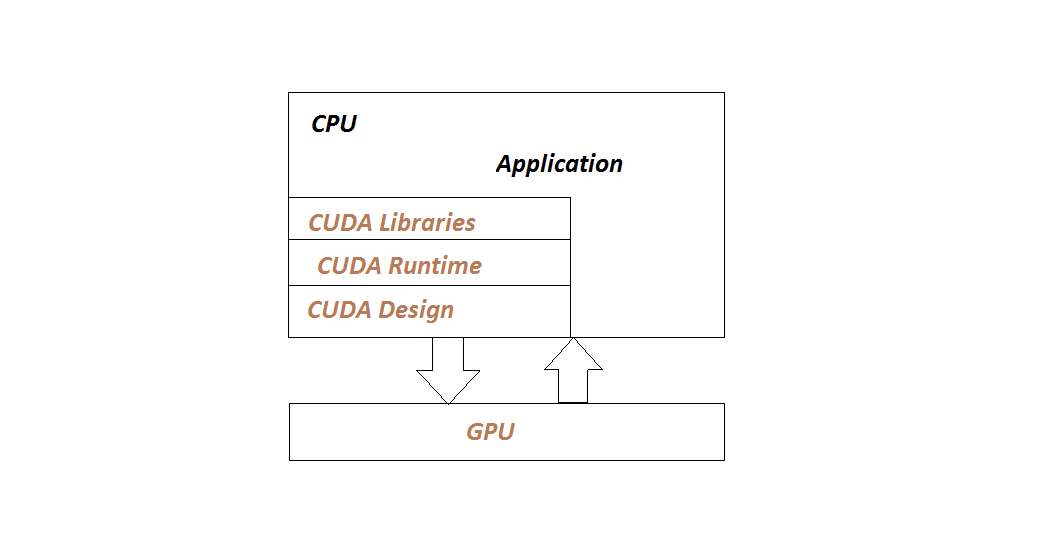
* CUDA Architecture:



* Programming Structure of GPU & CPU:
* CUDA Kernel:

The function which are executed on GPU are called as kernels.Kernels are full program or function invoke by the CPU and executed on GPU.A kernal is executed N number of times in parallel on GPU by using N number of threads.

Invocation: kernel\_name<<<grid,block>>>(argument,list);

kernel is defined as:

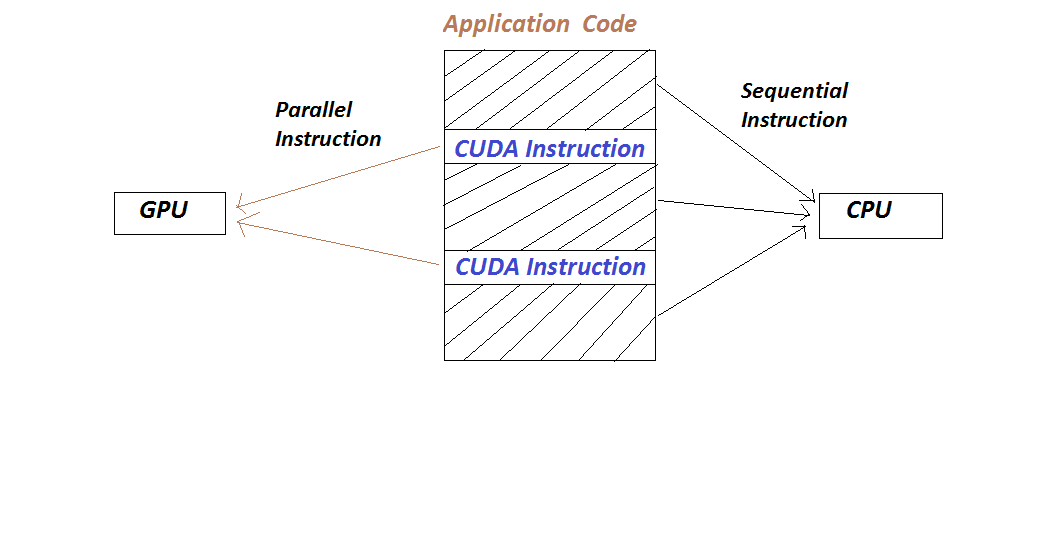
\_global\_voidk

ernel\_name(arguments)

