Reg. No. | | | | | | | | | | |

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

II SEMESTER M.TECH. (SOFTWARE ENGINEERING/COMPUTER NETWORKING & ENGINEERING)

END SEMESTER EXAMINATIONS, APRIL 2018

SUBJECT: ELECTIVE-III PARALLEL COMPUTATION AND APPLICATIONS [ICT 5241]

REVISED CREDIT SYSTEM

(27/04/2018)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer ALL the questions.
- Missing data may be suitably assumed.

1A. Write the complete efficient CUDA C program to multiply two matrices A (N x N), B (N x N) and store the result in C (NxN). Assume multiple blocks are used to handle the large input and shared memory is used to reduce the global memory traffic. 5

1B. With a necessary diagram, explain any three features of Kepler architecture. 3

1C. With an example, explain the need for synchronization barrier. How is it carried out in CUDA? 2

2A. Differentiate between the two broad categories of cache coherency protocols. With the help of a neat diagram, explain the MESIF protocol adopted in Nehalem micro-architecture. 5

2B. Explain the CUDA extended keywords for function declaration with an example code snippet. 3

2C. Differentiate between task parallelism and data parallelism with an example code snippet. 2

3A. With the neat diagram, explain the front-end pipeline of Nehalem micro-architecture. 5

3B. With the neat diagram explain the CUDA device memory model. 3

3C. With an example code snippet, explain Thrust interoperability. 2

4A. Write the equivalent efficient CUDA C program to compute var, SD using parallel approach.

```
#include <iostream>
#include <math.h>
#define MAXSIZE 2048
int main()
{
   float x[MAXSIZE];
   int i, n=MAXSIZE;
   float avg, var, sd, sum = 0,
   sum1 = 0;
   for (i = 0; i < n; i++){
      x[i]=rand()%2048;}
   for (i = 0; i < n; i++){
      sum = sum + x[i];}
   avg = sum / (float)n;
   for (i = 0; i < n; i++){
      sum1 = sum1 + ((x[i] - avg) * (x[i] - avg));}
   var = sum1 / (float)n;
   sd = sqrt(var);
   printf("var = %.2f\n", var);
   printf("SD  = %.2f\n", sd);}
```

Ensure that the program

i) uses multiple blocks of threads to handle the input data.

ii) dynamically allocates shared memory. 5

4B. Write the execution phases of the CUDA kernel that performs scan (prefix sum) on the input vector [5, 3, 2, 6, 8, 7, 1, 3, -4, 1, -5, 8, 3, 1, 10, -20], launched using <<4,4>>> execution configuration parameters. 3

4C In a certain program, 80% of the work is vectorizable. This program is run using 10 processing elements of SIMD machine. Under the assumption there are no additional overheads, what is the parallel speedup? 2

5A. What is thread divergence? Explain how it effects the performance of the CUDA program by considering the reduction(sum) algorithm for an input vector [4, 5, 2, 3, 1, 5, 6, 2, -1, -3, 3, 2, 6, 5, 7, 1]. Assume the kernel is launched with <<4,4>>> execution configuration parameters. Write the kernel function for an efficient reduction algorithm that minimizes the divergence. 5

5B. With a convolution CUDA kernel, explain how thread indices are mapped to 2D input data. 3

5C. For the below kernel code snippet, if the block size is 512 and warp size is 32, How many warp/s will have divergence, when the *phase* is equal to i)0 ii)16 iii) 32 iv)1024?

```
__shared__ float partialSum[SIZE];
partialSum[threadIdx.x] = X[blockIdx.x*blockDim.x+threadIdx.x];
unsigned int t = threadIdx.x;
for (unsigned int phase = 1; phase < blockDim.x; phase *= 2){
__syncthreads();
if (t % (2* phase) == 0)
partialSum[t] += partialSum[t+ phase];}
```

2