# Assignment No. 04

**Title of the Assignment:** Write a CUDA Program for:
1. Addition of two large vectors
2. Matrix Multiplication using CUDA

**The objective of the Assignment:** Students should be able to learn about parallel computing and
students should learn about CUDA (**Compute Unified Device Architecture**) and how it helps to
boost high-performance computations.

**Prerequisite:**
1. Basics of CUDA Architecture.
2. Basics of CUDA programming model.
3. CUDA kernel function.
4. CUDA thread organization

**Contents of Theory:**
1. **CUDA architecture:** CUDA is a parallel computing platform and programming model developed by NVIDIA. It allows developers to use the power of GPU (Graphics Processing Unit) to accelerate computations. CUDA architecture consists of host and device components, where the host is the CPU and the device is the GPU.

2. **CUDA programming model:** CUDA programming model consists of host and device codes. The host code runs on the CPU and is responsible for managing the GPU memory and launching the kernel functions on the device. The device code runs on the GPU and performs the computations.

3. **CUDA kernel function:** A CUDA kernel function is a function that is executed on the GPU. It is defined with the global keyword and is called from the host code using a launch configuration. Each kernel function runs in parallel on multiple threads, where each thread performs the same operation on different data.

4. **Memory management in CUDA:** In CUDA, there are three types of memory: global, shared, and local. Global memory is allocated on the device and can be accessed by all threads. Shared memory is allocated on the device and can be accessed by threads within a block. Local memory is allocated on each thread and is used for temporary storage.

5. **CUDA thread organization:** In CUDA, threads are organized into blocks, and blocks are organized into a grid. Each thread is identified by a unique thread index, and each block is identified by a unique block index.

6. **Matrix multiplication:** Matrix multiplication is a fundamental operation in linear algebra. It involves multiplying two matrices and producing a third matrix. The resulting matrix has dimensions equal to the number of rows of the first matrix and the number of columns of the second matrix.

CUDA stands for **Compute Unified Device Architecture**. It is a parallel computing platform and programming model developed by NVIDIA.CUDA allows developers to use the power of

the GPU to accelerate computations. It is designed to be used with C, C++, and Fortran programming languages. CUDA architecture consists of host and device components. The host is the CPU, and the device is the GPU. The CPU is responsible for managing the GPU memory and launching the kernel functions on the device.

A CUDA kernel function is a function that is executed on the GPU. It is defined with the global keyword and is called from the host code using a launch configuration. Each kernel function runs in parallel on multiple threads, where each thread performs the same operation on different data.
CUDA provides three types of memory: global, shared, and local. Global memory is allocated on the device and can be accessed by all threads. Shared memory is allocated on the device and can be accessed by threads within a block. Local memory is allocated on each thread and is used for temporary storage.
CUDA threads are organized into blocks, and blocks are organized into a grid. Each thread is identified by a unique thread index, and each block is identified by a unique block index.
CUDA devices have a hierarchical memory architecture consisting of multiple memory levels, including registers, shared memory, L1 cache, L2 cache, and global memory.
CUDA supports various libraries, including cuBLAS for linear algebra, cuFFT for Fast Fourier Transform, and cuDNN for deep learning.
CUDA programming requires a compatible NVIDIA GPU and an installation of the CUDA Toolkit, which includes the CUDA compiler, libraries, and tools.

**Conclusion:**
Hence, we have implemented Addition of two large vectors and Matrix Multiplication using CUDA.