Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL 576104

(Constituent College of Manipal University)

FIFTH SEMESTER B.Tech (CCE) DEGREE END SEMESTER EXAMINATIONS – NOV/DEC, 2015
SUBJECT: PARALLEL PROGRAMMING – ICT 355
(REVISED CREDIT SYSTEM)

TIME: 3 HOURS **30/11/2015** **MAX. MARKS: 50**

Instructions to candidates

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

1A. With example code snippets explain how GPU memory can be a limiting factor to parallelism?

1B. Explain the memory access pattern in Nehalem micro-architecture for local and remote memory access using suitable diagrams. Explain the memory access pattern and the terms local and remote access latencies considering two nodes.

1C. With reference to the traditional graphics pipeline, explain the need of unified shader architecture.

(5+3+2)

2A. Write the complete CUDA program to perform 1D convolution on a vector *V* with an input mask of size *M.* Consider the following factors when writing the program.

a. The number of elements is 1 million.
b. Allocate shared memory during runtime.
c. Ensure that the results are hazard free.
d. Perform all the calculations on the GPU.

2B. What is loop invariant analysis? Can the compiler safely make optimizations when a variable is external to a function and not local? Explain.

2C. Differentiate between CPUs and GPUs with respect to parallelism.

(5+3+2)

3A. Consider *Md* and *Nd* as two input matrices and *Pd* as the output matrix after matrix multiplication. Let *Width* = 4 be the size of the matrix and *Tile_Width* = 2 be the width of the tile.

i. What is the number of passes taken by the tiled matrix multiplication kernel?
ii. Write the kernel code for tiled matrix multiplication.
iii. Write the execution phases for the tiled matrix multiplication for all threads in block (1, 1).

3B. Write the CUDA kernel function to compute the inclusive prefix sum for an input vector of size N. For the above kernel, write the output vector for the input [5, 3, -6, 2, 7, 10, -2, 8].

3C. With a neat diagram differentiate the Kepler and Fermi grid management workflow.

(5+3+2)

4A. Consider the following list [10, 1, 8, -1, 0, -2, 3, 5, -2, -3, 2, 7, 0, 11, 0, 2].

i. Using the above set of numbers, explain the two approaches to find the maximum using parallel reduction.
ii. Write the kernel code for the parallel reduction that performs better among the two.

4B. What is Flynn's taxonomy? With suitable examples, explain the various methods by which data level parallelism is obtained?

4C. Explain the difference between inclusive and exclusive caches with suitable examples.

(5+3+2)

5A. With an example, explain Thrust interoperability. Write a complete Thrust program to calculate the sum of two matrices on a GPU.

5B. Explain how divergence works for the scenario given in Figure Q.5B in terms of the active warps and threads for every branching condition. Also, mention the functions that can be executed in parallel taking the warp granularity in consideration.

```
if ( threadIdx.x < 32)
{
        if(threadIdx.x < 16)
        {
                if(threadIdx.x < 8)
                        func_a1();
                else
                        func_a2();
        }
        else
        {
                func_b();
        }
}
```

Figure Q.5B

5C. Assume 0.1% of the runtime of a program is not parallelizable. This program is supposed to run on the Tianhe-2 supercomputer, which consists of 3,120,000 cores. Under the assumption that the program runs at the same speed on all of those cores, and there are no additional overheads, what is the parallel speedup on 30 and 3,000,000 cores?

(5+3+2)

6A. With an example show how to query the GPU device for any three properties. Write the complete CUDA program to compute row sum of a matrix using multiple blocks and shared memory.

6B. Write complete program to find the absolute minimum of an array of N values using CUDA library functions.

6C. What is loop unrolling? With an example explain the advantage of loop unrolling on a GPU?

(5+3+2)

****