Intel® FPGA SDK for OpenCL™ Pro Edition

Version 19.1 Release Notes

Updated for Intel® Quartus® Prime Design Suite: **19.1**

Send Feedback

RN-OCL004 | 2019.04.01
Latest document on the web: **PDF** | **HTML**

Contents

Send Feedback

1. Intel® FPGA SDK for OpenCL™ Pro Edition Version 19.1 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 19.1.

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:

- Non-`NULL` `global_work_offset` arguments are now supported in the `clEnqueueNDRangeKernel` function as described in the OpenCL Specification version 1.2.

- Enhanced the High Level Design Report in the following ways:

  — Added hierarchical menus to help you find information easily.

  — Added the following new reports:

    - The Block Viewer provides a granular view of the kernel.

    - The Cluster Viewer helps in viewing a cluster inside a block and all variables inside a cluster that have loop-carried dependency.

    - The Fmax II report lists key performance metrics on all blocks including scheduled f_{max}, sustainable II, block latency, and maximum interleaving iterations.

  — The Kernel Memory Viewer of the High Level Design Report is enhanced to help you report the following:

    - All user-defined arrays (including those implemented in registers and ROMs, or optimized away).

    - Arrays generated by inlining, unrolling, and disaggregation.

    - Implementation of a bank to help you understand the area cost of the memory system.

- Addressed areas with high overhead for smaller operations, for example, reduced overhead for a one-byte write by 33%.

---

[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.

- Improved host-mapped buffer efficiency by eliminating unnecessary copies between the device and the host program.

- Added a fast kernel-relaunch optimization feature to enable immediate launch of next kernel invocation. This feature helps in eliminating the overhead of launching the same kernel consecutively.

- Added the `speculated_iterations` pragma to direct the Intel FPGA SDK for OpenCL offline compiler to improve the performance of pipelined loops.

- Added the `__attribute__((scheduler_target_fmax_mhz(__x)))` source-level attribute to globally compile all kernels using the kernel-specific f_{max} target.

- Added the `__attribute__((max_concurrency(k))` memory attribute to control the copies of local arrays to reduce the area used while maintaining the throughput.

- An array declared in a multiply-inlined function or unrolled loop will now be duplicated along with the datapath.

- Added the following `aoc` command options:

  — `-no-hardware-kernel-invocation-queue` to direct the offline compiler to reduce kernel area use by removing kernel invocation queue in an OpenCL kernel.

  — `-global-ring` to override offline compiler's choice of optimal global memory interconnect topology and force a ring topology.

  — `-duplicate-ring` to improve throughput of kernel writes to global memory when offline compiler implements a ring topology.

  — `-hyper-optimized-handshaking=<auto | off>` to modify the handshaking protocol used in certain areas of your design targeting Intel Stratix® 10.

  — `-list-board-packages` to list Custom Platforms available in the system.

  — `-fmax=<fmax target in MHz>` to direct the offline compiler to globally compile all kernels using the kernel-specific f_{max} target.

- The fast emulator is now the recommended emulator. It includes performance optimizations and support for all functional hardware flags.

- Added a new memory attribute that allows you to specify whether a memory system should be implemented in M20Ks or MLABs. By default, the offline compiler now implements memory systems as `AUTO`.

- Expanded the offline compiler support for memory attributes:

  — Memory attributes can now be applied to constant memories (memories with loads only). This enables the creation of multi-ported ROMs, and forces ROMs into LUTs.

  — Memory attributes can now be applied to a member of a `struct`.

- Enhanced the compiler's inference of stall-free port sharing to identify more situations where sharing can be inferred.

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.

