Intensive Computation

Prof. A. Massini

June 4, 2021

End-of-term test

Student's Name

_____

***Matricola* number**

_____

Exercise 1 (5 points)	
Exercise 2 (4 points)	
Exercise 3 (4 points)	
Question 1 (4 points)	
Question 2 (4 points)	
Exercise 4 (4 points)	
Exercise 5 (4 points)	
Exercise 6 (4 points)	
Total (33 points)	

Exercise 1 (5 points) - GPU & CUDA

You need to write a kernel that operates on an ultrasound color image represented by a matrix of size **1440x1280x24** (color images: have 24 bits for representing the three colors red, green, blue with 8 bits). You would like to assign one thread to each matrix element. You would like 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 1.3.

b) How would you select the dimensions of a **2D grid** and **3D blocks** with the three sides all equal for your kernel, minimizing the number of idle threads? Consider a device having compute capability 3.5.

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)									

Total number of threads is $1440 \times 1280 \times 24 = 44\,236\,800$

a) CC 1.3 $\Rightarrow$ $512 = 2^9$ threads per block allowed

We can choose different sizes for the rectangular blocks, for example:

- $2^4 \times 2^5 \Rightarrow \left\lceil \frac{1440}{2^4} \right\rceil = 90 \quad \left\lceil \frac{1280 \times 24}{2^5} \right\rceil = 960$ [GRID 90 x 960, BLOCKS 16 x 32]
- $2^3 \times 2^6 \Rightarrow \left\lceil \frac{1440}{2^3} \right\rceil = 180 \quad \left\lceil \frac{1280 \times 24}{2^6} \right\rceil = 480$ [GRID 180 x 480, BLOCK 8 x 64]

} Both solutions do not produce idle threads

b) CC 3.5 $\Rightarrow$ $1024 = 2^{10}$ threads per block allowed

Blocks must be 3D with the three sides all equal $\Rightarrow$ $2^3 \times 2^3 \times 2^3 = 2^9$

So we have $\frac{1440}{8} \times \frac{1280}{8} \times \frac{24}{8} = 180 \times 160 \times 3$

The grid must be 2D, then we can arrange 3D blocks in the following ways

- $180 \times (160 \times 3) = 180 \times 540$
- $(180 \times 3) \times 160 = 540 \times 160$

} We do not have idle threads

Allowing 3D blocks have different sides, we could have 1024 threads per block. For example:

- GRID 90 x 160 $\longrightarrow$ with blocks 16 x 8 x 8
- GRID 180 x 80 $\longrightarrow$ with blocks 8 x 16 x 8

Exercise 2 (4 points) – Interconnection Networks – CLOS

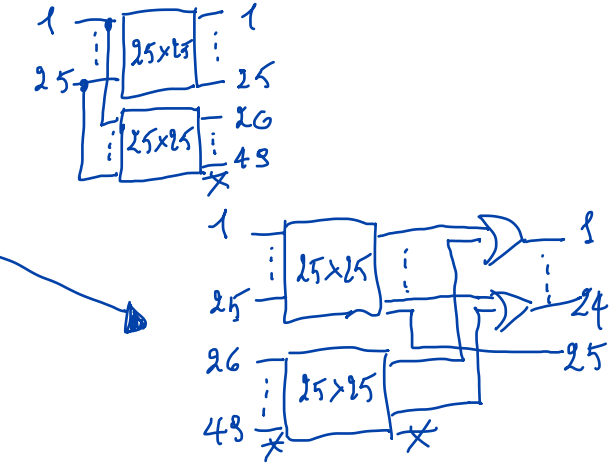
Design a Clos network of size 480 x 480, using in the first stage modules having 25 inputs. Consider both cases, strictly non-blocking and rearrangeable network.

