C-Based Design

The design complexity of next generation communication and video applications has outpaced traditional RTL methods that involve time consuming manual steps such as micro-architecture definition, handwritten RTL, and area/speed optimization through RTL synthesis. The Catapult™ C Synthesis tool moves hardware designers to a more productive abstraction level, enabling the efficient design of complex ASIC/FPGA hardware needed in next-generation, compute-intensive applications.

Algorithmic C Synthesis

Catapult C Synthesis is the only algorithmic synthesis tool to automatically generate optimized RTL hardware descriptions from untimed C++ up to 20x faster than traditional RTL methods. Effectively uniting the disparate domains of system designers and hardware designers, this high-level synthesis tool accelerates the creation of complex DSP functionality for next-generation applications in wireless, satellite and video, and image processing. By leveraging the same untimed C++ source typically developed by system designers, hardware designers can now automatically create a precise, repeatable path from C++ models to hardware much faster than with conventional manual methods. The result is an error-free flow that produces accurate RTL descriptions tuned to the RTL synthesis tool and vendor technology. Also, by enabling in-depth explo-
ration of multiple micro-architectures and interface scenarios, Catapult C Synthesis allows designers to produce hardware of the same and often superior quality than traditional hand-coded RTL methods.

Micro-Architecture What If Analysis

Traditional methods leave minimal, if any, time in the design schedule for performing what if evaluations of alternative micro-architectures. As a result, hardware designers are forced to compromise by limiting their selection to a few architectures in advance, which invariably results in non-optimal hardware.

By raising the design abstraction level, Catapult C Synthesis allows hardware designers to use the system designer’s C++ source code to automatically generate RTL code. This provides a single source for algorithm validation and hardware creation, as well as the ability to rapidly explore alternative micro-architectures for a given design. Catapult C Synthesis gives designers superior control, generating solutions based on user constraints and graphically displaying the results in a choice of X-Y plots, bar charts, tabular and schematic views. Users can thus make quick, cognizant decisions in terms of area and performance to deliver optimized hardware results.

Interface Synthesis

Unlike previous methods, Catapult C Synthesis does not require interface protocols to be embedded in the C++ source. Rather, it accepts a pure, untimed C++ description as its input, and uses patent-pending interface synthesis technology to control the timing and communications protocol on the design interface. This enables what-if interface analysis so designers can explore a full range of hardware interfaces such as streaming, single- or dual-port RAM, handshaking, FIFO, AMBA and many other built-in I/O components. Users can also target custom or proprietary I/O components using the Catapult C Library Builder™ tool.

Comprehensive Design Constraints

Catapult C Synthesis provides complete control over how a design is synthesized to hardware. From the same source, hardware designers use high-level constraints to create compact to highly parallel implementations. The architectural constraints window presents a graphical view of all ports, arrays, and loops in the design, and allows any or all of the following high-level constraints to be applied:

- Loop unrolling and pipelining
- Loop merging
- RAM, ROM, or FIFO array mapping
- Resource allocation and sharing
- Memory resources merging
- Memory bit-width re-sizing

Catapult C Synthesis combines automation with specific high-level constraints so the user can precisely control the hardware implementation to interactively converge on significantly better quality designs in far less time.

Algorithm and Architecture Analysis

The hierarchical Gantt chart is an algorithm and microarchitecture analysis tool. It essentially acts as a schematic viewer for the algorithm and provides the designer with information regarding data flow, component utilization and loop execution profiles. Issues such as memory bandwidth limitations, loop...
dependencies preventing parallelism, and data dependencies preventing optimal scheduling are quickly identified. Using this information, the designer can quickly identify hard-coded performance bottlenecks or inefficiencies in the design, cross-probing areas of interest back to the C++ source. By optimizing the algorithm and hardware implementation in tandem, designers can quickly converge on the optimal hardware implementation.

