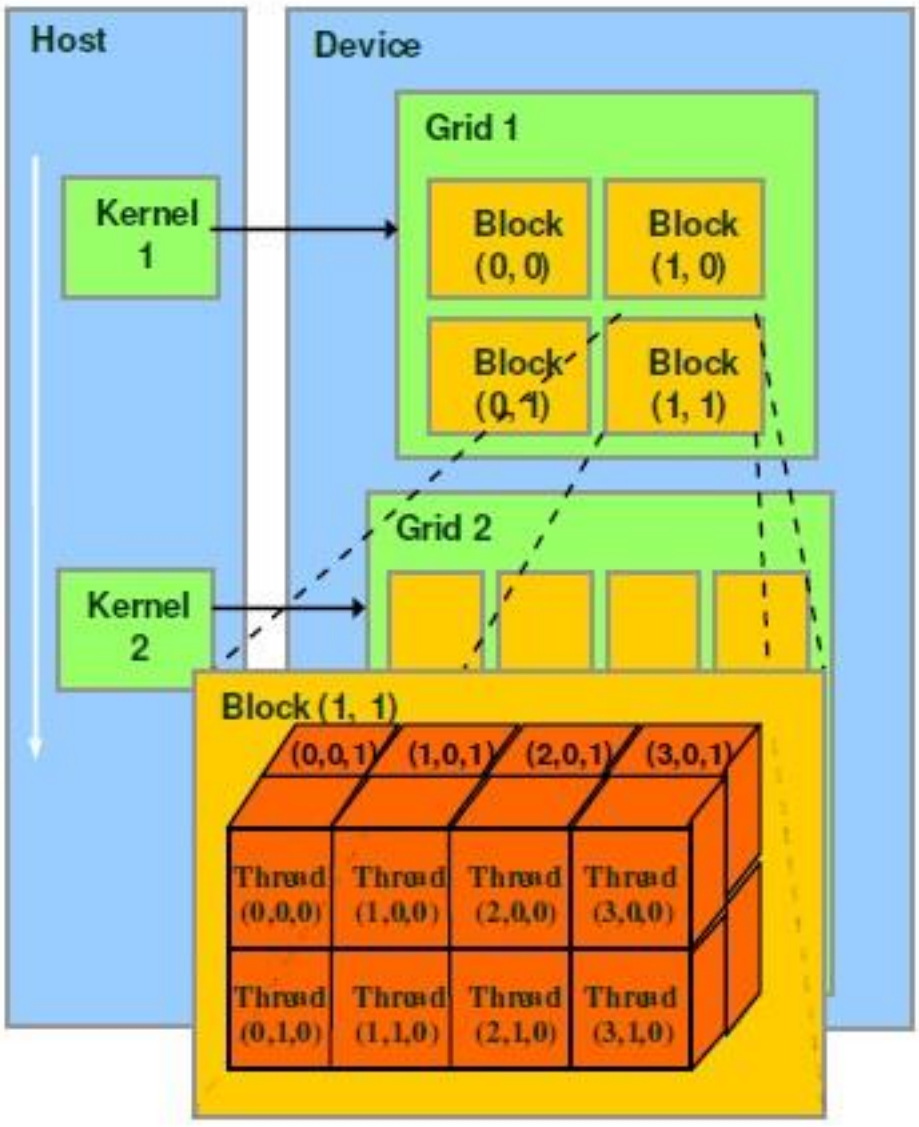
CUDA Thread Indexing sheet

Dept of MACS, NITK

If you are a CUDA parallel programmer but sometimes you cannot wrap your head around thread indexing in cuda environment



NVIDIA CUDA Thread Model

1D grid of 1D blocks

```
__device
int getGlobalIdx_1D_1D(){
      return blockIdx.x *blockDim.x + threadIdx.x;
}
```

1D grid of 2D blocks

```
__device
int getGlobalIdx_1D_2D(){
      return blockIdx.x * blockDim.x * blockDim.y
             + threadIdx.y * blockDim.x + threadIdx.x;
}
```

1D grid of 3D blocks

```
__device
int getGlobalIdx_1D_3D(){
      return blockIdx.x * blockDim.x * blockDim.y * blockDim.z
            + threadIdx.z * blockDim.y * blockDim.x
            + threadIdx.y * blockDim.x + threadIdx.x;
}
```

2D grid of 1D blocks

```
__device   int getGlobalIdx_2D_1D(){
      int blockId   = blockIdx.y * gridDim.x + blockIdx.x;
      int threadId = blockId * blockDim.x + threadIdx.x;
      return threadId;
}
```

2D grid of 2D blocks

```
 __device
int getGlobalIdx_2D_2D(){
      int blockId = blockIdx.x + blockIdx.y * gridDim.x;
      int threadId = blockId * (blockDim.x * blockDim.y)
                    + (threadIdx.y * blockDim.x) + threadIdx.x;
      return threadId;
}
```

2D grid of 3D blocks

```
__device
int getGlobalIdx_2D_3D(){
      int blockId = blockIdx.x + blockIdx.y * gridDim.x;
      int threadId = blockId * (blockDim.x * blockDim.y * blockDim.z)
                     + (threadIdx.z * (blockDim.x * blockDim.y))
                     + (threadIdx.y * blockDim.x) + threadIdx.x;
      return threadId;
}
```

3D grid of 1D blocks

```
__device
int getGlobalIdx_3D_1D(){
      int blockId = blockIdx.x + blockIdx.y * gridDim.x
                    + gridDim.x * gridDim.y * blockIdx.z;
      int threadId = blockId * blockDim.x + threadIdx.x;
      return threadId;
}
```

3D grid of 2D blocks

```
__device
int getGlobalIdx_3D_2D(){
      int blockId = blockIdx.x + blockIdx.y * gridDim.x
                    + gridDim.x * gridDim.y * blockIdx.z;
      int threadId = blockId * (blockDim.x * blockDim.y)
                     + (threadIdx.y * blockDim.x) + threadIdx.x;
      return threadId;
}
```

3D grid of 3D blocks

```
__device
int getGlobalIdx_3D_3D(){
      int blockId = blockIdx.x + blockIdx.y * gridDim.x
                     + gridDim.x * gridDim.y * blockIdx.z;
      int threadId = blockId * (blockDim.x * blockDim.y * blockDim.z)
                     + (threadIdx.z * (blockDim.x * blockDim.y))
                     + (threadIdx.y * blockDim.x) + threadIdx.x;
      return threadId;
}
```