

**Exhibit B-1: Evidence of Infringement of U.S. Patent No. 7,971,167 by Cadence Stratus HLS,
Including Stratus High-Level Synthesis (HLS), Genus Synthesis, Joules RTL Power, Architectural Exploration,
Flow Automation, Reporting, Optimize, Xcelium Simulation, Joules Power Analysis, Jasper, and Incisive**

Cadence directly infringes claims 1 and 2 under §271(a) by installing Stratus HLS on a semiconductor design support device for designing a semiconductor IC, and making and/or using the support device. For example, Cadence installs Stratus HLS on computers to test the functionality of Stratus HLS, as well as provide support to its customers. "The Cadence Global Customer Support team strives to provide flexible access to the expert support you need to successfully resolve any EDA challenge or technical issue. Whether it's fast access to our knowledgebase of solutions and technical documentation or direct collaboration between, your team and Cadence technical experts, Cadence is committed to providing efficient technical support that fits your needs." SDT00000687-689, https://www.cadence.com/en_US/home/support/support-process.html. For purposes of this chart, "Stratus HLS" includes Stratus High-Level Synthesis (HLS), Genus Synthesis, Joules RTL Power, Architectural Exploration, Flow Automation, Reporting, Optimize, Xcelium Simulation, Joules Power Analysis, Jasper, and Incisive (also referred to as the "Stratus HLS software," "Stratus HLS Product(s)" or the "Stratus HLS system"). CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5219, 5220; CDNS0005224-5229, [Accelerate Algorithm to Silicon Development with Stratus HLS](#) at 5226. The term "Stratus HLS" and any variants including, for example, "Stratus HLS software," "Stratus HLS Product(s)" and "Stratus HLS system" also includes any additional associated computer software and data, and any associated hardware, and processes and methods related thereto not specifically listed above.

Cadence indirectly infringes claims 1 and 2 under §271(b) by inducing its customers to install the Stratus HLS Product on a computer in order to make and/or use a semiconductor design support device for designing a semiconductor IC. See, e.g., SystemC Synthesis with Stratus HLS Training, available at SDT00000690-693, https://www.cadence.com/en_US/home/training/all-courses/86170.html. "Cadence® software is available through electronic distribution to customers with a current maintenance agreement and Cadence Online Support, or eDA-on-Tap website accounts. To stay up to date when selected product base and update releases are available, Cadence Online Support users may set up their Software Update Preferences. Software update notifications will provide an email containing the release information along with a link to an online order form." SDT00000683-686, https://www.cadence.com/en_US/home/support/software-downloads.html. See also SDT00000679-681, https://www.cadence.com/en_US/home/support/computing-platform-support.html ("Cadence works with our hardware platform partners to ensure that our products run on a variety of UNIX- and PC-based systems. For up-to-date information on operating system support, select the support matrix below. This matrix shows current Cadence® technology releases and the operating systems supported across UNIX and PC platforms.")

Cadence also indirectly infringes claims 1 and 2 under §271(c) by contributing to the infringement by its customers who make and/or use the claimed semiconductor design support device by installing the Stratus HLS Product on a computer. "Cadence® software is available through electronic distribution to customers with a current maintenance agreement and Cadence Online Support, or eDA-on-Tap website accounts. To stay up to date when selected product base and update releases are available, Cadence Online Support users may set up their Software Update Preferences. Software update notifications will provide an email containing the release information along with a link to an online order form." SDT00000683-686, https://www.cadence.com/en_US/home/support/software-downloads.html. The Stratus HLS Product is a material part of a semiconductor design support device for designing a semiconductor IC. The portion of the Stratus HLS Product that maps to claim 1 has no substantial non-infringing uses.

Cadence indirectly infringes claims 3 and 4 under §271(b) by inducing its customers to use the Stratus HLS Product to perform the manufacturing method. "Early results suggest that this flow will become the de facto standard method of developing and implementing algorithms targeted for silicon." https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/digital-design-signoff/cadence-stratus-hls-algorithm-wp.pdf. Customers utilize Stratus HLS to perform the manufacturing method in order to design and manufacture (including direct a fab to manufacture) a semiconductor integrated circuit. Cadence is a member of the TSMC EDA Alliance. "TSMC EDA Alliance, a key component of TSMC Open Innovation Platform® (OIP), reduces design barriers for customer's adoption of TSMC process technologies. By combining TSMC and EDA Alliance members R&D capability and resource, new generations of EDA solutions are enabled to be compliant to TSMC technology requirements." SDT00000682, https://www.tsmc.com/english/dedicatedFoundry/oip/eda_alliance.

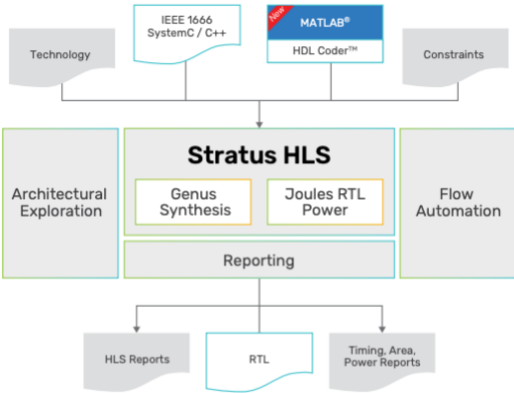
Cadence also indirectly infringes claims 3 and 4 under §271(c) by contributing to the infringement by its customers. Customers utilize the Stratus HLS Product to perform the manufacturing method to design and manufacture (or direct a fab to manufacture) a semiconductor integrated circuit. The Stratus HLS software is a

material part of performing the manufacturing method, and the portion of Stratus HLS involved in performing the claimed method has no substantial non-infringing uses.

US7971167	Stratus HLS
1[pre]. A semiconductor design support device for designing a semiconductor integrated circuit, the device comprising:	To the extent the preamble is limiting, a computer installed with Stratus HLS software is a semiconductor design support device for designing a semiconductor integrated device. Stratus HLS starts with a behavioral description, which is then converted into RTL. ▸ Synthesis of SystemC assertions and C++ asserts to SystemVerilog assertions (SVAs) CDNS0005219-5223, Stratus HLS Datasheet at 5220. Integrated with the Cadence verification suite, Stratus HLS supports automated mixed-language (SystemC and RTL) verification and debug including assertions, debugging, waveforms, and linkage back to the original SystemC design. CDNS0005219-5223, Stratus HLS Datasheet at 5221. Cadence® Stratus™ High-Level Synthesis (HLS) automatically creates high-quality register transfer level (RTL) design implementations for ASIC, system-on-chip (SoC), and FPGA targets from high-level IEEE 1666 SystemC™, C++, and MATLAB® descriptions. The proven successes of Stratus HLS in production designs around the world are testament to its consistently high-quality results, mature feature set, and complete design coverage. While most widely used for image processing, wireless, and machine learning (ML) applications, products built with Stratus HLS technology can be found in your home, automobile, and pockets. CDNS0005219-5223, Stratus HLS Datasheet at 5219.

US7971167	Stratus HLS
	<pre>graph TD Tech[Technology] --> StratusHLS SC[IEEE 1666 SystemC / C++] --> StratusHLS MATLAB[MATLAB® HDL Coder™] --> StratusHLS Constraints[Constraints] --> StratusHLS subgraph StratusHLS [Stratus HLS] AE[Architectural Exploration] GS[Genus Synthesis] JRP[Joules RTL Power] FA[Flow Automation] R[Reporting] end StratusHLS --> HR[HLS Reports] StratusHLS --> RTL[RTL] StratusHLS --> TAPR[Timing, Area, Power Reports]</pre> <i>Figure 1: Stratus HLS uses the Genus synthesis and Joules power engines to create high-quality RTL targeted to your technology and design constraints</i> CDNS0005219-5223, Stratus HLS Datasheet at 5219.
[a] a behavioral description configured to describe an algorithm of processing performed by hardware in a motion level;	The semiconductor design support device based on Stratus HLS receives and stores, and thus includes, a behavioral description configured to describe an algorithm of processing performed by hardware in a motion level. For example, behavioral descriptions can be provided in SystemC and C++ models. Stratus HLS supports untimed and timed SystemC and C++ models, including a mix of both, providing maximum flexibility to the designer. The output can be fully pipelined (new data each cycle), pipelined at reduced throughput (new CDNS0005219-5223, Stratus HLS Datasheet at 5220.

US7971167	Stratus HLS
	Behavioral IP Reuse Stratus HLS enables the creation and adaptation of behavioral IP, delivering on the promise of true IP reuse. Using Stratus HLS, the verified source code can be reused without modification for widely different process technologies, clock speeds, or PPA targets. Modifications to the algorithm, architecture, or interfaces can be made incrementally at a high level, where previously they required a complete RTL rewrite. Behavioral IP reuse with Stratus HLS significantly reduces overall design effort and maximizes return on investment (ROI). CDNS0005219-5223, Stratus HLS Datasheet at 5222. Further, HLS allows algorithmic code to be placed in the context of a hardware module: Features IEEE 1666 SystemC, C, C++ Synthesis Stratus HLS supports industry-standard IEEE 1666 SystemC as well as C and C++ semantics. The SystemC class library provides modularity, concurrency, and bit-accuracy, allowing designers to use familiar hierarchical decomposition techniques to manage design complexity. By supporting hierarchy and multiple threads, SystemC allows design and verification of algorithms and their interfaces operating concurrently. Stratus HLS also allows C or C++ algorithmic code to be automatically put into the context of a hardware module with hardware interfaces. CDNS0005219-5223, Stratus HLS Datasheet at 5220. Stratus Import The AES algorithm code that we removed from the Xtensa core is reused in this step. We take this behavioral C++ implementation of the AES algorithm and run it through Stratus' bdw_import utility, which will: ▶ Create SystemC wrappers around the C++ algorithm. This adds hardware-required constructs like clock, reset, and communication ports ▶ Create a stub testbench for the new accelerator ▶ Create a Stratus project configuration file The designer will then modify the Stratus project file to specify parameters such as clock frequency, reset mode, and technology libraries.