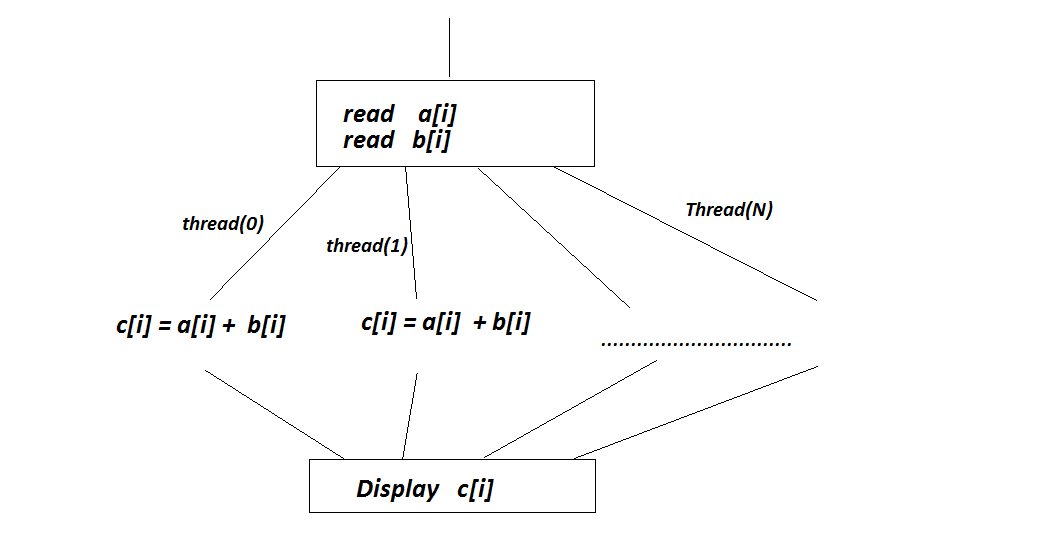
{

.........

}

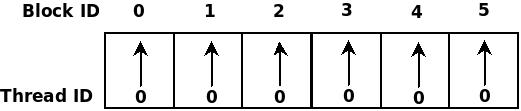


* Parallel Vector Addition:



Here three cases are considered for addition of two arrays:  
1. n blocks and one thread per block.  
2. 1 block and n threads in that block.  
3. m blocks and n threads per block.

**1. n blocks and one thread per block ( n=6)**

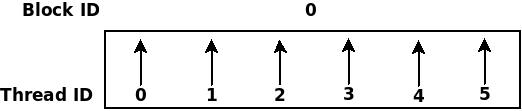


int id=blockIdx.x;

/\* blockIdx.x gives the respective block id which starts from 0 \*/

arradd<<<6,1>>>(d,e,f);

**2. One block and n threads in that block (In=6)**

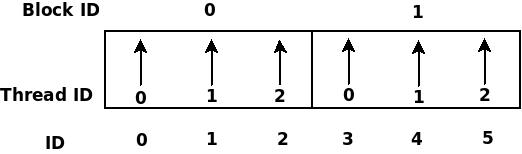


int id=threadIdx.x;

\* threadIdx.x gives the respective thread id which starts from 0 \*/

 arradd<<<1,6>>>(d,e,f);

**3. m blocks and n threads per block** **( m=2 and n=3)**



int id=blockIdx.x \* blockDim.x+threadIdx.x;  
/\* blockIdx.x gives the respective block id which starts from 0 \*/  
/\* threadIdx.x gives the respective thread id which starts from 0 \*/  
/\* blockDim.x gives the dimension of block i.e. number of threads in one block \*/

arradd<<<2,3>>>(d,e,f);