AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Sobol’s Quasi-Random Sequence

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 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. `QuasiRandomSequence`
   This runs the sample with default options –x 64 and –y 1.
2. `QuasiRandomSequence -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).
`-y`	`--dimension`	Number of dimensions.
`-x`	`--vector`	Number of vectors.
`-i`	`--iterations`	Number of iterations for kernel execution.

2 Description

To generate the j^{th} component of the points in a Sobol sequence, choose a primitive polynomial of some degree Sj in the field Z_2:

$$X^{Sj} + a_{1,j} \; X^{Sj-1} + a_{2,j} \, X^{Sj-2} + \ldots\ldots + a_{Sj-1,j} \; x + 1$$

where the coefficients $a_{1,j}$, $a_{2,j}$, ..., $a_{sj-1,j}$ are either 0 or 1. We define a sequence of positive integers { $m_{1,j}$, $m_{2,j}$, ... } by the recurrence relation

$$m_{k,j} = 2a_{1,j}m_{k-1,j} \oplus 2^2 a_{2,j}m_{k-2,j} \; \oplus \ldots\ldots \; \oplus 2^{Sj-1} \; a_{Sj-1,j}m_{k-Sj+1,j} \oplus 2^{Sj} \; m_{k-Sj,j} \; \oplus m_{k-Sj,j}$$

where ⊕ is the bit-by-bit exclusive-or operator. The initial values $m_{1,\,j}$, $m_{2,\,j}$, ... $m_{Sj,\,j}$ can be chosen freely, provided each $m_{k,\,j}$, $1 \le k \leqq Sj$, is odd and less than 2k. The so-called direction numbers { $V_{1,j}$, $V_{2,j}$, ... } are defined by $V_{k,j} = \frac{m_{k,j}}{2^k}$.

Then $x_{i,j}$, the j^{th} component of the i^{th} point in a Sobol sequence, is given by

$$x_{i,j} = i_1 v_{1,j} \; \oplus i_2 v_{2,j} \oplus .....$$

where i_k is the k^{th} digit from the right when i is written in binary i = { ...... i_3 i_2 i_1 } $_2$.

The primitive polynomials and direction numbers obtained based on various search criteria can be downloaded as text files from `http://www.maths.unsw.edu.au/~fkuo/sobol/` .

3 Implementation Details

Each thread block processes direction numbers of a single block, which represents a particular dimension, j.

First, all the precomputed direction numbers { $V_{1,j}$, $V_{2,j}$, ... } are cached into a local memory buffer from an array in global memory. Then, each work item calculates a Sobol's sequence value by using the equation

$$x_{i,j} = i_1 v_{1,j} \; \oplus i_2 v_{2,j} \oplus .....$$

where i_k is the k^{th} digit from the right when i is written in binary i = { ...... i_3 i_2 i_1 } $_2$, and i is the `local_id` of the work item.

4 Recommended Input Option Settings

For best performance, enter the following on the command line:

```
-d 10000 -x 1024 -i 10 -q -t
```

5 Reference

P. Bratley and B. L. Fox, “Algorithm 659: Implementing Sobol’s quasirandom sequence generator" in:ACM Transactions on Mathematical Software (TOMS), Vol. 14, Issue 1 (March 1988).

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