US7971167	Stratus HLS
	CDNS0005249-5255, Automated HW/SW Co-Design of DSP Systems Composed of Processors and Hardware Accelerators, at 5253 Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[b] an RTL description generated by reading the behavioral description, the RTL description configured to recognize a concept including register and clock synchronism particular to the hardware;	The semiconductor design support device based on Stratus HLS generates and stores, and thus includes, an RTL description generated by reading the behavioral description. Stratus utilizes the Genus Synthesis Solution to generate the RTL description from the behavioral description.  <i>Figure 1: Stratus HLS uses the Genus synthesis and Joules power engines to create high-quality RTL targeted to your technology and design constraints</i> CDNS0005219-5223, Stratus HLS Datasheet at 5219. The RTL description is configured to recognize a concept including register and clock synchronism particular to the hardware.

US7971167	Stratus HLS
	Design Closure Stratus HLS ensures easy timing closure for the generated RTL by exhaustively analyzing each path and scheduling operation so they fit in the given clock period. Stratus HLS uses patented datapath optimization technology and the embedded Genus synthesis to build all datapath components, multiplexers, and registers in the specified technology library to get accurate timing and area models. The user can control how aggressively Stratus HLS packs these operations into each clock period. Creating designs with Stratus HLS can save months of back-end effort by preventing timing closure problems. Integration with Genus physical synthesis allows early visibility and feedback into likely congestion problems, allowing the front-end designer to avoid problems in the back-end. CDNS0005219-5223, Stratus HLS Datasheet at 5222; see also SDT00000745-747, Genus Datasheet. Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[c] a latency analyzer configured to analyze a result of a logic simulation performed on the RTL description to calculate a latency in each block representing an operation in a predetermined unit in the behavioral description;	The semiconductor design support device based on Stratus HLS includes a latency analyzer configured to analyze a result of a logic simulation performed on the RTL description to calculate a latency in each block representing an operation in a predetermined unit in the behavioral description. ▸ Automated design and verification of hundreds of blocks with a consistent verification environment from TLM models through gates, including mixed-language (SystemC and RTL) simulation and debug CDNS0005219-5223, Stratus HLS Datasheet at 5220. Hierarchical Design Stratus HLS is applicable to a single block or complex hierarchy of modules, including both HLS and RTL blocks. Stratus design and verification automation allows the designer to synthesize one, some, or all of the modules and do mixed SystemC and RTL simulation and verification.

US7971167	Stratus HLS
	CDNS0005219-5223, Stratus HLS Datasheet at 5220. Moving from abstract MATLAB models to RTL descriptions has required manual conversion of MATLAB code into RTL. By definition, an RTL description expresses the cycle-accurate behavior. This step involves the design of a micro-architecture that accurately captures detailed cycle-by-cycle behavior and schedules operations among finite hardware resources to meet PPA goals in the implementation. To achieve optimal PPA, the micro-architectural solution space must be thoroughly explored—where the CDNS0005224-5229, Accelerate with Stratus HLS at 5226. Design Closure Stratus HLS ensures easy timing closure for the generated RTL by exhaustively analyzing each path and scheduling operation so they fit in the given clock period. Stratus HLS uses patented datapath optimization technology and the embedded Genus synthesis to build all datapath components, multiplexers, and registers in the specified technology library to get accurate timing and area models. The user can control how aggressively Stratus HLS packs these operations into each clock period. Creating designs with Stratus HLS can save months of back-end effort by preventing timing closure problems. Integration with Genus physical synthesis allows early visibility and feedback into likely congestion problems, allowing the front-end designer to avoid problems in the back-end. CDNS0005219-5223, Stratus HLS Datasheet at 5222. Stratus HLS includes the Genus Synthesis Solution engine. ▸ Genus logic synthesis and Joules power engines inside of Stratus HLS provide accurate timing, area, and power estimates CDNS0005219-5223, Stratus HLS Datasheet at 5220. Genus logic synthesis has timing awareness, including delay calculation. ▸ Automatic extraction of full timing and physical contexts for any subset of a design. Reduces iterations between unit-level and chip-/block-level synthesis by 2X or more. SDT00000737-739, Genus Synthesis Solution Datasheet at 737.

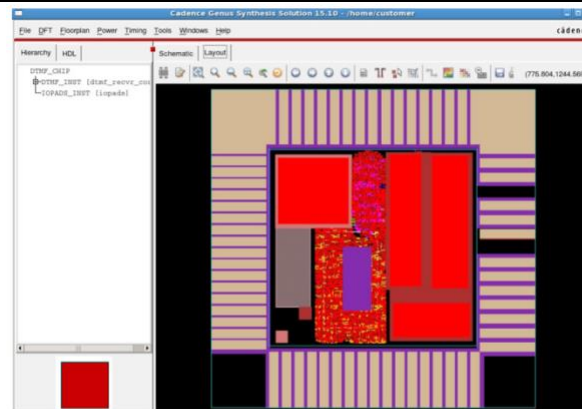


Figure 1: The Genus Synthesis Solution enables timing debug with physical interconnect knowledge built-in. Cross-probe to the physical viewer to see associated wirelengths, floorplan blockages, and estimated routing, and extract the chip-/block-level physical context for use in unit-level RTL design.

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 737.

- Unified GigaPlace™ engine, delay calculation, parasitic extraction, and timing-driven global routing with Cadence Innovus™ Implementation System, timing and wirelength between the tools correlate to within 5%

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 737.

Tight Correlation to Place and Route

The Genus Synthesis Solution shares several common engines with the Innovus Implementation System, including the GigaPlace engine, delay calculation, parasitic extraction, and timing-driven global routing. Timing and wirelength between the tools correlate tightly to within 5%, and global routing performance is 4X better. Both tools are critical for

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 738.

- Timing-driven physically aware multi-bit flop mapping
- Pipeline and general register retiming

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 738.

Register Retiming

The Genus Synthesis Solution can retime registers along pipelines and around sequential loops. Retiming can increase or decrease the number of flops along the retiming cut to achieve the best possible PPA tradeoff.

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 738.

Parasitic extraction and delay calculation. The Genus and Innovus solutions leverage unified parasitic extraction and delay calculation with full support for advanced-node waveform modeling.

These unified engines also extend into the Cadence Tempus™ Timing Signoff Solution, enabling truly convergent front-to-back modeling through the full Cadence digital implementation flow.

SDT00000740-744, [Genus Synthesis Solution Product Brief](#) at 742.

Stratus also uses Xcelium to simulate scenarios.

4. Analysis: In addition to HLS optimization, Stratus HLS launches Cadence's Genus™ Synthesis Solution, Xcelium™ Logic Simulator, and Joules™ RTL Power Solution to verify functionality and measure power consumption.

CDNS0005224-5229, [Accelerate Algorithm to Silicon Development with Stratus HLS](#) at 5228.

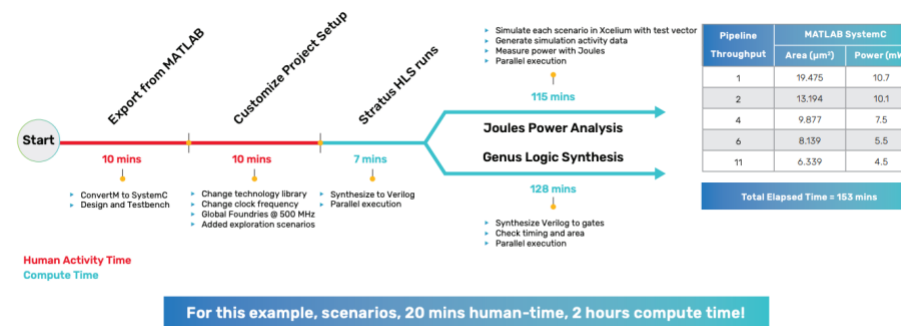
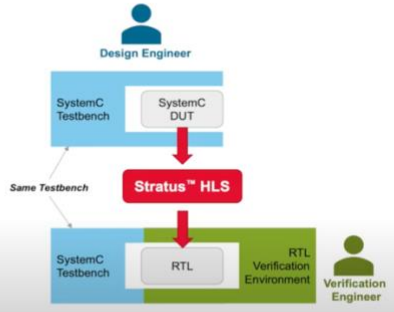


Figure 6: Testcase showing the productivity gain delivered by the integration—five micro-architectures explored with only 20-minutes of human activity time and 2 hours of compute time

CDNS0005224-5229, [Accelerate Algorithm to Silicon Development with Stratus HLS](#) at 5228.

