Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

VI SEMESTER B.TECH (COMPUTER SCIENCE AND ENGINEERING)
END SEMESTER EXAMINATIONS, APRIL/MAY 2017

SUBJECT: **PARALLEL COMPUTER ARCHITECTURE AND PROGRAMMING [CSE 3202]**

REVISED CREDIT SYSTEM

Time: 3 Hours 25-04-2017 MAX. MARKS: 50

Instructions to Candidates:

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

1A.With a neat diagram explain how the graphics pipeline arranged in early NVIDIA GeForce GPUs. What was the major improvement happened to these pipelines in the next decade? **4M**

1B. Without using Point to Point communication write an efficient MPI program which reads matrix *A* of size 4*x*4 and it produces a resultant matrix *RES* of size 4*x*4 such that every row elements of *A* are added with a key value. The key value for the first row is the maximum element of the last row and the key value for remaining row is the maximum element from the previous row of matrix *A*. Use 4 processes (including root) to perform this task.

Example: A RES

1	2	3	4	**7**	**8**	**9**	**10**
5	6	7	8	9	10	11	12
2	4	3	5	10	12	11	13
2	3	4	**6**	7	8	9	11

4M

1C. User wants to allocate some memory space which must be used during point to point communication in MPI. With the help of a code snippet explain how the user can achieve this. **2M**

2A.
```
#include "mpi.h"
#include <stdio.h>
void main(int argc, char *argv[])
{
  int  C=3;
  int  numtasks;
  MPI_Init(&argc, &argv);
  MPI_Comm_size(C, &numtasks);
  printf ("Number of tasks= %d \n", numtasks);
  MPI_Finalize();
}
```

Identify the error in the above code. Modify the above code to catch the error and display the error message. **2M**

2B. Write an OpenCL kernel code that accepts a string *S* consisting of *N* number of equal length words and generates the output string *RES* in the following manner. Each work-item should place all characters of a word from *S* in the proper position in *RES*. Write all the OpenCL APIs upto creation of buffers for the above kernel.

Example: S: HaI BaI SeE YoU
RES: HBSYaaeoIIEU **5M**

2C. Two independent Instruction streams A and B are given as below. Assume that the system has one Floating point ALU and three Integer ALUs. Using Simultaneous Multithreading show how the instructions are scheduled to these resources. **3M**

Instruction Stream A	Instruction Stream B
add a,b,c	add a,b,n
add d,b,c	fmul d,b,n
add f,a,d	mul g,h,a
mul g,d,h	add f,d,g
fmul k,g,f	fadd h,a,g
add m,g,k	add m,g,f

3A.Write OpenCL kernels to sort *N* numbers using odd-even transposition sorting. How will you create the kernel objects for these kernels. Write the part of host code which will clearly show how many iterations are required for sorting and how the kernels are invoked. **4M**

3B.Why do you think many-core design is given importance? How the OpenCL specification is defined? Explain. **4M**

3C.Explain the scope and lifetime of following type of variables used in CUDA? Also mention in which type of memory it is stored.

a) automatic array variables
b) shared variables **2M**

4A. Write a CUDA host program that reads a character type matrix *A* and integer type matrix *B* of size *m*x*n*. It produces an output string *STR* such that, every character of *A* is repeated *n* times (where *n* is the integer value in matrix *B* which is having the same index as that of the character taken in *A*). Solve this using 1D Block.

Example: A B

	A				B		
p	C	a	P	1	2	4	3
e	X	a	M	2	4	3	2

Output String STR: pCCaaaaPPPeeXXXXaaaMM **4M**

4B. Write the kernel code for Q4A.such that, every column of input matrix must be produced required number of times by one thread. **3M**

4C. With the help of a block diagram show how the 2D Grid of 1D Block can be represented in CUDA by mentioning blockId and threadId. How will you calculate the threadId of each thread? **3M**

5A. Write the main difference between MPI and PVM? Explain with an example how the virtual machine can be dynamically configured by a user application. **4M**

5B. Using n tasks (where n<=N), write a PVM program to find

1! + (1+2) + 3! + (1+2+3+4) + ………..+ N! or (1+2+…..+N) **4M**

5C Why do you think simple CUDA kernels will achieve only a small fraction of the potential speed of the underlying hardware? Assume NVIDIA X supports 98.4 gigabytes per second (GB/s) of global memory access bandwidth. With 8 bytes in each double precision floating-point value maximum how much data can be loaded per second and with CGMA Ratio of 1.0 how many maximum floating-point operations can be carried out by a kernel? How can you improve this? **2M**