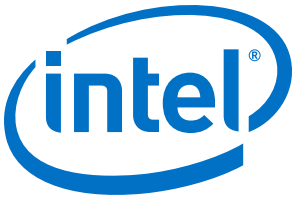

Intel® FPGA SDK for OpenCL™ Pro Edition

Version 18.1 Release Notes

Updated for Intel® Quartus® Prime Design Suite: **18.1**

RN-OCL004 | 2018.12.12
Latest document on the web: PDF | HTML

Contents

1. Intel® FPGA SDK for OpenCL™ Pro Edition Version 18.1 Release Notes

The *Intel® FPGA SDK for OpenCL™ Pro Edition Release Notes* provides late-breaking information about the Intel Software Development Kit (SDK) for OpenCL[1][2] Pro Edition and the Intel FPGA Runtime Environment (RTE) for OpenCL Pro Edition Version 18.1.

1.1. New Features and Enhancements

The Intel FPGA SDK for OpenCL Pro Edition and the Intel FPGA RTE for OpenCL Pro Edition include the following new features:

- Improved the OpenCL compiler front end.

  For details, see Improved Intel FPGA SDK for OpenCL Compiler Front End on page 3.
- Global memory is preserved on FPGA reconfiguration, when allowed by a BSP.
- Added support for full duplex PCIe, when supported by a BSP.
- Added support for Intel Stratix® 10 MX devices.
- Added preview of new fast emulator functions.
- For Linux operating systems, added preview of new co-simulation functions.
- Verified the backwards compatibility of the Intel FPGA SDK for OpenCL Pro Edition version 18.1 compiler with Intel Quartus® Prime Pro Edition Version 18.0 and Version 17.1.1.

  Backwards compatibility allows you to use older BSPs with newer OpenCL compilers. However, some newer OpenCL compiler features might not be available to use with older BSPs.

Improved Intel FPGA SDK for OpenCL Compiler Front End

The Intel FPGA SDK for OpenCL compiler has an improved front end that provides the following benefits:

- Improved error messages and warnings to help you debug and improve the code in your kernels.
- Enhanced OpenCL V1.0 specification conformance to help keep your kernel code portable.

---

(1) OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission of the Khronos Group™.

(2) The Intel FPGA SDK for OpenCL is based on a published Khronos Specification, and has passed the Khronos Conformance Testing Process. Current conformance status can be found at www.khronos.org/conformance.

You might need to adjust your code to comply with the enhanced OpenCL standards enforcement and other changes introduced by the new compiler front end. The updates to the Intel FPGA SDK for OpenCL compiler affect your kernel code in the following ways:

- The heuristics to determine when to automatically unroll loops have changed.

  If the new loop unrolling decisions affect your kernel negatively, use the `#prama unroll <N>` statement to specify a loop unroll factor.
- The OpenCL restriction against the use of variadic macros is now properly enforced. If your kernel code contains variadic macros, you now receive an error message when you compile your kernel.
- The OpenCL restriction against bit fields is now properly enforced. If your kernel code contains bit fields, you now receive an error message when you compile your kernel.
- String literals are in the address space for constants, as per the OpenCL V1.1 (and later) specification.

  While the Intel FPGA SDK for OpenCL conforms to the OpenCL V1.0 specification, the OpenCL V1.0 specification is ambiguous about where such string literal should reside. This ambiguity was resolved with the OpenCL V1.1 specification.
- Loop pragmas (for example, `#pragma unroll`) must immediately precede the loop that the pragma applies to.

  Previously, loop pragmas could be placed before elements such as statement labels on loops.
- Pragmas must appear in the function bodies in your code.

  Previously, you could place a pragma statement between a function declaration and the function body.
- Passing channels to functions by pointers is not permitted. Pass channels to functions by value.

  Previously, you would receive a warning messages when you passed channels by pointers. You now receive an error message.
- The following OpenCL and C restrictions are now strictly enforced:
  - An `enum` cannnot be incremented with the ++ operator.
  - The `channel` keyword must precede the type in a variable declaration.

    Previously, the `channel` keyword could appear before or after the type in a variable declaration.
  - You cannot use a `float` variable as the first expression in a ternary selection (`?:`) operator.
  - Any `inline` definition must also have an `extern` definition.

    You can use `static inline` definitions instead.

- Support for the following extensions and attributes is ended:
  - The `cl_intel_arbitrary_precision_integers` extension is not longer supported. You must use the `ihc_apint.h` header file instead.