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 OpenCL headers provided by the Intel FPGA SDK for OpenCL have been updated to the latest version distributed by Khronos.	Refer to the OpenCL 2.0 Headers in the *Intel FPGA SDK for OpenCL Pro Edition: Programming Guide* for more information.
The `cl.hpp` header is now deprecated.	Update your OpenCL host program code to use the `cl2.hpp` header that supports all versions of OpenCL.
The OpenCL host runtime now returns `CL_PROFILING_INFO_NOT_AVAILABLE` when you call `clGetEventProfilingInfo` function on a `cl_event` associated with a `cl_command_queue` that was not created with the `CL_QUEUE_PROFILING_ENABLE` property. Previous versions of the Intel FPGA SDK for OpenCL did not enforce this requirement of the OpenCL Specification version 1.2.	Update your OpenCL host program to create the `cl_command_queue` with the `CL_PROFILING_INFO_ENABLE` property if you want to enable profiling.
The formatting of `printf` command output of OpenCL vector types has been modified to be conformant with the OpenCL Specification version 1.2.	—
Mandatory requirement for all host programs that use Intel FPGA-specific APIs and enumerations.	If your host programs use Intel FPGA-specific APIs and enumerations, such as `clReadPipeIntelFPGA`, `clGetProfileInfoIntelFPGA`, and other APIs that end with `IntelFPGA`, you must explicitly include the `CL/cl_ext_intelfpga.h` header file in your source code.
The host pipe feature requires the use of OpenCL 2.0 features.	Enable support for OpenCL 2.0 APIs in all of your host programs that use host pipes in their source code. Refer to the OpenCL 2.0 Headers in the *Intel FPGA SDK for OpenCL Pro Edition: Programming Guide* for more information.
In previous versions of Intel FPGA SDK for OpenCL, the offline compiler automatically disabled the local memory replication for all memory systems if it estimated an M20K utilization of over 100%. The offline compiler no longer disables the local memory replication automatically.	To restore the behavior of disabling the local memory replication automatically, use the `-no-local-mem-replication` option in the `aoc` command.
Automatic loop unrolling is now disabled.	If you want to enable it, use LLVM options `-unroll-threshold` and `-unroll-partial-threshold` in the `aoc` command. Recommended settings: • For optimization levels `-O0`, `-O1`, and `-O2`, set `-unroll-threshold=150` and `-unroll-partial-threshold=150`. • For optimization level `-O3`, set `-unroll-threshold=300` and `-unroll-partial-threshold=150`.
Support for passing pipes or channels by reference is removed.	Update your code to pass pipes or channels by value.
The `restrict` keyword is renamed to `__restrict`.	Update your code to use the `__restrict` keyword instead of the `restrict` keyword.

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

Related Information

OpenCL 2.0 Headers

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 19.1.

Description	Workaround
The OpenCL fast emulator does not support the `uses_global_work_offset` kernel attribute. A warning may be emitted at compilation time to notify you that the attribute is ignored.	Ignore the warning message because it does not affect kernel execution in the fast emulator.
When debugging OpenCL kernel code with the fast emulator using `gdb` on a Linux system, you may see the message `1: No such file or directory`.	Copy your kernel source file to a file called `1` and restart your debugging session.
Links to the OpenCL help pages are either broken or incorrect.	Refer to the *Intel FPGA SDK for OpenCL Pro Edition Best Practices Guide*.
The OpenCL emulator does not return `CL_INVALID_GLOBAL_OFFSET` when a kernel with the `uses_global_work_offset(0)` kernel attribute set is enqueued with a non-zero or non-`NULL` `global_work_offset` argument.	Set the `global_work_offset` argument to `NULL` or zero in all work group dimensions so that this issue does not change the kernel output between the emulator and hardware.

OpenCL kernels with names longer than 61 characters might fail in the Intel Quartus® Prime Pro Edition 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 sub-buffers and their parent buffers, changes written to one might not be reflected in the other.	Unmapping and mapping a buffer forces the sub-buffers and their parent buffers to be synced. Unmapping and mapping a buffer between buffer uses should prevent this issue.
In the OpenCL runtime, making more than one OpenCL context in a multithreaded environment might cause a segmentation fault.	—

continued...

Send Feedback

1. Intel® FPGA SDK for OpenCL™ Pro Edition Version 19.1 Release Notes
RN-OCL004 | 2019.04.01

Description	Workaround
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 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`.

Latest Known Intel FPGA SDK for OpenCL Software Issues

For additional known issue information for the current Intel FPGA SDK for OpenCL version and for previous versions, refer to 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 19.1.

Table 1.   **Issues Resolved in the Intel FPGA SDK for OpenCL and the Intel FPGA RTE for OpenCL Version 19.1**

Customer Service Request Numbers
2205691404

1.6. Software Patches Included in this Release

No software patches in this release.

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

Document Version	Intel Quartus Prime Version	Changes
2019.04.01	19.1	Initial release.

Send Feedback