NAME: MOHAMMAD MANAN GHANI
ROLL NO. : 18BCS050
SUB : Parallel & Distributed Computing
Date : 10-12-2021

Qno.1) Ans

CUDA is a parallel computing platform and an API model. It can be used to utilize the power of GPUs to perform general computing tasks such as multiplying matrices & in performing other linear algebra operations, instead of just doing graphical calculations

CUDA program using C for matrix multiplications:

```
# include <stdio.h>
# include <math.h>

# define TILE_WIDTH

void Matrix_mul(float *Md, float *Nd, float *Pd,
                const int WIDTH)
{
    unsigned int column = TILE_width * blockIdx.x
                          + threadIdx.x;
    unsigned int row    = TILE_WIDTH * blockIdx.y
                          + threadIdx.y;
    for(int i=0 ; i<WIDTH ; i++)
    {
        Pd[row*WIDTH + colu] + Md[row*width + i]*
                                [i*width + column];
    }
}
```

Q no. 3) Ans

$$\text{Total clock cycles} = \sum_{i=1}^{1024} (2+2*i)$$

$$= 2*1024 + 1024*1025$$

$$= 1051648$$

Time reqd. for 1st processor $= \sum_{i=1}^{32} (2+2*i)$

// // // 2nd processor $= \sum_{i=993}^{1024} (2+2*i)$

$$= 64608$$

$$\therefore \text{speed up} = \frac{\text{Uniprocessor system time}}{\text{32-processor time}}$$

$$= 1051648 / 64608$$

$$= 16.3$$

Now, for parallel program,

```
PAR for (L=1; L<32, L++)
    {  for (I=(L-1)*16+1; I<=L*16; I++)
        {  SUM(I) = 0
           for (J=1; J<=I, J++)
              SUM[I] = SUM[I] + I
        }
for (I=(64-L)*16+1; I<=((64-L)+1)*16, I++)
    {  sum[I] = 0
```

for (J=1 , J<=I , J++)

SUM[I] + SUM[I] + I

{ }

Total cycles = $\sum_{I=(L-1)*16+1}^{L*16} (2+2I)$

$= \sum_{I=(64-1)*16+1}^{(64-L+1)*16} (2+2I)$

$= 32864$

$\therefore$ Speed up $= 1051648/32864$

$= 32$

Qno2) Ans Required SIMD algorithm :

For j=1 to n Do

Par for k=1 to n Do

$I^{Cik} = 0$ (vector load)

for j=1 to n Do

Par for k=1 to n Do

$Cik = Cik + a_{ij} \cdot b_{jk}$ (vector multiply)

End j loop

End i loop

Qno.4) Ans MPI program to find factorial of a given integer(n):

```
#include <stdio.h>
# include <mpi.h>
   int main (int argc , char **argv)
     { int myid, numprocs, i, n, lrm, j, mod;
   int fact,rslt=1;
      MPI_Init (&argc, &argv);
      MPI_comm_size(MPI_comm_WORLD, &numprocs);
     MPI_comm_rank(MPI_comm_WORLD, &myid);
int unused attribute ((unused));
   if (myid ==0) {
          printf ("Enter the No.=> \n");
          unused = scanf ("%d", &n);
     }
MPI_Bcast (&n,1, MPI_INT,0, MPI_comm_WORLD);
 lm = n/numprocs ;
    mod = n % numprocs;
for (i= myid*lm+1 ; i< myid*lm+lm; i++)
   { rslt =rslt * i;
      if (mod != 0 && myid == numprocs-1)
   {
```

```
for (j=1 ; j<=(myid+1)*lm +mod ; j++)
{
    rslt = rslt *j ;
}
{printf
MPI_Reduce(&rslt, &fact, 1, MPI_int, MPI_Prod,
                0, MPI_comm_WORLD);
if (myid == 0) {
    printf(" The requised factorial value=%d", fact);
    }
MPI_Finalize();
return 0;
}
```