    Arbitrary precision integers up to 64 bits are still supported through non-templated types.

    You cannot have `ap_int` literals longer than 32 bits (or 64 bits when suffixed with `L`).
  - Support for the deprecated `num_vector_lanes` attribute is ended.
- The `restrict` type qualifier is replaced with the `__restrict` keyword.
- The bank selection bits specified in the `bankbits` memory attribute must be specified in ascending or descending order. If the bank selection bits are not ordered, you receive an error message.
- On Windows systems, object files and libraries that depend on Microsoft Visual Studio runtime libraries must be built with the `/MD` Microsoft compiler option.

1.2. Operating System Support

Information about OS support for the Intel FPGA SDK for OpenCL is available on the Operating System Support page of the Intel FPGA website.

Related Information

Operating System Support

1.3. Changes to Software Behavior

Items listed in the following table represent cases in which the behaviors of the current release of the Intel FPGA SDK for OpenCL and the Intel FPGA RTE for OpenCL differ from the previous version.

Description	Required Actions
The `restrict` keyword is renamed to `__restrict`.	Update your code to use the `__restrict` keyword instead of the `restrict` keyword.
Support for passing pipes or channels by reference is removed.	Update your code to pass pipes or channels by value.

Items listed in the following table represent cases in which the behaviors of the current release of the Intel FPGA SDK for OpenCL Custom Platform Toolkit and Reference Platforms differ from the previous version.

Description	Workaround
N/A	N/A

1.4. Known Issues and Workarounds

This section provides information about known issues that affect the Intel FPGA SDK for OpenCL and the Intel FPGA RTE for OpenCL Version 18.1.

Description	Workaround
OpenCL kernels with names longer than 61 characters might fail in the Quartus compiler with an error similar to the following error: `Error (16045): Instance "...|<long_kernel_name>_cra_slave_inst" instantiates undefined entity "<long_kernel_name>_function_cra_slave" File: <filename> Line: <linenumber>`	Reduce the size of the OpenCL kernel name.
Valid OpenCL kernel pipes cannot be passed as arguments in some cases. The symptom is the runtime will receive a `CL_INVALID_BUFFER_SIZE (-61)` error when you enqueue your kernel.	Modify your design to use channels instead of pipes.
The emulator runtime results in an assertion error if a kernel is enqueued 16,000 times.	Do not enqueue a kernel more than 16,000 times.
When alternatively using subbuffers and their parent buffers, changes written to one might not be reflected in the other.	Unmapping and mapping a buffer forces the subbuffers and their parent buffers to be synced. Unmapping and mapping a buffer between buffer uses should prevent this issue.
After you set up the Installable Client Driver (ICD) and the FPGA Client Driver (FCD) on an Intel SoC Custom or Reference Platform, the Intel FPGA SDK for OpenCL `aocl link-config` and `aocl linkflags` utilities do not return the correct library paths.	To obtain the correct information on libraries and paths, concatenate the results returned by the `aocl ldflags` and `aocl ldlibs` utilities.
In the OpenCL runtime, making more than one OpenCL context in a multithreaded environment might cause a segmentation fault.	—
When you compile kernel code for the fast emulator platform, some diagnostic messages might not be printed. Some kernel compilations might fail without printing any useful error messages.	Compile your kernel code for the default emulator (by removing the `-fast-emulator` flag from your `aoc` command) to view any relevant diagnostic messages. After you address any compilation warnings or errors,recompile your kernel code for the fast emulator platform.

In some cases, an error occurs when reading data from a channel directly into a variable in the __local address space.
For example, the following code causes an error:
`#pragma OPENCL EXTENSION cl_intel_channels : enable`
`channel int TOKEN_STREAM;`
`__kernel void consumer ()`
`{`
`  __local int base_address;`
`  base_address =`
`read_channel_intel( TOKEN_STREAM );`
`}`
	Store the data into a private variable first, then copy the data to the variable in the __local address space.
	For example:
	`#pragma OPENCL EXTENSION cl_intel_channels :`
	`enable`
	`channel int TOKEN_STREAM;`
	`__kernel void consumer ()`
	`{`
	`  int temp_storage;`
	`  __local int base_address;`
	`  temp_storage =`
	`read_channel_intel( TOKEN_STREAM );`
	`  base_address = temp_storage;`
	`}`

