General Purpose Computing on Graphic Processing Units (GPGPU) For HPC
EE 5885 – 04, TR: 9:35 AM – 10.50 AM, EN 5022
Dept. of Electrical Engineering and Computer Science, Credits: 3.0
TEST 2, Part 2, Format: Take Home, Duration: 12/03/2022 by 4:00 PM
Points: 50

Segmented Parallel Scan Algorithm

Template Project Name: SegmentedParallelScan

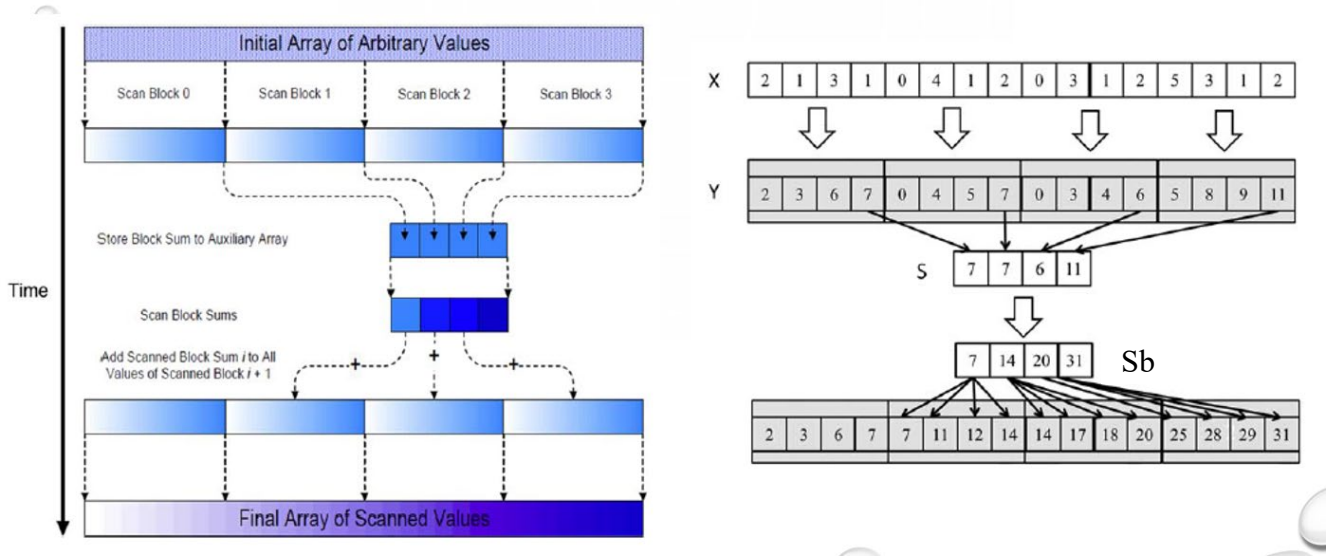

Figure 1 Segmented Parallel Scan Architecture

Information:

- Predefined constant ***B*** depicts the block size. The block sizes can be 256 or 512 or 1024
- Predefined constant ***VECTOR_SIZE*** is the product of ***B*** and ***Z***, where ***Z*** is an integer multiplier
- The value of the ***Z*** has to be less than or equal to ***B***
- __global__ void SingleBlockScan(float* In, float* Out, const int SIZE)
  GPU Kernel to perform a parallel scan of only a single block with a maximum of 1024 elements using the Efficient Kogge Stone Algorithm. It is the same kernel that was implemented during the class and can only scan a single block

- __global__ void MultipleBlockScanIntermediateOutput(float* In, float* AuxOut, float *blockSum, const int SIZE)
  GPU Kernel to perform a scan of individual blocks and produce intermediate output and output of the last element in each block. This kernel should have Vector ***X*** as the input and produce Vectors ***Y*** and ***S*** in Figure 1.

- The vector ***Sb*** in Figure 1 should be generated using the SingleBlockScan kernel

- __global__ void AddBlockSumValues(float* AuxIn, float* blockSum, const int SIZE)
  GPU Kernel to add the block sum of the last element to the Intermediate Outputs to produce the final answer. This kernel should have Vectors ***Y*** and ***Sb*** as inputs and produce the final output vector in Figure 1 representing the Parallel Scan result.

Coding Tasks:

Implement code of the functions listed below:

- `__global__ void MultipleBlockScanIntermediateOutput(float* In, float* AuxOut, float *blockSum, const int SIZE)`
- `__global__ void AddBlockSumValues(float* AuxIn, float* blockSum, const int SIZE)`
- `__host__ void Helper_Scan(float* Input, float* Output, float* RefOutputData, int SIZE)`

Note: Examine the function OnInitializeInputData to help with debugging.

Submissions:

- Project with commented code (**Note: If the code implemented by you is not commented appropriately, a penalty of 10 points is assigned**) **Points: 35**

- Collect the performance data for the following data sizes: **Points: 10**

		Execution Times					
B	**Z**	CPU (msecs)	MultipleBlockScan (usecs)	SingleBlockScan (usecs)	AddBlock (usecs)	Total GPU (Only Kernel) Time (usecs)	Speedup
256	**64**						
256	**128**						
256	**256**						
512	**128**						
512	**256**						
512	**512**						
1024	**256**						
1024	**512**						
1024	**1024**						

- Explain the accuracy of the GPU computation with increasing vector size and determine the precision (decimal place) matching the results of the CPU computation. **Points: 5**