## Intensive Computation

Prof. A. Massini

April 20, 2022

Midterm test

Student's Name

## Matricola number

| Exercise 1 (4 points) |  |
| :--- | :--- |
| Exercise 2 (4 points) |  |
| Exercise 3 (5 points) |  |
| Exercise 4 (4 points) |  |
| Exercise 5 (5 points) |  |
| Exercise 6 (4 points) |  |
| Question 1 (4 points) |  |
| Question 2 (2 points) |  |
| Total (32 points) |  |

## Exercises 1 (4 points) Performance equation

Suppose we have made the following measurements, where we are considering Arithmetic and Logic instructions - AL - and the subset of AL of only Integer Multiplications and Divisions - IMD:

Frequency of AL operations $=45 \%$
Average CPI of AL operations $=3$
Average CPI of other instructions $=2.2$
Frequency of IMD = 10\%
CPI of IMD = 9

Assume that the two design alternatives are to decrease the CPI of AL to 2.5 or to decrease the average CPI of IMD operations to 3.4.
Compare these two design alternatives using the processor performance equation and compute the speedup in both cases, determining which alternative is more cost-effective.

## Exercises 2 (4 points) Amdhal Law

The following measurements are recorded with respect to the different instruction classes for the instruction set running a given set of benchmark programs:

| Instruction Type | Instruction Count (millior | Cycles per Instructio |
| :--- | :---: | :---: |
| Arithmetic and logic | 6 | 6 |
| Load and store | 8 | 2 |
| Branch | 5 | 6 |
| Others | 7 | 5 |

Assume that "Arithmetic and logic" instructions can be modified so that they take 4 cycles per instruction instead of 6, and "Branch" instructions can be modified so that they take 5 cycle per instruction instead of 6 as in the table. Compute the speedup obtained by introducing only one enhancement and both enhancements using the Amdhal law.

How many cycles should the "Branch" instructions consist of if we wanted modify only this subset of instructions but get the same speedup obtained applying both of the above enhancements?

## Exercise 3 (5 points) - GPU \& CUDA

| Technical specifications | Compute capability (version) |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 1.0 | 1.1 | 1.2 | 1.3 | 2.x | 3.0 | 3.5 | 3.7 | 5.0 | 5.2 |
| Maximum dimensionality of grid of thread blocks | 2 |  |  |  | 3 |  |  |  |  |  |
| Maximum x-dimension of a grid of thread blocks | 65535 |  |  |  |  | $2^{31}-1$ |  |  |  |  |
| Maximum y-, or z-dimension of a grid of thread blocks | 65535 |  |  |  |  |  |  |  |  |  |
| Maximum dimensionality of thread block | 3 |  |  |  |  |  |  |  |  |  |
| Maximum x- or y-dimension of a block | 512 |  |  |  | 1024 |  |  |  |  |  |
| Maximum z-dimension of a block | 64 |  |  |  |  |  |  |  |  |  |
| Maximum number of threads per block | 512 |  |  |  | 1024 |  |  |  |  |  |
| Warp size | 32 |  |  |  |  |  |  |  |  |  |
| Maximum number of resident blocks per multiprocessor | 8 |  |  |  |  |  | 16 |  | 32 |  |
| Maximum number of resident warps per multiprocessor | 24 |  | 32 |  | 48 | 64 |  |  |  |  |
| Maximum number of resident threads per multiprocessor | 768 |  | 1024 |  | 1536 | 2048 |  |  |  |  |
| Technical specifications | 1.0 | 1.1 | 1.2 | 1.3 | 2.x | 3.0 | 3.5 | 3.7 | 5.0 | 5.2 |
|  | Compute capability (version) |  |  |  |  |  |  |  |  |  |

You need to write a kernel for a computational fluid dynamics problem that operates on a matrix of size $\mathbf{3 2 5 1 0 x 3 2 5 1 0}$. You would like to assign one thread to each matrix element, and your thread blocks to use the maximum number of threads per block possible on your device.
a) How would you select the dimensions of a 2D grid and 2D rectangular blocks for your kernel, minimizing the number of idle threads? Consider a device having compute capability 2.x.
b) How would you select the dimensions of a 2D grid and 2D square blocks for your kernel, minimizing the number of idle threads? Consider a device having compute capability 3.7.

## Exercise 4 (4 points) - Circuit time and area

A particular way to implement an adder is given by the carry-select adder, generally consisting of ripple carry adders and multiplexers. A 7-bit carry-select adder with a variable block size is shown in the figure.
i) Compute the time (propagation delay) and area required by the 7-bit carry-select adder.
ii) Compare the 7-bits carry-select adder and the standard binary ripple-carry adder (that is compute


## Exercise 5 ( 5 points) - Number representation

- Represent the natural number range $[0 ; 3799]$ using the residue number system, considering:
* the conventional choice consisting of 3 moduli $\left\{2^{n}-1 ; 2^{n} ; 2^{n}+1\right\}$
* a moduli set consisting of 5 moduli.
- Give an estimation of the representational efficiency in both cases.
- Represent $\mathrm{A}=48$ in both residue systems defined and in the mixed radix representation for one of them at your choice.

Exercise 6 (4 points) Draw the scheme of the pipelined signed multiplier and show the product between the two 2 's complement operands $A=-7$ and $B=5$.

## Question 1 (4 points)

Briefly describe the main characteristics of the vector architectures.
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$
$\qquad$

## Question 2 (2 points)

Show how to obtain a unique ID for each thread by using the block ID and thread ID, in the case of:

- a 1D grid and 3D blocks
- a 2D grid and 2D blocks

