

**Exhibit A-1: Infringement of U.S. Patent No. 7,603,636 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, 7, 8, and 13 under §271(a) by making and/or using an assertion generating system by installing Stratus HLS software on a computer-readable medium as part of a computer. For example, Cadence installs Stratus HLS software on computers it uses 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 1, 2; 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)" or "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, 7, 8, and 13 under §271(b) by inducing its customers to make and/or use an assertion generating system by installing the Stratus HLS Product on customer-provided computer-readable media for generating assertion descriptions for validation of a semiconductor IC. See, e.g., https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/synthesis/stratus-high-level-synthesis.html; 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 indirectly infringes claim 14 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. See also SDT00000690-693, https://www.cadence.com/en_US/home/training/all-courses/86170.html. 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 1, 7, 8, 13, and 14 under §271(c) by contributing to the infringement by its customers. Customers directly infringe these claims for the same reasons stated above in connection with §271(b). The Stratus HLS Product is a material part of an assertion generation system for generating system for generating assertion descriptions for validation of a semiconductor IC, as well as material to performing the claimed method. The portions of the Stratus HLS Product that map to and/or are involved in performing the steps recited in claims 1, 7, 8, 13, and 14 have no substantial non-infringing uses.

US7603636	Stratus HLS
1[pre]. An assertion generating system that generates an assertion description which is used for assertion verification of a semiconductor integrated circuit, comprising:	To the extent the preamble is limiting, a computer upon which Stratus HLS software is installed is an assertion generating system that generates an assertion description which is used for assertion verification of a semiconductor integrated circuit. ▸ 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 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.
[a] a specification inputting unit that generates design data or specifications and a document for confirming a specification of the semiconductor integrated circuit by graphically editing the specification of the semiconductor integrated circuit based on user operations;	The Stratus HLS system includes a specification inputting unit, including software and/or hardware that generates design data or specifications and a document for confirming a specification of the semiconductor integrated circuit by graphically editing the specification of the semiconductor integrated circuit based on user operations. Stratus HLS generates design data or specifications and a document for confirming a specification of the semiconductor integrated circuit. Stratus HLS allows users to graphically edit the specification of the semiconductor integrated circuit based on user operations. The graphics user interface (GUI) incorporates an integrated development environment (IDE). An integrated development environment typically includes a source-code editor, build automation tools, and a debugger. The CDNS0005219-5223, Stratus HLS Datasheet describes its GUI:

GUI

The Stratus GUI incorporates an IDE, making SystemC development easy and intuitive for new users and advanced users alike. In addition to typical IDE features, the Stratus IDE makes it easy to quickly create new models using pre-defined design templates to reduce design and debugging time.

The Stratus analysis environment includes SystemC and RTL source linking, control and dataflow graphs, schematic viewer, and pipeline analysis, as well as QoR reporting and visualization to judge the impact of architectural optimizations. Although most commonly used via the GUI, this analysis is also available via the Stratus Tcl API.

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

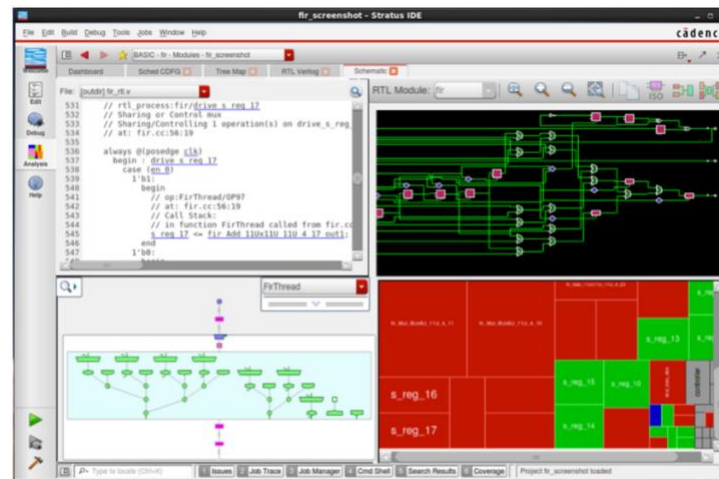
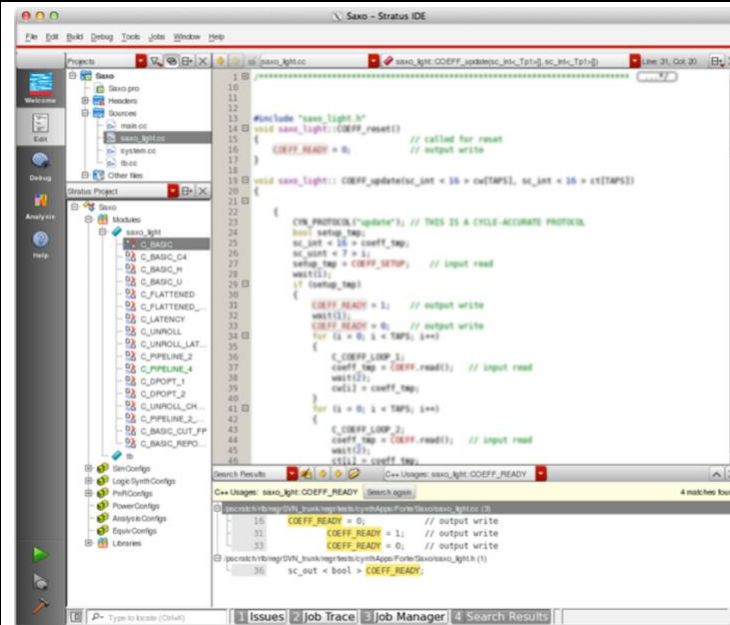
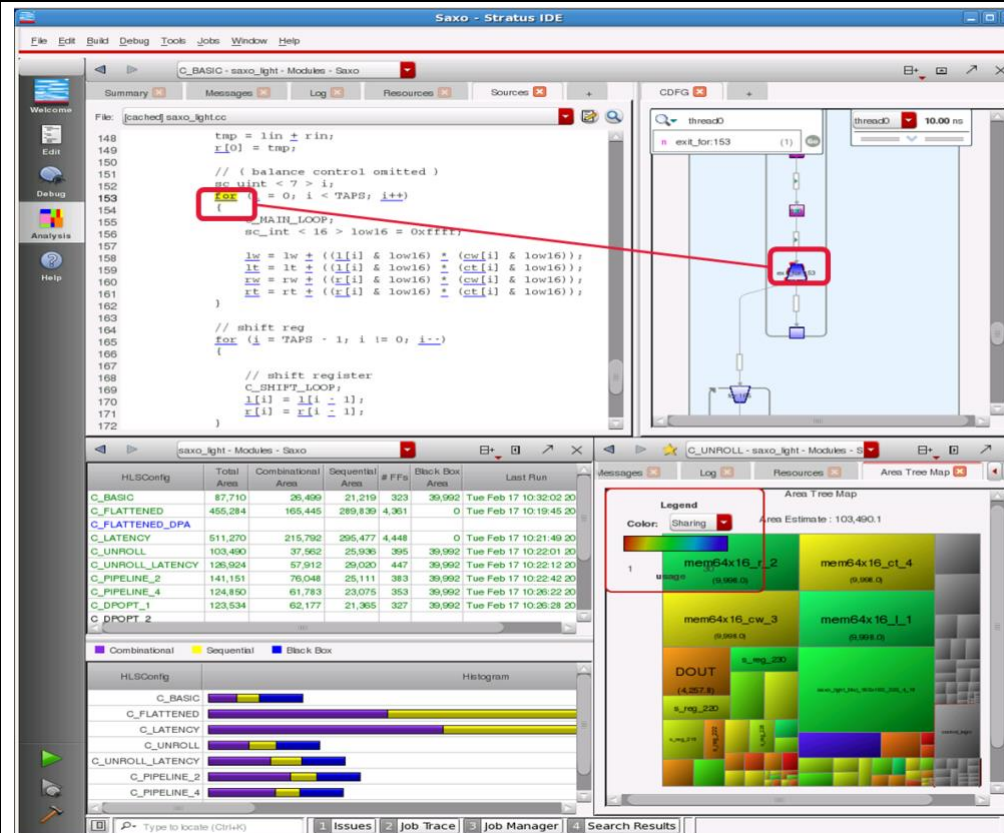


Figure 2: Complete graphical analysis with links to source code

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



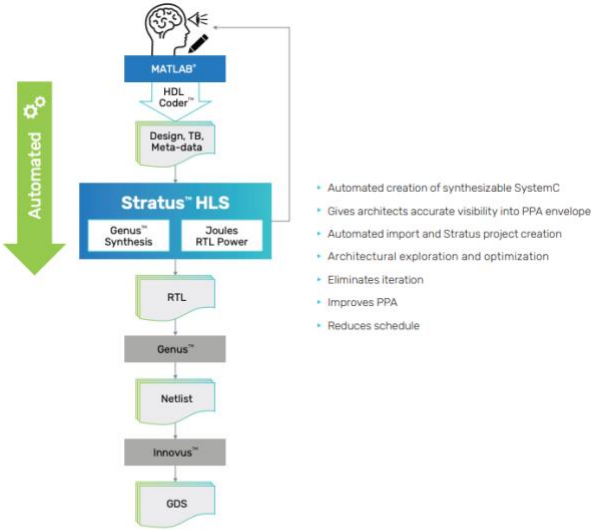
CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5221.

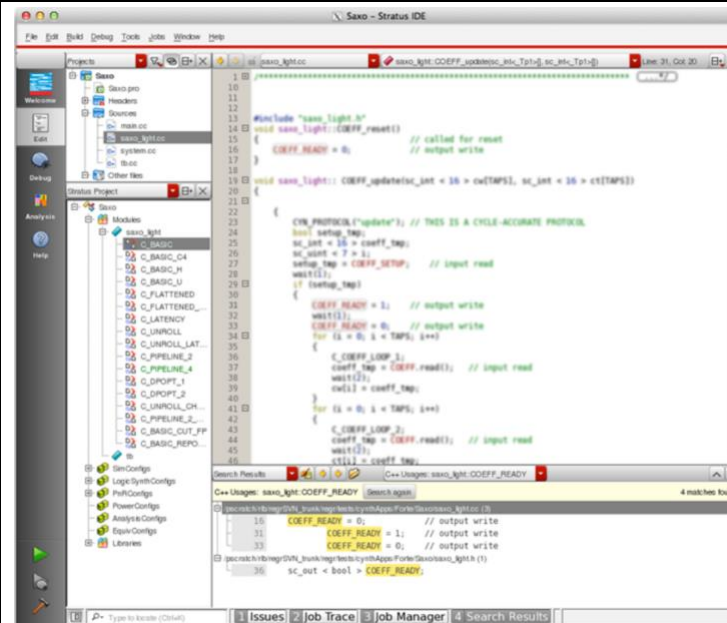


Tal Laviv, How to Dramatically Reduce Time from Architecture Spec to Tapeout, LinkedIn (Mar. 10, 2020), <https://www.linkedin.com/pulse/how-dramatically-reduce-time-from-architecture-spec-tapeout-laviv/>, SDT00000661-670 [hereinafter Reduce Time to Tapeout].

