OpenCL™ Driver for Intel® HD, Iris™, and Iris™ Pro Graphics for Linux -- Release Notes[1]

Version Information

This document covers the Intel® OpenCL Linux graphics device driver version r5.0-BUILD_ID, hereafter referred to as the intel-opencl-r5.0 driver, where BUILD_ID refers to the build ID of the distributed files.

Overview

The intel-opencl-r5.0 driver for Linux exposes the general-purpose parallel compute capabilities of Intel® graphics for OpenCL applications.

This release provides OpenCL 2.0 support for 5th, 6th and 7th generations of Intel® Core™ and Xeon™ processors with Intel® Processor Graphics Technology not previously disabled by the BIOS or motherboard settings, OpenCL 1.2 support for Intel® Atom™ processor E3900 series, Intel Pentium N4200, J4000 Intel Celeron J3000, N3350 and the following extensions:

- cl_intel_accelerator
- cl_intel_advanced_motion_estimation (version 2; see notes below)
- cl_intel_device_side_avc_motion_estimation
- cl_intel_driver_diagnostics
- cl_intel_media_block_io
- cl_intel_motion_estimation
- cl_intel_planar_yuv
- cl_intel_packed_yuv
- cl_intel_required_subgroup_size
- cl_intel_subgroups
- cl_intel_subgroups_short
- cl_intel_va_api_media_sharing
- cl_khr_3d_image_writes
- cl_khr_byte_addressable_store
- cl_khr_depth_images
- cl_khr_fp16 (5th generation Intel® Core™ processors and above)
- cl_khr_fp64 (5th generation Intel® Core™ processors and above)
- cl_khr_global_int32_base_atomics
- cl_khr_global_int32_extended_atomics
- cl_khr_icd
- cl_khr_image2d_from_buffer (5th generation Intel® Core™ processors and above)
- cl_khr_local_int32_base_atomics
- cl_khr_local_int32_extended_atomics
- cl_khr_mipmap_image

- cl_khr_mipmap_image_writes
- cl_khr_spir
- cl_khr_subgroups (6th generation Intel® Core™ processors and above)

System Requirements

The intel-opencl-r5.0 driver enables OpenCL 1.2 or 2.0 on the GPU/CPU for the following Intel® processors:

- Intel® 5th, 6th or 7th generation Core™ processor
- Intel Pentium N4200, J4000 and Intel Celeron J3000, N3350
- Intel® Atom™ processor E3900 series
- Intel® Xeon® v4, or Intel® Xeon® v5 Processors with Intel® Graphics Technology enabled by the BIOS or motherboard settings

In addition to the above processor requirements, proper operation also requires Linux kernel source code patches included with Intel® driver to be applied to the kernel. See the intel-opencl-r5 driver installation document for more information on how to apply the patches and to verify that the system meets the necessary requirements.

Changes Since intel-opencl-r4.1

- cl_intel_va_api_media_sharing – added new full NV12 format support
- Added Centos 7.3 support
- Added Ubuntu 16.04.2 with vanilla 4.8 kernel support – **no kernel patching needed**

Changes Since intel-opencl-r4.0

- Functional quality improvements

Changes Since intel-opencl-r3.1

The intel-opencl-r4.1 driver includes the following new features introduced since the intel-opencl-r3.1 release:

- VTune support for 7th generation Intel® Core™ processors
- updated patches for the Linux 4.7 and 4.4 kernels
- cl_intel_device_side_avc_motion_estimation extension to support AVC VME built-in functions callable from OpenCL kernels giving greater flexibility for expert developers and potential performance improvements for simpler VME operations compared to the built-in kernels provided with the cl_intel_motion_estimation and cl_intel_advanced_motion_estimation extensions. The new functions can be used to implement the pre-ENC and ENC stages of an encode pipeline using Intel® GPUs.
- cl_intel_media_block_io extension to augment the block read/write functionality available in the Intel® vendor subgroup extensions, cl_intel_subgroups and cl_intel_subgroups_short, by the specification of additional built-in functions to facilitate the reading and writing of flexible 2D regions from images. This API allows for the explicit specification of the width and height of the image regions. The primary use for this extension is to support the reading of edge texels (or image elements) of neighboring macro-blocks as described in the Intel vendor extension cl_intel_device_side_avc_motion_estimation.
- cl_intel_planar_yuv - Adds native support for the Planar YUV (YCbCr) image format which can be used with the Device-Side AVC Motion Estimation and Video Enhancement extensions.
- cl_intelx_video_enhancement (6th generation Intel® Core™ processors and above) preview extension

- cl_intelx_video_enhancement_camera_pipeline (6th generation Intel® Core™ processors and above) preview extension
- cl_intelx_video_enhancement_color_pipeline (6th generation Intel® Core™ processors and above) preview extension

The video enhancement, video enhancement camera pipeline, and video enhancement color pipeline preview extensions are described in section Preview OpenCL Extensions.

