AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

OpenCL - DX11 Interoperability

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 default executables are placed in `$<APPSDKSamplesInstallPath>\samples\opencl\bin\x86` for 32-bit builds and `$<APPSDKSamplesInstallPath>\samples\opencl\bin\x86_64\` for 64-bit builds.

Before building this sample:

- Ensure you have the DirectX SDK or the SDK for Windows 8 installed.

Type the following command(s).

1. `SimpleDX11`
   Produces the image of a moving sine wave with a bitmap in the background.
2. `SimpleDX11 -h`
   This prints the help message.

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`	Display all command options and their respective meanings.
`-q`	`--quiet`	Quiet mode. Suppresses all text output.
`-e`	`--verify`	Verify results against reference implementation.
	`--dump`	Dump binary image for all devices.
	`--load`	Load binary image, and execute on device.
	`--flags`	Specify compiler flags to build the kernel.
`-p`	`--platformId`	Select the platform ID to be used (0 to N-1, where N is the number of available platforms).
`-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.

2 Implementation Details

You must have a graphics driver and an OpenCL runtime that support the Direct3D 11 media sharing extension. To verify that they are installed, run `clinfo` from the command line, and check if your device supports the `cl_khr_d3d11_sharing` extension.

The following are the key functions of SimpleDX11:

```
clCreateFromD3D11BufferKHR

clEnqueueAcquireD3D11ObjectsKHR

clEnqueueReleaseD3D11ObjectsKHR

clGetDeviceIDsFromD3D11KHR
```

3 Algorithm

The following equation is used to compute the sine wave shown in the program:

```
y = sin(x + t) × cos(x + t);
```

where x is the position on the X axis, and t is the time.

4 Sample Design

The input of this sample is a `.bmp` picture. This example reads in the data of the picture, calculates a sine sector using the OpenCL kernel, and renders and shows this picture using D3D11.

Steps:

1. Create a D3D11 offscreen surface using the D3D11 API `D3D11CreateDeviceAndSwapChain`, and initialize DirectX11 as shown in Figure 1.

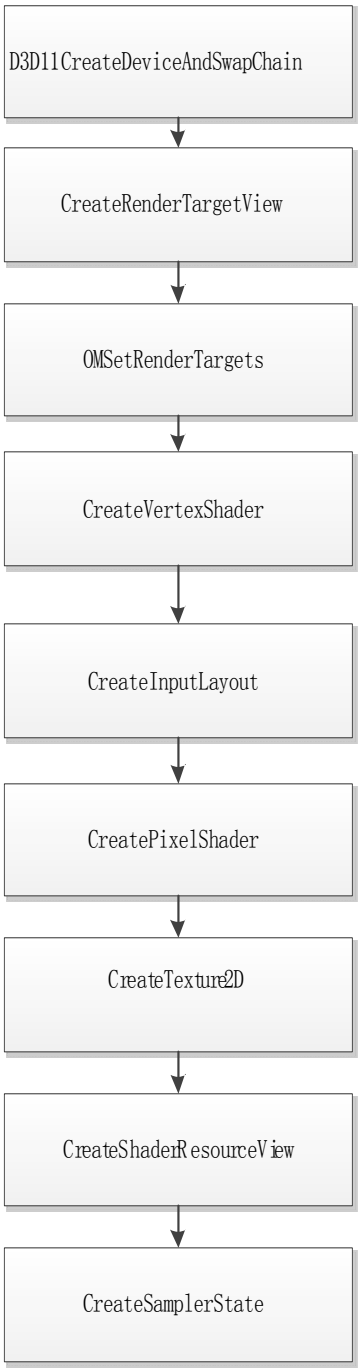


Figure 1 Initializing DX

2. Set up the Opencl environment as shown in Figure 2. Use `vetexBufPtr` to create an ID3D11Buffer with a pointer. Use `IASetVertexBuffers` to bind it to the device. Use `pfn_clCreateFromD3D11BufferKHR` to create an Opencl buffer with DirectX11 for sufrace sharing.