US7971167	Stratus HLS
	The Xcelium simulator offers the following benefits aimed at accelerating system development: • Multi-core simulation improves runtime while also reducing project schedules: The third generation Xcelium simulator is built on the technology acquired from Rocketick and is the only fully-released simulator based on production-proven parallel simulation technology. It speeds runtime by an average of 3X for register-transfer level (RTL) design simulation, 5X for gate-level simulation and 10X for parallel design for test (DFT) simulation, potentially saving weeks to months on project schedules. • Broad applicability: The Xcelium simulator supports modern design styles and IEEE standards, enabling engineers to realize performance gains without recoding. • Easy to use: The Xcelium simulator's compilation and elaboration flow assigns the design and verification testbench code to the ideal engines and automatically selects the optimal number of cores for fast execution speed. • Incorporates several new patent-pending technologies to improve productivity: New features that speed overall SoC verification time include SystemVerilog testbench coverage for faster verification closure and parallel multi-core build. "Verification is often the primary cost and schedule challenge associated with getting new, high-quality products to market," said Dr. Anirudh Devgan, senior vice president and general manager of the Digital & Signoff Group and the System & Verification Group at Cadence. "The Xcelium simulator combined with JasperGold® Apps, the Palladium® Z1 Enterprise Emulation Platform and the Protium™ S1 FPGA-Based Prototyping Platform offer customers the strongest verification suite on the market, enabling engineers to accelerate the pace of innovation." Cadence News Release, Cadence Launches Xcelium Parallel Simulator, the Industry's First Production-Proven Parallel Simulator, (Feb. 27, 2017), at SDT00000734-736. Verification Tasks in a HLS Flow • SystemC® DUT – Algorithm verification – Design checking / linting – Code coverage to ensure testbench covers all design code • RTL DUT (by design engineer) – Performance testing (latency, throughput) – PPA assessment (via logic synth, power estimation) – Functional equivalence – Connectivity/interoperability with other blocks • RTL DUT (by verification engineer) – Feed-forward verification metrics from designer – Potentially reuse some assets, as needed – Continue though verification flow to signoff  13 © 2020 Cadence Design Systems, Inc. All rights reserved. cadence SDT00000748, TensorFlow to RTL with High-Level Synthesis -- Cadence Design Systems at 14:50 The Stratus HLS GUI allows users to analyze individual modules synthesized to RTL and displays information including latency for each module.

Synthesis and RTL Simulation

- Stratus™ HLS synthesizes each module to RTL
 - Targets specified technology and clock period
 - Each module synthesized according to given settings and configurations
- RTL is simulated with the same testbench used for the behavioral SystemC® model
- RTL simulation...
 - Verifies network performance (latency, throughput)
 - Verifies functional equivalence between the behavior and synthesized RTL
 - Ensures blocks working together (e.g., no deadlock)
 - Generates waveforms for power analysis

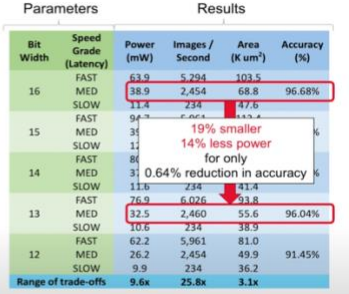
Module Name	RTL Verilog	Area	Latency	Power	Run Time
B_S_TLM	1024	passed	0.0000		
B_W_PFM	18781	passed	0.0000		
B_V_FAST	1,076,754	800,964,020	passed	0.0000	
CONV1 RTL_V_FAST	1	27,418	880	0.0001	
PROB1 RTL_V_FAST	1	17,532	1,575	0.0000	
CONV2 RTL_V_FAST	1	420,404	5,228	0.0438	
PROB2 RTL_V_FAST	1	48,758	4,827	0.0102	
PCL RTL_V_FAST	1	387,888	35,303	0.1010	
CL RTL_V_FAST	1	198,036	7,424	0.0007	

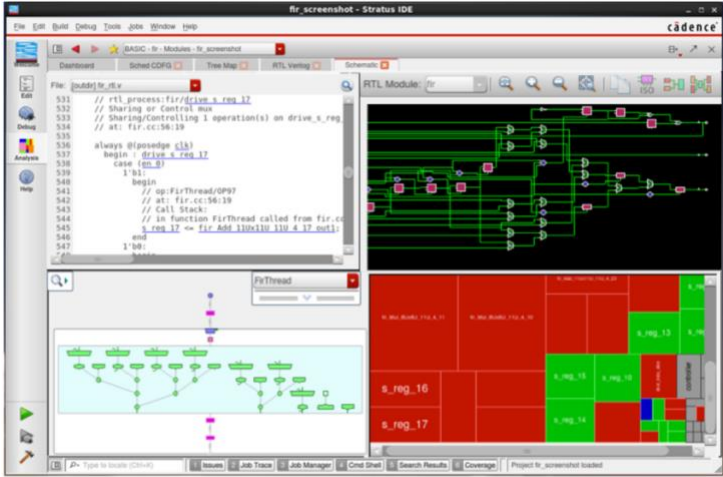
SDT00000748, [TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems](#) at 16:55

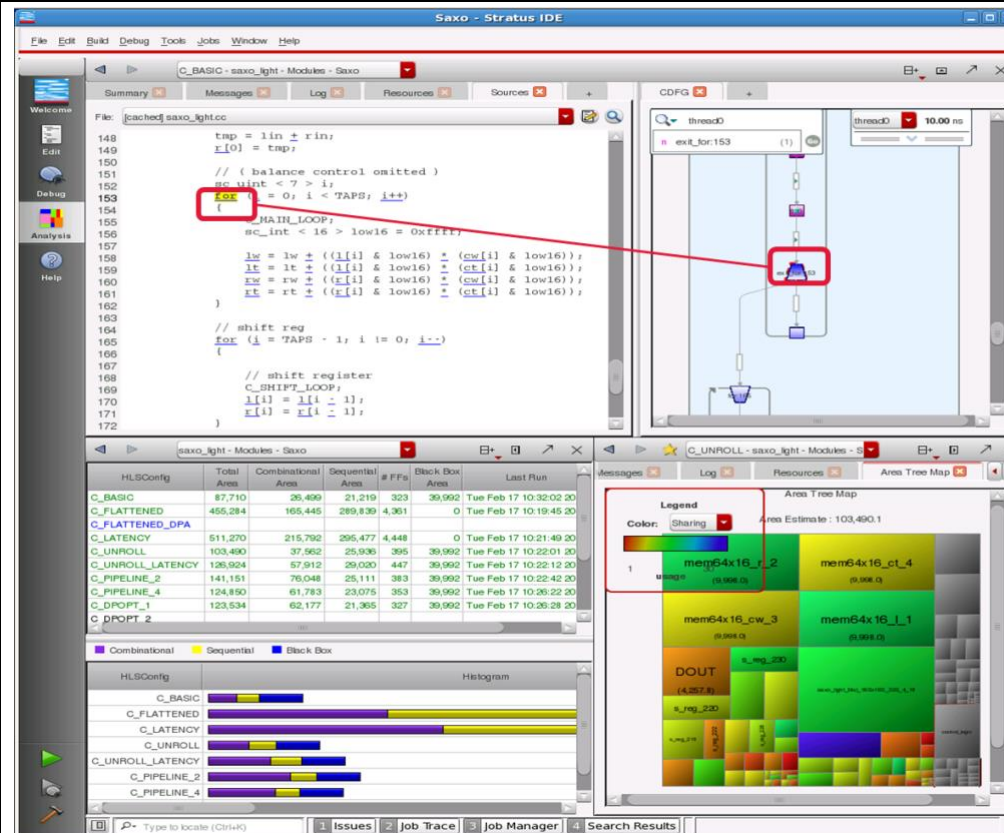
RTL Performance and PPA Assessment

- Power and area measured by implementation tools
- Enabled by **turnkey automation**
 - Simulation with automatic setup of TLM, behavioral SystemC® and RTL Verilog
 - Logic synthesis with Genus™ solution for area and timing estimates
 - Power measurement with Joules™ solution using simulation waveforms

SDT00000748, [TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems](#) at 18:10

US7971167	Stratus HLS
	Design Space Exploration Results - Multiple bit widths implemented for each SystemC® module - Larger bit widths give better accuracy - Smaller bit widths give smaller hardware with reduced power consumption - Multiple synthesis configurations for each SystemC module - Tradeoff latency and hardware area - Slow and small vs fast and large - Easy to find the best solution that meets your requirements  SDT00000748, TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems at 20:07 Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[d] a correspondence table in which each block in the behavioral description corresponds to a state in the RTL description; and	The semiconductor design support device based on Stratus HLS includes a correspondence table in which each block in the behavioral description corresponds to a state in the RTL description. Automated design and verification of hundreds of blocks with a consistent verification environment from TLM models through gates, including mixed-language (SystemC and RTL) simulation and debug CDNS0005219-5223, Stratus HLS Datasheet at 5220.

