

CADENCE C-TO-SILICON COMPILER HIGH-LEVEL SYNTHESIS

Cadence® C-to-Silicon Compiler automatically generates synthesizable RTL for both datapath and control functionality from timed and untimed C/C++/SystemC® algorithm descriptions. Achieving quality of results at or above the 90th percentile of manual RTL design while slashing engineering effort by up to 90%, C-to-Silicon Compiler bridges the gap between design complexity and efficiency of RTL code generation.

C-TO-SILICON COMPILER

C-to-Silicon Compiler increases new design productivity and significantly eases legacy design reuse by starting design at a high level of abstraction. Its tight integration with the downstream Cadence Encounter® implementation and Cadence Incisive® verification flows ensures that the final SoC design verifiably meets the product specification. C-to-Silicon Compiler also enables early software development and faster hardware/software co-verification by generating fast hardware models (FHM) early in the design cycle (see Figure 1).

From the original SystemC input code, C-to-Silicon Compiler automatically generates control/dataflow graphs, timing-analysis reports from its embedded RTL compiler logic synthesis tool, and other key design information—all interlinked via a unique, behavior-structure-timing database that records each design state, tool state, and optimization decision during the synthesis

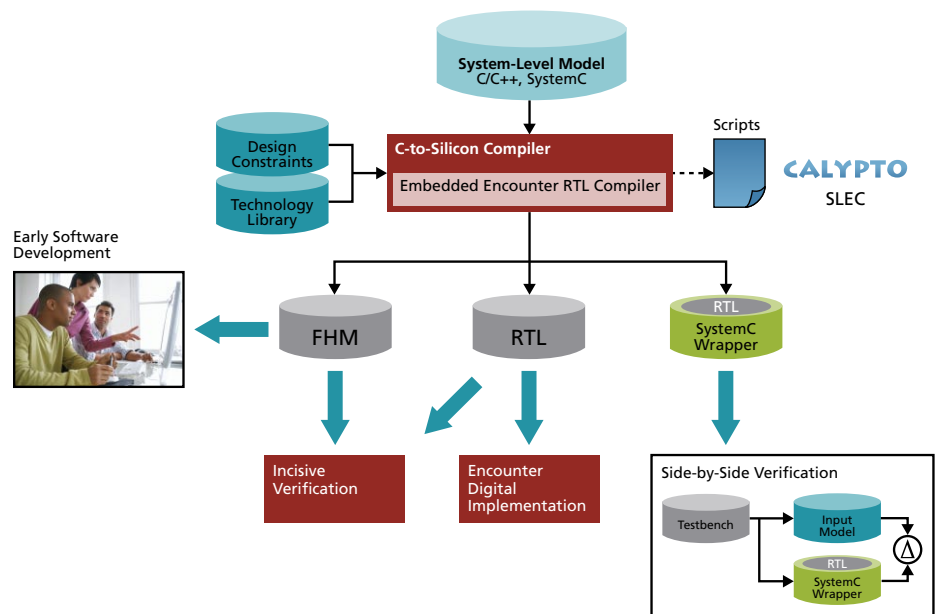


Figure 1: C-to-Silicon Compiler high-level synthesis flow

process. This patent-pending database technology allows designers to track each design transformation from the original SystemC input through to RTL and enables incremental synthesis, all within a single, integrated environment.

Integrated with the Encounter digital IC implementation flow, C-to-Silicon Compiler uses full-context, gate-level timing feedback from embedded logic synthesis to rapidly achieve the desired results. Embedded logic synthesis also ensures that the generated RTL will synthesize exactly as predicted.

C-to-Silicon Compiler is the first high-level synthesis technology to deliver four critical capabilities in one package:

- **Embedded logic synthesis (ELS)** enables parallel optimization of control and datapath logic, delivering better-than-average human quality of results (QoR)
- **Behavior-structure-timing (BST)** database enables true incremental synthesis and much faster design and verification turnaround time
- **Constraint-functionality separation (CFS)** enables reuse across multiple applications and process technologies
- **Auto-generated fast hardware models (FHM)** accelerate verification and enable hardware/software co-development

Hardware architects and RTL designers can use these capabilities to:

Develop RTL faster and with significantly less effort

Conventional high-level synthesis tools lack automated downstream feedback, resulting in multiple, time-consuming manual iterations that limit exploration.

