Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

A Constituent Institution of Manipal University

V SEMESTER B.TECH. (COMPUTER AND COMMUNICATION ENGINEERING)

END SEMESTER EXAMINATIONS, NOV/DEC 2016

SUBJECT: PARALLEL PROGRAMMING [ICT 3153]

REVISED CREDIT SYSTEM
(29/11/2016)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

- Answer **ALL** the questions.
- Missing data if any, may be suitably assumed.

1A.	With suitable diagrams, explain the key features of Kepler GPU architecture.	5
1B.	With suitable code snippets, explain any three algorithms available in CUDA Thrust library.	3
1C.	With suitable examples, explain the differences between data parallelism and task parallelism.	2
2A.	Write the complete CUDA C program to perform 1D convolution using shared memory. Write the necessary comments highlighting each steps/phases. For the above program, write the execution phases for an input array of [20, 10, 8, 1, 0, 8, 3, 2, 0, 5, 2, 2, 5, 3, 1, 2] with an input mask of [1,2,1,3,1], assuming that the kernel is launched with <<< 4,4 >>> execution configuration parameters.	5
2B.	With the help of neat diagram, explain how Nehalem core pipeline unit performs fetching and pipe-lining of instructions.	3
2C.	With suitable code snippets, explain the lifetime and scope of any two CUDA variable type qualifiers.	2
3A.	Differentiate between the two cache coherency protocols that are classified based on the way multiple copies of the same block are located. With the help of a neat diagram, explain the MESIF protocol adopted in Nehalem micro-architecture.	5
3B.	With suitable code snippets, explain the common compiler optimization techniques.	3
3C.	Write complete program to find the absolute maximum of an array of N values using CUDA library functions.	2

4A. Write the CUDA kernel to reduce 1D input array of size N using shared memory with least divergence. The output should include maximum of M elements. i.e for the input array [2, 1, 8, 1, 0, 4, 4, 2, 0, 3, 1, 2, 5, 3,1, 2] and M=4, the output should be [8,4,3,5]. Write the necessary comments highlighting each steps/phases. For the above kernel and example, explain how shared memory usage improves the performance. **5**

4B. With suitable code snippets, explain any three CUDA C keywords for function declarations. Write the CUDA C kernel to add two matrices and store the result in third matrix. Assume that multiple 1D blocks of threads are launched to handle the huge data input. Every thread should compute the sum of each row. **3**

4C. With suitable code snippets, explain how error handling can be done in CUDA C programs. **2**

5A. Write the CUDA C kernel to compute exclusive prefix scan for 1D array elements. Show the execution phases of the above kernel for the input: 7, 6, 10, -18, 0, -9, 15 12, 5, 3, 4, 8, 7, 3, 1, 2. Assume the kernel is launched with <<< 4, 4 >>> execution configuration parameters. **5**

5B. With the necessary examples explain the classification of computer architectures using Flynn's taxonomy. **3**

5C. With the parallel vector addition CUDA C program and necessary diagrams, explain the compilation process of a CUDA program. **2**