King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No 1 – Fall 2021

Question 1

1. Give the definition of Parallel computing and Parallel Programming

Parallel computing: is a form of computation in which many calculations are carried out
simultaneously.
Parallel Programming: Decomposing a programming problem into tasks and Deploy the tasks on multiple processors and run them simultaneously

2. Enumerate and give a brief description of the main opportunities of parallelism.

3. Enumerate and give a brief description of the main aspects of parallel computing.

4. What are the main differences between Distributed and Parallel Computing.

Q2:
1-Instruction Level Parallelism:
Hidden Parallelism in computer programs

2-Single computer level:
Multi core computers: Chip multi processors
Dual core, Quad core, GP GPU
Multi processor computers: Symmetric multi processors
Supercomputers

3-Multiple computers level:
Clusters(Known Configs at compile time), Servers, Grid computing(Daynamic can change at run time)

Q3
1-Parallel Computers Architecture

2-Algorithms and applications
Reasoning about performance
Designing parallel algorithms.

3-Parallel Programming
Paradigms
Programming Models
Programming languages
Frameworks
Dedicated environments

Q4
1- in distributed computing the procssors are geograhpic distant but in parallel computing they are in the same machine
2- in distributed computing the aim is to make services avilable and realiable but in parallel computing the aim is to increase the preformance
3- in parallel computing interaction is Fine grained with low overhead but in distributed computing interaction is Coarse grained and heavier weight

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No 2 – Spring 2021

Question

1. Give the flynn's classification of computers.

   SISD , SIMD , MISD , MIMD

2. Use an example to explain the differences between the ***SIMD*** and ***MIMD*** computers.

3. Explain the main differences between the **Blocking non-buffered** and the **Non-Blocking non-buffered** send/receive operations of the message passing paradigm.

4. Let's consider that a root process has N child processes. Let's consider that the root process has an array called ***Data*** of size N. Explain the following operations using the array Data.
   a. The root executes the operation ***broadcast*** of the message passing paradigm.

   b. The root executes the operation ***scatter*** of the message passing paradigm.

   c. The root executes the operation ***gather*** of the message passing paradigm.

5. Describe the ***Task Farming*** and the ***Divide-and-Conquer*** programming models and explain the main differences between them.

Q2:
SIMD: Every proccsor has same instruction stream but each proccsor has it's own data stream
MIMD: Every proccsor has it's own instruction stream and data stream

Q3:
Blocking non-buffered: The sender when he send a send operation he will be blocked until the reciver match a recive operation
Non-Blockiing non-bufferd: The sender sends a send operation and then he will continue proccsing until the reciver send an interuption signal

Q4:
A- The Same Data will be copied to each proccses
B- The Data will be split between the procces
C- The Data will be gathered and joined in the root

Q5:
In divide and conquer the sub-tasks have the same nature and will be recursivly decomposed
In master-slave the sub-tasks may have diffrent nature

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No 3 – Fall 2021

Question

1. What GPGPU stands for and what does it mean.

General Purpose Graphical Procces Unit, It means that the gpu does the procceses that are usally done by cpu

2. Why CUDA is said Heterogeneous computing.

Becasue there are some portaion of the code that is done in serial by the cpu and other portaion of the code is done in parallel by gpu

3. Give the definition of the following terms:

   a. Device: GPU and it's memory

   b. Kernel. the portaion of the code that will be preformed on the device

   c. Grid of thread blocks. it's a computing model of cuda it consist of orginazing threads in diffrent blocks where each of them is composed of set of threads

   d. Warp. groups of 32 thread that always excute same instruction

4. Explain the parallel programming model of CUDA.

Grids composed of set of blocks evry block composed of set threads, threads are grouped of warps that excute the same instruction warps are time sliced

5. Enumerate and explain the different types of memory adopted by CUDA.

1- register (on chip, fast, per thread, store data up to 32bit)

2- local memory (slow, DRAM ,not cached, per thread)

3- shared memory (on chip , fast , per block, not cached)

4- global memory (DRAM , not cached , per grid)

5- constant (DRAM , read-only , cached , per grid)

6- texture memory (DRAM , cached , read-only)

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No 4 – Spring 2021

Question 1

1. Let’s consider 2 integer Arrays A and B of dimension N. Let’s consider that we would like to write a C program that runs in parallel and that computes the sum of the 2 arrays:

   C[i] = A[i] + B[i]

   a. Write the kernel (called ***kernel_1***) that will run on 1 Block of N threads.

```
__global__ void add( int *a, int *b, int *c){
c[threadIdx.x ] = a[ threadIdx.x ] + b[ threadIdx.x];
}
```

   b. Write another kernel (called ***kernel_2***) that will run on N blocks with 1 thread each.

```
__global__ void add( int *a, int *b, int *c){
c[blockIdx.x ] = a[blockIdx.x ] + b[blockIdx.x]
}
```

   c. Write the main program that will call both kernels.

Page 14 pdf

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No 4 – Spring 2021

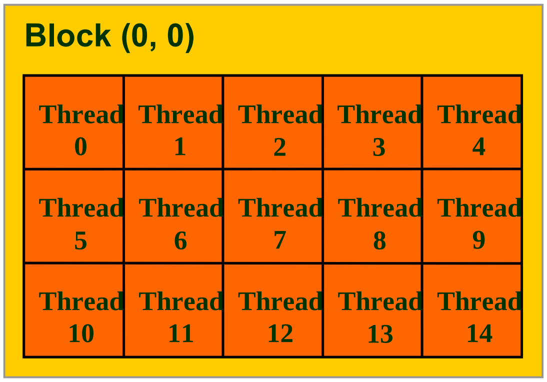
Question 2

Let's consider 2 integer Arrays A and B of dimension N. Let's consider that we would like to write a C program that runs in parallel and that computes the sum of the 2 arrays:

