AMD **Accelerated** Parallel Processing
TECHNOLOGY

SAMPLE

Range Minimum Query

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. Ensure that the OpenCL 2.0 environment is installed.

Type the following command(s).

1. `RangeMinimumQuery`
   This command runs the program with the default options.
2. `RangeMinimumQuery -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].
-x	--numInputs	Array Length

Short Form	Long Form	Description
`-m`	`--startIndex`	Start Index for RMQ <default value for m = 10>
`-n`	`--endIndex`	Last Index for RMQ <default value for n=99999999>
`-l`	`--localsize`	Number of work-items per work-group (should be 2 ^ N)

2 Introduction

This sample demonstrates the ability in OpenCL 2.0 and higher versions to pass a pointer with an offset as a kernel argument. This was a restriction in OpenCL 1.2.

The sample uses `clSetKernelArgSVMPointer` to pass a Shared Virtual Memory (SVM) pointer with a specific offset as an argument value to kernel.

The sample also demonstrates another feature of OpenCL 2.0, which allows the NDRange to be flexible and a non-multiple of the workgroup size (unlike in OpenCL 1.2, in which NDRange needed to be a multiple of the workgroup size).

This sample also demonstrates how to use `work_group_reduce_<op>`, a work-group level built-in function in OpenCL 2.0.

3 Implementation

This sample implements the Range Minimum Query (RMQ) problem.

3.1 RMQ Problem

Given an array A[0…n-1] of n objects, a range minimum query or RMQ asks for position of a minimum element in some sub-array A[i…j].

For example:

Given A = [6, 5, 2, 5, 4, 3, 1, 9, 3], then answer to RMQ for A [3, 7] = [5, 4, 3, 1, 9] is 6, as A [6] = 1.

3.2 Implementation

The aim is to find minimum value in a given range/sub-set of the input values. The sample first creates a SVM buffer and populates the buffer with some random values. The SVM pointer along with the specified offset is passed as an argument value to *Range Minimum Query* kernel.

By default, the offset is set to 10, SVM pointer value specified as the argument value is (pointer+10), kernel launches (n-1 – offset = n-11) work-items, and returns minimum value within A [10, n-1]. User can also specify the range of sub-array within which minimum value has to be found. In this case, if user specifies some range, say for example A [i, j] : 0 <= i <= j < n, then SVM pointer value specified as the argument value would be (pointer+i), kernel would launch (j + 1 - i) work-items, and return minimum value within A [i, j].

The sample uses `work_group_reduce_min`, a work-group level built-in function to find minimum value within a work-group.

4 References

1. https://www.khronos.org/registry/cl/specs/opencl-2.0.pdf
2. https://www.khronos.org/registry/cl/specs/opencl-2.0-openclc.pdf
3. http://wcipeg.com/wiki/Range_minimum_query

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