CO316-Assignment3

Hardik Rana-16CO138
Harshal Shinde-16CO223

QUESTION1 - RGB TO GRAYSCALE

1.How many floating operations are being performed in your color conversion kernel?

Ans: There are total **Width * Height * 5 floating operations** being performed in our color conversion kernel where we have 3 multiplications and 2 additions per RGB pixel.

2.Which format would be more efficient for color conversion: a 2D matrix where each entry is an RGB value or a 3D matrix where each slice in the Z axis represents a color. I.e. is it better to have color interleaved in this application? can you name an application where the opposite is true?

Ans: 2D matrix is better for applications like color conversion where data from all the three channels are required for every pixel. However for applications like blurring an image where data from different channels are required separately, it's convenient to store them in different layers and parallelly process them separately.An example where 3D matrix is better would be an **image manipulation software** working separately on different color channels.

3.How many global memory reads are being performed by your kernel?

Ans: There are total **Width * Height** global memory reads being performed by our kernel.If each color is read separately, then we have total **Width *Height * (No. Of Channels)** global memory reads.

4.How many global memory writes are being performed by your kernel?

Ans: There are total **Width * Height** writes being performed by our kernel.

5.Describe what possible optimizations can be implemented to your kernel to achieve a performance speedup.

Ans: SIMD instructions can be implemented. Branching statements like if can be removed and two kernel functions can be used instead.

QUESTION2 -MATRIX MULTIPLICATION OF TWO LARGE MATRICES

Assumption:

K is the common dimension of the two matrices. i.e. M x K and K x N are the dimensions of the input matrices

1.How many floating operations are being performed in your matrix multiply kernel?

Ans: We will have total **M * N * (2K-1)** floating operations in our matrix multiply Kernel, where we have **K multiplications** and **K-1 additions** per cell in the output matrix.

2.How many global memory reads are being performed by your kernel?

Ans: There are total **2 * M * N * K** global memory reads being performed by our kernel.

3.How many global memory writes are being performed by your kernel?

Ans: There are total **M*N** writes being performed by our kernel.

4.Describe what possible optimizations can be implemented to your kernel to achieve a performance speedup.

Ans: This can optimized by using shared memory to implement tiled matrix Multiplication.

QUESTION3 -TILED DENSE MATRIX MULTIPLICATION

Assumption:

K is the common dimension of the two matrices. i.e. M x K and K x N are the dimensions of the input matrices

1.How many floating operations are being performed in your matrix multiply kernel?explain.

Ans: We will have total **M * N * (2K-1)** floating operations in our matrix multiply Kernel, where we have **K multiplications** and **K-1 additions** per cell in the output matrix.

2.How many global memory reads are being performed by your kernel?explain.

Ans: There are total **M * N * K/(Tile Width)** global memory reads being performed by our kernel.Tiling reduces global memory reads by a factor of the tile width.

3.How many global memory writes are being performed by your kernel?explain.

Ans: There are total **M*N** writes being performed by our kernel.

4.Describe what further optimizations can be implemented to your kernel to achieve a performance speedup.

Ans: The Multiplications can also run parallely. The sum can be then calculated using parallel reduction for each cell in the output matrix.

5.Compare the implementation difficulty of this kernel compared to the previous MP. What difficulties did you have with this implementation?

Ans: The major difficulty involved working with border cases. Finding when the threads should be synced was also difficult.

6.Suppose you have matrices with dimensions bigger than the max thread dimensions. Sketch an algorithm that would perform matrix multiplication algorithm that would perform the multiplication in this case.

Ans: Matrix can be divided into tiles with dimensions smaller than the thread limit. Multiple kernels calls can then be made to facilitate complete matrix Multiplication.

7.Suppose you have matrices that would not fit in global memory. Sketch an algorithm that would perform matrix multiplication algorithm that would perform the multiplication out of place.

Ans: A version of tiled matrix multiplicaton can be used. Load only a tile onto the global memory and split it into further subtiles using shared memory. Call the kernel several times to operate on each tile.