepts such as function inlining and loop unrolling to provide a complete coverage (line, branch and expression) picture for the design. Catapult Coverage writes its coverage data to the Mentor UCDB (Unified Coverage Database) to provide the designer with a complete set of post-processing tools for merging, ranking, reporting, and connecting to a test plan that is unified with RTL. Using Catapult Coverage on the HLS source achieves the equivalent coverage metrics automatically for the resulting RTL and closes RTL coverage.

Catapult Automatic C++ to RTL Co-Verification

Catapult provides an automated verification flow (SCVerify) that verifies the original C++/SystemC design against the synthesized RTL. This flow auto-generates a SystemC test infrastructure that reuses the original C++/SystemC testbench to verify the RTL, providing designers with a push-button unit test solution to quickly sanity-check the RTL so it can then be handed off to the downstream RTL integration and verification teams. Catapult also provides an automated verification flow that automatically generates a complete UVM environment including random sequences.
Catapult is Proven in 1,000’s of Projects to Reduce Complete Project Time by 50%

Catapult was released in 2004 as a C++ based ASIC synthesis tool for datapath-dominated wireless communication hardware. Since 2004, Catapult has evolved into a C++/SystemC synthesis tool with support for virtually any FPGA or ASIC digital hardware design type, and it has been proven in thousands of projects to cut overall design time in half.

Catapult-generated hardware can be found in hundreds of millions of cell phones, tablets, cars, computers, printers, cameras, gaming consoles, and satellites. Due to this extensive customer experience, Catapult has been optimized to work with existing RTL linting, coverage closure, synthesis, and ECO flows. An example is illustrated in a whitepaper with Qualcomm that demonstrates how Catapult and HLS works within very strict corporate RTL verification, synthesis, and ECO flows.

Catapult is Proven for Large Designs

Catapult designs are often very large and complex systems and subsystems. For example, Google recently used Catapult to synthesize their VP9 video encoder and decoder, a complex, 8 million gate design with over 150 leaf blocks with an balanced mix of both datapath and control logic. To accommodate this level of complex hardware, Catapult has both top-down and bottom-up hierarchical design management capabilities. Designers can focus on the blocks that they are working on while locking down other regions of the design, allowing an efficient design flow.

Proven to Easily Adapt to Last Minute Design Changes

One of the biggest advantages of using HLS is the ability to reuse and modify functionality that would not be possible with an RTL implementation. Because the specifics of the timing, registers, and datapaths are not contained in the source, but specified during the HLS synthesis process, big changes in specification can be both implemented and verified while still staying on schedule. In a recent whitepaper from NVIDIA, they describe two changes that they were able to implement within months of RTL freeze that would have been impossible using a traditional RTL flow.

System Requirements and Compatibility

Languages: VHDL 87, 93 & 97 and Verilog 95 & 2001, SystemVerilog, C++
Platforms: Windows 7, Linux Red Hat Enterprise 5 and 6
Memory: 2 GB minimum

For the latest product information, call us or visit: www.mentor.com/hls-lp

©2017 Mentor Graphics Corporation, all rights reserved. This document contains information that is proprietary to Mentor Graphics Corporation and may be duplicated in whole or in part by the original recipient for internal business purposes only, provided that this entire notice appears in all copies. In accepting this document, the recipient agrees to make every reasonable effort to prevent unauthorized use of this information. All trademarks mentioned in this document are the trademarks of their respective owners.