Strictly non-blocking

$m \geq 2n - 1 = 2 \cdot 25 - 1 = 49$

$r = \lceil \frac{N}{n} \rceil = \lceil \frac{480}{25} \rceil = 20$

- stage 1: 20 modules 25×49
- stage 2: 49 modules 20×20
- stage 3: 20 module 49×25



Rearrangeable non-blocking

$m \geq n = 25$

$r = \lceil \frac{N}{n} \rceil = 20$

- stages 1 and 3: 20 modules 25×25
- stage 2: 25 modules 20×20

Compare the cost of the crossbar 480 x 480 and the Clos network, strictly non-blocking and rearrangeable non-blocking, designed in the previous point.

Cost of crossbar $C = 480 \times 480 = 230.400$

Cost of strictly non-blocking Clos network

$C_{NB} = (20 \times 2 \times (25 \times 25)) + (49 \times 20 \times 20) + (20 \times 2 \times (25 \times 25)) = 25.000 + 19.600 + 25.000 = 69.600$

Cost of rearrangeable nonblocking Clos network

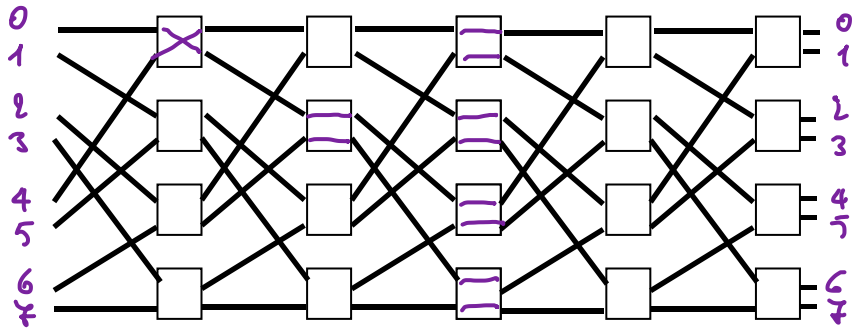
$C_{R\text{-}NB} = (20 \times 25 \times 25) + (25 \times 20 \times 20) + (20 \times 25 \times 25) = 12.500 + 10.000 + 12.500 = 35.000$

$GAIN_{NB} = \frac{230.400}{69.600} = 3,31$

$GAIN_{R\text{-}NB} = \frac{230.400}{35.000} = 6,58$

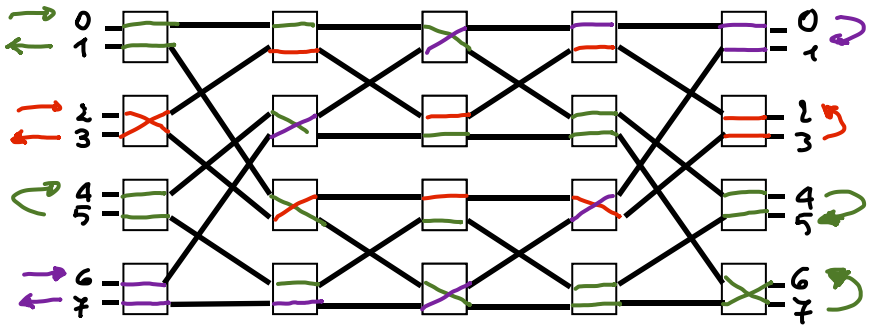
Exercise 3 (4 points) – Interconnection networks – (2 log N - 1) MIN

Draw a Shuffle-Shuffle network of size N=8 using the switches below. Is it possible to realize the permutation $P = \begin{pmatrix} 01\ 23\ 45\ 67 \\ 46\ 32\ 75\ 01 \end{pmatrix}$ setting all the switches in the central stage on the straight state?



Let us consider connection request 0 → 4
Input 0 can reach only the upper port of switches in the middle stage, both in case the switch in the first stage is set to straight or to cross
On the other hand, output 4 can be reached from the lower output port of switches in the middle stage
Hence, at least one switch of the middle stage should be set to cross.

Which multistage network with 2logN -1 stages would you use to route the permutation P above, that guarantees to find a solution? Draw the network and show the connections.



The Benes network, consisting of a Baseline cascaded with a reverse Baseline, can route any permutation using the looping algorithm.