Changes Since intel-opencl-r2.0

In addition to the changes since the intel-opencl-r3.1 release, the intel-opencl-r4.1 driver includes the following new features introduced since the intel-opencl-r2.0 release:

- OpenCL 2.0 support for 7th generation Intel® Core™ processors
- OpenCL 1.2 support for Intel Pentium J4000 and Intel Celeron J3000[2]
- support for a patched Linux 4.7 kernel
- deprecation of the Linux 3.10.0 and 4.1 kernels
- 4th generation Intel® Core™ processors are not formally supported in this release
- cl_intel_driver_diagnostics extension allows the driver to pass additional strings containing diagnostic information. The diagnostic messages can help to understand how the driver works and can provide guidance to modify an application to improve performance.
- cl_intel_subgroups_short extension to support improved performance of applications operating on 16-bit data types by extending the subgroup functions described in the cl_intel_subgroups extension to support the 16-bit integer data types short and ushort.

Changes Since 16.5 Release Included with Intel® Media Server Studio 2017 R2

In addition to the changes since the intel-opencl-r4.1 the intel-opencl-r5.0 driver includes the following new features introduced since the 16.5 release:

- support for the MSS 2017 R3 media release using the patched Linux 4.4 kernel[3]

System using the Intel® Media Server Studio 2017 or R2 releases are not supported and must be upgraded to MSS 2017 R3. See the MSS 2017 R3 documentation for installation procedures and hardware requirements.

Changes Since 16.5 Release Included with Intel® Media Server Studio 2017

In addition to the changes since the intel-opencl-r2.0 and intel-opencl-r3.1 release, the intel-opencl-r4.1 driver includes the following new features introduced since the 16.5 release:

- cl_intel_driver_diagnostics extension
- cl_intel_subgroups_short extension
- support for the MSS 2017 R2 media release using the patched Linux 4.4 kernel[4]

---

[2] Enabling GPU OpenCL for supported Intel Atom, Celeron and Pentium series of processors requires the patched Linux 4.7 kernel as well as a kernel option provided by the boot loader. See section Known Workarounds for more information.
[3] System utilizing the Intel® Media Server Studio 2017 R2 must not use included Linux 4.7 kernel patches.
[4] System utilizing the Intel® Media Server Studio 2017 R2 must not use included Linux 4.7 kernel patches.

System using the Intel® Media Server Studio 2017 release are not supported and must be upgraded to MSS 2017 R2. See the MSS 2017 R2 documentation for installation procedures and hardware requirements.

Supported Configurations

Intel validates the intel-opencl-r5.0 driver on CentOS 7.2 and 7.3 when running the following 64-bit kernels:

- Linux 4.7 kernel patched for OpenCL
- Linux 4.4 kernel patched for MSS 2017 R3

Intel also validates Ubuntu 16.04.2 when running the following 64-bit kernel:

- Ubuntu 16.04.2 vanilla 4.8 kernel

Ubuntu 16.04 with the default kernel works fairly well but some core features (i.e. device enqueue, SVM memory coherency, VTune support) won't work without kernel patches. This configuration has been minimally validated to prove that it is viable to suggest **for experimental use**, but it is not fully supported or certified.

Although Intel validates and provides technical support only for the above Linux kernels on CentOS 7.2 and 7.3, other distributions may be adapted by utilizing our generic operating system installation steps as well as MSS 2017 R3 installation steps.

The CPU OpenCL solution is also packaged with the intel-opencl-r5.0 driver. The combined GPU/CPU platform has been validated with this release.

Preview OpenCL Extensions

This driver release contains optionally accessible preview extensions permitting developers to expose new funcionality to OpenCL kernels. Once enabled using an environment variable the preview extensions will be listed in the device extension list provided by clGetDeviceInfo. All preview extensions can be identified by the pattern cl_intelx_ in the extension name.

Developers utilizing the preview extensions are encouraged to provide feedback and to review future releases for changes. Additionally:

- A preview extension is not a final commitment. We may modify the functionality and/or interface or remove it completely. Developers should not depend on the existence of preview extensions in shipping drivers and should not enabled these features on end-user platforms as part of any application installation process.
- Preview extensions may not receive the same amount of testing as production features. If you have enabled a preview extension, please only file bug reports for that extension. If you suspect a bug in non-preview functionality, it would be helpful to confirm that issue with and without preview extensions enabled.
- We appreciate and want all feedback! Your feedback enables us to improve the product and provide you with the best product possible. Please use the feedback channels provided elsewhere in this document, or contribute to the following forum: https://software.intel.com/en-us/forums/opencl

