Intel® High Level Synthesis Compiler Pro Edition

Version 19.4 Release Notes

Updated for Intel® Quartus® Prime Design Suite: 19.4


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>
<thead>
<tr>
<th>Title and Description</th>
<th>PRO</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Notes</td>
<td>Link</td>
<td>Link</td>
</tr>
<tr>
<td>Getting Started Guide</td>
<td>Link</td>
<td>Link</td>
</tr>
<tr>
<td>User Guide</td>
<td>Link</td>
<td>Link</td>
</tr>
<tr>
<td>Reference Manual</td>
<td>Link</td>
<td>Link</td>
</tr>
<tr>
<td>Best Practices Guide</td>
<td>Link</td>
<td>Link</td>
</tr>
<tr>
<td>Quick Reference</td>
<td>Link</td>
<td>Link</td>
</tr>
</tbody>
</table>

1.1. New Features and Enhancements

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

<table>
<thead>
<tr>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Windows only) Compiling a design in a directory with a long path name can result in compile failures.</td>
<td>Compile the design in a directory with a short path name.</td>
</tr>
<tr>
<td>(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.</td>
<td>Move the tutorials and examples to a short path name before trying to run them.</td>
</tr>
<tr>
<td>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.</td>
<td>Ensure that you use the same -march option value for both the compilation with the -c command option stage and the linking stage.</td>
</tr>
<tr>
<td>Applying the hls_merge memory attribute to an array declared within an unrolled or partially unrolled loop creates an unexpectedly wide memory.</td>
<td>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.</td>
</tr>
<tr>
<td>Slave memories cannot be implemented as MLABs. They can be implemented only as M20K blocks.</td>
<td>N/A</td>
</tr>
<tr>
<td>In the Function Memory Viewer high-level design report, some function-scoped memories might appear as &quot;optimized away&quot;.</td>
<td>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.</td>
</tr>
</tbody>
</table>

continued...
## Description

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:

```c
#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;
}
```

## Workaround

Use `__pragma` instead of `#pragma`. For example, the following compiles successfully with the `fpga_crossgen` command:

```c
#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;
}
```

### 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.

### Libraries that target OpenCL and are written in HLS cannot use streams.

### Libraries written in HLS or OpenCL are not supported on Microsoft Windows* operating systems.

### Library functions that use the following HLS features cannot be emulated:
- `hls_float` data type
- `ac_fixed` data type
- System of tasks

### 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:

```c
// 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:

```c
// 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>
```

### (Windows only) When you run the `init_hls.bat` script, you might receive the following warning message:

**WARNING:**
Detected Microsoft Visual C++ 2017 installation. The i++ compiler is not tested with this version.

You can safely ignore this warning.
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

<table>
<thead>
<tr>
<th>Intel HLS Compiler Version</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.2</td>
<td>Intel High Level Synthesis Compiler Version 19.2 Release Notes</td>
</tr>
<tr>
<td>19.1</td>
<td>Intel High Level Synthesis Compiler Version 19.1 Release Notes</td>
</tr>
<tr>
<td>18.1</td>
<td>Intel High Level Synthesis Compiler Version 18.1 Release Notes</td>
</tr>
<tr>
<td>18.0</td>
<td>Intel High Level Synthesis Compiler Version 18.0 Release Notes</td>
</tr>
<tr>
<td>17.1</td>
<td>Intel High Level Synthesis Compiler Version 17.1 Release Notes</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Document Version</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
</table>