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.

US7603636	Stratus HLS
	 Figure 1: The integration allows algorithm developers to refine their algorithms with accurate data-driven PPA measurements that correlate well with downstream implementation CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5225. MATLAB is easy to use and contains comprehensive toolboxes for multiple application domains, so it is used widely for architectural exploration and algorithm development. However, we need register transfer language (RTL) for the hardware implementation of these algorithms in an IC or ASIC. So, it is required to convert the models implemented in high-level languages such as MATLAB, C/C++, and SystemC™ to RTL. CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5226. 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] a computer readable storage device having a first storing unit that stores the design data generated by the specification inputting unit;	The Stratus HLS system includes a computer readable storage device having a first storing unit, including hardware and/or software that stores the design data generated by the specification inputting unit.



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

An IDE would require a computer, which would have a computer readable storage device, such as a disk or flash drive.

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 property generating unit that generates a property which verifies the specification of the semiconductor integrated circuit by reading the design data generated by the specification inputting unit from the first storing unit and using the read design data;

The Stratus HLS system includes a property generating unit, including hardware and/or software that generates a property which verifies the specification of the semiconductor integrated circuit by reading the design data generated by the specification inputting unit from the first storing unit and using the read design data.

- 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.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the *synthesize_asserts* feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.



Figure 5: Assertion synthesis flow

S. Dahir, "Using HLS to Improve Design-for-Verification of Multi-Pipeline Designs with Resource Sharing," DVCON 2021 at 5-6, available <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].

Additionally, Jasper and Jasper BPS "Automatically classifies properties to drastically reduce the need for manual review of generated properties, automatically extracts properties from simulation traces—with or **without knowledge of the RTL**, automatically classifies the derived properties (assert/assume/cover), and Automatically ranks the derived properties as high-/medium-/low-value, and supports further qualification by leveraging our formal engines under-the-hood." SDT00000660, https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-gold-verification-platform/behavioral-property-synthesis-app.html.

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 second storing unit that stores the property generated by the property generating unit, wherein the second storing unit may or may not be a part of the computer readable storage device; and

The Stratus HLS system includes a second storing unit, including hardware and/or software that stores the property generated by the property generating unit, wherein the second storing unit may or may not be a part of the computer readable storage device.

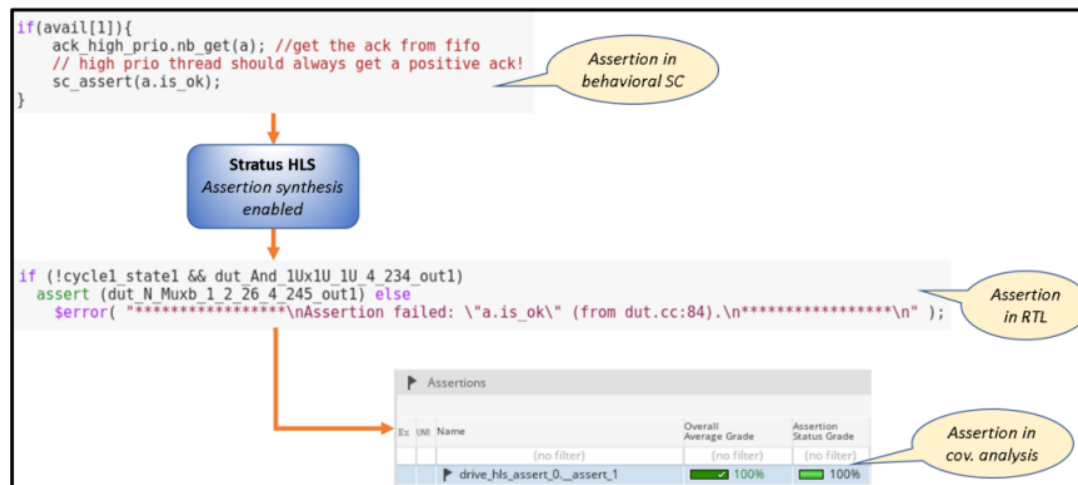
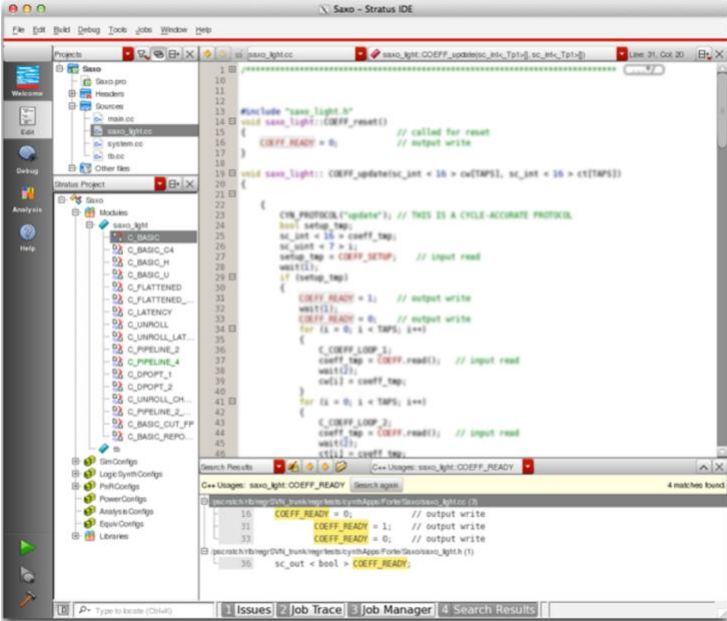
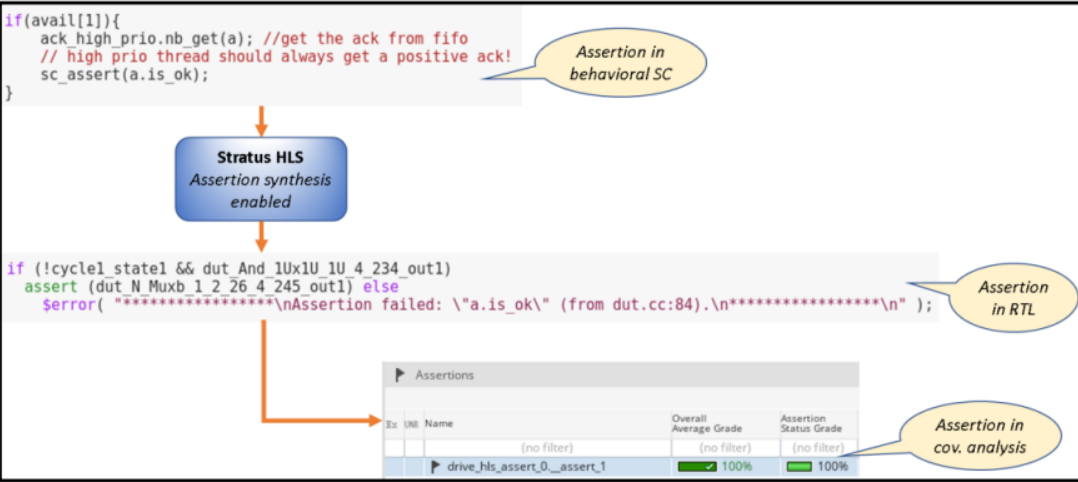
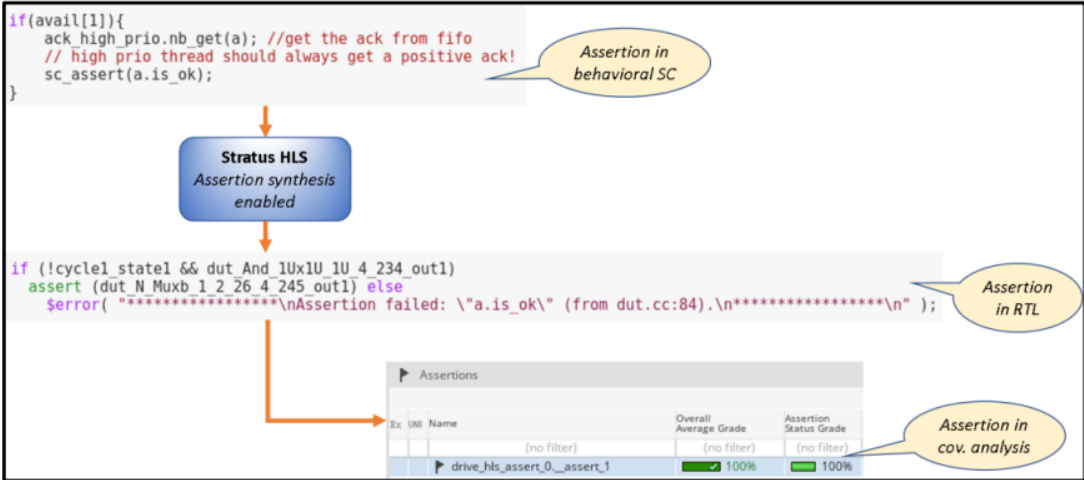


Figure 5: Assertion synthesis flow

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>.

US7603636	Stratus HLS
	 CDNS0005219-5223, Stratus HLS Datasheet at 5221. 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] an assertion generating unit that automatically converts the property into an assertion description if the property is to be verified during assertion verification by reading the property generated by the property generating unit from the second storing unit, wherein	The Stratus HLS system includes an assertion generating unit, including hardware and/or software that automatically converts the property into an assertion description if the property is to be verified during assertion verification by reading the property generated by the property generating unit from the second storing unit. ▸ 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.

US7603636	Stratus HLS
	 Figure 5: Assertion synthesis flow 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. 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.
[f] the property generated by the property generating unit is a selection condition with respect to a state transition, a logic value of at least one or more signals, or at least one or more signals in the design data.	In Stratus HLS, the property generated by the property generating unit is a selection condition with respect to a state transition, a logic value of at least one or more signals, or at least one or more signals in the design data. Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the <i>synthesize_asserts</i> feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.