Instructions for Preview OpenCL Features

In Linux, you can use an environmental variable with prefix: "OCL_" in the following manner

```
$ OCL_EnablePreviewFeatures=1 path/to/oclapplication
```

or

```
$ export OCL_EnablePreviewFeatures=1
```

This release provides preview support for the following Video Enhancement HW block functionality for expert developers as the preview extensions:

- cl_intelx_video_enhancement - Exposes a fixed-function pipeline consisting of Denoise, Deinterlace and Hot Pixel Correction stages. It comes together with a number of per-frame and per-block statistics that can be accessed, including global noise estimate statistics, spatial-temporal motion measure, denoise history, histograms, per command and other statistics that the Video Enhancement pipeline generates. Includes support for NV12 image format. The features are based on built-in kernels infrastructure and the cl_intel_accelerator extension.

- cl_intelx_video_enhancement_camera_pipeline - Exposes a fixed-function pipeline of operations working on camera inputs in Bayer formats: Black Level Correction, Vignette, White Balance Correction, Denoise, Hot Pixel Correction and Demosaic. The extension reuses the infrastructure defined in the cl_intelx_video_enhancement extension.

- cl_intelx_video_enhancement_color_pipeline - Extends the cl_intelx_video_enhancement and cl_intelx_video_enhancement_camera_pipeline extensions with color processing and enhancement pipeline, consisting of the following stages: Color Correction Matrix, Forward Gamma Correction, Front-End Color Space Conversion, Skin-tone Detection and Enhancement, Gamut Compression, Adaptive Contrast Enhancement, Total Color Correction, Process Amplifier, Back-end Color Space Conversion and Gamut Expansion / Color Correction.

Package Contents

The following files are included with the intel-opencl-r5.0 driver distribution:

- intel-opencl-r5.0-BUILD_ID-* (.rpm and .tar.xz)
  - the OpenCL 2.0 ICD loader, the OpenCL 2.0 ICD for Intel® HD, Iris, and Iris Pro graphics, and the Intel® OpenCL 2.0 driver and runtime for Intel® processor graphics
  - kernel mode driver patches based against specific distributions or reference kernels
- intel-opencl-devel-r5.0-BUILD_ID-* (.rpm and .tar.xz)
  - optional OpenCL 2.0 development files for compiling OpenCL applications
- intel-opencl-cpu-r5.0-BUILD_ID-* (.rpm and .tar.xz)
  - Intel® OpenCL 2.0 runtime for Intel® processor

Linux kernel 4.7 patch description

Intel validates OpenCL driver with the full set of the 17 patches for Linux 4.7 kernel.

#	Title	type	comment
1	drm/i915: Add L3_LRA_1 Register to cmdparser whitelist	performance	HSW L3 cache
2	drm/i915: Android MOCS on top of upstream	performance	SKL Mocs
3	drm/i915: Exec flag to force non IA-Coherent cache for Gen9+	functional	Fine grain
4	Revert "drm/i915/skl: Fix spurious gpu hang with gt3/gt4 revs"	functional	Fine grain
5	drm/i915: OA regs configuration for MDAPI TBS	functional	VTune support
6	drm/i915 - Allow a render node to use perfmon ioctl	functional	VTune support
7	drm/i915: Fix ref counting for RC6 WA BB alloc/dealloc	functional	VTune support
8	drm/i915/perfmon: Fixup locking on error path in RC6 WA BB alloc	functional	VTune bug fix
9	drm/i915: Remove unused "valid" parameter from pte_encode	functional	bug fix
10	drm/i915/gtt: Split gen8_ppgtt_clear_pte_range	functional	bug fix

11	drm/i915/gtt: Free unused lower-level page tables	functional	bug fix
12	drm/i915/gtt: Mark tlbs dirty on clear	functional	bug fix
13	drm/i915/gtt: Fix pte clear range	functional	bug fix
14	drm/i915: Add support for OCL Turbo Boost	performance	enhancement
15	Subject: drm/i915/bxt: Enable Pooled EU support	functional	BXT specific
16	drm/i915/bxt: Add WaEnablePooledEuFor2x6	functional	BXT specific
17	Implement WaDisablePooledEuLoadBalancingFix	functional	BXT specific

Known Workarounds

- For workloads that take longer than 1.5 seconds the i915 hang check will reset the GPU, output a kernel message for logging, and clear any pending work items. When necessary, the i915 hang check can be disabled on demand with

  ```
  $ sudo bash -c 'echo N > /sys/module/i915/parameters/enable_hangcheck'
  ```

  Although the GPU will no longer reset when executing with hang checks disabled, sufficiently large workloads may stall other GPU tasks such as screen updates. These situations can be recovered from by manually resetting the GPU with

  ```
  $ sudo bash -c 'echo 1 > /sys/kernel/debug/dri/0/i915_wedged'
  ```

- The trade-off between GPU busy (GPU being fed) vs. latency is that the driver might internally choose to submit or flush after *n* commands being queued and this is an expected behavior. Currently the driver is forced to flush after *n*=8 commands are queued.