US7971167	Stratus HLS
	Design Closure Stratus HLS ensures easy timing closure for the generated RTL by exhaustively analyzing each path and scheduling operation so they fit in the given clock period. Stratus HLS uses patented datapath optimization technology and the embedded Genus synthesis to build all datapath components, multiplexers, and registers in the specified technology library to get accurate timing and area models. The user can control how aggressively Stratus HLS packs these operations into each clock period. Creating designs with Stratus HLS can save months of back-end effort by preventing timing closure problems. Integration with Genus physical synthesis allows early visibility and feedback into likely congestion problems, allowing the front-end designer to avoid problems in the back-end. CDNS0005219-5223, Stratus HLS Datasheet at 5222. Stratus HLS supports graphical analysis of the RTL with links to source code.  The screenshot shows the Cadence Stratus IDE interface. On the left, the 'RTL Module: fir' pane displays Verilog code for a FIR filter. The middle pane shows a logic diagram with interconnected blocks and signals. The bottom-right pane shows a floorplan with various components labeled, such as 's_reg_16', 's_reg_17', 's_reg_15', 's_reg_14', 's_reg_13', 's_reg_12', 's_reg_11', 's_reg_10', 's_reg_9', 's_reg_8', 's_reg_7', 's_reg_6', 's_reg_5', 's_reg_4', 's_reg_3', 's_reg_2', 's_reg_1', 's_reg_0', 's_reg_neg_1', 's_reg_neg_2', 's_reg_neg_3', 's_reg_neg_4', 's_reg_neg_5', 's_reg_neg_6', 's_reg_neg_7', 's_reg_neg_8', 's_reg_neg_9', 's_reg_neg_10', 's_reg_neg_11', 's_reg_neg_12', 's_reg_neg_13', 's_reg_neg_14', 's_reg_neg_15', 's_reg_neg_16', 's_reg_neg_17', 's_reg_neg_18', 's_reg_neg_19', 's_reg_neg_20', 's_reg_neg_21', 's_reg_neg_22', 's_reg_neg_23', 's_reg_neg_24', 's_reg_neg_25', 's_reg_neg_26', 's_reg_neg_27', 's_reg_neg_28', 's_reg_neg_29', 's_reg_neg_30', 's_reg_neg_31', 's_reg_neg_32', 's_reg_neg_33', 's_reg_neg_34', 's_reg_neg_35', 's_reg_neg_36', 's_reg_neg_37', 's_reg_neg_38', 's_reg_neg_39', 's_reg_neg_40', 's_reg_neg_41', 's_reg_neg_42', 's_reg_neg_43', 's_reg_neg_44', 's_reg_neg_45', 's_reg_neg_46', 's_reg_neg_47', 's_reg_neg_48', 's_reg_neg_49', 's_reg_neg_50', 's_reg_neg_51', 's_reg_neg_52', 's_reg_neg_53', 's_reg_neg_54', 's_reg_neg_55', 's_reg_neg_56', 's_reg_neg_57', 's_reg_neg_58', 's_reg_neg_59', 's_reg_neg_60', 's_reg_neg_61', 's_reg_neg_62', 's_reg_neg_63', 's_reg_neg_64', 's_reg_neg_65', 's_reg_neg_66', 's_reg_neg_67', 's_reg_neg_68', 's_reg_neg_69', 's_reg_neg_70', 's_reg_neg_71', 's_reg_neg_72', 's_reg_neg_73', 's_reg_neg_74', 's_reg_neg_75', 's_reg_neg_76', 's_reg_neg_77', 's_reg_neg_78', 's_reg_neg_79', 's_reg_neg_80', 's_reg_neg_81', 's_reg_neg_82', 's_reg_neg_83', 's_reg_neg_84', 's_reg_neg_85', 's_reg_neg_86', 's_reg_neg_87', 's_reg_neg_88', 's_reg_neg_89', 's_reg_neg_90', 's_reg_neg_91', 's_reg_neg_92', 's_reg_neg_93', 's_reg_neg_94', 's_reg_neg_95', 's_reg_neg_96', 's_reg_neg_97', 's_reg_neg_98', 's_reg_neg_99'. The bottom status bar shows 'Project fir_screenshots loaded'. <i>Figure 2: Complete graphical analysis with links to source code</i> Stratus HLS Datasheet at 2.



Tal Laviv, How to Dramatically Reduce Time from Architecture Spec to Tapeout, LinkedIn (Mar. 10, 2020), <https://www.linkedin.com/pulse/how-dramatically-reduce-time-from-architecture-spec-tapeout-laviv/>, SDT00000661-670 [hereinafter Reduce Time to Tapeout].

Integrated with the Cadence verification suite, Stratus HLS supports automated mixed-language (SystemC and RTL) verification and debug including assertions, debugging, waveforms, and linkage back to the original SystemC design.

CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5221.

Stratus HLS uses coverage databases to enable tracing RTL descriptions to the originating behavioral descriptions.

By turning on the *rtl_annotation* feature in Stratus™ HLS, the uncovered RTL code lines are easily traced back to the originating behavioral SystemC® code lines. This saves weeks of verification effort by quickly identifying weaknesses in the TB and/or possible bugs in the DUT, and also by allowing developers to debug the involved logic on the higher abstraction behavioral/algorithmic SystemC® model.

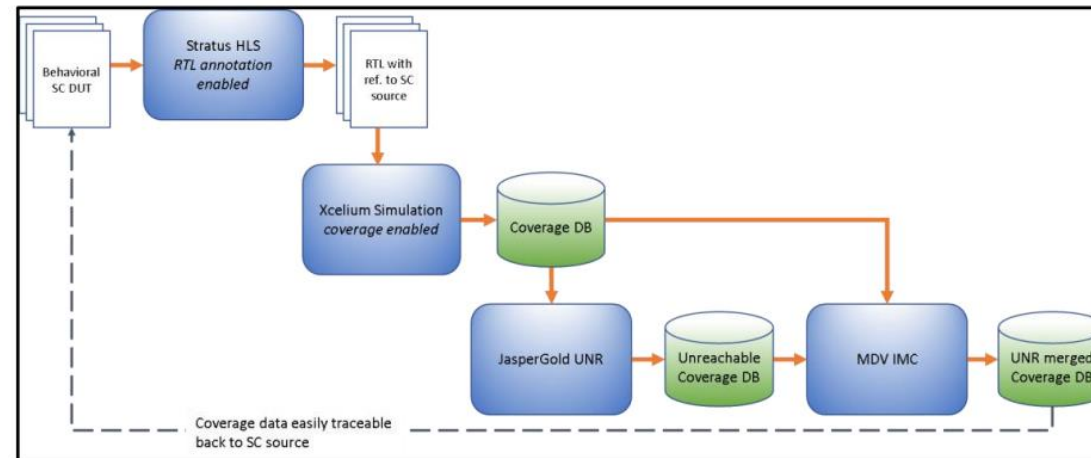


Figure 8: UNR analysis flow with traceability back to SC source

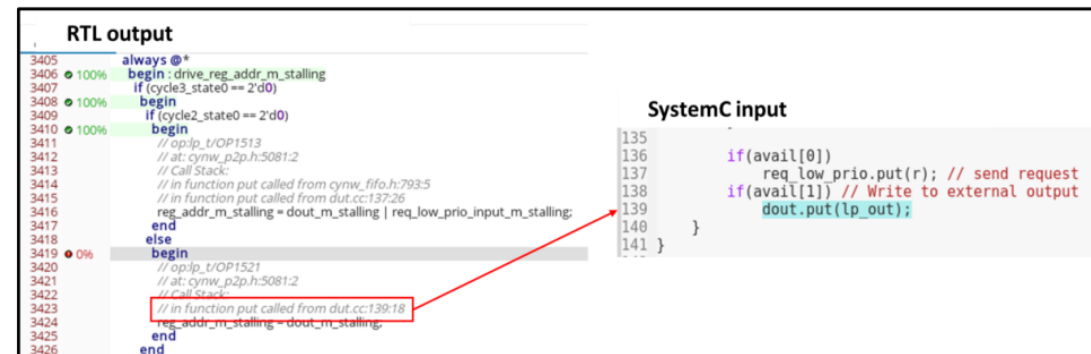


Figure 9: RTL annotation for traceability back to originating SystemC code lines

S. Dahir, "Using HLS to Improve Design-for-Verification of Multi-Pipeline Designs with Resource Sharing," DVCON 2021 at 6-7, available at <https://dvcon-proceedings.org/wp-content/uploads/using-hls-to-improve-design-for-verification-of-multi-pipeline-designs-with-resource-sharing.pdf.pdf>, at SDT00000671-678 [hereinafter Design-for-Verification].

US7971167	Stratus HLS
	HLS for SystemC involves a sequence of algorithms to realize an implementation-based model: • elaboration • input synthesis directives and constraints • characterization of the hardware resources for all operations • scheduling operations to clocks • generation of the RTL model To enable SystemC synthesis, additional "hardware-centric" features were needed – e.g., modules, ports, signal, processes, bit-accurate datatypes, communication channels, and clocks. SystemC synthesis directives are also unique, offering (optional) user guidance on: • program loop interpretation (e.g., "UNROLL_LOOP") • resource allocation (i.e., binding operations to resources) • cycle scheduling (e.g., pipelining evaluations, latency) • allocation and mapping to registers • reset behavior • creation of finite-state machine states and transitions • definition of data channels (e.g., point-to-point interfaces, FIFO's, etc.) • pin-level protocols for data communication (with SystemC function calls through an event on a "SC_port") Even with cycle-accurate definitions for protocols and controls, the base algorithm models are still abstract – SystemC for HLS maintains its design and verification productivity. Daniel Nenni, Update on SystemC for High-Level Synthesis, SemiWiki (Mar. 4, 2019), https://semiwiki.com/eda/cadence/8080-update-on-systemc-for-high-level-synthesis/, at SDT00000749-753. Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[e] a correspondence table generator configured to generate the correspondence table.	The semiconductor design support device based on Stratus HLS includes a correspondence table generator configured to generate the correspondence table. Design Closure Stratus HLS ensures easy timing closure for the generated RTL by exhaustively analyzing each path and scheduling operation so they fit in the given clock period. Stratus HLS uses patented datapath optimization technology and the embedded Genus synthesis to build all datapath components, multiplexers, and registers in the specified technology library to get accurate timing and area models. The user can control how aggressively Stratus HLS packs these operations into each clock period. Creating designs with Stratus HLS can save months of back-end effort by preventing timing closure problems. Integration with Genus physical synthesis allows early visibility and feedback into likely congestion problems, allowing the front-end designer to avoid problems in the back-end.

CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5222.

Stratus HLS supports graphical analysis of the RTL with links to source code.

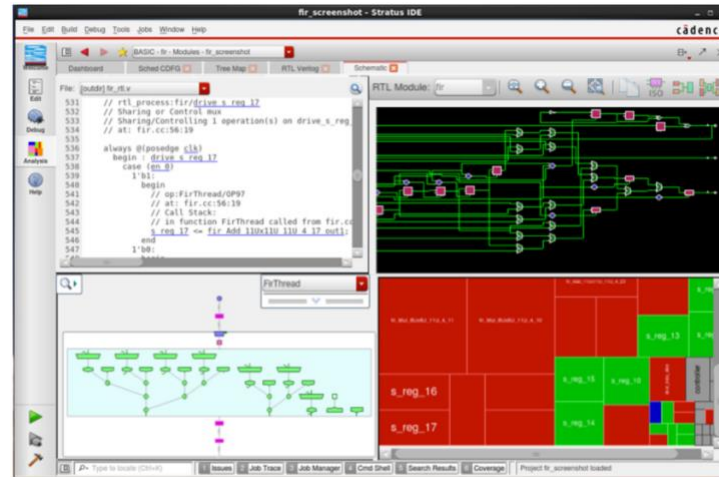
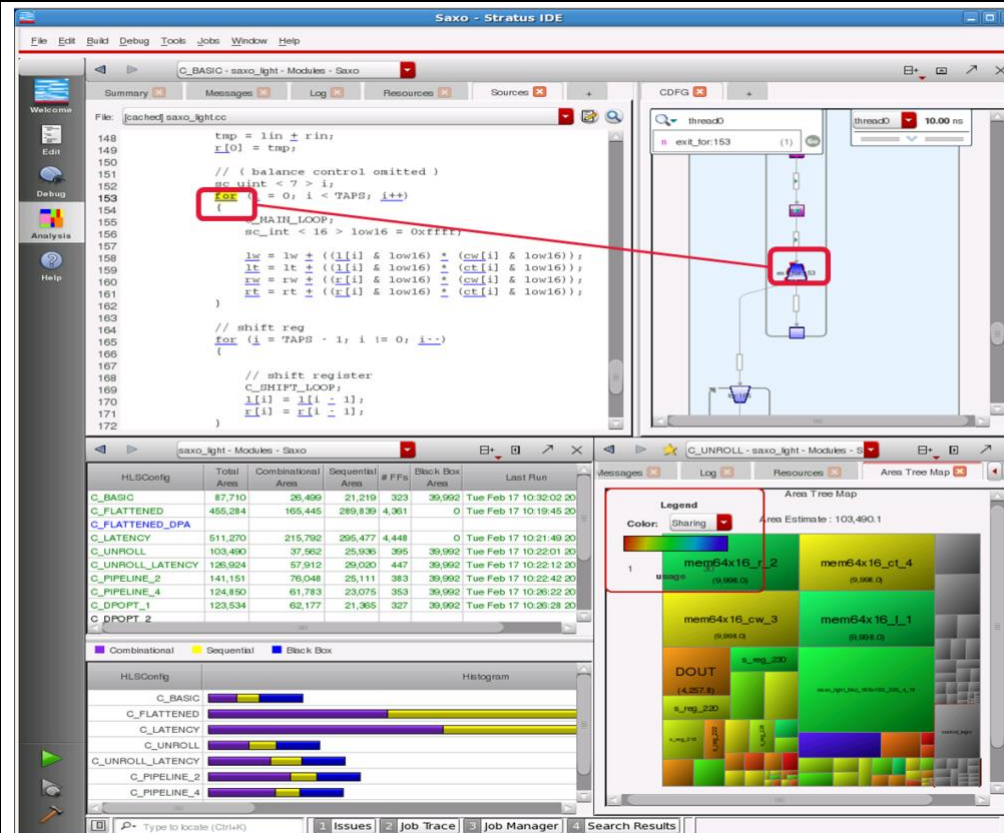


Figure 2: Complete graphical analysis with links to source code

CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5220.



SDT0000661-670, Reduce Time to Tapeout, <https://www.linkedin.com/pulse/how-dramatically-reduce-time-from-architecture-spec-tapeout-laviv/>

Integrated with the Cadence verification suite, Stratus HLS supports automated mixed-language (SystemC and RTL) verification and debug including assertions, debugging, waveforms, and linkage back to the original SystemC design.

CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5221.

By turning on the `rtl_annotation` feature in Stratus™ HLS, the uncovered RTL code lines are easily traced back to the originating behavioral SystemC® code lines. This saves weeks of verification effort by quickly identifying weaknesses in the TB and/or possible bugs in the DUT, and also by allowing developers to debug the involved logic on the higher abstraction behavioral/algorithmic SystemC® model.

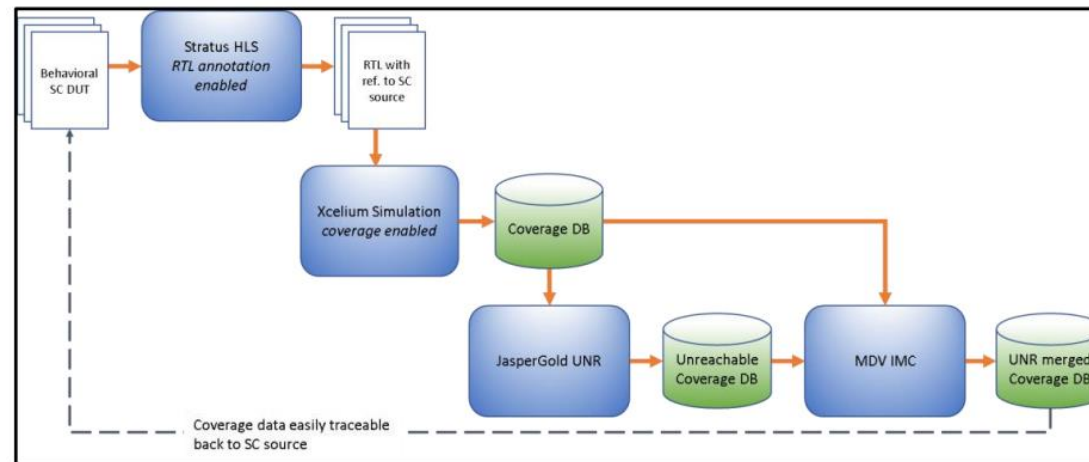


Figure 8: UNR analysis flow with traceability back to SC source

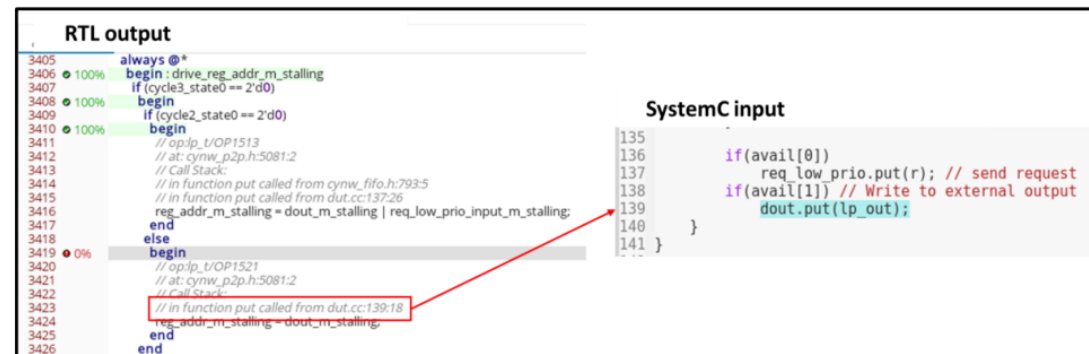


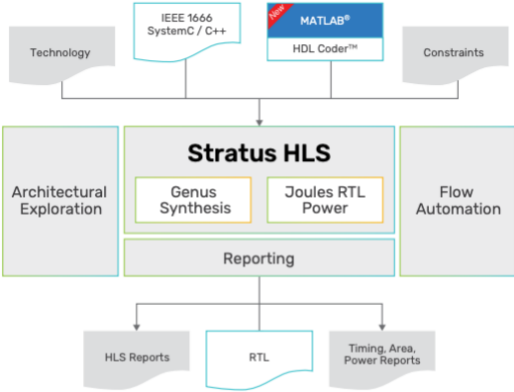
Figure 9: RTL annotation for traceability back to originating SystemC code lines

