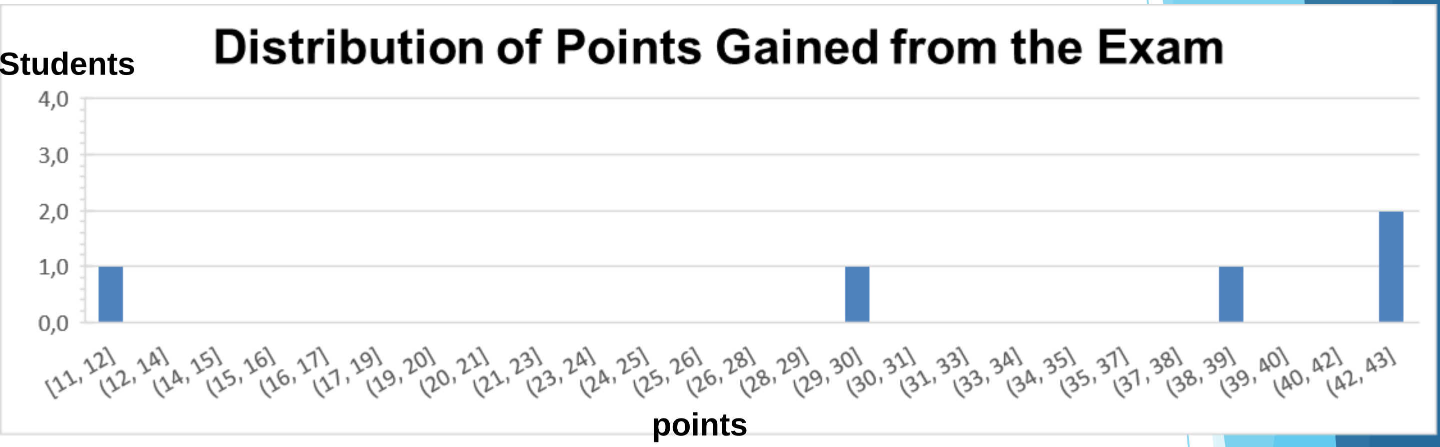
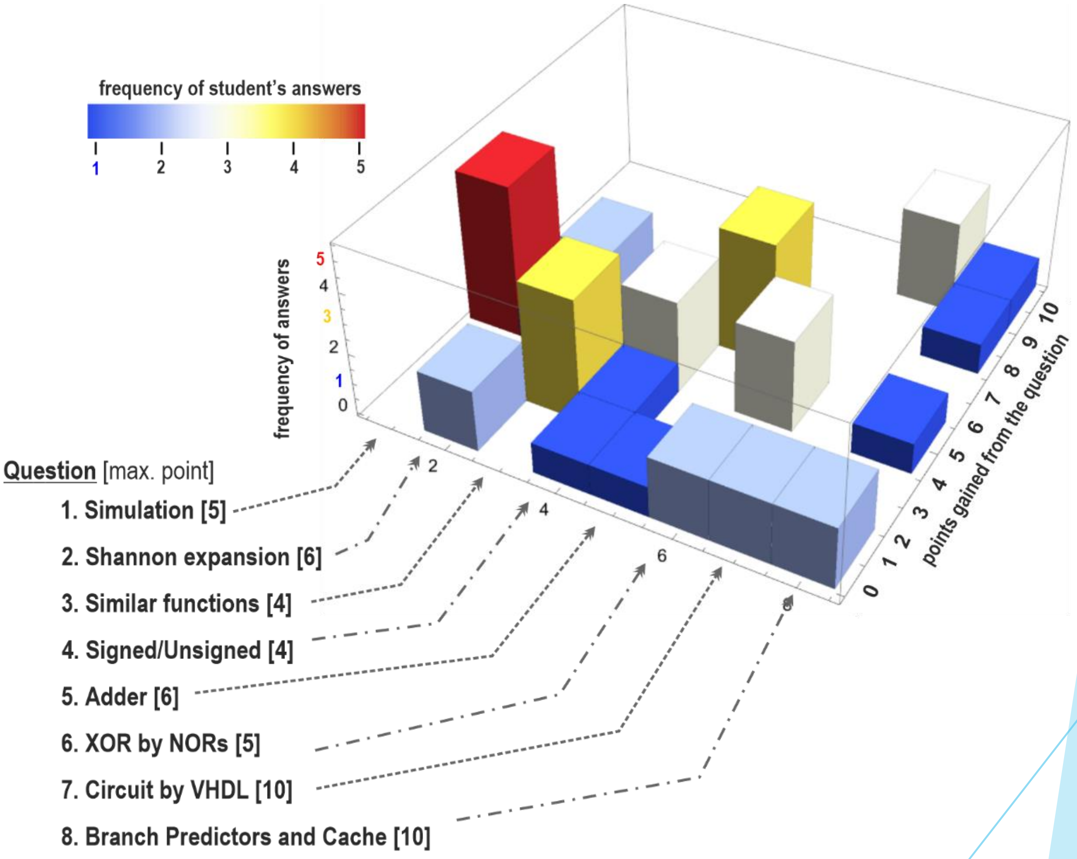
Statistical results of the LSP written exam on Monday 20 January 2025



Distribution of Points Gained from the Exam



Question [max. point]

1. Simulation [5]
2. Shannon expansion [6]
3. Similar functions [4]
4. Signed/Unsigned [4]
5. Adder [6]
6. XOR by NORs [5]
7. Circuit by VHDL [10]
8. Branch Predictors and Cache [10]

Surname and first name :..