C[index] = A[index] + B[index];

For every configuration of the grid of thread blocks described below, give the statement that computes the index for each each thread:

1. The grid is composed of 1 block and threads should have ids as in the following figure:

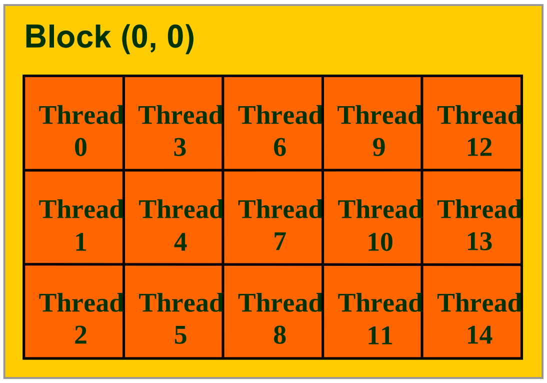


Block (0, 0)

Thread 0	Thread 1	Thread 2	Thread 3	Thread 4
Thread 5	Thread 6	Thread 7	Thread 8	Thread 9
Thread 10	Thread 11	Thread 12	Thread 13	Thread 14

int index = threadidx.x + (blockdim.x * threadidx.y);

2. The grid is composed of 1 block and threads should have ids as in the following figure:



Block (0, 0)

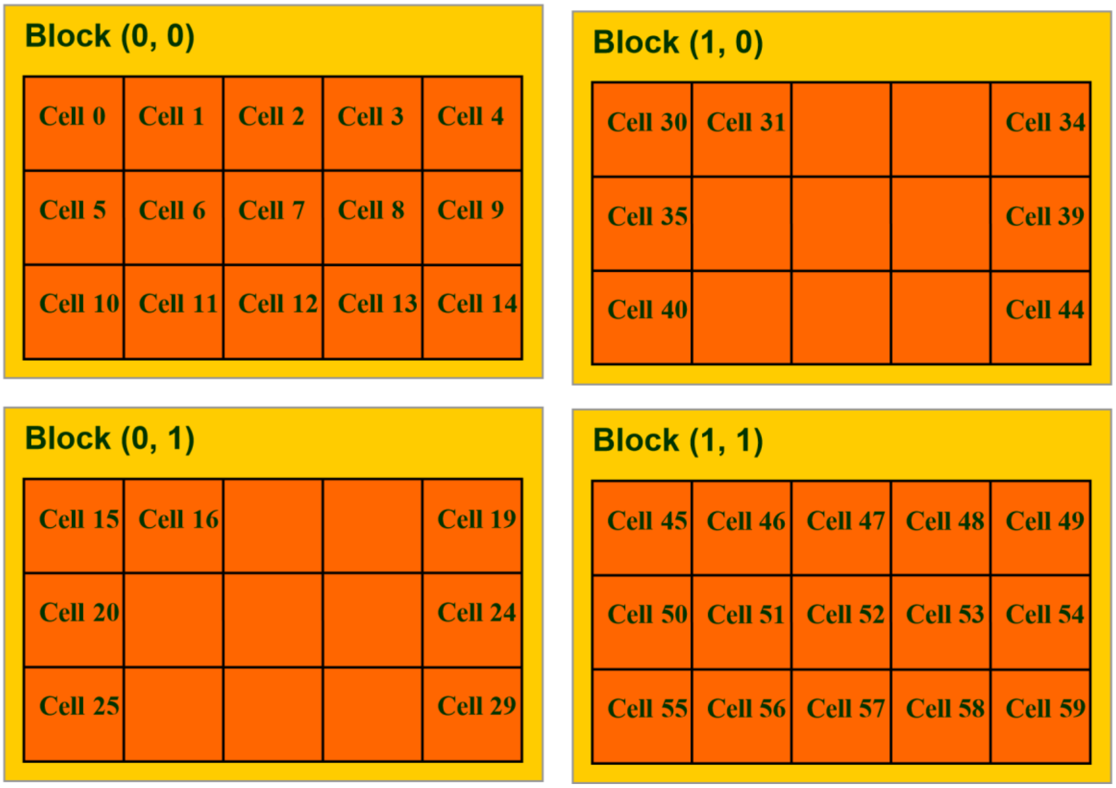
Thread 0	Thread 3	Thread 6	Thread 9	Thread 12
Thread 1	Thread 4	Thread 7	Thread 10	Thread 13
Thread 2	Thread 5	Thread 8	Thread 11	Thread 14

int index = threadidx.y + (blockdim.y * threadidx.x);

King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC453 – Parallel Processing – Tutorial No 5 – Fall 2021

Question 1

We would like to run a kernel on grid configured as M * N matrix of thread blocks. Every thread handles only one cell. Give the statement that calculates the ***cell_id*** for each thread as shown in the following figure:



Block (0, 0)

Cell 0	Cell 1	Cell 2	Cell 3	Cell 4
Cell 5	Cell 6	Cell 7	Cell 8	Cell 9
Cell 10	Cell 11	Cell 12	Cell 13	Cell 14

Block (1, 0)

Cell 30	Cell 31			Cell 34
Cell 35				Cell 39
Cell 40				Cell 44

Block (0, 1)

Cell 15	Cell 16			Cell 19
Cell 20				Cell 24
Cell 25				Cell 29

Block (1, 1)

Cell 45	Cell 46	Cell 47	Cell 48	Cell 49
Cell 50	Cell 51	Cell 52	Cell 53	Cell 54
Cell 55	Cell 56	Cell 57	Cell 58	Cell 59

cell_id **=** (threadIdx.x+threadIdx.y*blockdim.x)
+(blockIdx.y+blockIdx.x*gridDim.y)*(blockDim.x*blockDim.y)