SDT00000671-678, Design-for-Verification, <https://dvcon-proceedings.org/wp-content/uploads/using-hls-to-improve-design-for-verification-of-multi-pipeline-designs-with-resource-sharing.pdf.pdf>

See also "Timing, Area, Power Reports":

US7971167	Stratus HLS
	SDT00000748, TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems at 16:55. Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
3. A manufacturing method for manufacturing a semiconductor integrated circuit, the method comprising:	To the extent the preamble is limiting, Stratus HLS users perform a manufacturing method for manufacturing a semiconductor integrated circuit. Cadence® Stratus™ High-Level Synthesis (HLS) automatically creates high-quality register transfer level (RTL) design implementations for ASIC, system-on-chip (SoC), and FPGA targets from high-level IEEE 1666 SystemC™, C++, and MATLAB® descriptions. The proven successes of Stratus HLS in production designs around the world are testament to its consistently high-quality results, mature feature set, and complete design coverage. While most widely used for image processing, wireless, and machine learning (ML) applications, products built with Stratus HLS technology can be found in your home, automobile, and pockets. CDNS0005219-5223, Stratus HLS Datasheet at 5219.
[a] describing an algorithm of processing performed by hardware in a motion level to generate a behavioral description;	Stratus HLS users describe an algorithm of processing performed by hardware in a motion level to generate a behavioral description. For example, behavioral descriptions can be provided in SystemC and C++ models. Stratus HLS supports untimed and timed SystemC and C++ models, including a mix of both, providing maximum flexibility to the designer. The output can be fully pipelined (new data each cycle), pipelined at reduced throughput (new CDNS0005219-5223, Stratus HLS Datasheet at 5220.

US7971167	Stratus HLS
	Behavioral IP Reuse Stratus HLS enables the creation and adaptation of behavioral IP, delivering on the promise of true IP reuse. Using Stratus HLS, the verified source code can be reused without modification for widely different process technologies, clock speeds, or PPA targets. Modifications to the algorithm, architecture, or interfaces can be made incrementally at a high level, where previously they required a complete RTL rewrite. Behavioral IP reuse with Stratus HLS significantly reduces overall design effort and maximizes return on investment (ROI). CDNS0005219-5223, Stratus HLS Datasheet at 5222. Further, HLS allows algorithmic code to be placed in the context of a hardware module: Features IEEE 1666 SystemC, C, C++ Synthesis Stratus HLS supports industry-standard IEEE 1666 SystemC as well as C and C++ semantics. The SystemC class library provides modularity, concurrency, and bit-accuracy, allowing designers to use familiar hierarchical decomposition techniques to manage design complexity. By supporting hierarchy and multiple threads, SystemC allows design and verification of algorithms and their interfaces operating concurrently. Stratus HLS also allows C or C++ algorithmic code to be automatically put into the context of a hardware module with hardware interfaces. CDNS0005219-5223, Stratus HLS Datasheet at 5220.

US7971167	Stratus HLS
	Stratus Import The AES algorithm code that we removed from the Xtensa core is reused in this step. We take this behavioral C++ implementation of the AES algorithm and run it through Stratus' bdw_import utility, which will: ▶ Create SystemC wrappers around the C++ algorithm. This adds hardware-required constructs like clock, reset, and communication ports ▶ Create a stub testbench for the new accelerator ▶ Create a Stratus project configuration file The designer will then modify the Stratus project file to specify parameters such as clock frequency, reset mode, and technology libraries. CDNS0005249-5255, Automated HW/SW Co-Design of DSP Systems Composed of Processors and Hardware Accelerators, at 5253. Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[b] reading the behavioral description to generate an RTL description for recognizing a concept including register and clock synchronism particular to the hardware;	Stratus HLS users direct the Stratus HLS software to read the behavioral description to generate an RTL description for recognizing a concept including register and clock synchronism particular to the hardware. Stratus utilizes the Genus Synthesis Solution to generate the RTL description from the behavioral description.  <i>Figure 1: Stratus HLS uses the Genus synthesis and Joules power engines to create high-quality RTL targeted to your technology and design constraints</i> CDNS0005219-5223, Stratus HLS Datasheet at 5219. The RTL description is configured to recognize a concept including register and clock synchronism particular to the hardware.

US7971167	Stratus HLS
	Design Closure Stratus HLS ensures easy timing closure for the generated RTL by exhaustively analyzing each path and scheduling operation so they fit in the given clock period. Stratus HLS uses patented datapath optimization technology and the embedded Genus synthesis to build all datapath components, multiplexers, and registers in the specified technology library to get accurate timing and area models. The user can control how aggressively Stratus HLS packs these operations into each clock period. Creating designs with Stratus HLS can save months of back-end effort by preventing timing closure problems. Integration with Genus physical synthesis allows early visibility and feedback into likely congestion problems, allowing the front-end designer to avoid problems in the back-end. CDNS0005219-5223, Stratus HLS Datasheet at 5222; see also SDT00000745-747, Genus Datasheet. Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[c] analyzing a result of a logic simulation performed on the RTL description to calculate a latency in each block representing an operation in a predetermined unit in the behavioral description;	Stratus HLS users direct the Stratus HLS software to analyze a result of a logic simulation performed on the RTL description to calculate a latency in each block representing an operation in a predetermined unit in the behavioral description. ▸ Automated design and verification of hundreds of blocks with a consistent verification environment from TLM models through gates, including mixed-language (SystemC and RTL) simulation and debug CDNS0005219-5223, Stratus HLS Datasheet at 5220. Hierarchical Design Stratus HLS is applicable to a single block or complex hierarchy of modules, including both HLS and RTL blocks. Stratus design and verification automation allows the designer to synthesize one, some, or all of the modules and do mixed SystemC and RTL simulation and verification.

US7971167	Stratus HLS
	CDNS0005219-5223, Stratus HLS Datasheet at 5220. Moving from abstract MATLAB models to RTL descriptions has required manual conversion of MATLAB code into RTL. By definition, an RTL description expresses the cycle-accurate behavior. This step involves the design of a micro-architecture that accurately captures detailed cycle-by-cycle behavior and schedules operations among finite hardware resources to meet PPA goals in the implementation. To achieve optimal PPA, the micro-architectural solution space must be thoroughly explored—where the CDNS0005224-5229, Accelerate with Stratus HLS at 5226. Design Closure Stratus HLS ensures easy timing closure for the generated RTL by exhaustively analyzing each path and scheduling operation so they fit in the given clock period. Stratus HLS uses patented datapath optimization technology and the embedded Genus synthesis to build all datapath components, multiplexers, and registers in the specified technology library to get accurate timing and area models. The user can control how aggressively Stratus HLS packs these operations into each clock period. Creating designs with Stratus HLS can save months of back-end effort by preventing timing closure problems. Integration with Genus physical synthesis allows early visibility and feedback into likely congestion problems, allowing the front-end designer to avoid problems in the back-end. CDNS0005219-5223, Stratus HLS Datasheet at 5222. Stratus HLS includes the Genus Synthesis Solution engine. ▸ Genus logic synthesis and Joules power engines inside of Stratus HLS provide accurate timing, area, and power estimates CDNS0005219-5223, Stratus HLS Datasheet at 5220. Genus logic synthesis has timing awareness (including delay calculation) and as such is aware of the latency of operations in the behavioral description. ▸ Automatic extraction of full timing and physical contexts for any subset of a design. Reduces iterations between unit-level and chip-/block-level synthesis by 2X or more. SDT00000737-739, Genus Synthesis Solution Datasheet at 737.

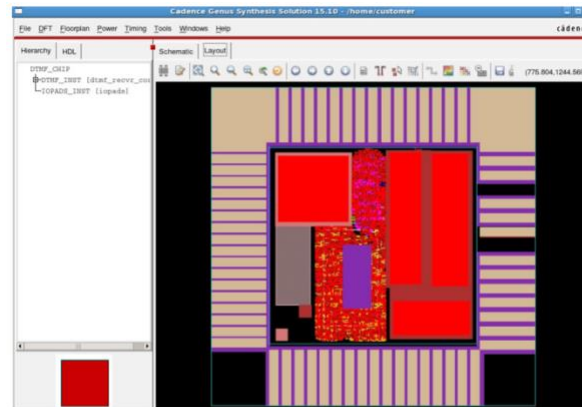


Figure 1: The Genus Synthesis Solution enables timing debug with physical interconnect knowledge built-in. Cross-probe to the physical viewer to see associated wirelengths, floorplan blockages, and estimated routing, and extract the chip-/block-level physical context for use in unit-level RTL design.

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 737.

- Unified GigaPlace™ engine, delay calculation, parasitic extraction, and timing-driven global routing with Cadence Innovus™ Implementation System, timing and wirelength between the tools correlate to within 5%

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 737.

Tight Correlation to Place and Route

The Genus Synthesis Solution shares several common engines with the Innovus Implementation System, including the GigaPlace engine, delay calculation, parasitic extraction, and timing-driven global routing. Timing and wirelength between the tools correlate tightly to within 5%, and global routing performance is 4X better. Both tools are critical for

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 738.

- Timing-driven physically aware multi-bit flop mapping
- Pipeline and general register retiming

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 738.

Register Retiming

The Genus Synthesis Solution can retime registers along pipelines and around sequential loops. Retiming can increase or decrease the number of flops along the retiming cut to achieve the best possible PPA tradeoff.

