



Intel[®] High Level Synthesis Compiler Pro Edition

Version 19.4 Release Notes

Updated for Intel[®] Quartus[®] Prime Design Suite: **19.4**



RN-1146 | 2020.01.31

Latest document on the web: [PDF](#) | [HTML](#)



Contents

1. Intel® High Level Synthesis Compiler Pro Edition Version 19.4 Release Notes.....	3
1.1. New Features and Enhancements.....	3
1.2. Intel High Level Synthesis Compiler Pro Edition Prerequisites.....	4
1.3. Known Issues and Workarounds.....	5
1.4. Software Issues Resolved.....	7
1.5. Intel High Level Synthesis Compiler Pro Edition Release Notes Archives.....	7
1.6. Document Revision History for Intel HLS Compiler Pro Edition Version 19.4 Release Notes.....	7



1. Intel® High Level Synthesis Compiler Pro Edition Version 19.4 Release Notes

The *Intel® High Level Synthesis Compiler Pro Edition Release Notes* provide late-breaking information about the Intel High Level Synthesis Compiler Pro Edition Version 19.4.

For the most recent Standard Edition release notes, see the [Intel High Level Synthesis Compiler Standard Edition Release Notes](#).

About the Intel HLS Compiler Documentation Library

Documentation for the Intel HLS Compiler is split across a few publications. Use the following table to find the publication that contains the Intel HLS Compiler information that you are looking for:

Table 1. Intel High Level Synthesis Compiler Documentation Library

Title and Description	PRO	STD
Release Notes Provide late-breaking information about the Intel HLS Compiler.	Link	Link
Getting Started Guide Get up and running with the Intel HLS Compiler by learning how to initialize your compiler environment and reviewing the various design examples and tutorials provided with the Intel HLS Compiler.	Link	Link
User Guide Provides instructions on synthesizing, verifying, and simulating intellectual property (IP) that you design for Intel FPGA products. Go through the entire development flow of your component from creating your component and testbench up to integrating your component IP into a larger system with the Intel Quartus Prime software.	Link	Link
Reference Manual Provides reference information about the features supported by the Intel HLS Compiler. Find details on Intel HLS Compiler command options, header files, pragmas, attributes, macros, declarations, arguments, and template libraries.	Link	Link
Best Practices Guide Provides techniques and practices that you can apply to improve the FPGA area utilization and performance of your HLS component. Typically, you apply these best practices after you verify the functional correctness of your component.	Link	Link
Quick Reference Provides a brief summary of Intel HLS Compiler declarations and attributes on a single two-sided page.	Link	Link

1.1. New Features and Enhancements

The Intel High Level Synthesis Compiler Pro Edition Version 19.4 includes the following new features:

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.



- Added the `ihc::launch_always_run` System of Tasks API to launch a task function at component power-on or reset and continuously execute the function.
- Added the `--daz` command option to disable subnormal support in double-precision floating-point computations
- Added the `--rounding` command option to control the rounding scheme used for double-precision adders, multipliers, and dividers.

Changes in Software Support

In addition to the new features and enhancements, the Intel High Level Synthesis Compiler Pro Edition Version 19.4 has the following changes:

- The Intel High Level Synthesis Compiler documentation library has been split into separate libraries for Pro Edition and Standard Edition. For details, see [About the Intel HLS Compiler Documentation Library](#) on page 3.

1.2. Intel High Level Synthesis Compiler Pro Edition Prerequisites

The Intel HLS Compiler Pro Edition is part of the Intel Quartus® Prime Design Suite. You can install it as part of your Intel Quartus Prime software installation or install it separately. It requires Intel Quartus Prime and additional software to use.

For detailed instructions about installing Intel Quartus Prime software, including system requirements, prerequisites, and licensing requirements, see [Intel FPGA Software Installation and Licensing](#).

The Intel HLS Compiler requires the following software in addition to Intel Quartus Prime:

C++ Compiler

For Linux, install GCC compiler and C++ Libraries version 5.4.0.

