AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Radix Sort

1 Overview

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.

Use the command line to change to the directory where the executable is located. The pre-compiled sample executable is at `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86\` for 32-bit builds, and `$(AMDAPPSDKSAMPLESROOT)\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `RadixSort`
   Runs with default option x = 8192.
2. `RadixSort -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 device.
	--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).
-d	--deviceId	Select deviceId to be used (0 to N-1, where N is the number of available devices).
-x	--count	Element count.
-i	--iterations	Number of iterations for kernel execution.

2 Introduction

Radix-based sorting algorithms treat keys as multi-digit numbers in which each digit is an integer with a value ranging from 0 to *m*, where *m* is the radix. A 32-bit integer, for example, could be treated as a 4-digit number with radix m = $2^{32/4} = 2^8 = 256$. Radix sort works by breaking keys into digits and sorting one digit at a time, starting with the *least* significant digit. The radix *m* is usually chosen to minimize the running time; it is highly dependent on the implementation and the number of keys being sorted.

The Radix Sort algorithm is divided into 3 phases:

1. Calculate the histogram of an unsorted array.
2. Prescan the histogram bins.
3. Rank and permute to keys to get a sorted array.

See reference [1] for more details on serial and parallel Radix Sort algorithms.

3 Implementation Details

The implemented Radix sort breaks keys (32 integers) into 8-bit digits and sorts one 8-bit digit at a time, starting with the least significant digit. It loops four times to complete sorting. In each i^{th} loop, the following three phases sort the input array using i^{th} 8-bit digit.

1. Calculate histogram bins.

   The input array is divided into blocks of *N* * *M* elements. Where *M* is the radix (*M* is 256 for an 8-bit digit), and *N* is the number of work-items in a block. In this case, *N* = 16. Each work-item calculates its histogram bin from the allotted 256 elements and passes this histogram to next phase.

2. Prescan histogram bins.

   In this phase the histogram bins are prescanned column-wise, where histogram bins are arranged in the following way.

   There are *B* * *N* histogram bins, where *B* is the number of blocks, and *N* is the number of work-items in a block. Histogram bins are arranged such that the 0^{th} block bin comes first, and the $(B - 1)^{th}$ block comes last. Each block's histogram bins are arranged so that the 0^{th} work-item bin comes first, and $(N - 1)^{th}$ work-item bin comes last.

   The prescanned histogram is passed to next phase.

3. Rank and permute keys to get the sorted array.

   Eack work-item permutes the allotted 256 elements by using its prescanned histogram bins.

4 References

1. Marcho Zagha and Guy E. Blelloch. "Radix Sort For Vector Multiprocessor." in: *Conference on High Performance Networking and Computing*, pp. 712-721, 1991.
2. Guy E. Blelloch, *Prefix Sums and Their Applications*, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1990.

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**