C-to-Silicon Compiler tracks each design change and its area/performance impact at every design level, enabling comprehensive design space exploration. Designers can quickly and easily iterate multiple, different RTL micro-architecture options to identify the optimum micro-architecture with the desired area and timing QoR.

Execute engineering change orders with minimum effort

Other high-level synthesis tools force designers to re-synthesize the entire design whenever any change is made. C-to-Silicon Compiler incremental synthesis enables designers to implement changes and engineering change orders (ECOs) incrementally, without disturbing the rest of the design, and avoid repeating their entire design/verification flow on the whole design.

Optimize results for the broadest possible set of designs

Other high-level synthesis tools typically analyze control logic and datapath logic separately, and fail to produce sufficiently accurate timing estimates, forcing engineers to design the two types of logic separately and integrate them manually. C-to-Silicon Compiler analyzes and synthesizes both control logic and datapath logic together, and it performs the tradeoffs necessary to achieve the optimum area and performance.

Maximize design reusability

Conventional high-level synthesis tools require implementation-specific pragmas, synthesis directives, and extensions to be embedded in the input source code, making it difficult to reuse for different implementations. The patent-pending constraint functionality separation

technology in C-to-Silicon Compiler strictly separates the functional description from design constraints. Separate synthesis directive files guide C-to-Silicon Compiler toward different implementations, ensuring that the original function model remains “golden.”

Perform faster, more reliable verification

The patent-pending FHM generation technology in C-to-Silicon Compiler creates cycle-accurate FHMs, functionally equivalent to the RTL that simulate in the Incisive simulation environment 80–90% as fast as the original untimed input model. These FHMs enable faster verification and earlier hardware-software co-design.

BENEFITS

- Enables control and datapath micro-architecture exploration to determine optimum design tradeoffs
- Provides earlier feedback on implementation feasibility, area, and performance
- Creates better-than-manual-quality RTL for new designs with much less effort
- Performs fast ECOs without full re-synthesis and re-verification
- Easily retargets designs for different applications and manufacturing processes
- Accelerates design closure with far fewer iterations
- Enables faster simulation and verification
- Allows engineers to develop software earlier with FHM virtual prototype

FEATURES

- Accepts a wide range of C/C++/SystemC coding styles and constructs, including templates, classes, user-defined types, and certain types of pointers
- Automatically generates synthesizable IEEE-1364 Verilog® and synthesis scripts for Encounter RTL Compiler global synthesis
- Automatically generates I/O cycle-accurate simulation models, assertions, and scripts for simulation
- BST database tracks all data transformations from SystemC source files all the way to logic synthesis, enabling analysis and mapping of RTL data back to logic SystemC source code
- CFS maintains independence between functionality and design constraints

- Incremental synthesis enables more design exploration and faster ECO turnaround time
- ELS enables parallel control-datapath optimization—better than average human QoR
- Integrated, interactive GUI provides complete environment for synthesis, analysis, and debug; affords maximum user control of the high-level synthesis process and visualization of results
- Automatically generates SystemC “wrappers” to enable RTL verification with SystemC testbenches
- Provides integrated/tested flow and scripts for Calypto SLEC
- Supports OSCI® 1.0 transaction-level modeling (TLM) constructs

PLATFORM

- Linux 32/64-bit

CADENCE SERVICES AND SUPPORT

- Cadence application engineers can answer your technical questions by telephone, email, or Internet—they can also provide technical assistance and custom training
- Cadence certified instructors teach more than 70 courses and bring their real-world experience into the classroom
- More than 25 Internet Learning Series (iLS) online courses allow you the flexibility of training at your own computer via the Internet
- SourceLink® online customer support gives you answers to your technical questions—24 hours a day, 7 days a week—including the latest in quarterly software rollups, product change release information, technical documentation, solutions, software updates, and more

For more information,
Email us at
info@cadence.com
or visit
www.cadence.com

cadence™

Cadence Design Systems, Inc.

CORPORATE HEADQUARTERS

2655 Seely Avenue
San Jose, CA 95134
P: +1.800.746.6223 (*within US*)
+1.408.943.1234 (*outside US*)
F: +1.408.943.5001
www.cadence.com