AMD **Accelerated** Parallel Processing
TECHNOLOGY

SAMPLE

Buffer-image Interop

1 Overview

1.1 Location

`$<APPSDKSamplesInstallPath>\samples\opencl\cl\`

1.2 How to Run

See the *Getting Started Guide* for how to build samples. You first must compile the sample.

Use the command line to change to the directory where the executable is located. The precompiled sample executable is at `$<APPSDKSamplesInstallPath>\samples\opencl\bin\x86\` for 32-bit builds, and `$<APPSDKSamplesInstallPath>\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `BufferImageInterop`
   Produces the image, applies Sepia Filter and then outputs the mirror image of the filtered image.
2. `BufferImageInterop -h`
   This command prints the help file.

1.3 Command Line Options

Table 1 lists, and briefly describes, the command line options.

Table 1 Command Line Options

Short Form	Long Form	Description
-h	--help	Shows all command options and their respective meanings.
	--device [cpu|gpu]	Devices on which the OpenCL kernel is to be run. Acceptable values are cpu or gpu.
-q	--quiet	Quiet mode. Suppresses all text output.
-e	--verify	Verify results against reference implementation.
-t	--timing	Print timing-related statistics.
-v	--version	AMD APP SDK version string.
	--dump [filename]	Dump the binary image for all devices.
	--load [filename]	Load the binary image and execute on the device.
	--flags [filename]	Specify the filename containing the compiler flags for building the kernel.
-i	--iterations	Number of iterations.
-p	--platformId	Select the platformId to be used[0 to N-1 where N is number platform s available].
-d	--deviceId	Select deviceId to be used[0 to N-1 where N is number devices available].

2 Introduction

This sample demonstrates the interop between the OpenCL Image object and the buffer object. This interop is just a wrapper and there is no additional data copy involved. The following extensions support this Buffer- Image interop:

1. cl_khr_image2d_from_buffer
2. cl_khr_image2d_from_buffer_read_only

cl_amd_image2d_from_buffer_read_only (on AMD platforms)

The sample workflow is as follows:

1. Read the input image into an OpenCL buffer.
2. Apply the Sepia tone filter on the image pixel data.
3. Wrap the output buffer of Sepia tone around as an image. For this task, the buffer is passed as an input to the `clCreateImage` function by using the `cl_image_desc` structure.
4. Reverse the image created as above.
5. Save the output image.

The Image buffer interop is a core feature in the OpenCL 2.x platform, while it was an optional extension in OpenCL 1.x. Based on the device's OpenCL version, the sample checks for the image-buffer interop extension for 1.x platforms and skips this test for 2.x platforms.

3 Implementation

The input image is first read into an OpenCL buffer. The Sepia tone Filter is applied on this image data. The pseudo-code for Sepia tone is as follows:

```
for each inputPixel  in the Image
do
outputRed = (inputRed * .393) + (inputGreen *.769) + (inputBlue * .189)
outputGreen = (inputRed * .349) + (inputGreen *.686) + (inputBlue * .168)
outputBlue = (inputRed * .272) + (inputGreen *.534) + (inputBlue * .131)
      write outputPixel(outputRed,outputGreen,OutputBlue)
end for
```

The buffer image interop is achieved by wrapping the output buffer of Sepia tone around an image object. For this task, the buffer is passed as input to the `clCreateImage` function by using the `cl_image_desc` structure.

```
imageDesc.buffer = outputImageBuffer;
```

The image thus created is reversed. The pseudocode for Image Reverse is as follows:

```
for each p(x,y) in inputimage
do
//where x varies from 1 to w and
//y varies from 1 to h
Write P (x,y) to P1(x,y ) in outputimage
Write P(x,y) to P2(w -x,y) in outputimage
```

```
end for
```

4 References

1. The OpenCL Extension Specification (ver 1.2, rev 19) document (Page 129).

2. Sepia tone filter: http://xjaphx.wordpress.com/2011/06/21/image-processing-photography-sepia-toning-effect/.

Contact

Advanced Micro Devices, Inc.
One AMD Place
P.O. Box 3453
Sunnyvale, CA, 94088-3453
Phone: +1.408.749.4000

For AMD Accelerated Parallel Processing:
URL: developer.amd.com/appsdk
Developing: developer.amd.com/
Support: developer.amd.com/appsdksupport
Forum: developer.amd.com/openclforum

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. The information contained herein may be of a preliminary or advance nature and is subject to change without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

Copyright and Trademarks