SDT00000737-739, [Genus Synthesis Solution Datasheet](#) at 738.

Parasitic extraction and delay calculation. The Genus and Innovus solutions leverage unified parasitic extraction and delay calculation with full support for advanced-node waveform modeling.

These unified engines also extend into the Cadence Tempus™ Timing Signoff Solution, enabling truly convergent front-to-back modeling through the full Cadence digital implementation flow.

SDT00000740-744, [Genus Synthesis Solution Product Brief](#) at 742.

Stratus also uses Xcelium to simulate scenarios.

4. Analysis: In addition to HLS optimization, Stratus HLS launches Cadence's Genus™ Synthesis Solution, Xcelium™ Logic Simulator, and Joules™ RTL Power Solution to verify functionality and measure power consumption.

CDNS0005224-5229, [Accelerate Algorithm to Silicon Development with Stratus HLS](#) at 5228.

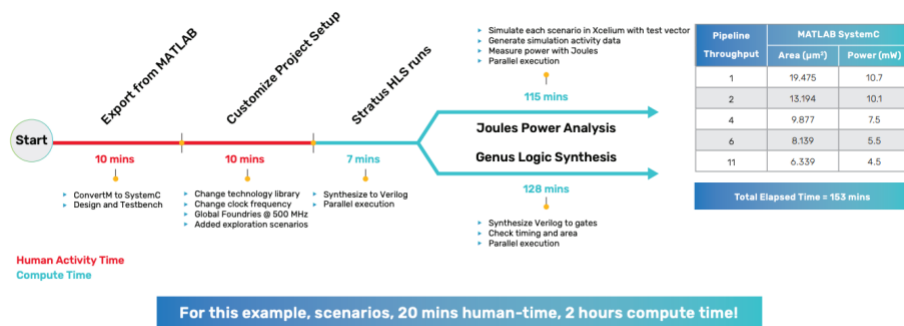
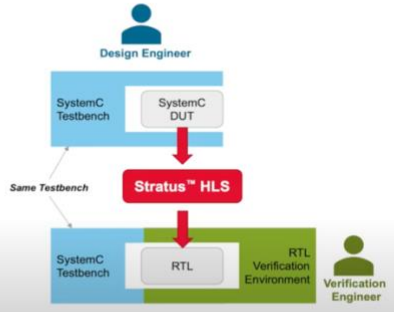


Figure 6: Testcase showing the productivity gain delivered by the integration—five micro-architectures explored with only 20-minutes of human activity time and 2 hours of compute time

CDNS0005224-5229, [Accelerate Algorithm to Silicon Development with Stratus HLS](#) at 5228.

US7971167	Stratus HLS
	The Xcelium simulator offers the following benefits aimed at accelerating system development: • Multi-core simulation improves runtime while also reducing project schedules: The third generation Xcelium simulator is built on the technology acquired from Rocketick and is the only fully-released simulator based on production-proven parallel simulation technology. It speeds runtime by an average of 3X for register-transfer level (RTL) design simulation, 5X for gate-level simulation and 10X for parallel design for test (DFT) simulation, potentially saving weeks to months on project schedules. • Broad applicability: The Xcelium simulator supports modern design styles and IEEE standards, enabling engineers to realize performance gains without recoding. • Easy to use: The Xcelium simulator's compilation and elaboration flow assigns the design and verification testbench code to the ideal engines and automatically selects the optimal number of cores for fast execution speed. • Incorporates several new patent-pending technologies to improve productivity: New features that speed overall SoC verification time include SystemVerilog testbench coverage for faster verification closure and parallel multi-core build. "Verification is often the primary cost and schedule challenge associated with getting new, high-quality products to market," said Dr. Anirudh Devgan, senior vice president and general manager of the Digital & Signoff Group and the System & Verification Group at Cadence. "The Xcelium simulator combined with JasperGold® Apps, the Palladium® Z1 Enterprise Emulation Platform and the Protium™ S1 FPGA-Based Prototyping Platform offer customers the strongest verification suite on the market, enabling engineers to accelerate the pace of innovation." SDT00000734-736, Cadence Launches Xcelium Parallel Simulator, the Industry's First Production-Proven Parallel Simulator Verification Tasks in a HLS Flow • SystemC® DUT – Algorithm verification – Design checking / linting – Code coverage to ensure testbench covers all design code • RTL DUT (by design engineer) – Performance testing (latency, throughput) – PPA assessment (via logic synth, power estimation) – Functional equivalence – Connectivity/interoperability with other blocks • RTL DUT (by verification engineer) – Feed-forward verification metrics from designer – Potentially reuse some assets, as needed – Continue though verification flow to signoff  © 2020 Cadence Design Systems, Inc. All rights reserved. SDT00000748, TensorFlow to RTL with High-Level Synthesis -- Cadence Design Systems at 14:50 The Stratus HLS GUI allows users to analyze individual modules synthesized to RTL and displays information including latency for each module.

Synthesis and RTL Simulation

- Stratus™ HLS synthesizes each module to RTL
 - Targets specified technology and clock period
 - Each module synthesized according to given settings and configurations
- RTL is simulated with the same testbench used for the behavioral SystemC® model
- RTL simulation...
 - Verifies network performance (latency, throughput)
 - Verifies functional equivalence between the behavior and synthesized RTL
 - Ensures blocks working together (e.g., no deadlock)
 - Generates waveforms for power analysis

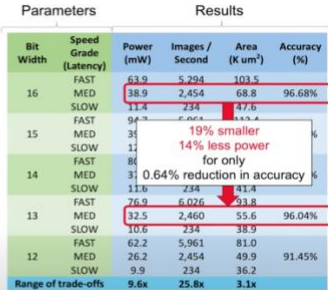
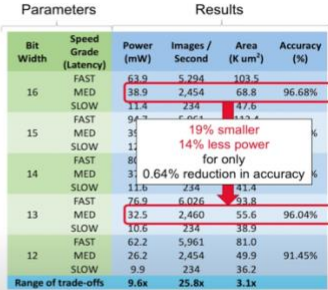
Module Name	Area (mm²)	Latency (ns)	Power (mW)
TLM	1024	0.0000	0.0000
PFM	1,076,754	0.0000	0.0000
FAST	800,964,020	0.0000	0.0000
CONV1 RTL_V FAST	1	27,418	0.0000
PROB1 RTL_V FAST	1	17,532	0.0000
CONV2 RTL_V FAST	1	420,404	0.0000
PROB2 RTL_V FAST	1	48,759	0.0000
POL RTL_V FAST	1	387,889	0.0000
OL RTL_V FAST	1	199,036	0.0000

SDT00000748, [TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems](#) at 16:55

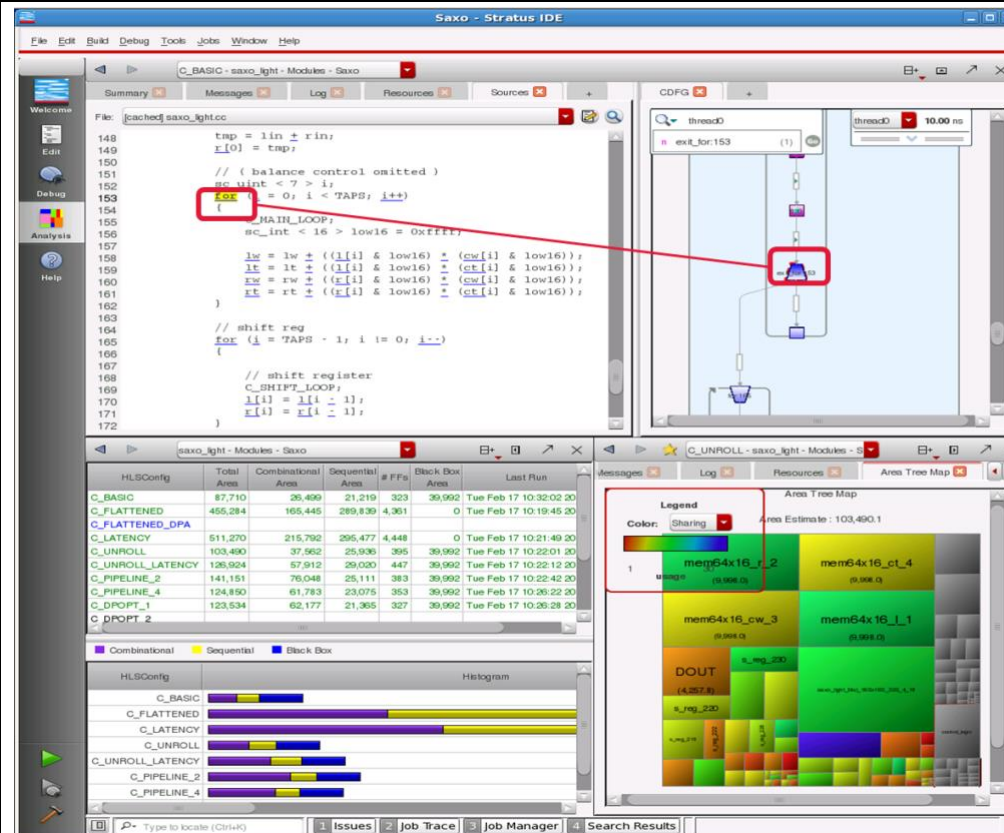
RTL Performance and PPA Assessment

- Power and area measured by implementation tools
- Enabled by **turnkey automation**
 - Simulation with automatic setup of TLM, behavioral SystemC® and RTL Verilog
 - Logic synthesis with Genus™ solution for area and timing estimates
 - Power measurement with Joules™ solution using simulation waveforms

SDT00000748, [TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems](#) at 18:10

US7971167	Stratus HLS
	Design Space Exploration Results - Multiple bit widths implemented for each SystemC® module - Larger bit widths give better accuracy - Smaller bit widths give smaller hardware with reduced power consumption - Multiple synthesis configurations for each SystemC module - Tradeoff latency and hardware area - Slow and small vs fast and large - Easy to find the best solution that meets your requirements  SDT00000748, TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems at 20:07 Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[d] designing a circuit based on the calculated latency;	Design Space Exploration Results - Multiple bit widths implemented for each SystemC® module - Larger bit widths give better accuracy - Smaller bit widths give smaller hardware with reduced power consumption - Multiple synthesis configurations for each SystemC module - Tradeoff latency and hardware area - Slow and small vs fast and large - Easy to find the best solution that meets your requirements  SDT00000748, TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems at 20:07 Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
[e] creating a layout of the circuit;	In order to manufacture the circuit, Stratus HLS users must create a layout of the circuit. Alternatively, Stratus HLS users may direct a third party, such as a foundry, to perform this step on their behalf.