Symbolic Analysis and Optimization

Catapult C Synthesis is driven by a mature, second-generation high-level synthesis technology developed by Mentor Graphics. The customer-proven technology employs advanced symbolic analysis and optimization techniques including sequential constant propagation, variable lifetime, loop boundary and array index analysis, as well as bit width and memory bandwidth optimization. These built-in features allow Catapult C Synthesis to automatically reduce operator bit widths and efficiently share components and resources to build smaller, more efficient hardware.

Integrated RTL Synthesis and Simulation

Catapult C Synthesis produces both cycle-accurate and RTL netlists in either VHDL, Verilog, or SystemC along with simulation and synthesis scripts for Design Compiler®, Precision® Synthesis, and ModelSim®. The tool also takes advantage of RTL synthesis features like automatic RAM inferencing. For example, when designers map arrays in the C++ source to block RAMs, Catapult C Synthesis generates the correct RTL structure to leverage RAM inferencing in the RTL synthesis tool. It also takes advantage of RTL synthesis features like automatic RAM interfacing. For example, when designers map arrays in .

Predictable Timing Closure

Catapult C maximizes design performance by structuring essential FSM control logic off the timing-critical data path. Catapult C constructs highly optimized data paths by leveraging the native technology-specific operators used by the downstream RTL synthesis tools such as DesignWare for Design Compiler. This methodology ensures precise knowledge of data path delays leading to correct-by-construction timing through RTL and physical synthesis.

Integrated SystemC Environment

Catapult C Synthesis provides integrated verification for hardware designers working in a SystemC environment. The tool supports SystemC transaction level systems by generating transactors that synchronize timed RTL with a transaction or sequential test environment. This connection allows a single SystemC-based testing environment for the entire design flow. Catapult C Synthesis also has testbench generation which automatically compares the C++ input to the RTL output, providing debug information for a specific synchronization point in the case of a simulation mismatch. Users can then use ModelSim to analyze the mixed VHDL, Verilog and SystemC to identify the mismatch.

Catapult C Synthesis supports SystemC-only simulation by generating a behavioral SystemC model for cycle- and pin-accurate simulation in a SystemC only environment. Like the behavioral VHDL and Verilog generated by Catapult C Synthesis, the
Catapult C supports SystemC transaction level systems by generating transactors that synchronize timed RTL with a transaction or sequential test environment. This connection allows a single SystemC-based testing environment for the entire design flow.

interface of the SystemC model simulates the same as the RTL generated by the tool, but the core of the design is optimized to simulate 20 to 100 times faster than RTL.

**Intuitive Project and Solution Management**

Catapult C Synthesis automatically creates a new solution each time users modify the C++ source or apply a new constraint, preserving previous solutions for subsequent analysis and optimization. The tool also generates X-Y plots and bar charts, as well as tables to allow detailed comparison between different solutions. Users can choose any of the solutions to automatically generate optimized RTL.

**Catapult C Library Builder**

Catapult C Library Builder collects detailed characterization data from the downstream RTL synthesis tools with specific target technology libraries. This allows Catapults C Synthesis to precisely schedule hardware resources, chain operators, infer multi-cycle components, and quickly provide accurate area, latency, and throughput estimates without spending costly time and effort going through RTL synthesis. Catapult C Library Builder also allows designers to leverage custom components including memories, IP, DesignWare, and existing RTL.

**Customer-Proven Algorithmic C Synthesis**

Catapult C Synthesis has already been instrumental in many successful tapeouts from major hardware design companies worldwide. The mature, second-generation algorithmic C synthesis environment automatically generates error-free RTL from untimed C++ up to 20x faster than traditional manual methods. Using pure, untimed C++ to describe functional intent, designers move up to a more productive abstraction level for designing complex ASIC or FPGA hardware typically found in next-generation, compute-intensive applications.

The tool’s advanced what if analysis allows hardware designers to fully and interactively explore the microarchitecture and interface design space, yielding high-performance hardware that rivals hand-coded design quality. Catapult C Synthesis unites two distinct domains — system-level design and hardware design — and when combined with Mentor Graphics ModelSim simulation tools, lays the foundation for next-generation electronic system level (ESL) design.

**Platforms Supported**