GPU and Heterogeneous Systems – A.Y. 2021-22

Scuola di Ingegneria Industriale e dell'Informazione
Instructor: Prof. Antonio Miele

June 28, 2022 – **FIRST PART OF THE EXAM**

Surname:	Name:	Person Code:

Question	**1**	**2**	**3**	**4**	**5**	**OVERALL**
Max score	**3**	**3**	**3**	**3**	**3**	**15**
Score						

Instructions:

- **Duration: 30 minutes**
- This first part of the exam is "closed book". The students are not allowed to consult any course material and notes.
- No extra devices (e.g., phones, iPad) are allowed. Please, shut down and store any electronic device.
- Students are not allowed to communicate with any other ones.
- Students can write in pen or pencil, any color, but avoid writing in red.
- Any violation of the above rules will lead to the invalidation of the test.

Question 1
Describe the benefits of warp interleaving in NVIDIA GPU and how it is managed.

Question 2
What will the following program fragment print on the screen? How many threads are created?

REMEMBER: nested kernels always complete before the parent one, so we will see first the printf with the highest depth

```
__global__ void foo(int iSize, int iDepth) {
  int tid = threadIdx.x;

  if (iSize > 1) {
    int nthreads = iSize/2;
    if(tid == 0 && nthreads > 0){
      foo<<<1, nthreads>>>(nthreads, iDepth+1);
      cudaDeviceSynchronize();
    }
    __syncthreads();
  }
  printf("Recursion=%d: Hi from thread %d block %d\n", iDepth, tid, blockIdx.x);
}

int main(){
  /*...*/
  int iSize = 4;
  foo<<<1, iSize>>>(iSize, 0);
  /*...*/
}
```

Question 3
Briefly describe CUDA memory model; for each component specify name, type of usage, type of access (read/write or read only) and scope.

Question 4
Comment the benefits of the unified coherent memory architecture implemented in AMD heterogeneous systems w.r.t. the memory organization of a traditional architecture having a discrete GPU.

Question 5
What is the key optimization to speed up the convolution process?

GPU and Heterogeneous Systems – A.Y. 2021-22

Scuola di Ingegneria Industriale e dell'Informazione
Instructor: Prof. Antonio Miele

June 28, 2022 – **SECOND PART OF THE EXAM**

Surname:	Name:	Personal Code:

Question	**1**	**2**	**3**	**OVERALL**
Max score	**5**	**5**	**6**	**16**
Score				

Instructions:

- **Duration: 1 hour and 15 minutes**
- This second part of the exam is "open book". The students are allowed to use any material and notes.
- The students are allowed to use the laptop and the tablet. No extra devices (e.g., phones) are allowed. Please, shut down and store not allowed electronic devices.
- Students are not allowed to communicate with any other one or use Internet.
- Students can write in pen or pencil, any color, but avoid writing in red.
- Students can also use the laptop to code the test solution. In this case, please pay attention to the instructor's instructions to submit the test solution.
- Any violation of the above rules will lead to the invalidation of the test.

Question 1
Implement a basic CUDA kernel function to accelerate the compute-intensive function in the following C program.

Question 2
Modify the main function to execute the CUDA kernel function defined in the former question. Set block size to 32.

Question 3
Implement a new CUDA kernel function to accelerate the compute-intensive function in the following C program by using the shared memory.

The source code can be downloaded from: https://miele.faculty.polimi.it/findpeaks.c

```
/*
* The kernel function to accelerate receives in input a vector of positive integers,
* called A, together with its size, and a second empty vector of integers, B, of the
* same size.
* For each element i in A, the function saves in B[i] the value 1 if A[i] is greater
* than all the neighbor values with an index between (i-DIST) and (i+DIST), bounds
* included and if they exist; 0 otherwise. DIST is a constant value defined with a
* macro.
* The main function is a dummy program that receives as an argument the vector size,
* instantiates and populates randomly A, invokes the above function, and shows
* results.
*/

#include <stdio.h>
#include <stdlib.h>

#define MAXVAL 100
#define DIST 10

void compute(int *V, int *R, int num);

//kernel function: identify peaks in the vector
void compute(int *V, int *R, int num) {
  int i, j, ok;
  for(i=0; i<num; i++){
    for(j=-DIST, ok=1; j<=DIST; j++){
      if(i+j>=0 && i+j<num && j!=0 && V[i]<=V[i+j])
        ok=0;
    }
    R[i] = ok;
  }
}

int main(int argc, char **argv) {
  int *A;
  int *B;
  int dim;
  int i;

  //read arguments
  if(argc!=2){
    printf("Please specify sizes of the input vector\n");
    return 0;
  }
  dim=atoi(argv[1]);

  //allocate memory for the three vectors
  A = (int*) malloc(sizeof(int) * dim);
  B = (int*) malloc(sizeof(int) * dim);

  //initialize input vectors
  /*code omitted for the sake of space*/

  //execute on CPU
  compute(A, B, dim);

  //print results
  /*code omitted for the sake of space*/

  free(A);
  free(B);

  return 0;
}
```