AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Image Overlap

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.

Install the OpenCL 1.2.

Use the command line to change to the directory where the executable is located. The pre-compiled 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. `ImageOverlap`
   This executes a copy operation using the GPU texture unit (Image), as well as fill operations on the image object.
2. `ImageOverlap -h`
   This 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 meaning.
	`--device`	Devices on which the program 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.
	`--dump`	Dump binary image for all devices.
	`--load`	Load binary image, and execute on devices.
	`--flags`	Specify compiler flags to build the kernel.
`-p`	`--platformId`	Select platformId to be used (0 to N-1, where N is the number of available platforms).

Short Form	Long Form	Description
-d	--deviceID	Select device Id to be used (0 to N-1, where N is the number of available devices).
-v	--version	AMD APP SDK version string.
-i	--iterations	Number of iterations for kernel execution.

2 Introduction

This sample shows how to load a .bmp file into the GPU using image_2d. It also shows how to fill a certain location with one color using clEnqueueFillImage. The kernel execution overlaps two images by adding the RGB value of every pixel: (R1 + R2) (G1 + G2) (B1 + B2).

Figure 1 shows the two images before overlapping.

Figure 1 Images Before Overlapping

Figure 2 shows the result of the overlapping of the two images in Figure 1.

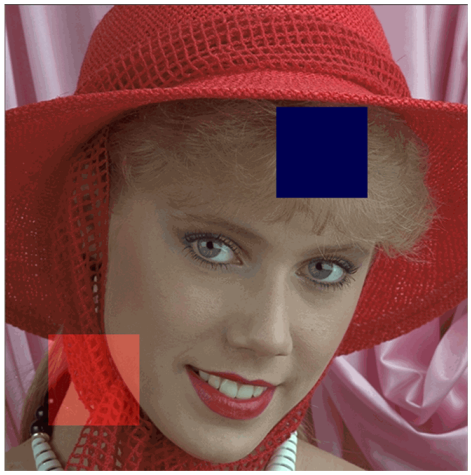

Figure 2 Resulting Overlapped Images

3 Implementation Details

Copy one image into the GPU using `clCreateImage`. Then, fill a 100x100 blue block on the image using `clEnqueueFillImage`. Fill an empty image with a red block. Use `clEnqueueMarkerWithWaitList` to wait until the two fill operations are completed; then, overlap the two images.

4 Environment

This Sample must run in the OpenCL 1.2 environment. The following APIs are from OpenCL 1.2:

- `clCreateImage`

  Creates a 1D image, 1D image buffer, 1D image array, 2D image, 2D image array and 3D image object, based on different flags.

- `clEnqueueFillImage`

  Enqueues a command to fill an image object with a specified color.

- `clEnqueueMarkerWithWaitList`

  Enqueues a marker command that waits for a list of events to complete. If the list is empty, it waits for all commands previously enqueued in command queue to complete before it completes.

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