Question 1 (4 points)

Briefly explain the **unit-gate model** giving the gate-count and gate-delay, used to evaluate time and area of circuits based on gates AND, OR, NOT and other gates derived from them.

Question 2 (4 points)

Explain what CUDA threads are and how they are organized in terms of blocks and grids. Describe how to obtain a unique ID for each thread by using the block ID and thread ID, in the case of a 2D grid and 3D blocks.

ID = (blockId.x + blockId.y * gridDim.x) *
(blocDim.x + blockDim.y + blockDim.z)
+(threadId.z*(blockDim.x * blockDim.y))
+ (treadId.y * blockDim.x)
+ threadId.x

Exercises 4 (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 (millions)	Cycles per Instruction
Arithmetic and logic	5	6
Load and store	8	3
Branch	6	5
Others	6	4

Assume that instructions "*Branch*" can be modified so that they take 4 cycle per instruction instead of 5 as in the table. Compute the speedup obtained by introducing this enhancement using the **Amdhal law**.

How many cycles should "*Arithmetic and logic*" instructions consist of to reach at least the same speedup obtained modifying the number of cycles of "~~*Arithmetic and logic*~~ Branch" instructions as above?

AMDHAL LAW $S = \dfrac{1}{(1 - FRAC) + \dfrac{FRAC}{SPEEDUP}}$

TOTAL CYCLES

$T = \underbrace{5 \cdot 6}_{A\&L} + \underbrace{8 \cdot 3}_{L/S} + \underbrace{6 \cdot 5}_{BRANCH} + \underbrace{6 \cdot 4}_{OTHERS} = 108$

- BRANCH

$FRAC_B = \dfrac{6 \cdot 5}{108} = 0{,}278$ $\quad SPEEDUP_B = \dfrac{5}{4} = 1{,}25$

$$S = \frac{1}{(1 - 0{,}278) + \frac{0{,}278}{1{,}25}} = \frac{1}{0{,}722 + 0{,}222} = \frac{1}{0{,}944} = 1{,}06$$

- $FRAC_{A\&L} = \dfrac{5 \cdot 6}{108} = 0{,}278$

$$\frac{1}{(1 - 0{,}278) + \frac{0{,}278}{6/c'}} = 1{,}06 \qquad \frac{1}{0{,}722 + \frac{0{,}278}{6} c'} = 1{,}06 \qquad 0{,}722 + \frac{0{,}278}{6} \cdot c' = \frac{1}{1{,}06}$$

$$c' = (0{,}943 - 0{,}722) \cdot \frac{6}{0{,}278} = 0{,}221 \cdot 21{,}582 = 4{,}769$$

Hence, to reach at least the same speedup obtained from the Branch optimization the CPI of A&L instructions must be less than or equal to 4,769

Exercises 5 (4 points) Performance equation

Suppose we have made the following measurements, where we are considering Arithmetic and FP (Floating Point) instructions:
Frequency of Arithmetic operations = 20%
Average CPI of Arithmetic operations = 4.2
Average CPI of other instructions = 2.8

Frequency of FP operations = 30%
CPI of FP = 5.6

Assume that the two design alternatives are to decrease the CPI of FP to 3.5 or to decrease the average CPI of Arithmetic operations to 2.4.
Compare these two design alternatives using the processor **performance equation** and compute the speedup in both cases.

- $CPI_{orig} = \sum_{i=1}^{n} CPI_i \times \frac{IC_i}{IC} = (4,2 \times 20\%) + (2,8 \times 80\%) = 0,84 + 2,24 = 3,08$

$CPI_{new\ Arith} = (2,4 \times 20\%) + (2,8 \times 80\%) = 0,48 + 2,24 = 2,72$

$CPI_{new\ FP} = CPI_{orig} - freq_{FP}\ (CPI_{FP\ old} - CPI_{FP\ new}) =$

$= 3,08 - 30\% (5,6 - 3,5) = 3,08 - 0,63 = 2,45$

- $SPEEDUP = \frac{CPI_{orig}}{CPI_{new}}$ $SPEEDUP_{FP} = \frac{3,08}{2,45} = 1,26$ $SPEEDUP_{Arith} = \frac{3,08}{2,72} = 1,13$

