AMD **Accelerated** Parallel Processing TECHNOLOGY

SAMPLE

Matrix Transpose

1 Overview

1.1 Location

`$<AMDAPPSDKSamplesInstallPath>\samples\opencl\cl\1.x`

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 `$<AMDAPPSDKSamplesInstallPath>\samples\opencl\bin\x86` for 32-bit builds and `$<AMDAPPSDKSamplesInstallPath>\samples\opencl\bin\x86_64\` for 64-bit builds.

Type the following command(s).

1. `MatrixTranspose`
   Transforms an abitrary array of 64.
2. `MatrixTranspose -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 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).
`-v`	`--version`	AMD APP SDK version string.
`-x`	`--width`	Width/height dimension of the square matrix.
`-b`	`--blockSize`	Use local memory of dimensions blockSize x blockSize.
`-i`	`--iterations`	Number of iterations for kernel execution.

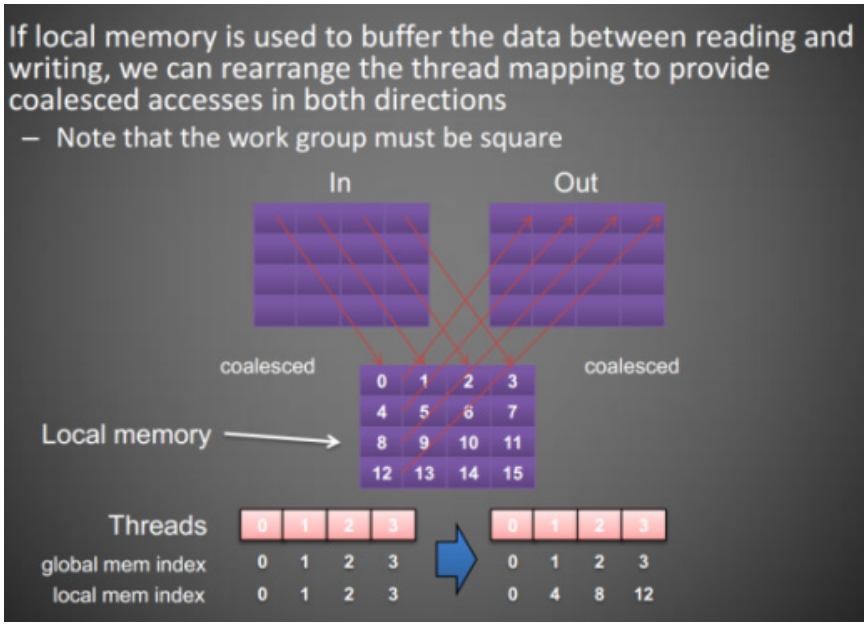
2 Introduction

In linear algebra, the transpose of a matrix A is another matrix A^T (also written A', A^{tr} or A^t). Formally, the i^{th} row, j^{th} column element of A^T is the j^{th} row, i^{th} column element of A.

If A is an m × n matrix then A^T is an n × m matrix.

3 Implementation

The implementation takes by default the workgroup of size 256 in 16 X 16 format. If the first row of work-items reads the input matrix in a coalesced manner, the elements read must be written into different rows but into the same column. Such an implementation will cause global memory channel conflicts and will be very slow.



To handle this situation, Local Data Share (LDS) is used as an intermediary buffer. A workgroup puts a block of memory from the input matrix into this LDS block. This write can occur in a coalesced manner. The elements of the block are transposed inside the LDS. Even this approach will cause LDS bank conflicts, but the conflicts are far less severe. The LDS block can now be written back to the global memory in a coalesced manner, as the matrix is already transposed.

4 References

1. http://en.wikipedia.org/wiki/Transpose
2. https://www.cvg.ethz.ch/teaching/2011spring/gpgpu/GPU-Optimization.pdf

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