Reg. No.									

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

VI SEMESTER B.TECH. (COMPUTER SCIENCE AND ENGINEERING)

END SEMESTER EXAMINATIONS, APR 2019

PARALLEL COMPUTER ARCHITECTURE AND PROGRAMMING [CSE 3202]

REVISED CREDIT SYSTEM
(27/04/2019)

Time: 3 Hours MAX. MARKS: 50

Instructions to Candidates:

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

1A. The raster operation (ROP) stage in fixed-function graphics pipeline in NVIDIA GeForce GPU performs the final raster operations on the pixels. Explain. **2M**

1B. Differentiate hyper threading and multithreading in heterogeneous computing system. **4M**

1C. 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.

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

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

4M

2A. Write a MPI program which reads an encrypted string *STR* consisting of *N* unequal length words. Help the user in decrypting to an output string *RSTR* for which the information is hidden inside the input string *STR*. Each word of *STR* contains one special symbol *S* and a number *KEY*. Using *N* number of processes (including root) extract this information. After extracting every process will generate the symbol *S*, *KEY* number of times. The final resultant string *RSTR* should be displayed in the root process as shown in the example.

Input string STR : C*a3n Y5ou% Fi+n1d2 &M1e1
Output string RSTR : ***
%%%%%
++++++++++++
&&&&&&&&&&&&

5M

2B. With help of example code snippet write the difference between following collective communication routines:

i) MPI_Allgather( ) and MPI_Alltoall( )
ii) MPI_Reduce( ) and MPI_Scan( ) **5M**

Also give the example code snippet to measure the performance of a MPI application.

3A. Write an efficient OpenCL kernel code that reads a matrix A of size *n* x *n*. Rotate the matrix by 180° degree clockwise.

Example :

Input			Output		
1	2	3	9	8	7
4	5	6	6	5	4
7	8	9	3	2	1

3M

3B. A programmer wants to check the time taken for kernel execution for different ways of kernel implementation in case of 180° matrix rotation kernel. Help the programmer by writing the procedure with all OpenCL functions with program structure. **2M**

3C. Convert the input matrix *A* into output matrix *B* of size *N* x *N* by considering elements of matrix *A* into 4 equal parts. 1^{st} and 4^{th} part of elements should be square of its element, 2^{nd} and 3^{rd} part should be cube of its element. Write an OpenCL kernel code by using element number of threads.

Example : Input : N=4

A

3	8	2	5
2	3	5	6
2	4	3	1
3	2	1	5

B

9	64	8	125
4	9	125	216
8	64	9	1
27	8	1	25

5M

4A. Write a OpenCL kernel which multiply 0^{th} element of array by 3^0, multiply 1^{st} element of array by 3^1, multiply 2^{nd} element of array by 3^2 and so on multiply $(N-1)^{th}$ element of array by $3^{(N-1)}$. Create the required buffer(s) for the kernel and write the OpenCL API to execute this kernel. **3M**

4B. What is data parallelism? List any four engineering problems that you know which has data parallelism. **2M**

4C. There is a binary tree of height *L*. If any node is absent the corresponding value is -1. Write a parallel CUDA program to determine whether the binary tree is a binary search tree and display in main. **5M**

5A. Write a CUDA kernel for determining the total number of Fibonacci numbers among 8 neighbors of a non-border element in a matrix of size *M* x *N*. The remaining elements in output matrix should be 0. All the elements of the output matrix will have to be generated in parallel.

M=3, N=5

Input :					Output:				
1	2	3	4	5	0	0	0	0	0
6	7	8	9	10	0	5	3	4	0
11	12	13	14	15	0	0	0	0	0

5M

5B. Write a PVM program in which Sender sends an integer and a char array, and Receiving Worker receives the integer and char array. As an acknowledgement, receiver sends a random number to the sender and sender must accept it. **5M**