

Note: Make sure that CUDA is installed and working. For more information check: Lesson 7.10 Installing and running CUDA

- You are given the code for vector addition program in CUDA

Code is given

- Write a similar program for multiplying three vectors.

Solution is given

Some points to consider:

- Change the SIZE to different values and observe the results
 - If you make SIZE too big, make sure to change data types in the code appropriately
 - If not done, then integer variables used in loops or other parts will cause error
- Why do we need to do $(\text{int})\text{ceil}((\text{float})\text{SIZE} / \text{threads})$ and not just $\text{SIZE}/\text{threads}$?
 - If $\text{SIZE}/\text{threads}$ is done then there might be loss in integer conversion during division, this might result in lower number of threads than expected
 - So we convert to float and take the ceiling to get higher value for threads and cast to int
- Why do we need to do *if (thread_id < n)* in the kernel?
 - If we have higher number of threads than vector size, it might try to access vector index greater than the size of the vector
- Why do we need to do

$\text{thread_id} = \text{blockIdx.x} * \text{blockDim.x} + \text{threadIdx.x}$
and not $\text{thread_id} = \text{threadIdx.x};$

Since we are using block size greater than one, we need to get the correct global id using blockIdx and blockDim. If there was only one block, then thread_id=threadIdx.x will work.

Common Pitfalls for Students and Instructors

- Mistakes in proper allocation of memory in CPU and GPU
- Mistakes in writing kernel
- Students might be confused with using proper number of threads and blocks
- Students might make mistake in copying data from CPU to GPU or vice versa