US7603636	Stratus HLS
	 Figure 5: Assertion synthesis flow 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. Additionally, Jasper and Jasper BPS "Automatically classifies properties to drastically reduce the need for manual review of generated properties, automatically extracts properties from simulation traces—with or without knowledge of the RTL, automatically classifies the derived properties (assert/assume/cover), and Automatically ranks the derived properties as high-/medium-/low-value, and supports further qualification by leveraging our formal engines under-the-hood." SDT00000660, https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-gold-verification-platform/behavioral-property-synthesis-app.html. 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.
7. A circuit verifying system, comprising:	To the extent the preamble is limiting, a computer upon which Stratus HLS software is installed is a circuit verifying system. ▸ Synthesis of SystemC assertions and C++ asserts to SystemVerilog assertions (SVAs) CDNS0005219-5223, Stratus HLS Datasheet at 5220.

US7603636	Stratus HLS
	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.
[a] an assertion generating system that generates an assertion description which is used for assertion verification of a semiconductor integrated circuit, comprising:	The Stratus HLS system includes an assertion generating system that generates an assertion description which is used for assertion verification of a semiconductor integrated circuit. ▸ 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 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.
[a-1] a specification inputting unit that generates design data or specifications and a document for confirming a specification of the semiconductor integrated circuit by graphically editing the specification of the semiconductor integrated circuit based on user operations;	The assertion generating system of Stratus HLS includes a specification inputting unit, including hardware and/or software that generates design data or specifications and a document for confirming a specification of the semiconductor integrated circuit by graphically editing the specification of the semiconductor integrated circuit based on user operations. The graphics user interface (GUI) incorporates an integrated development environment (IDE). An integrated development environment typically includes a source-code editor, build automation tools, and a debugger. The CDNS0005219-5223, Stratus HLS Datasheet describes its GUI:

GUI

The Stratus GUI incorporates an IDE, making SystemC development easy and intuitive for new users and advanced users alike. In addition to typical IDE features, the Stratus IDE makes it easy to quickly create new models using pre-defined design templates to reduce design and debugging time.

The Stratus analysis environment includes SystemC and RTL source linking, control and dataflow graphs, schematic viewer, and pipeline analysis, as well as QoR reporting and visualization to judge the impact of architectural optimizations. Although most commonly used via the GUI, this analysis is also available via the Stratus Tcl API.

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

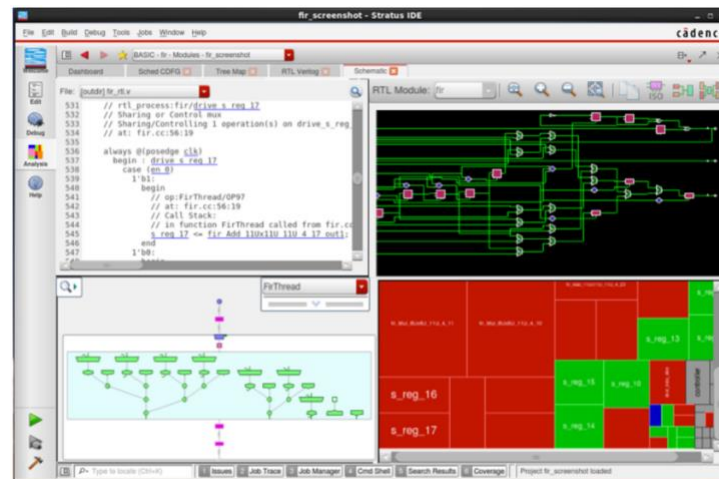
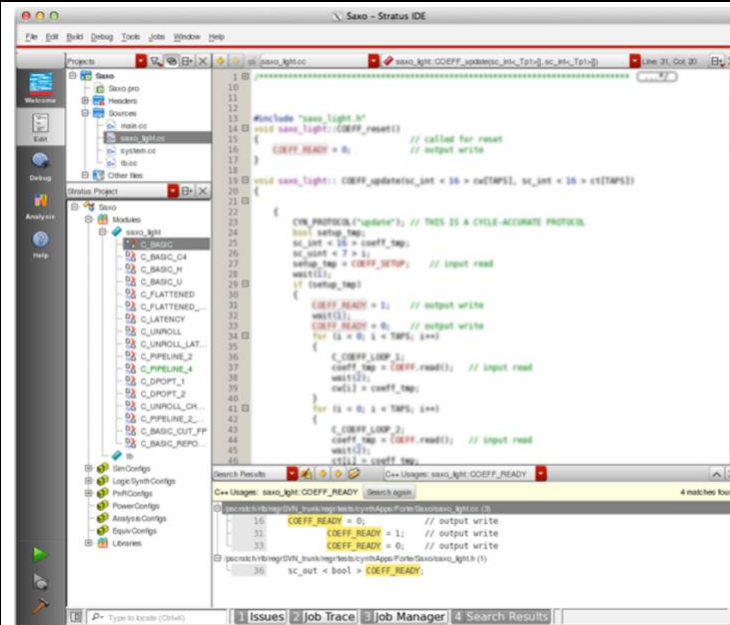
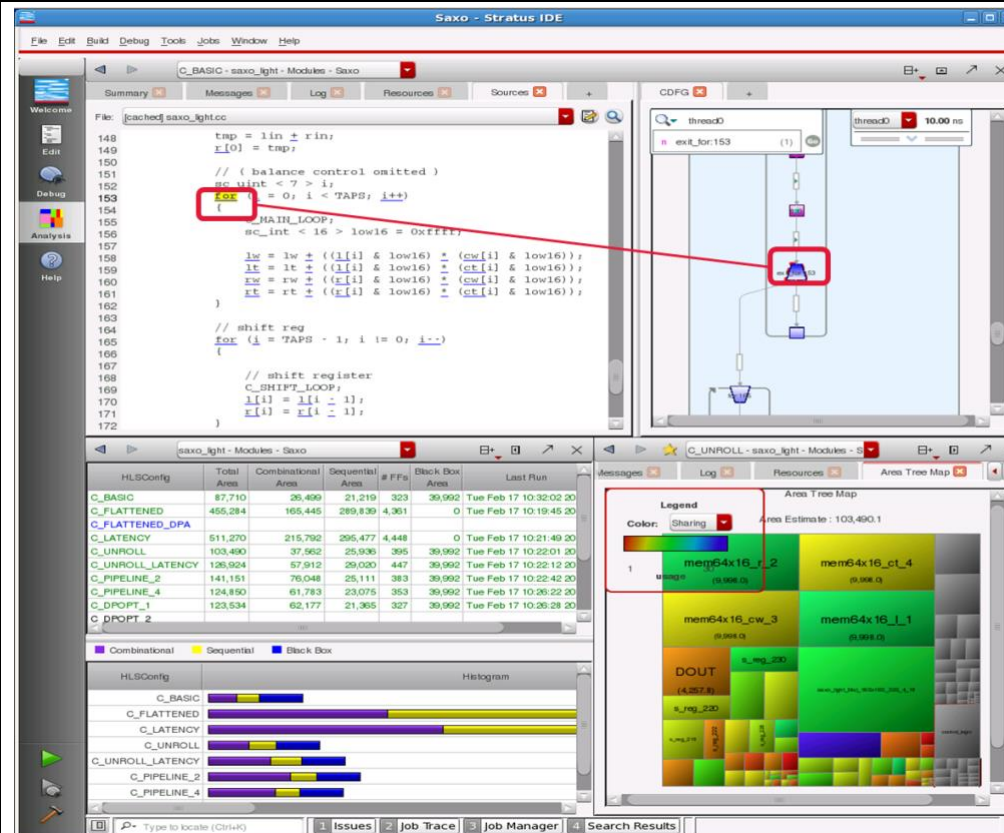


Figure 2: Complete graphical analysis with links to source code

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



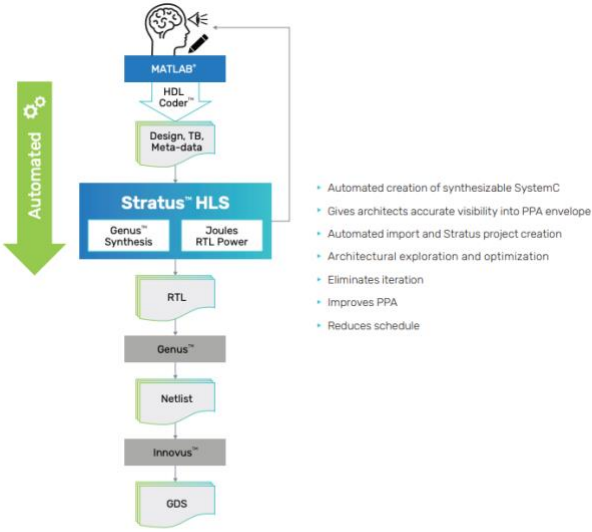
CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5221.

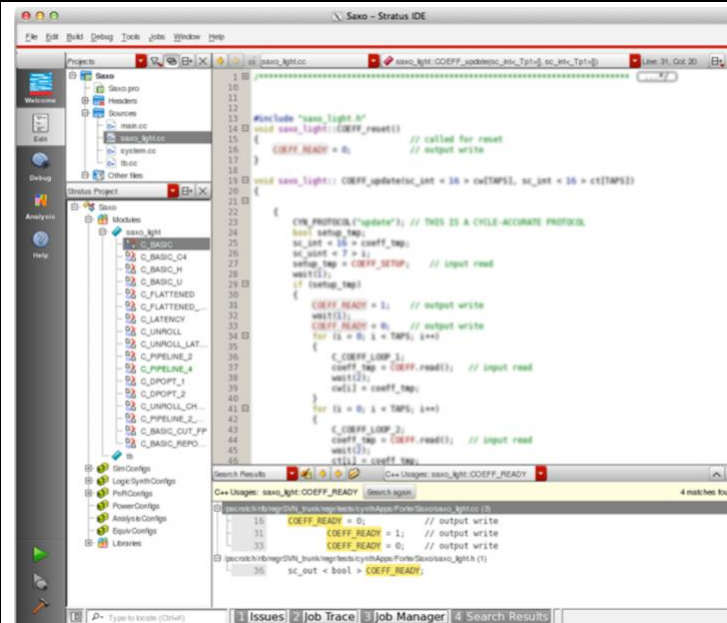


SDT00000661-670, Reduce Time to Tapeout, <https://www.linkedin.com/pulse/how-dramatically-reduce-time-from-architecture-spec-tapeout-laviv/>.

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.

US7603636	Stratus HLS
	 Figure 1: The integration allows algorithm developers to refine their algorithms with accurate data-driven PPA measurements that correlate well with downstream implementation CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5225. MATLAB is easy to use and contains comprehensive toolboxes for multiple application domains, so it is used widely for architectural exploration and algorithm development. However, we need register transfer language (RTL) for the hardware implementation of these algorithms in an IC or ASIC. So, it is required to convert the models implemented in high-level languages such as MATLAB, C/C++, and SystemC™ to RTL. CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5226. 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.
[a-2] a computer readable storage device having a first storing unit that stores the design data generated by the specification inputting unit;	The assertion generating system of Stratus HLS includes a computer readable storage device having a first storing unit that stores the design data generated by the specification inputting unit.