US7971167	Stratus HLS
[f] generating a photomask based on the layout of the circuit;	In order to manufacture the circuit, Stratus HLS users must generate a photomask based on the layout of the circuit. Alternatively, Stratus HLS users may direct a third party, such as a foundry, to perform this step on their behalf.
[g] transferring a pattern of the photomask onto a semiconductor wafer; and	In order to manufacture the circuit, Stratus HLS users must transfer a pattern of the photomask onto a semiconductor wafer. Alternatively, Stratus HLS users may direct a third party, such as a foundry, to perform this step on their behalf.
[h] generating a correspondence table in which each block in the behavioral description corresponds to a state in the RTL description.	Stratus HLS users direct the Stratus HLS software to generate a correspondence table in which each block in the behavioral description corresponds to a state in the RTL description. Design Closure Stratus HLS ensures easy timing closure for the generated RTL by exhaustively analyzing each path and scheduling operation so they fit in the given clock period. Stratus HLS uses patented datapath optimization technology and the embedded Genus synthesis to build all datapath components, multiplexers, and registers in the specified technology library to get accurate timing and area models. The user can control how aggressively Stratus HLS packs these operations into each clock period. Creating designs with Stratus HLS can save months of back-end effort by preventing timing closure problems. Integration with Genus physical synthesis allows early visibility and feedback into likely congestion problems, allowing the front-end designer to avoid problems in the back-end. CDNS0005219-5223, Stratus HLS Datasheet at 5222. Stratus HLS supports graphical analysis of the RTL with links to source code.



SDT0000661-670, Reduce Time to Tapeout, <https://www.linkedin.com/pulse/how-dramatically-reduce-time-from-architecture-spec-tapeout-laviv/>.

Integrated with the Cadence verification suite, Stratus HLS supports automated mixed-language (SystemC and RTL) verification and debug including assertions, debugging, waveforms, and linkage back to the original SystemC design.

CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5221.

By turning on the `rtl_annotation` feature in Stratus™ HLS, the uncovered RTL code lines are easily traced back to the originating behavioral SystemC® code lines. This saves weeks of verification effort by quickly identifying weaknesses in the TB and/or possible bugs in the DUT, and also by allowing developers to debug the involved logic on the higher abstraction behavioral/algorithmic SystemC® model.

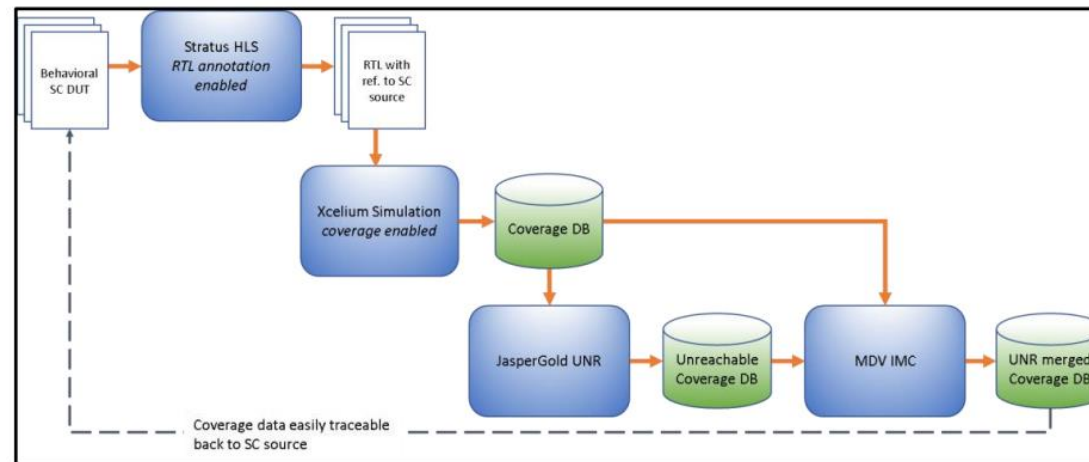


Figure 8: UNR analysis flow with traceability back to SC source

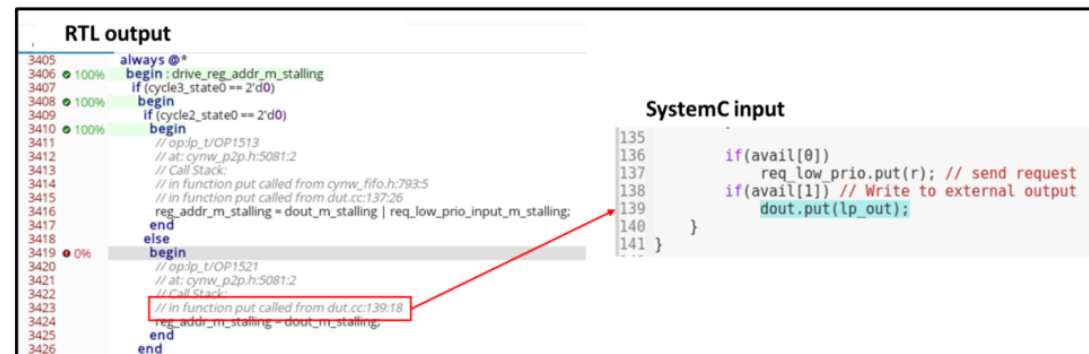
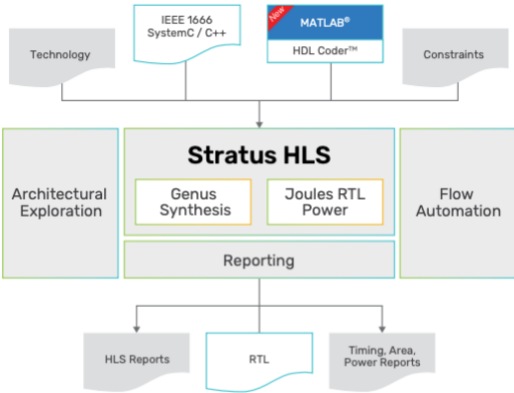


Figure 9: RTL annotation for traceability back to originating SystemC code lines

SDT00000671-678, Design-for-Verification, <https://dvcon-proceedings.org/wp-content/uploads/using-hls-to-improve-design-for-verification-of-multi-pipeline-designs-with-resource-sharing.pdf.pdf>.

See also "Timing, Area, Power Reports":

US7971167	Stratus HLS
	 <i>Figure 1: Stratus HLS uses the Genus synthesis and Joules power engines to create high-quality RTL targeted to your technology and design constraints</i> CDNS0005219-5223, Stratus HLS Datasheet at 5219. Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.
4. The manufacturing method according to claim 3, further comprising:	As shown above, Stratus HLS users perform the manufacturing method of claim 3.
referring to the result of the logic simulation and the correspondence table to calculate the latency in each block in the behavioral description.	Stratus HLS users direct the Stratus HLS software to refer to the result of the logic simulation and the correspondence table to calculate the latency in each block in the behavioral description. In Stratus HLS, “RTL is simulated with the same testbench used for the behavioral SystemC model.” “The RTL simulation ... verifies network performance (latency, throughput)” and “verifies functional equivalence between the behavior and synthesized RTL.”

Synthesis and RTL Simulation

- **Stratus™ HLS synthesizes each module to RTL**
 - Targets specified technology and clock period
 - Each module synthesized according to given settings and configurations
- **RTL is simulated with the same testbench used for the behavioral SystemC® model**
- **RTL simulation...**
 - Verifies network performance (latency, throughput)
 - Verifies functional equivalence between the behavior and synthesized RTL
 - Ensures blocks working together (e.g., no deadlock)
 - Generates waveforms for power analysis

[illegible]

15 © 2020 Cadence Design Systems, Inc. All rights reserved.

cādence®

SDT00000748, [TensorFlow to RTL with High-Level Synthesis – Cadence Design Systems](#) at 16:55

Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to describe functionality of the Accused Products fully and accurately with respect to this limitation and, accordingly, Plaintiff reserves the right to supplement these infringement contentions once such information is made available to Plaintiff.