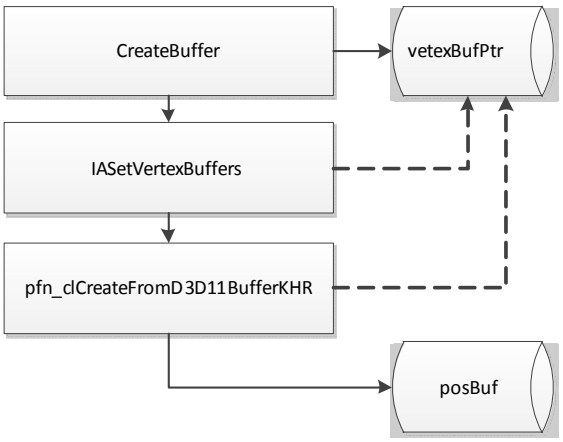


Figure 2 Setting Up CL

3. Run the kernel as shown in Figure 3. Before invoking `clEnqueueNDRangeKernel`, use `pfn_clEnqueueAcquireD3D11ObjectsKHR` to acquire the D3D11 object. After using `clEnqueueNDRangeKernel`, use `pfn_clEnqueueReleaseD3D11ObjectsKHR` to release the D3D11 object.

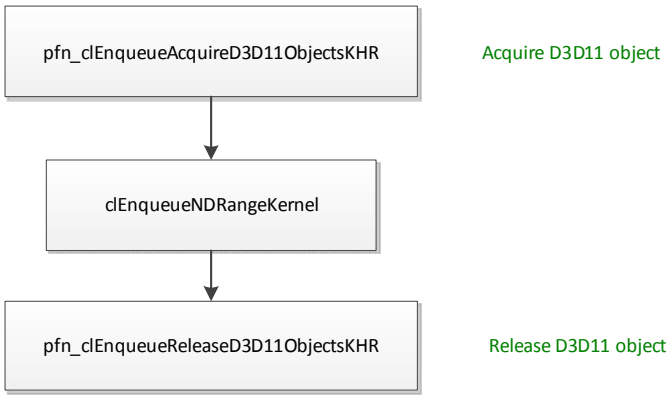
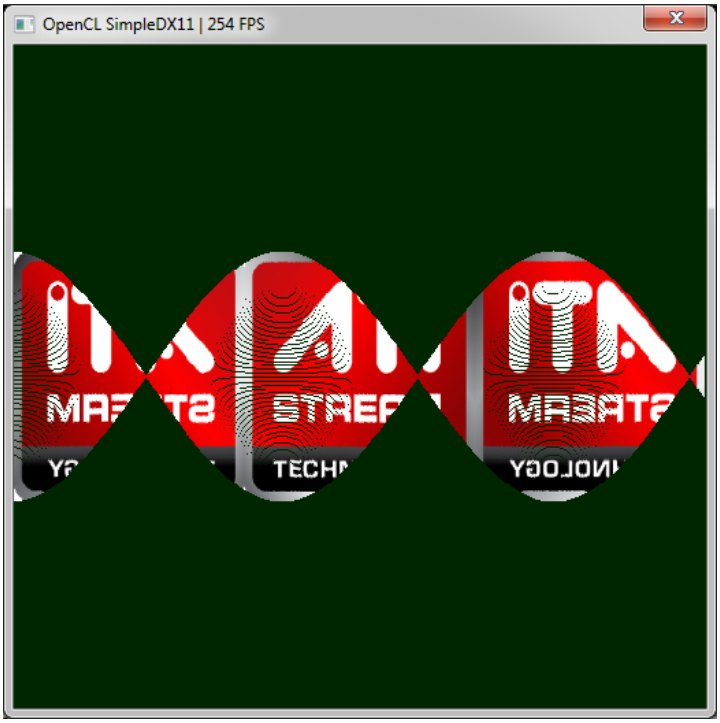


Figure 3 Running the Kernel

4. Copy the picture surface into the texture surface, and set the vertex for rendering.
5. Call the D3D11 function `present` to show the picture.

5 Sample Effect

The screen shots below show sample effects.



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/
Forum: developer.amd.com/openclforum