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

An IDE would require a computer, which would have a computer readable storage device, such as a disk or flash drive.

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.

[a-3] a property generating unit that generates a property which verifies the specification of the semiconductor integrated circuit by reading the design data generated by the specification inputting unit from the first storing unit and using the read design data;

The assertion generating system of Stratus HLS includes a property generating unit, including hardware and/or software that generates a property which verifies the specification of the semiconductor integrated circuit by reading the design data generated by the specification inputting unit from the first storing unit and using the read design data.

- 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.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the *synthesize_asserts* feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.

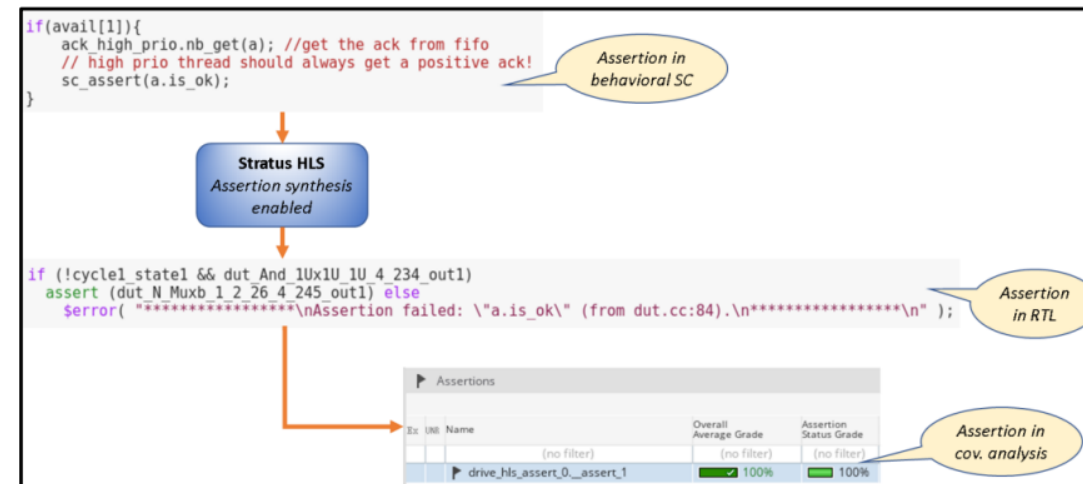


Figure 5: Assertion synthesis flow

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>.

Additionally, Jasper and Jasper BPS "Automatically classifies properties to drastically reduce the need for manual review of generated properties, automatically extracts properties from simulation traces—with or **without knowledge of the RTL**, automatically classifies the derived properties (assert/assume/cover), and Automatically ranks the derived properties as high-/medium-/low-value, and supports further qualification by leveraging our formal engines under-the-hood." SDT00000660, https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-gold-verification-platform/behavioral-property-synthesis-app.html.

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.

[a-4] a second storing unit that stores the property generated by the property generating unit, wherein the second storing unit may or may not be a part

The assertion generating system of Stratus HLS includes a second storing unit that, including hardware and/or software stores the property generated by the property generating unit, wherein the second storing unit may or may not be a part of the computer readable storage device.

of the computer readable storage device; and

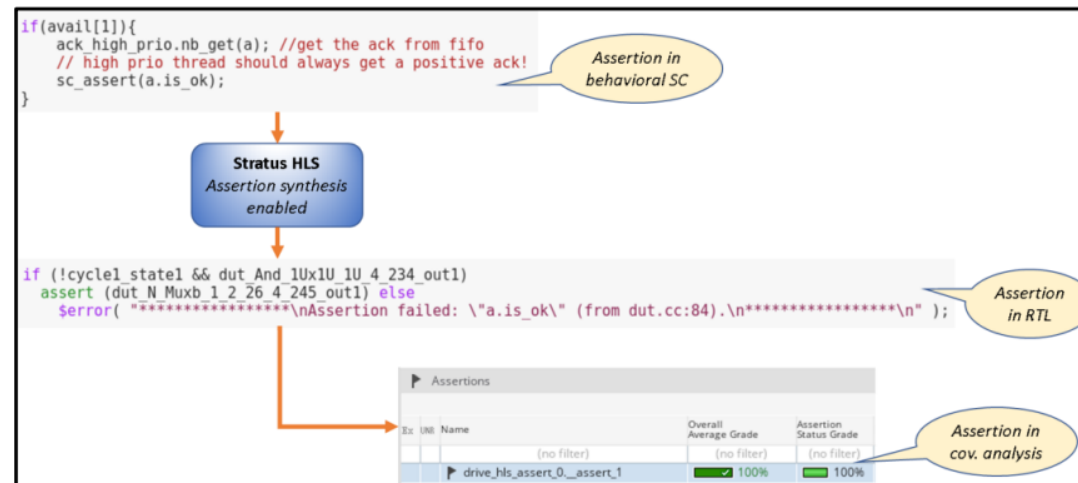
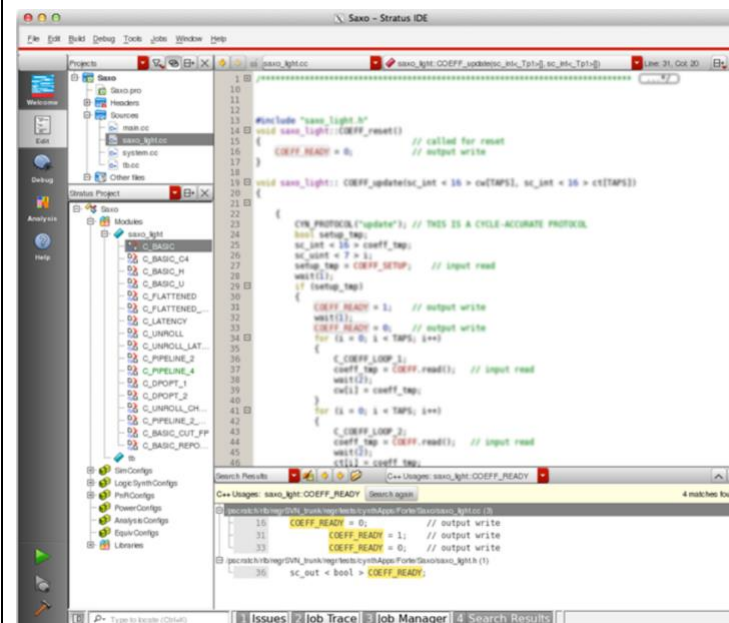
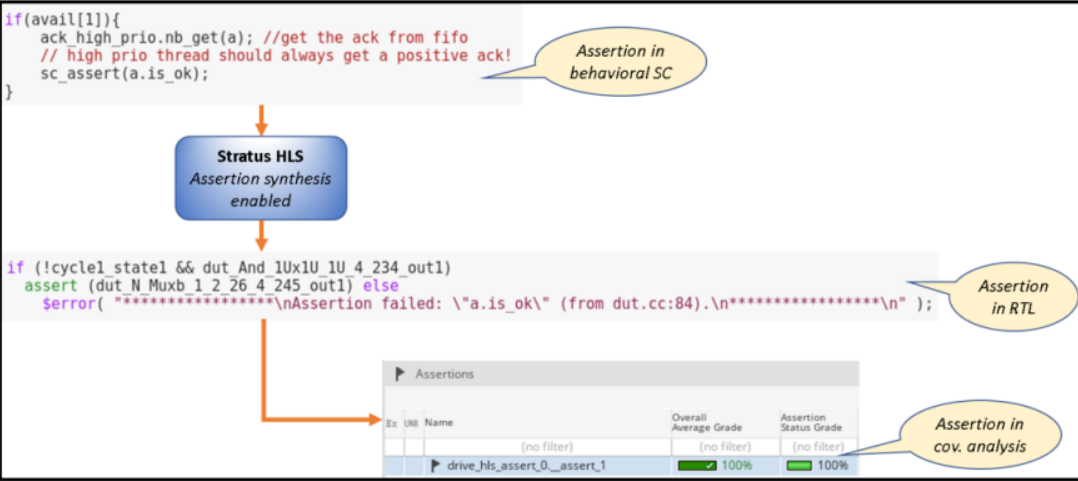


Figure 5: Assertion synthesis flow

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>.



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

US7603636	Stratus HLS
	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.
[a-5] an assertion generating unit that automatically converts the property into an assertion description if the property is to be verified during assertion verification by reading the property generated by the property generating unit from the second storing unit, wherein	The assertion generating system of Stratus HLS includes an assertion generating unit, including hardware and/or software that automatically converts the property into an assertion description if the property is to be verified during assertion verification by reading the property generated by the property generating unit from the second storing unit. ► 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.  Figure 5: Assertion synthesis flow 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. 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] the property generated by the property generating unit is a selection condition with respect to a state transition, a logic value of at least one or more signals, or at least one or more signals in the design data; and

In Stratus HLS, the property generated by the property generating unit is a selection condition with respect to a state transition, a logic value of at least one or more signals, or at least one or more signals in the design data.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the *synthesize_asserts* feature. This allows the designer to communicate effectively and clearly the design intend and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.

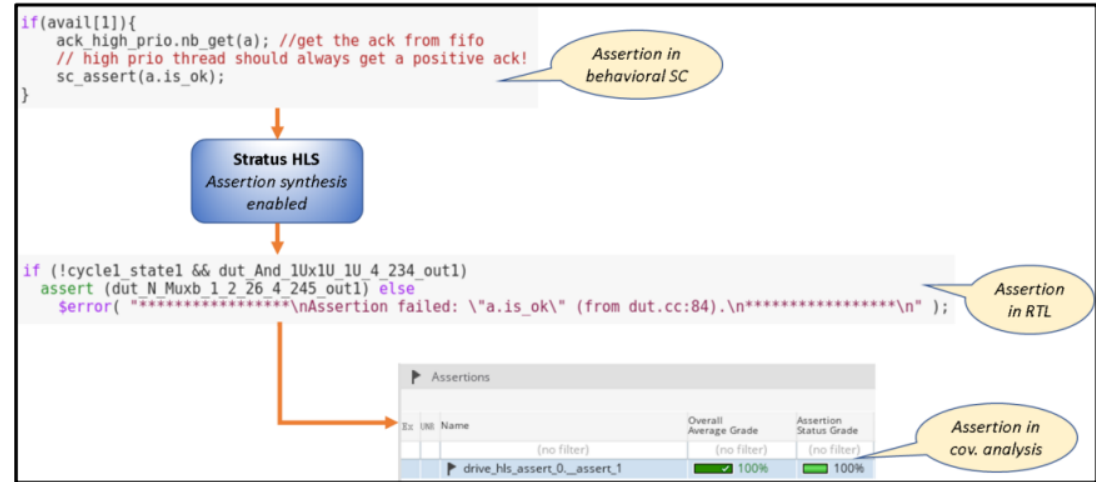


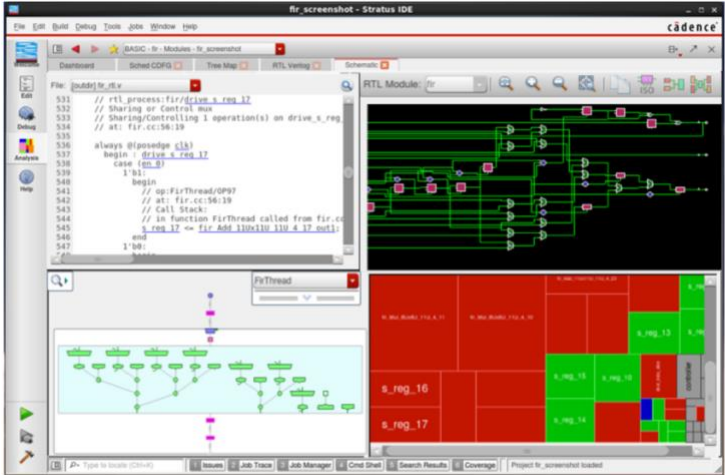
Figure 5: Assertion synthesis flow

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>.