- You must install these libraries manually. See [Installing the Intel HLS Compiler on Linux Systems](#) for instructions.

Important: The Intel HLS Compiler software does not support versions of the GCC compiler other than those specified for the edition of the software.

For Windows, install one of the following versions of the Microsoft Visual Studio Professional, depending on your edition of Intel Quartus Prime software:

- Microsoft Visual Studio 2017 Professional
- Microsoft Visual Studio 2017 Community

Important: The Intel HLS Compiler software does not support versions of Microsoft Visual Studio other than those specified for the edition of the software.

Mentor Graphics* ModelSim* Software

On Windows and RedHat Linux systems, you can install the ModelSim* software from the Intel Quartus Prime software installer. The available options are:

- ModelSim - Intel FPGA Edition
- ModelSim - Intel FPGA Starter Edition



Alternatively, you can use your own licensed version of Mentor Graphics* ModelSim software.

On RedHat Linux systems, ModelSim software requires the Red Hat development tools packages. Additionally, any 32-bit versions of ModelSim software (including those provided with Intel Quartus Prime) require additional 32-bit libraries. The commands to install these requirements are provided in [Installing the Intel HLS Compiler on Linux Systems](#).

On SUSE Linux systems, you must use your own licensed version of Mentor Graphics ModelSim software.

For information about all the ModelSim software versions that the Intel software supports, refer to the *EDA Interface Information* section in the Software and Device Support Release Notes for your edition of Intel Quartus Prime

Related Information

- [Intel High Level Synthesis Compiler Getting Started Guide](#)
- [Supported Operating Systems](#)
- [Software Requirements](#)
in *Intel FPGA Software Installation and Licensing*
- [EDA Interface Information \(Intel Quartus Prime Pro Edition\)](#)
- [Mentor Graphics ModelSim Website](#)

1.3. Known Issues and Workarounds

This section provides information about known issues that affect the Intel HLS Compiler Pro Edition Version 19.4.

Description	Workaround
(Windows only) Compiling a design in a directory with a long path name can result in compile failures.	Compile the design in a directory with a short path name.
(Windows only) A long path for your Intel Quartus Prime installation directory can prevent you from successfully compiling and running the Intel HLS Compiler tutorials and example designs.	Move the tutorials and examples to a short path name before trying to run them.
When you use the <code>-c</code> command option to have separate compilation and linking stages in your workflow, and if you do not specify the <code>-march</code> option in the linking stage (or specify a different <code>-march</code> option value), your linking stage might fail with or without error messages.	Ensure that you use the same <code>-march</code> option value for both the compilation with the <code>-c</code> command option stage and the linking stage.
Applying the <code>hls_merge</code> memory attribute to an array declared within an unrolled or partially unrolled loop creates an unexpectedly wide memory.	Avoid using the <code>hls_merge</code> memory attribute in unrolled loops. If you need to merge memories in an unrolled loop, explicitly declare an array of struct type.
Slave memories cannot be implemented as MLABs. They can be implemented only as M20K blocks.	N/A
In the Function Memory Viewer high-level design report, some function-scoped memories might appear as "optimized away".	None. When a file contains functions that are components and functions that are not components, all function-scoped variables are listed in the Function Memory List pane, but only variables from components have information about them to show in the Function Memory View pane.

continued...