- The 4.7 linux kernel has preliminary hardware support for Intel® Atom™ processor E3900 series, Intel Pentium N4200, J4000 Intel Celeron J3000, N3350. To enable the OpenCL functionality for those platforms you need to add the parameter to the kernel command line:

  ```
  i915.preliminary_hw_support=1
  ```

Known Limitations

When creating vaSurface using vaCreateSurfaces the resulting surface may have larger size then requested. This is caused by surface size alignment requirement for better performance.

This needs to be taken into account when creating surfaces for OCL sharing and VME functionality. Motion Estimation kernel will sample data outside of surface size. Normally sampler clamps the out of bounds access and VME kernel works assuming this is the case. However when the vaSurface has increased size because of alignment, pixel accesses that would normally be handled as out of bounds will instead read the data from rows or columns used for padding. Data in those columns is undefined and will skew the results on the bottom and right size of the surface. To achieve desired behavior user must copy rightmost column data into the extra columns and bottom row into the extra rows. From kernel perspective this will be as if the image did not have extra padding and access was clamped at the requested image size. Simplified order of these operations is provided below.

Create new surface of desired size using vaCreateSurfaces:

```
vaCreateSurfaces(vaDisplay, VA_FOURCC_NV12, imageWidth, imageHeight, &vaSurface, 1, NULL, 0);
```

Create VA image from the surface and write user image data to it. Example below is using VA_FOURCC_NV12 format, the exact implementation will differ depending on the format. Width and height parameters should be the same as passed to vaCreateSurfaces.

```
void WriteToVASurface(VADisplay vaDisplay, VASurfaceID vaSurface, const size_t width, const
size_t height, const std::vector<cl_uchar> &data) {
VAImage vaImage;
vaDeriveImage(vaDisplay, vaSurface, &vaImage);

cl_uchar *ptr;
vaMapBuffer(vaDisplay, vaImage.buf, (void**)&ptr);

size_t elemSize = sizeof(cl_char);
size_t pitch = vaImage.pitches[0] / elemSize;
size_t lineSize = width * elemSize;

for (size_t y = 0; y < height; ++y){
    memcpy(ptr + vaImage.offsets[0] + y * pitch, &data.at(y * width), lineSize);

    // Copy last element of the column into the extra space
    for (size_t x = width; x < pitch; ++x){
        memcpy(
        ptr + vaImage.offsets[0] + y * pitch + x * elemSize,
        ptr + vaImage.offsets[0] + y * pitch + (width - 1) * elemSize,
        elemSize);
        }
    }

    // Copy bottom row into the extra space
    for (size_t y = height; y < vaImage.height; ++y){
        memcpy(ptr + vaImage.offsets[0] + y * pitch, ptr + vaImage.offsets[0] + (height - 1) *
                                        pitch, lineSize);
    }

    vaUnmapBuffer(vaDisplay, vaImage.buf);
    vaDestroyImage(vaDisplay, vaImage.image_id);
}
```

Now the surface can be shared with OCL.

```
clCreateFromVA_APIMediaSurfaceINTEL(context, CL_MEM_READ_WRITE, &vaSurface, 0, 0);
```

After these steps the enqueued motion estimation kernel will provide correct results.

Known Issues

- CV SDK R1 has illegal type casting in ocl kernels (uchar4 -> uint) resulting in 90% performance drop in ScaleImage, WarpAffine, WarpPerspective and 5 conformance tests failures:
  - OCL_OVX_CONFORMANCE_MT_TILED_GAUSSIANPYRAMID
  - OCL_OPENVX_CONFORMANCE.OCL_OVX_CONFORMANCE_MT_TILED_GRAPH
  - OCL_OPENVX_CONFORMANCE.OCL_OVX_CONFORMANCE_MT_TILED_SCALE
  - OCL_OPENVX_CONFORMANCE.OCL_OVX_CONFORMANCE_MT_TILED_WARPAFFINE
  - OCL_OPENVX_CONFORMANCE.OCL_OVX_CONFORMANCE_MT_TILED_WARPPERSPECTIVE

  This issue is planned to be fixed in the next CV SDK release.

- Ubuntu 16.04.2 with vanilla 4.8 kernel does not support Vtune for OpenCL workloads.

Feedback and Support

This user-mode driver and kernel patch set are focused on OpenCL compute use cases. Unless otherwise specified, interoperability with other drivers, operating systems, or platform features is not verified or

supported. We welcome feedback to continue to make this product better. Please direct your feedback, including feature requests, through your primary Intel product support channels.

Legal

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20170630