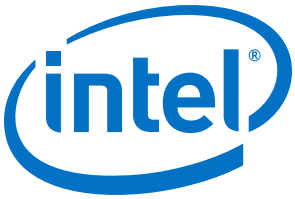

Intel® High Level Synthesis Compiler Pro Edition

Version 19.3 Release Notes

Updated for Intel® Quartus® Prime Design Suite: **19.3**

RN-1146 | 2019.09.30
Latest document on the web: **PDF** | **HTML**

Subscribe
Send Feedback

Contents

1. Intel® High Level Synthesis Compiler Pro Edition Version 19.3 Release Notes

The *Intel® High Level Synthesis Compiler Release Notes* provide late-breaking information about the Intel High Level Synthesis Compiler Pro Edition Version 19.3.

For Intel High Level Synthesis Compiler Standard Edition release notes, see Intel High Level Synthesis Compiler Version 18.1 Release Notes.

1.1. New Features and Enhancements

The Intel High Level Synthesis Compiler Pro Edition Version 19.3 includes the following new features:

- Added arbirtrary-precision floating point support with new `hls_float` data type.
- Expanded support for creating object libraries to include creating libraries from HLS code (Linux only).
- Added `#pragma clang fp rasssoc` and `#pragma clang fp contract` pragmas to enable block-level control of fp-relaxed and fpc optimizations.
- Enhanced loop controls by adding the `max_interleaving` pragma.
- Enhanced memory architecture controls with new `hls_max_replicates` memory attribute.
- Updated and improved component memory architecture tutorials.
- Enhanced high-level design reports with Schedule Viewer (alpha)
- Added new controls for controlling the styles of LSUs used for accessing variable-latency Avalon® Memory-Mapped (MM) Master interfaces.
- Added support in task functions for Avalon MM Master interfaces.
- Added `hls_fpga_reg()` to enable expert users.
- Added support for SUSE Linux Enterprise Server 12.
- Added support for Microsoft Visual Studio 2017.

Changes in Software Support

In addition to the new features and enhancements, the Intel High Level Synthesis Compiler Pro Edition Version 19.3 has the following changes:

- Removed support for Microsoft Visual Studio 2015.

  Use a support edition of Microsoft Visual Studio 2017 instead.
- Deprecated `hls_numports_readonly_writeonly` memory attribute.

  Use the `hls_max_replicates` memory attribute instead.

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 19.3.

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 `-c` command option to have separate compilation and linking stages in your workflow, and if you do not specify the `-march` option in the linking stage (or specify a different `-march` option value), your linking stage might fail with or without error messages.	Ensure that you use the same `-march` option value for both the compilation with the `-c` command option stage and the linking stage.
Applying the `hls_merge` memory attribute to an array declared within an unrolled or partially unrolled loop creates an unexpectedly wide memory.	Avoid using the `hls_merge` 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.
In the high-level design reports, block latency information is missing from the f_{MAX} II report, and might be incorrect in the Graph Viewer.	Check block latency in the Schedule Viewer in the high-level design reports.

When developing a library, if you have a `#define` defining a value that you use later in a `#pragma`, the `fpga_crossgen` command fails.
For example, the following code cannot be compiled by the `fpga_crossgen` command:
`#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];`
	Use `__pragma` instead of `#pragma`.
	For example, the following compiles successfully with the `fpga_crossgen` command:
	`#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;`
	`}`

continued...

Description	Workaround

`}`
`return sum;`
`}`	

Some high-level design reports fail in Microsoft* Internet Explorer*.	Use one of the following browsers to view the reports: • Google Chrome* • Microsoft Edge* • Mozilla* Firefox*
Libraries that target OpenCL* and are written in HLS might cause OpenCL kernels that include the library to have a more conservative incremental compilation.	N/A
Libraries that target OpenCL and are written in HLS cannot use streams.	N/A
Libraries written in HLS or OpenCL are not supported on Microsoft Windows* operating systems.	N/A
Library functions that use the following HLS features cannot be emulated: • `hls_float` data type • `ac_fixed` data type • System of tasks	N/A
The following tutorial is not supported: `<quartus_installdir>/hls/examples/tutorials/best_practices/stall_enable`	N/A
The Intel HLS Compiler fails when you include an RTL library that contains an RTL file with a filename with a length of less than less than 9 characters, including the file extension.	When you create an RTL library that targets HLS, ensure that the library filename is at least 9 characters long.

Using a `struct` of a single `ac_int` data type in steaming interface that uses packets (`ihc::usesPackets<true>`) does not work.
For example, the following code snippet does not work:
`// class definition`
`class DataType {`
`ac_int<155, false> data;`
`...`
`}`
`// stream definition`
`typedef ihc::stream_in<DataType,`
`ihc::usesPackets<true>,`
`ihc::usesEmpty<true>`
`> DataStreamIn;`
	To use this combination in your design, obey the following restrictions:
	• The internal `ac_int` data size must be multiple of 8
	• The stream interface type declaration must specify `ihc::bitsPerSymbol<8>`
	For example, the following code snippet works:
	`// 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>`

1.4. Software Issues Resolved

The following issues were corrected or otherwise resolved in the Intel HLS Compiler Pro Edition Version 19.3.

Table 1. Issues Resolved in the Intel HLS Compiler Pro Edition Version 19.3

Customer Service Request Numbers						
00387783	00399997	00400618	00422693	00434721	00448913	11387326
11409593						

1.5. Intel High Level Synthesis Compiler Release Notes Archives

Intel HLS Compiler Version	User Guide
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 19.3 Release Notes

Document Version	Intel Quartus Prime Version	Changes
2019.09.30	19.3	• Initial release.