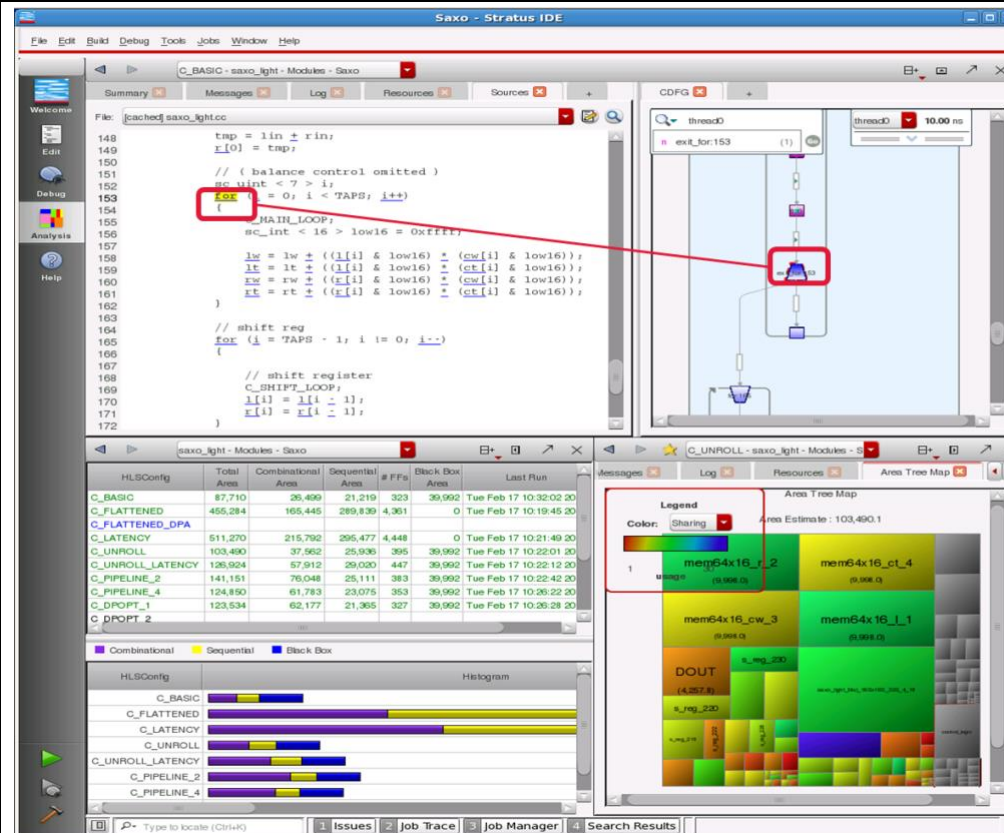
Additionally, Jasper and Jasper BPS "Automatically classifies properties to drastically reduce the need for manual review of generated properties, automatically extracts properties from simulation traces—with or **without knowledge of the RTL**, automatically classifies the derived properties (assert/assume/cover), and Automatically ranks the derived properties as high-/medium-/low-value, and supports further qualification by leveraging our formal engines under-the-hood." SDT00000660, https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-gold-verification-platform/behavioral-property-synthesis-app.html.

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.

US7603636	Stratus HLS
[c] wherein the circuit verifying system executes assertion verification of the semiconductor integrated circuit by using the assertion description generated by the assertion generating system.	In Stratus HLS, the circuit verifying system executes assertion verification of the semiconductor integrated circuit by using the assertion description generated by the assertion generating system. ▸ 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 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.
8. A computer readable medium encoded with a program for causing a computer provided in an assertion generating system, that is capable of generating an assertion description for assertion verification of a semiconductor integrated circuit, to execute:	To the extent the preamble is limiting, a computer readable medium encoded with Stratus HLS software satisfies this language. Stratus HLS causes a computer provided in an assertion generating system to generate an assertion description. ▸ 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 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.
[a-1] a specification inputting step that generates design data of the semiconductor integrated circuit by graphically editing a specification of	The computer upon which Stratus HLS is installed executes a specification inputting step that generates design data of the semiconductor integrated circuit by graphically editing a specification of the semiconductor integrated circuit based on user operations.

US7603636	Stratus HLS
the semiconductor integrated circuit based on user operations and	The graphics user interface (GUI) incorporates an integrated development environment (IDE). An integrated development environment typically includes a source-code editor, build automation tools, and a debugger. The Stratus HLS Datasheet describes its GUI: GUI The Stratus GUI incorporates an IDE, making SystemC development easy and intuitive for new users and advanced users alike. In addition to typical IDE features, the Stratus IDE makes it easy to quickly create new models using pre-defined design templates to reduce design and debugging time. The Stratus analysis environment includes SystemC and RTL source linking, control and dataflow graphs, schematic viewer, and pipeline analysis, as well as QoR reporting and visualization to judge the impact of architectural optimizations. Although most commonly used via the GUI, this analysis is also available via the Stratus Tcl API. CDNS0005219-5223, Stratus HLS Datasheet at 5221.  The screenshot displays the Stratus IDE interface. On the left, a source code editor shows Verilog code for a module named 'fir'. The top right pane shows the 'RTL Module' view with a block diagram of the FIR filter architecture. The bottom right pane shows a graphical analysis, likely a resource utilization or QoR report, with a color-coded block diagram. The bottom left pane shows a 'FIRThread' control flow graph. The interface includes a menu bar, a toolbar, and a status bar at the bottom. <i>Figure 2: Complete graphical analysis with links to source code</i> CDNS0005219-5223, Stratus HLS Datasheet at 5220.

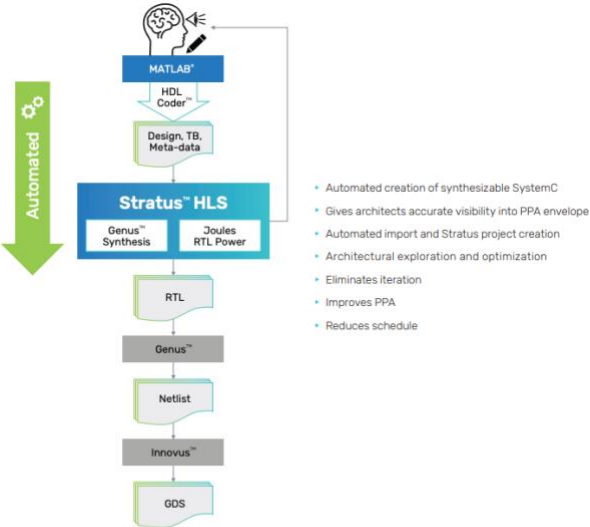


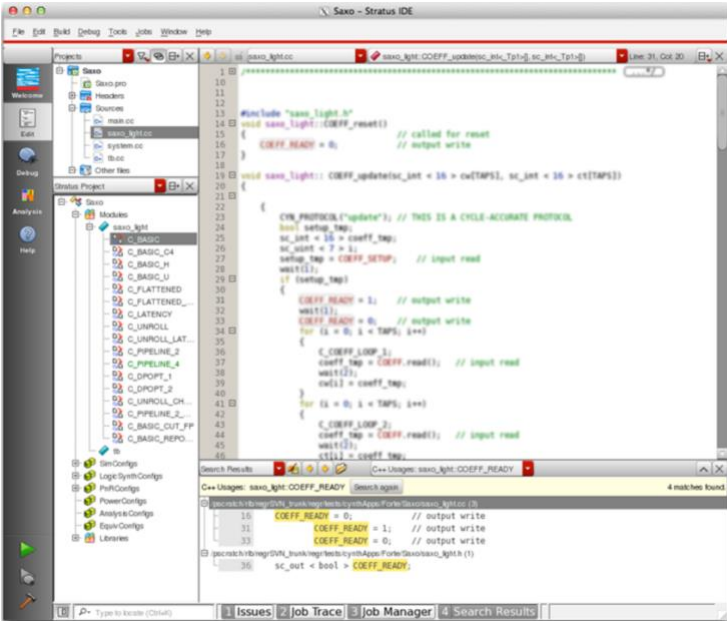


SDT00000661-670, Reduce Time to Tapeout, <https://www.linkedin.com/pulse/how-dramatically-reduce-time-from-architecture-spec-tapeout-laviv/>.

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.

US7603636	Stratus HLS
	 Figure 1: The integration allows algorithm developers to refine their algorithms with accurate data-driven PPA measurements that correlate well with downstream implementation CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5225. MATLAB is easy to use and contains comprehensive toolboxes for multiple application domains, so it is used widely for architectural exploration and algorithm development. However, we need register transfer language (RTL) for the hardware implementation of these algorithms in an IC or ASIC. So, it is required to convert the models implemented in high-level languages such as MATLAB, C/C++, and SystemC™ to RTL. CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5226. 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.
[a-2] inputs the design data in storage;	As part of the specification inputting step, the computer upon which Stratus HLS is installed inputs the design data in storage.

US7603636	Stratus HLS
	 CDNS0005219-5223, Stratus HLS Datasheet at 5221. An IDE would require a computer, which would have a computer readable storage device, such as a disk or flash drive. 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-1] a property generating step that reads the design data generated at the specification inputting step from the storage and	The computer upon which Stratus HLS is installed executes a property generating step that reads the design data generated at the specification inputting step from the storage. ► 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.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the `synthesize_asserts` feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.



Figure 5: Assertion synthesis flow

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>.

