Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL 576104

(Constituent College of Manipal University)

FIFTH SEMESTER B.Tech (CCE) DEGREE MAKEUP EXAMINATION – DEC, 2015/ JAN, 2016
SUBJECT: PARALLEL PROGRAMMING – ICT 355
(REVISED CREDIT SYSTEM)

TIME: 3 HOURS **02/01/2016** **MAX. MARKS: 50**

Instructions to candidates
- Answer any **FIVE FULL** questions.
- Missing data, if any, may be suitably assumed.

1A. Write a complete CUDA C program to perform matrix multiplication using multiple blocks. Consider the following factors when writing the program.
a. The number of elements is 10,000.
b. Let the *Block_Size* =16 and *Width* = 32.
c. Program must include code for input data allocation, data transfer and kernel invocation.
d. Ensure that the results are hazard free.
e. Perform all the calculations on the GPU.

1B. How has the commodity hardware evolved over time? Explain the advantages and disadvantages of each evolution.

1C. Give an overview of the execution unit of Nehalem's OOEE.

(5+3+2)

2A. Consider the set of numbers [10, 1, 8, -1, 0, -2, 3, 5, -2, -3, 2, 7, 0, 11, 0, 2].
i. Explain how interleaved and contiguous parallel sum reduction work using the above list.
ii. Which parallel reduction performs better? Explain.

2B. What is snooping? How can you classify the snooping protocols? Explain.

2C. With an example, explain the need of a SYNC barrier. How is it done in CUDA?

(5+3+2)

3A. Explain different CUDA variable type qualifiers with an example for each.

3B. Assume a CUDA device with 8 blocks and 1024 threads per SM and allows upto 512 threads in each block. How do you calculate the number of warps in an SM? Also, analyse the utilization of SM resources based on warp scheduling for matrix multiplication using 8X8 and 16X16 blocks.

3C. With a neat diagram, explain the CUDA memory hierarchy.

(5+3+2)

4A. Compare the GPU Kepler and Fermi architecture based on the following features:
i. Dynamic Parallelism
ii. HyperQ

4B. Write the CUDA kernel to perform exclusive prefix sum for the input vector. For the above kernel write the execution phases for the input [4, 2, -30, 20, 17, 80, -12, 8].

4C. Explain the technique to handle dependencies and introduce nondependent instructions to hide arithmetic latency on GPUs.

(5+3+2)

5A. Explain Thrust functor with an example. Write a complete CUDA program using the library functions to perform square root of squares of a sequence of N input elements using functors.

5B. What is the need of a loop stream detector? Which unit in the Nehalem Cores FEP supports the detector? How does it differ from that of the Core2? Explain.

5C. Explain Turbo Boost 2.0 technology using a suitable diagram. Indicate the states corresponding to the frequencies in the diagram.

(5+3+2)

6A. What is global memory bandwidth? How do you allocate 1KB of characters using CUDA APIs on the three GPU memory hierarchies? Given the following CUDA code, rewrite the code to improve bandwidth to global memory and, if applicable, shared memory. Assume c is stored in row-major order, so c[i][j] is adjacent to c[i][j+1].

```
N = 512; NUMBLOCKS = 512/64;
float a[512], b[512], c[512][512];
__global__ void compute(float *a, float *b, float *c) {
        int tx = threadIdx.x;
        int bx = blockIdx.x;
        for (j = bx*64; j< (bx*64)+64; j++)
          a[tx] = a[tx] - c[tx][j] * b[j];
}
```

6B. Write the CUDA kernel function to perform 2D convolution operation on N x N matrix using a single 2D block.

6C. An array of type float elements is to be processed in a one-element-per-thread fashion by a GPU. Write an execution configuration for the following scenarios:

a. The array is 1-D and of size N. The target GPU has 8 SMs, each with 16 SPs.
b. The array is 2-D and of size NxN. The target GPU has 5 SMs, each with 48 SPs.

(5+3+2)

****