ATI STREAM COMPUTING SAMPLE

Sobol's Quasi-Random Sequence

1 Overview

1.1 Location

`$(ATISTREAMSDKSAMPLESROOT)\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 `$(ATISTREAMSDKSAMPLESROOT)\samples\opencl\bin\x86` for 32-bit builds and `$(ATISTREAMSDKSAMPLESROOT)\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.
`-q`	`--quiet`	Quiet mode. Suppresses all text output.
`-e`	`--verify`	Verify results against reference implementation.
`-v`	`--verbose`	Verbose output.
`-t`	`--timing`	Print timing.
`-x`	`--width`	Specify number of vectors.
`-y`	`--height`	Specify number of dimensions.
	`--device`	Devices on which the program is to be run. Acceptable values are cpu or gpu.

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 $\oplus$ 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 $V_{k,j} = \dfrac{m_{k,j}}{2^k}$.

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

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

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

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

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 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 Stream Computing:
URL: www.amd.com/stream
Questions: streamcomputing@amd.com
Developing: ATI_Stream_SDK_Help_Request
Forum: www.amd.com/streamdevforum