GPGPU Problem Sheet 1
High Performance Computing

Prof. Dr. V. Lindenstruth
D. Rohr

SS 2010

1 . Exercise .

a) *Take the OpenCL sample provided for vector addition and alter it to do a matrix multiplication $C = A \cdot B$. For simplicity restrict to square matrices of size $n \times n$. Store the matrices in a linear array such that the $(i,j)^{-th}$ entry can be accessed at $matrix[j \cdot n + i]$. Use 2-dimensional work-groups and start n^2 work-items. Each work item shall calculate exactly one entry of C determined by its global id.*

b) *Benchmark your program for $n = 1024$ and the following work-group sizes:* (512, 1), (256, 2), (128, 4), (64, 8), (32, 16), (16, 32), (8, 64), (4, 128), (2, 256), (1, 512) *as well as* (256, 1), (128, 2), (64, 4), (32, 8), (16, 16), (8, 32), (4, 64), (2, 128), (1, 256). *Think of a way to visualize the memory access by work-items within one work-group that are executed simultaneously. Use the coalescing rules to explain the effect of the work-group size on performance.*

c) *What would happen if the matrix would be accessed in a transposed way, i.e. the $(i,j)^{-th}$ entry would be $matrix[i \cdot n + j]$?*

2 . Exercise .

As a more complex example add a local memory cache to you matrix multiplication program. Take a fixed work-group size of (16, 16). *Consider the work-group calculating the sub-matrix $\tilde{C}_{k,l}$ of C, with $\tilde{C}_{k,l} = \{c_{i,j} \mid 16 \cdot k \leq i < 16 \cdot (k+1), 16 \cdot l \leq j < 16 \cdot (l+1)$. The calculation of $\tilde{C}_{k,l}$ can be split into steps of calculating products of sub-matrices of A and B, i.e.*

$$\tilde{C}_{k,l} = \sum_{s=0}^{\frac{n}{16}-1} \tilde{A}_{k,s} \cdot \tilde{B}_{s,l}.$$

The submatrices $\tilde{A}_{k,s}$ and $\tilde{B}_{s,l}$ can be cached in a local memory cache of 256 *single precision floats. Try to implement such a matrix multiplication variant and benchmark it against the version that was implemented before. Refer to the OpenCL specification to learn how to declare variables in local memory and how to add barriers where the work-items within one work-group are synchronized.*