Description	Workaround
<p>When developing a library, if you have a #define defining a value that you use later in a #pragma, the fpga_crossgen command fails.</p> <p>For example, the following code cannot be compiled by the fpga_crossgen command:</p> <pre data-bbox="240 457 797 674"> #define unroll_factor 5 int foo(int array_size) { int tmp[100]; int sum =0; //pragma unroll unroll_factor #pragma ivdep array(tmp) safelen(unroll_factor) for (int i=0;i<array_size;i++) { sum+=tmp[i]; } return sum; } </pre>	<p>Use __pragma instead of #pragma.</p> <p>For example, the following compiles successfully with the fpga_crossgen command:</p> <pre data-bbox="831 407 1388 623"> #define unroll_factor 5 int foo(int array_size) { int tmp[100]; int sum =0; //pragma unroll unroll_factor __pragma ivdep array(tmp) safelen(unroll_factor) for (int i=0;i<array_size;i++) { sum+=tmp[i]; } return sum; } </pre>
<p>Some high-level design reports fail in Microsoft* Internet Explorer*.</p>	<p>Use one of the following browsers to view the reports:</p> <ul data-bbox="831 724 1019 800" style="list-style-type: none"> • Google Chrome* • Microsoft Edge* • Mozilla* Firefox*
<p>Libraries that target OpenCL* and are written in HLS might cause OpenCL kernels that include the library to have a more conservative incremental compilation.</p>	<p>N/A</p>
<p>Libraries that target OpenCL and are written in HLS cannot use streams.</p>	<p>N/A</p>
<p>Libraries written in HLS or OpenCL are not supported on Microsoft Windows* operating systems.</p>	<p>N/A</p>
<p>Library functions that use the following HLS features cannot be emulated:</p> <ul data-bbox="240 1087 483 1171" style="list-style-type: none"> • hls_float data type • ac_fixed data type • System of tasks 	<p>N/A</p>
<p>Using a struct of a single ac_int data type in steaming interface that uses packets (ihc::usesPackets<true>) does not work.</p> <p>For example, the following code snippet does not work:</p> <pre data-bbox="240 1308 797 1497"> // class definition class DataType { ac_int<155, false> data; ... } // stream definition typedef ihc::stream_in<DataType, ihc::usesPackets<true>, ihc::usesEmpty<true> > DataStreamIn; </pre>	<p>To use this combination in your design, obey the following restrictions:</p> <ul data-bbox="831 1241 1356 1316" style="list-style-type: none"> • The internal ac_int data size must be multiple of 8 • The stream interface type declaration must specify ihc::bitsPerSymbol<8> <p>For example, the following code snippet works:</p> <pre data-bbox="859 1360 1388 1598"> // class definition class DataType { ac_int<160, false> data; // data width must be multiple of 8 ... } // stream definition typedef ihc::stream_in<DataType, ihc::usesPackets<true>, ihc::usesEmpty<true>, ihc::bitsPerSymbol<8> > DataStreamIn; // added ihc::bitsPerSymbol<8> </pre>
<p>(Windows only) When you run the init_hls.bat script, you might receive the following warning message:</p> <pre data-bbox="240 1686 797 1745"> WARNING: Detected Microsoft Visual C++ 2017 installation. The i++ compiler is not tested with this version. </pre>	<p>You can safely ignore this warning.</p>



1.4. Software Issues Resolved

No issues associated with customer service requests were corrected or otherwise resolved in Intel HLS Compiler Pro Edition Version 19.4. This release does contain other bug fixes and security updates.

1.5. Intel High Level Synthesis Compiler Pro Edition Release Notes Archives

Intel HLS Compiler Version	Title
19.4	Intel High Level Synthesis Compiler Pro Edition Version 19.4 Release Notes
19.3	Intel High Level Synthesis Compiler Pro Edition Version 19.3 Release Notes
19.2	Intel High Level Synthesis Compiler Pro Edition Version 19.2 Release Notes
19.1	Intel High Level Synthesis Compiler Pro Edition Version 19.1 Release Notes
18.1	Intel High Level Synthesis Compiler Version 18.1 Release Notes
18.0	Intel High Level Synthesis Compiler Version 18.0 Release Notes
17.1	Intel High Level Synthesis Compiler Version 17.1 Release Notes

1.6. Document Revision History for Intel HLS Compiler Pro Edition Version 19.4 Release Notes

Document Version	Intel Quartus Prime Version	Changes
2020.01.31	19.4	<ul style="list-style-type: none">Corrected publication titles in Intel High Level Synthesis Compiler Pro Edition Release Notes Archives on page 7.
2019.12.16	19.4	<ul style="list-style-type: none">Initial release.