IF FP INSTRUCTIONS ARE NOT INCLUDED IN THE SET OF ARITHMETIC INSTRUCTIONS

- $CPI_{orig} = (4,2 \times 20\%) + (5,6 \times 30\%) + (2,8 \times 50\%) = 3,32$

$CPI_{new\ Arithm} = (2,4 \times 20\%) + (5,6 \times 30\%) + (2,8 \times 50\%) = 3,56$ $SPEEDUP_{Arithm} = 1,11$

$CPI_{new\ FP} = (4,2 \times 20\%) + (3,5 \times 30\%) + (2,8 \times 30\%) = 3,29$ $SPEEDUP_{FP} = 1,19$

Hence, the new FP design is more convenient than the new Arithmetic operation design in both scenarios

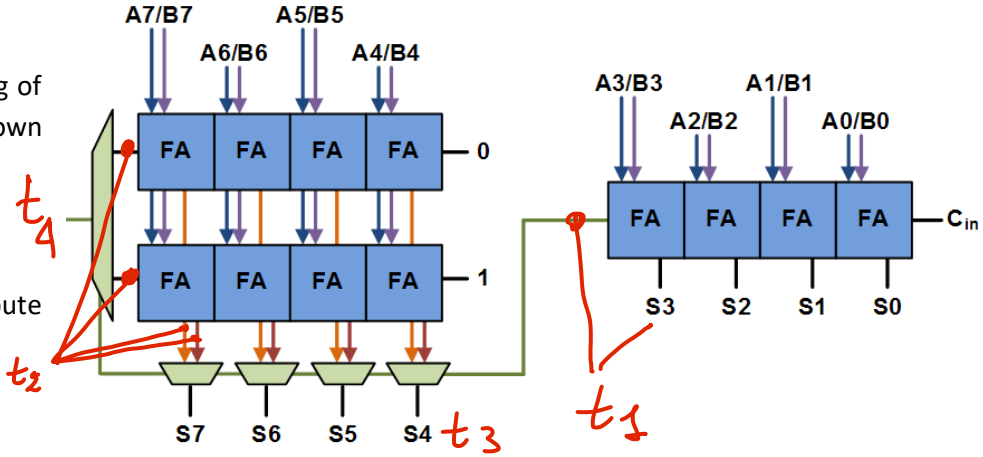
Exercise 6 (2+2 points) - Circuit time and area

CSA

A particular way to implement an adder is given by the ***carry-select adder***, generally consisting of ripple carry adders and multiplexers. A 8-bit carry-select adder with a uniform block size of 4 is shown in the figure.

i) Compute the ***time*** (propagation delay) and ***area*** required by the 8-bit carry-select adder.

ii) Compare the 8-bits carry-select adder and the standard binary ripple-carry adder (that is compute the **speedup** both for time and area).



- DELAY CSA

$t_1 = t_2 = 4\,T_{FA} = 16\,T_{GATE}$

$t_3 = t_4 = t_1 + T_{MUX} = T_1 + 2\,T_{GATE} = 18\,T_{GATE} = T_{CSA}$

- AREA CSA

$A_{CSA} = 12\,A_{FA} + 5\,A_{MUX(2-A-1)} = 12 \cdot 7\,A_{GATE} + 5 \cdot 3\,A_{GATE} = (84+15)\,A_{GATE} = 99\,A_{GATE}$

- DELAY RCA

$T_{RCA} = 8\,T_{FA} = 32\,T_{GATE}$

- AREA RCA

$A_{RCA} = 8\,A_{FA} = 56\,A_{GATE}$

- SPEEDUP CSA VS RCA

$S_T = \frac{T_{RCA}}{T_{CSA}} = \frac{32}{18} \simeq 1{,}78$

$S_A = \frac{A_{RCA}}{A_{CSA}} = \frac{56}{99} = 0{,}57$