Additionally, Jasper and Jasper BPS "Automatically classifies properties to drastically reduce the need for manual review of generated properties, automatically extracts properties from simulation traces—with or **without knowledge of the RTL**, automatically classifies the derived properties (assert/assume/cover), and Automatically ranks the derived properties as high-/medium-/low-value, and supports further qualification by leveraging our formal engines under-the-hood." SDT00000660, https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-gold-verification-platform/behavioral-property-synthesis-app.html.

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-2] generates a property which verifies the specification of the

As part of the property generating step, the computer upon which Stratus HLS is installed generates a property which verifies the specification of the semiconductor integrated circuit using the read design data.

US7603636	Stratus HLS
semiconductor integrated circuit using the read design data and	Stratus HLS automates the design and verification flow from transaction-level modeling (TLM) to gates. The architecture exploration involves proposing changes to the design constraints, optimization, simulation, and results analysis to achieve the desired PPA. At a high level, it comprises the below exploration steps, yielding multiple architectures in an effort to find an implementation that best meets design requirements: ▸ Define and simulate the high-level description in C++ or SystemC ▸ Apply HLS and synthesis constraints for optimization and RTL generation CDNS0005219-5223, Stratus HLS Datasheet at 5220. 2. Customize the Setup: Stratus HLS' Behavioral Design Workbench (BDW) import function reads the design, testbench, and metadata and creates a Stratus HLS project. The user configures the Stratus HLS project by specifying the technology library, design constraints, and exploration settings. 3. Exploration: Once configured, Stratus HLS explores the solution space. Exploration is governed by a Stratus HLS Tcl control file that defines how constraints are changed and imposed on the candidate micro-architecture designs. Accelerate the Algorithm to Silicon Development with Stratus HLS (cadence.com) at 5. ▸ 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. Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the <i>synthesize_asserts</i> feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.

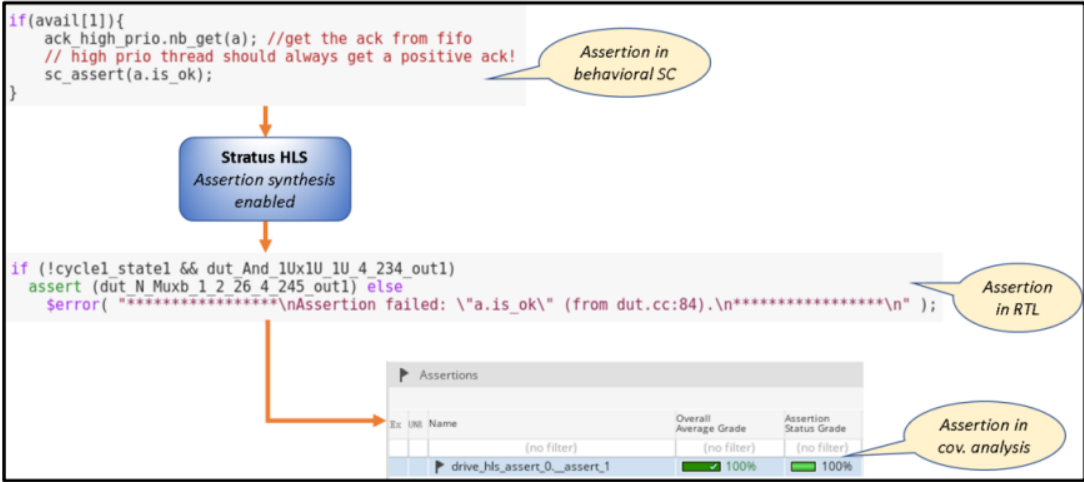


Figure 5: Assertion synthesis flow

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>. 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-3] inputs the property in the storage; and

As part of the property generating step, the computer upon which Stratus HLS is installed inputs the property in the storage.

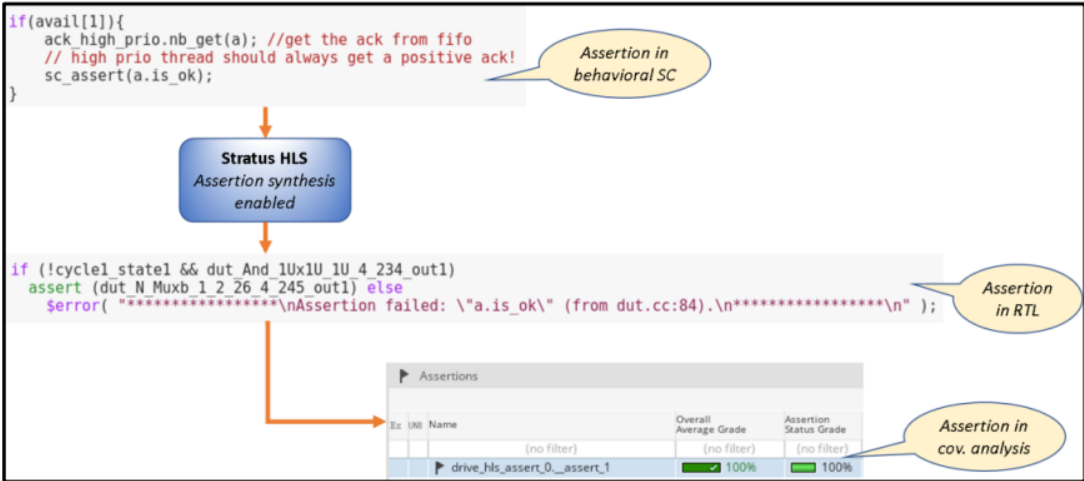
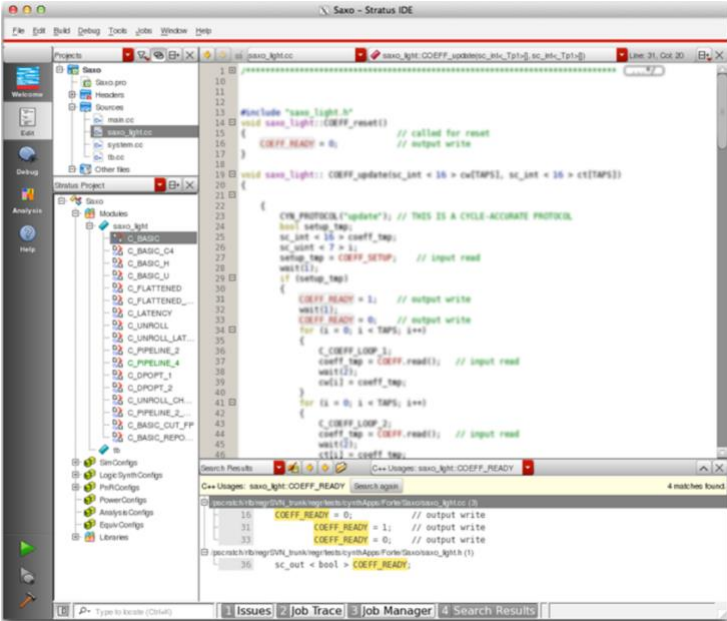
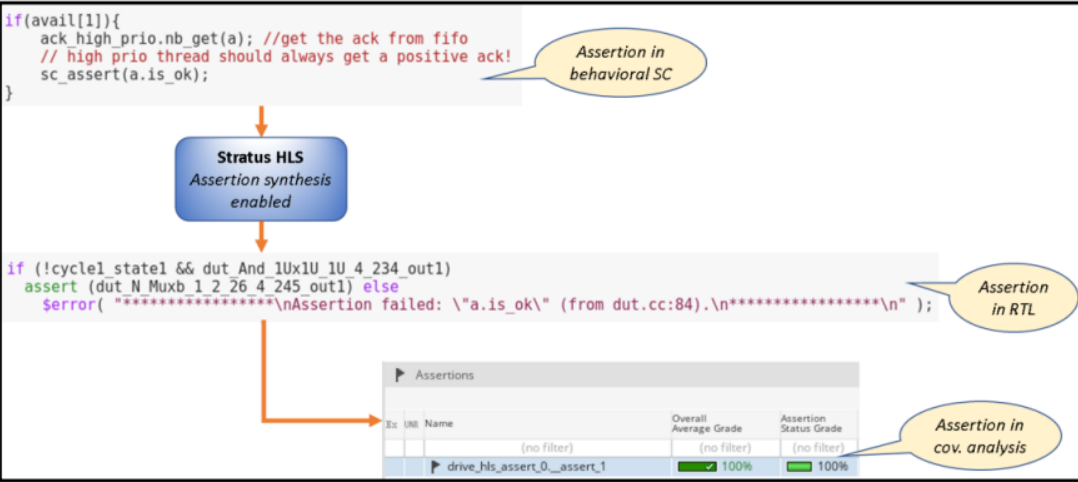


Figure 5: Assertion synthesis flow

US7603636	Stratus HLS
	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.  CDNS0005219-5223, Stratus HLS Datasheet at 5221. 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-1] an assertion generating step that reads the property generated at the property generating step from the storage and	The computer upon which Stratus HLS is installed executes an assertion generating step that reads the property generated at the property generating step from the storage.

US7603636	Stratus HLS
	 Figure 5: Assertion synthesis flow 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. Cadence confidential information (e.g., product documentation, source code, etc.) may be necessary to fully and accurately describe functionality of the Accused Products 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-2] automatically converts the property into an assertion description if the property is to be verified during assertion verification,	As part of the assertion generating step, the computer upon which Stratus HLS is installed automatically converts the property into an assertion description if the property is to be verified during assertion verification. ► 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.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the *synthesize_asserts* feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.



Figure 5: Assertion synthesis flow

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>.

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-1] wherein the property generated by the property generating step is a selection condition with respect to a state transition, a logic value of at least one or more signals, or at least one or more signals in the design data.

Properties and assertions for describing the design include selection conditions with respect to state transitions, logic values, and signals in the design data.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the *synthesize_asserts* feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.



Figure 5: Assertion synthesis flow

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>.

Additionally, Jasper and Jasper BPS "Automatically classifies properties to drastically reduce the need for manual review of generated properties, automatically extracts properties from simulation traces—with or **without knowledge of the RTL**, automatically classifies the derived properties (assert/assume/cover), and Automatically ranks the derived properties as high-/medium-/low-value, and supports further qualification by leveraging our formal engines under-the-hood."

SDT00000660, https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-gold-verification-platform/behavioral-property-synthesis-app.html.

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.

13. The computer readable medium as claimed in claim 8, wherein the specification inputting step expands the design data into a graphic structure and inputs the graphic structure in the storage.

As part of the specification inputting step, Stratus HLS expands the design data into a graphic structure and inputs the graphic structure in the storage.