Exam Jan 20, 2025 - write here only your answers

Do not write here

1. Inputs A, B and C have values shown in the figure for times t_0, t_1, t_2, t_3, t_4. Write values of X and Y outputs. The intervals between input changes are long so that we can neglect the delays of gates. 5

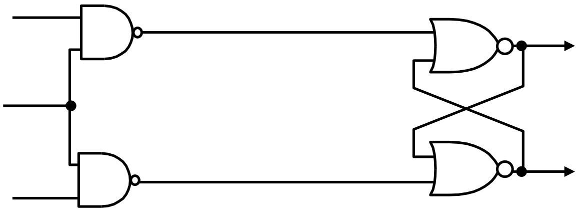
A = ..0..| ..0..|..1..|..1.. |..0..|

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

B = .1..|..0..|..1..|..1.. |..1..|

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

C = ..1..|..0..|..0..|..1.. |..1..|

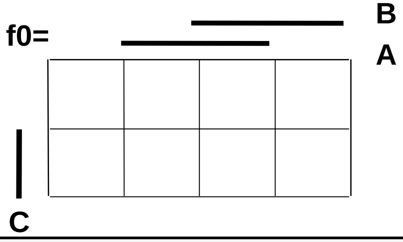


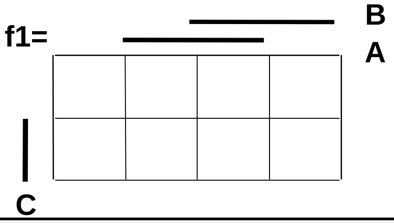
X = ...|... ....|... ....| ......|... ...|

t_0..|..t_1...|.. t_2..|..t_3..|..t_4..|..

Y = ...|... ....|... ....| ......|.. ....|

2. Decompose function X=f(A,B,C, X) from question 1 with the aid of Shannon's expansion into X= (not X and f0(A, B,C)) or (X and f1(A,B,C)) . Write f0 and f1 functions as Karnaugh maps: 6





3. Mark all logic functions that have another equivalent logic function here*:* 4

y1<=((not A or C) and B and not D) or (A and D); y1 □

y2<=((B and (not A or C)) or D) and (A or not D); y2 □

y3<= (A or C or D) and (A or B) and (not D or C); y3 □

y4<=(((not A and not D) or (A and C)) and B) or (A and D); y4 □

4. . If we perform the addition 1023+1023+1023+1023 by a 10-bit adder, what will be its result converted to a decimal number as: 4

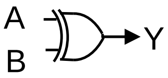
a) unsigned .... b) signed in two's-complement............. ...

5. Complete the schema of the full one-bit adder. 6

Carry$_{in}$
A
B
S
Carry$_{out}$

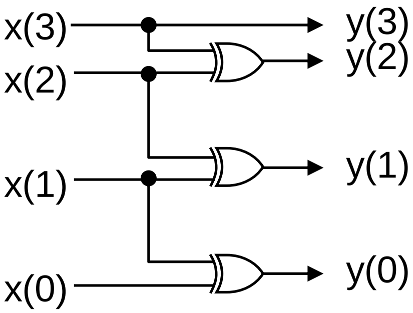
Do not write here

6. You have only 2-input NOR gates. Using only them, create XOR.
Hint: Write Y=XOR(A, B) as a logic function and apply De Morgan law.



5

7. Describe the function of the circuit in the figure using only concurrent VHDL statements, i.e., without sequential statements.
You will get full points only if you use a **single** statement in the section begin-end of the architecture.
You will lose 1 point for each additional command.



.

library ieee; use ieee.std_logic_1164.all;

use ieee.numeric_std.all;

entity Test20250120q7 is **port(**

....

....

);

end **entity**;

architecture rtl of Test20250120q7 is

begin

....

....

....

....

....

....

....

end architecture;

10

Do not write here

8. Consider a simple C program designed for testing the performance of a processor:

```
int i, j; double arr[2000];
for (i=0; i<5; i++) {  for (j=0; j<2000; j++) arr[j]++;
                    }
```

A) How many erroneous branch predictions occur in the program if we assume that for-loops with constant ranges are optimally compiled by **do-while loops** and the processor only uses:

6

1-bit predictors with default states of Not-Taken, **NT**. Misses=..............

