AMD **Accelerated** Parallel Processing TECHNOLOGY

# SAMPLE

# Device Fission 11 Ext

## 1 Overview

Note that this sample is essential the same as the DeviceFission sample, except that this sample uses OpenCL 1.1 and the cl_ext_device_fission extension; the DeviceFission sample uses the OpenCL 1.2 device fission functionality.

### 1.1 Location

`$(AMDAPPSDKSAMPLESROOT)\samples\opencl\cl\app`

### 1.2 How to Run

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

Ensure that the OpenCL 1.2 environment is installed.

Use the command line to change to the directory where the executable is located. The default executables are placed in `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86` for 32-bit builds and `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `DeviceFission`
   This tests the kernel execution on multi-devices asynchronously with default option `-x 1024 -g 1.`.
2. `DeviceFission -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` | 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 the CPU.. |
| | `--loadgpu` | Load GPU binary image, and execute on the GPU. |
| | `--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 |
|---|---|---|
| -v | --version | AMD APP SDK version string. |
| -x | --length | Length of the input array. |
| -d | --deviceId | Select deviceId to be used (0 to N-1, where N is the number of available devices). |
| -g | --cpu2gpu | Migrate the memory object from a sub-device to the GPU before executing kernels. 0 is disable; 1 is enable. |
| -c | --cpu2cpu | Migrate the memory object from one sub-device to another sub-device before executing kernels. 0 is disable; 1 is enable. |

# 2 Introduction

In this sample, a CPU device is partitioned into two sub-devices by using `clCreateSubDevices`. We create only one input buffer. One of the sub-devices is in charge of writing data into the input buffer; then, two CPU sub-devices (and, if it exists, the GPU) must execute the kernels using only the data from that input buffer.

Typically, the device writing to the input buffer has the highest priority to use the buffer. To change this priority, use `clEnqueueMigrateMemObjects` to move the input buffer and choose which device has the higher priority.

# 3 Environment

This sample must be run in the OpenCL 1.2 environment. The following APIs are part of OpenCL 1.2:

- `clCreateSubDevices`
  Creates an array of sub-devices that each reference a non-intersecting set of compute units within a GPU, according to a partition scheme given by the API parameter *Properties*.
- `clEnqueueMigrateMemObjects`
  Enqueues a command to indicate the device with which a set of memory objects are to be associated.

IMPORTANT:

- When running this sample, if there are no GPU device, it uses a CPU to replace the GPU.
- The cpu2gpu mode is not supported if there is no GPU.
- When loading a binary image for the CPU using `--load`, ensure that the binary image for the GPU is loaded at the same time using `--loadgpu`.
- The cpu2gpu and cpu2cpu modes cannot be enabled concurrently. To enable cpu2cpu, ensure cpu2gpu is disabled first.

**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