GUI

The Stratus GUI incorporates an IDE, making SystemC development easy and intuitive for new users and advanced users alike. In addition to typical IDE features, the Stratus IDE makes it easy to quickly create new models using pre-defined design templates to reduce design and debugging time.

The Stratus analysis environment includes SystemC and RTL source linking, control and dataflow graphs, schematic viewer, and pipeline analysis, as well as QoR reporting and visualization to judge the impact of architectural optimizations. Although most commonly used via the GUI, this analysis is also available via the Stratus Tcl API.

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

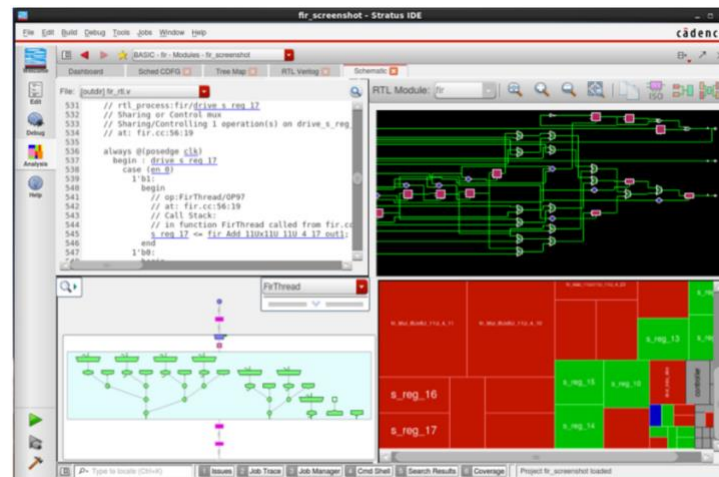
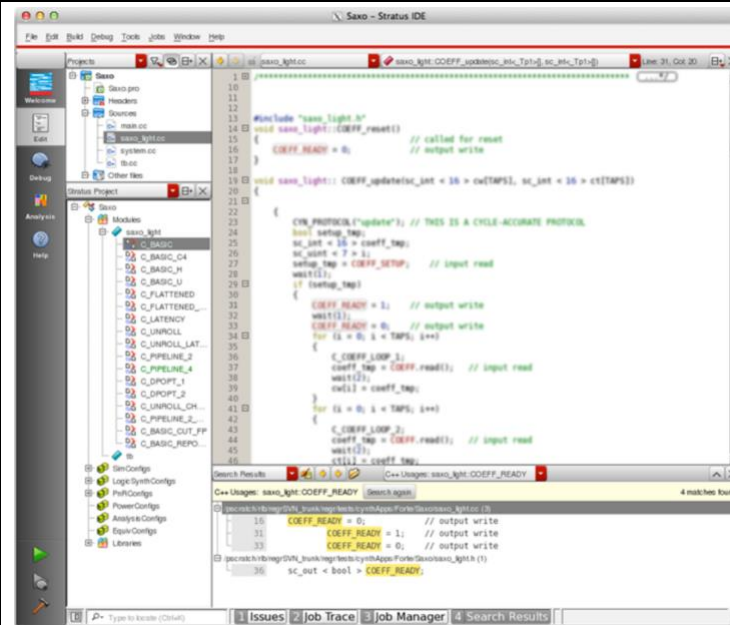
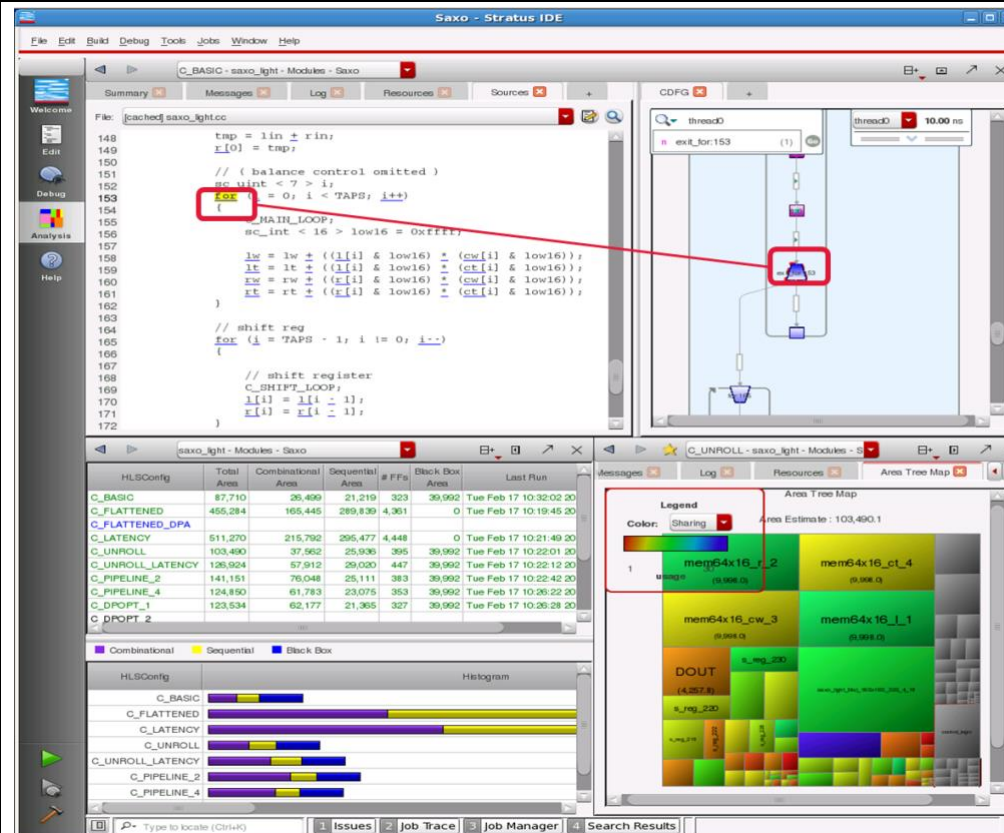


Figure 2: Complete graphical analysis with links to source code

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



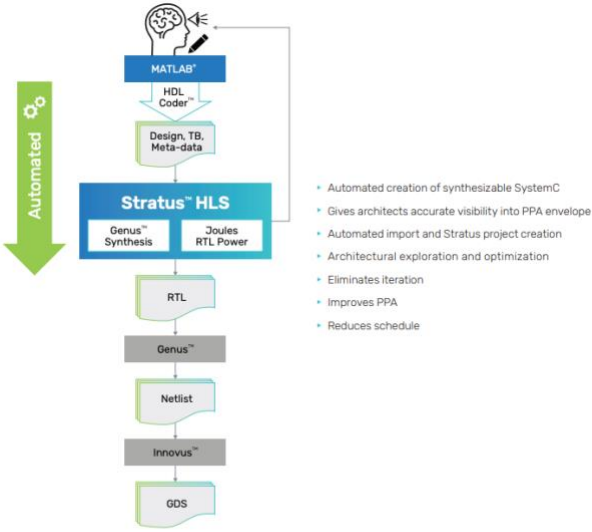
CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5221.



SDT00000661-670, 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.

US7603636	Stratus HLS
	 Figure 1: The integration allows algorithm developers to refine their algorithms with accurate data-driven PPA measurements that correlate well with downstream implementation CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5225. MATLAB is easy to use and contains comprehensive toolboxes for multiple application domains, so it is used widely for architectural exploration and algorithm development. However, we need register transfer language (RTL) for the hardware implementation of these algorithms in an IC or ASIC. So, it is required to convert the models implemented in high-level languages such as MATLAB, C/C++, and SystemC™ to RTL. CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5226. 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.
14 [pre]. A manufacturing method of a semiconductor device, comprising:	To the extent the preamble is limiting, Stratus HLS users perform a manufacturing method for manufacturing a semiconductor integrated circuit. Users produce and verify an RTL design using Stratus HLS, and may direct a fab to perform the actual manufacturing steps.

US7603636	Stratus HLS
	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 1.
[a] a designing step that designs an integrated circuit having a predetermined function;	Stratus HLS users perform a designing step that designs an integrated circuit having a predetermined function. 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 CDNS0005219. With Stratus HLS, engineering teams can quickly design and verify high-quality RTL implementations from abstract IEEE 1666 synthesizable SystemC, C, C++, or MATLAB models. The models can be easily created using the Stratus integrated design environment (IDE). Stratus synthesizable SystemC IP provides simulation and synthesis models for common bus-based and point-to-point communication protocols as well as common mathematical operations and datatypes. 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.

US7603636	Stratus HLS
[b] a simulating step that simulates the integrated circuit by using an assertion generated by an assertion generating method; and	Stratus HLS users perform a simulating step that simulates the integrated circuit by using an assertion generated by an assertion generating method. ▶ 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. ▶ 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 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 manufacturing step that manufactures a semiconductor device based on specifications of the integrated circuit, and	Stratus HLS users perform (or direct a foundry or fab to perform) a manufacturing step that manufactures a semiconductor device based on specifications of the integrated circuit. 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] wherein the assertion generating method comprises:	Stratus HLS users perform the assertion generating method, as described below.
[d-1] a specification inputting step that generates design data of the semiconductor integrated circuit by graphically editing a specification of the semiconductor integrated circuit based on user operations and inputs the design data in storage;	Stratus HLS users perform a specification inputting step that generates design data of the semiconductor integrated circuit by graphically editing a specification of the semiconductor integrated circuit based on user operations and inputs the design data in storage. The graphics user interface (GUI) incorporates an integrated development environment (IDE). An integrated development environment typically includes a source-code editor, build automation tools, and a debugger. The Stratus HLS Datasheet describes its GUI:

GUI

The Stratus GUI incorporates an IDE, making SystemC development easy and intuitive for new users and advanced users alike. In addition to typical IDE features, the Stratus IDE makes it easy to quickly create new models using pre-defined design templates to reduce design and debugging time.

The Stratus analysis environment includes SystemC and RTL source linking, control and dataflow graphs, schematic viewer, and pipeline analysis, as well as QoR reporting and visualization to judge the impact of architectural optimizations. Although most commonly used via the GUI, this analysis is also available via the Stratus Tcl API.