2-bit predictors with default states of Weakly Taken, **WT**. Misses=..............

B) Let the program run on a 64-bit processor that has a **directly mapped data cache** of **32 kilobytes** with **blocks of length 4 words** and the array "**arr**" starts at address **0x100000**.

4

What will be the number of cache misses triggered by accesses to **arr** if the cache is empty at the beginning?

Cache misses ……………………………..

10

Results

1.

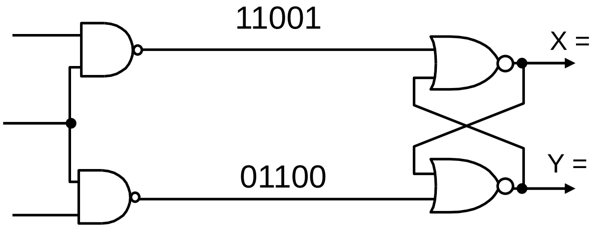
A = **..0..| ..0..|..1..|..1.. |..0..|**

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

B = **.1..|..0..|..1..|..1.. |..1..|**

t_0..|..t_1..|.. t_2..|..t_3..|..t_4..|..

C = **..1..|..0..|..0..|..1.. |..1..|**



X = ...0 ...|...0 ....|...1 ....| ..1....|...0 ...|

t_0..|..t_1...|.. t_2..|..t_3..|..t_4..|..

Y = ... 1...|...0 ....|... 0....| ..0....|.. .1...|

2.

f0= **B** **A**

0	0	1	0
0	0	0	0

C

f1= **B** **A**

0	0	1	0
0	0	1	0

C

3.

y1<=((not A or C) and B and not D) or (A and D);
y2<=((B and (not A or C)) or D) and (A or not D);
y3<= (A or C or D) and (A or B) and (not D or C);
y4<=(((not A and not D) or (A and C)) and B) or (A and D);

y1 ■
y2 ■
y3 □
y4 ■

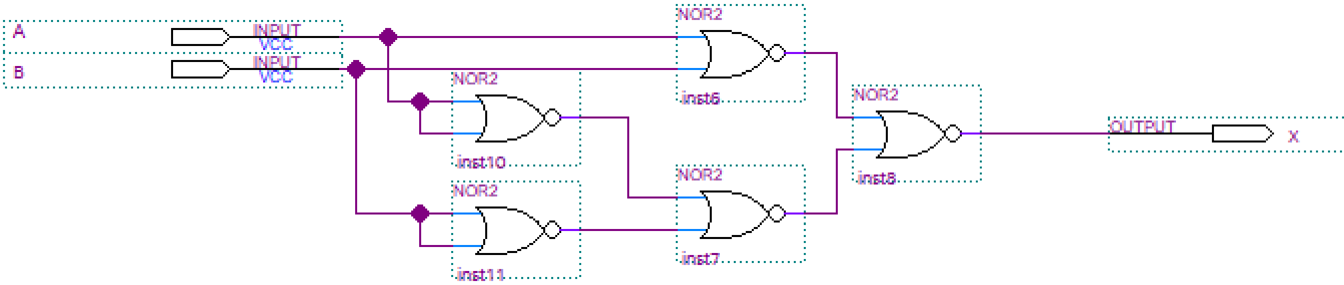
4. .If we add 1023+1023+1023+1023 by a 10-bit adder, what will be its 10-bit result converted to a decimal number.

a) unsigned ..(1024-1)*4=1024-4=**1020**.......... b) signed in two's-complement.. 1023 represents -1, 4*-1=**-4**......... ...

Note: 1024 = 0 at 10-bit adder. Thus, N*1024=0, for any integer N, see Binary Prerequisite

5. *lecture 13*

6.



7.

```
library ieee; use ieee.std_logic_1164.all; use ieee.numeric_std.all;
entity Test20240605q7 is port( x: in std_logic_vector(3 downto 0); y:out std_logic_vector(3 downto 0)); end entity;
architecture rtl of Test20240605q7 is
begin
        y<= ('0'& x(3 downto 1)) xor x;  -- the lecture 4th, slide 8
end architecture rtl;
```

8. for-loop: 1-bit predictors, init. Not-Taken, NT 2 wrong predictions in outer for loop + 5*2(inner) = **12**,

2-bit predictors, init. WT, 1 wrong prediction in outer for loop + 5*1(inner loop) = **6**

2000 elements of the double array need **2000*** **8** (double) = **16000** bytes **- the 32 kB cache** can contain the whole array. Thus, its first read will only generate cache misses, the following will have cache hits! The array elements are loaded into the cache in whole blocks during the first read. Each block contains 4 words (a word has 8 bytes on a 64-bit processor). 1st access of any block always generates a cache miss; the other three words from the already loaded block result in cache hits. The array begins at the address dividable by 4 (size of blocks), i.e., cache misses

= array_elements/size_of_block = **2000/4 = 500.**