On Linux platforms, the installation script of the Intel FPGA SDK for OpenCL (`setup_pro.sh`) does not invoke the Intel Code Builder for OpenCL installer after installing Intel Quartus Prime and the Intel FPGA SDK for OpenCL. Intel Code Builder for OpenCL is required if you wish to use the Intel Code Builder for OpenCL Plug-in or the fast emulator for OpenCL.	After running the `setup_pro.sh` script, manually install the Intel Code Builder for OpenCL package by running the following file: `intel_sdk_for_opencl_setup_<installer_version_number>.run` For example, `intel_sdk_for_opencl_setup_7.0.0.3101.run`. This file is available in the components subdirectory created when you extract the installer `.tar` file.

This section provides information about known issues that affect the current release of the Intel FPGA SDK for OpenCL Custom Platform Toolkit and Reference Platforms. These issues might also affect Custom Platforms you create for use with the Intel FPGA SDK for OpenCL.

Description	Workaround
Race conditions can occur between enqueue and dequeue buffer operations and host pipe operations. These conditions can results in incorrect data being read or written.	Manually make sure that no enqueue or dequeue buffer operations occur in parallel with host pipe API calls. A way to ensure that buffer operations to do not occur in parallel with host pipe API calls is to do buffer operations as blocking calls before the first host pipe operation and after you are certain that the last host pipe operation has been completed (for example, all the data is read back).
For Windows, when the host application queries the number of devices, calls to `clGetDeviceIDs` return 128 devices regardless of the actual number of devices present. *Note:* You can find the actual available devices at the beginning of the device list returned by `clGetDeviceIDs`. This issue affects the Intel Arria® 10 GX FPGA Development Kit Reference Platform and the Intel Stratix 10 GX FPGA Development Kit Reference Platform.	Perform one of the following workarounds: • Rewrite the host application to limit the query for `clGetDeviceIDs` to the actual number of devices. • Rewrite the host application to use `clGetDeviceInfo` to query which devices are available. Calling `clGetDeviceInfo` with the `CL_DEVICE_AVAILABLE` flag correctly reports that extraneous devices are unavailable. • Rewrite the host application to only call `clCreateContext` with the actual number of devices. Calling `clCreateContext` with extraneous devices will fail with the error `CL_DEVICE_NOT_AVAILABLE`. • Set the environment variable *CL_OVERRIDE_NUM_DEVICES_INTELFPGA* to the correct number of devices. Doing so fixes the erroneous behavior of `clGetDeviceIDs`.

For additional known issue information for the current Intel FPGA SDK for OpenCL version, refer to the Knowledge Base web page.
Additional Known Software Issues Affecting the Intel FPGA SDK for OpenCL Version 18.1

Latest Known Intel FPGA SDK for OpenCL Software Issues

You can find known issue information for previous Intel FPGA SDK for OpenCL versions on the Knowledge Base web page.

Related Information

Knowledge Base

1.5. Software Issues Resolved

The following issues were corrected or otherwise resolved in the Intel FPGA SDK for OpenCL and the Intel FPGA RTE for OpenCL Version 18.1.

Table 1. Issues Resolved in the Intel FPGA SDK for OpenCL and the Intel FPGA RTE for OpenCL Version 18.1

Customer Service Request Numbers						
11332051	11346929	11363755	11386675	11386675	11408173	11411327
11414858						

1.6. Software Patches Included in this Release

Table 2. Software Patches Included in the Intel FPGA SDK for OpenCL

Software Version	Patch	Customer Service Request Number
Intel FPGA SDK for OpenCL version 18.0	0.28cl	11386675

1.7. Document Revision History of the Intel FPGA SDK for OpenCL Pro Edition Release Notes

Document Version	Intel Quartus Prime Version	Changes
2018.12.12	18.1	• The following items were incorrectly listed as known issues for Version 18.1: — Passing a struct as a function. Passing a struct as a function does not cause a compiler error as the earlier versions of this document indicated. — Support in OpenCL Profiler for kernels with multiple channel call sites. You can use the OpenCL Profiler on kernels with multiple channel call sites without receiving a compilation error. — Restriction on having kernel source files named `kernel.cl`. This restriction was lifted.
2018.10.17	18.1	• Corrected typos. • Removed statement of support for dynamic channel indices. This feature was accidentally included in the list of features added. Dynamic channel indices is not fully supported in Version 18.1.
2018.09.24	18.1	Initial release.