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

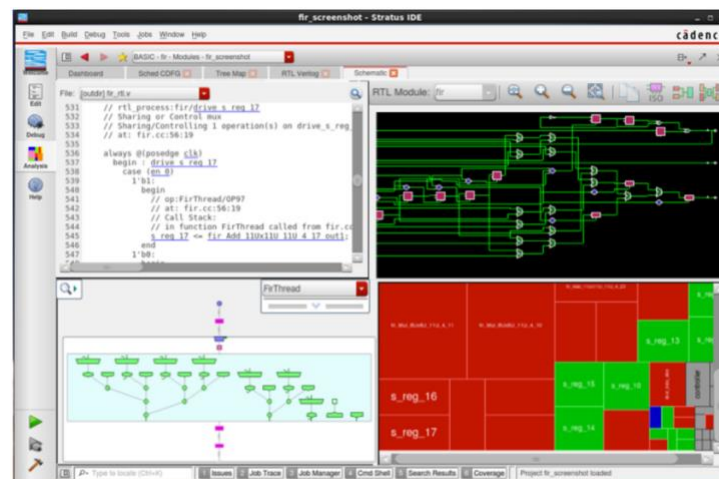
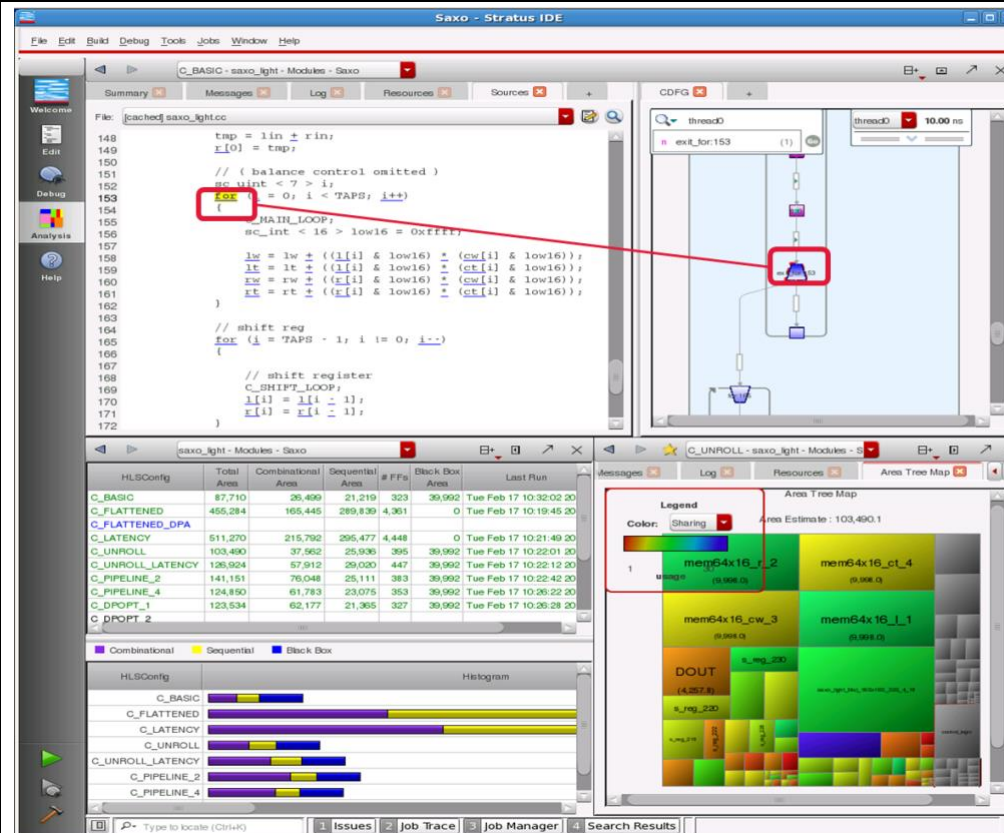


Figure 2: Complete graphical analysis with links to source code

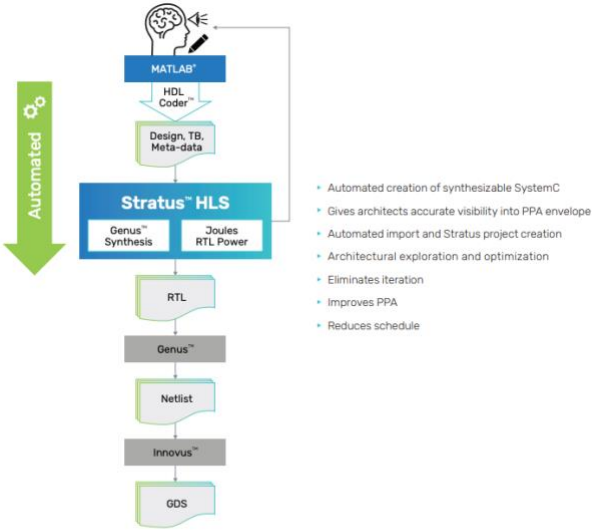
CDNS0005219-5223, [Stratus HLS Datasheet](#) at 5220.



SDT00000661-670, 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.

US7603636	Stratus HLS
	 Figure 1: The integration allows algorithm developers to refine their algorithms with accurate data-driven PPA measurements that correlate well with downstream implementation CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5225. MATLAB is easy to use and contains comprehensive toolboxes for multiple application domains, so it is used widely for architectural exploration and algorithm development. However, we need register transfer language (RTL) for the hardware implementation of these algorithms in an IC or ASIC. So, it is required to convert the models implemented in high-level languages such as MATLAB, C/C++, and SystemC™ to RTL. CDNS0005224-5229, Accelerate Algorithm to Silicon Development with Stratus HLS at 5226. 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-2] a property generating step that reads the design data generated at the specification inputting step from the storage and generates a property which verifies the specification of the semiconductor integrated circuit using the read design data and inputs the property in the storage; and	Stratus HLS users perform a property generating step that reads the design data generated at the specification inputting step from the storage and generates a property which verifies the specification of the semiconductor integrated circuit using the read design data and inputs the property in the storage. ▸ 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.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the *synthesize_asserts* feature. This allows the designer to communicate effectively and clearly the design intent and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.



Figure 5: Assertion synthesis flow

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>.

Additionally, Jasper and Jasper BPS "Automatically classifies properties to drastically reduce the need for manual review of generated properties, automatically extracts properties from simulation traces—with or **without knowledge of the RTL**, automatically classifies the derived properties (assert/assume/cover), and Automatically ranks the derived properties as high-/medium-/low-value, and supports further qualification by leveraging our formal engines under-the-hood." SDT00000660, https://www.cadence.com/en_US/home/tools/system-design-and-verification/formal-and-static-verification/jasper-gold-verification-platform/behavioral-property-synthesis-app.html.

US7603636	Stratus HLS																				
	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-3] an assertion generating step that reads the property generated at the property generating step from the storage and automatically converts the property into an assertion description if the property is to be verified during assertion verification, wherein	Stratus HLS users perform an assertion generating step that reads the property generated at the property generating step from the storage and automatically converts the property into an assertion description if the property is to be verified during assertion verification. ► 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. <pre>if(avail[1]){ ack_high_prio.nb_get(a); //get the ack from fifo // high prio thread should always get a positive ack! sc_assert(a.is_ok); }</pre> Assertion in behavioral SC Stratus HLS Assertion synthesis enabled <pre>if (!cycle1 state1 && dut_And_1Ux1U_1U_4_234_out1) assert (dut_N_Muxb_1_2_26_4_245_out1) else \$error("*****\nAssertion failed: \"a.is_ok\" (from dut.cc:84).\n*****\n");</pre> Assertion in RTL <table><tr><th colspan="5">Assertions</th></tr><tr><th>Ex</th><th>Unit</th><th>Name</th><th>Overall Average Grade</th><th>Assertion Status Grade</th></tr><tr><td></td><td></td><td>(no filter)</td><td>(no filter)</td><td>(no filter)</td></tr><tr><td>▶</td><td></td><td>drive_hls_assert_0_assert_1</td><td>100%</td><td>100%</td></tr></table> Assertion in cov. analysis Figure 5: Assertion synthesis flow 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. 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.	Assertions					Ex	Unit	Name	Overall Average Grade	Assertion Status Grade			(no filter)	(no filter)	(no filter)	▶		drive_hls_assert_0_assert_1	100%	100%
Assertions																					
Ex	Unit	Name	Overall Average Grade	Assertion Status Grade																	
		(no filter)	(no filter)	(no filter)																	
▶		drive_hls_assert_0_assert_1	100%	100%																	

[d-4] the property generated by the property generating step is a selection condition with respect to a state transition, a logic value of at least one or more signals, or at least one or more signals in the design data.

With Stratus HLS, the property generated by the property generating step is a selection condition with respect to a state transition, a logic value of at least one or more signals, or at least one or more signals in the design data.

Another way to improve DFV is for designers to add assertions in their code to identify illegal conditions. For example, the high-priority thread should never get a NACK response on a memory access request. Using Stratus™ HLS, SystemC/C++ assertions can be synthesized into the generated RTL implementation by turning on the *synthesize_asserts* feature. This allows the designer to communicate effectively and clearly the design intend and assumptions to the verifiers. The verifiers would use these assertions as targets to be verified and covered in their testbenches. Assertions are clear coverage analysis points that can be reviewed using assertion coverage features in simulation. The following figure shows an assertion in the input behavioral SystemC® design, and the equivalent Verilog assertion synthesized into the produced RTL, and also the assertion coverage report that confirms that this design feature was exercised and verified by the verification test cases.

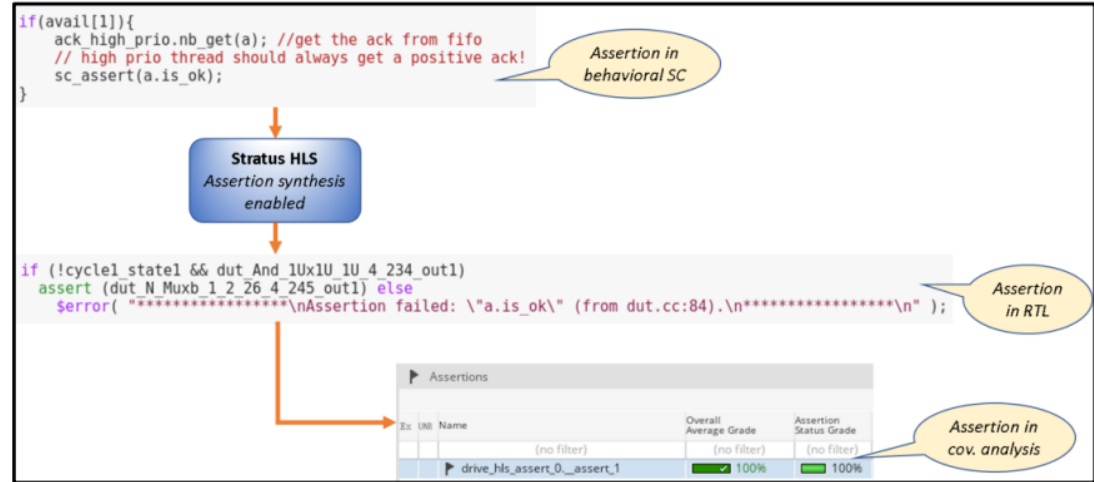


Figure 5: Assertion synthesis flow

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>.

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.