



Intel[®] High Level Synthesis Compiler

Version 19.2 Release Notes

Updated for Intel[®] Quartus[®] Prime Design Suite: **19.2**



RN-1146 | 2019.07.01

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



Contents

1. Intel® High Level Synthesis Compiler Version 19.2 Release Notes.....	3
1.1. New Features and Enhancements.....	3
1.2. Intel High Level Synthesis Compiler Prerequisites.....	3
1.3. Known Issues and Workarounds.....	5
1.4. Software Issues Resolved.....	5
1.5. Intel High Level Synthesis Compiler Release Notes Archives.....	6
1.6. Document Revision History for Intel HLS Compiler Version 19.2 Release Notes.....	6

1. Intel® High Level Synthesis Compiler Version 19.2 Release Notes

The *Intel® High Level Synthesis Compiler Release Notes* provide late-breaking information about the Intel High Level Synthesis Compiler Version 19.2.

1.1. New Features and Enhancements

The Intel High Level Synthesis Compiler Version 19.2 includes the following new features:

- PRO Added new pipelining controls:
 - The `disable_loop_pipelining` pragma helps you control loop pipelining.
 - The `hls_disable_component_pipelining` attribute helps you control component pipelining.
- PRO Added support for the following math functions:

<code>acosh/acoshf</code>	<code>asinh/asinhf</code>	<code>atanh/atanhf</code>
<code>log1pf</code>	<code>logb/logbf</code>	<code>erf/erff</code>
<code>erfc/erfcf</code>	<code>tgamma/tgammaf</code>	<code>lgamma/lgammaf</code>
<code>lgamma_r/lgamma_rf</code>	<code>remainder/remainderf</code>	<code>remquo/remquof</code>
<code>copysign/copysignf</code>	<code>nextafter/nextafterf</code>	<code>fma/fmaf</code>
<code>fpclassify/fpclassifyf (Linux only)</code>	<code>isnormal/isnormalf (Linux only)</code>	<code>signbit/signbitf (Linux only)</code>
<code>isunordered (Linux only)</code>	<code>maxmag/maxmagf</code>	<code>minmag/minmagf</code>
<code>mad/madf</code>	<code>rootn/rootnf</code>	<code>oclnan/oclnanf</code>
<code>fract/fractf</code>	<code>isordered/isorderedf</code>	

- STD Intel HLS Compiler Standard Edition Version 19.2 is not available.

1.2. Intel High Level Synthesis Compiler Prerequisites

The Intel HLS Compiler 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 one of the following versions of the GCC compiler and C++ libraries, depending on your edition of Intel Quartus Prime software:

- **PRO** 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.
- **STD** GCC compiler and C++ Libraries version 4.4.7
These libraries are included in the version of Linux supported by the Intel HLS Compiler.

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:

- **PRO** Microsoft Visual Studio 2015 Professional
- **PRO** Microsoft Visual Studio 2015 Community
- **STD** Microsoft Visual Studio 2010 Professional

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

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 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](#).

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 \(Intel Quartus Prime Standard Edition\)](#)
- [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 High Level Synthesis Compiler Version 19.2.

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 rare cases, the compiler might fail to initialize internal linkage global variables (such <code>static</code> or <code>const</code> global variables) in the generated hardware. This failure can result in functional differences between the generated hardware and x86 emulation results.	If an internal linkage global variable requires initialization through a function, mark the function as <code>constexpr</code> . If it is not possible to mark the function as <code>constexpr</code> , declare and initialize the global variable inside the component using it. The <code>bit_fill_hex</code> functions of the <code>ac_int.h</code> header are currently not marked as <code>constexpr</code> . If you need to, you can mark the functions as <code>constexpr</code> with the <code>__HLS_CONSTEXPR__</code> macro.

1.4. Software Issues Resolved

The following issues were corrected or otherwise resolved in the Intel HLS Compiler Pro Edition Version 19.2.

Table 1. Issues Resolved in the Intel HLS Compiler Version

Customer Service Request Numbers						
00387783	00402868	00422693	00428138			



1.5. Intel High Level Synthesis Compiler Release Notes Archives

Intel HLS Compiler Version	User Guide
19.1	Intel High Level Synthesis Compiler 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 Version 19.2 Release Notes

Document Version	Intel Quartus Prime Version	Changes
2019.07.01	19.2	<ul style="list-style-type: none">Initial release.