Key to Final Exam S1
Computer Architecture

Duration: 1 hr. 30 min.

Last name: .... First name: .... Group: ....

Write answers only on the worksheet.
Do not show any calculation unless you are explicitly asked.
Do not use a pencil or red ink.

Exercise 1 (2 points)

Convert the following numbers from the source form into the destination form. Do not write down the result in a fraction or a power form (e.g. write down 0.25 and not ¼ or 2^{-2}).

Number to Convert	Source Form	Destination Form	Result
101111011.01001	Binary	Decimal	**379.28125**
2CD.48	Hexadecimal	Decimal	**717.28125**
750	Decimal	Base 6	**3250**
BB8.68	Hexadecimal	Base 8	**5670.32**

Exercise 2 (5 points)

Perform the following 8-bit binary operations (the two operands and the result are 8 bits wide). Then, convert the result into unsigned and signed decimal values. If an overflow occurs, write down 'ERROR' instead of the decimal value.

Operation	Binary Result	Decimal Value	
		Unsigned	Signed
10000111 + 10101101	00110100	**ERROR**	**ERROR**
01010010 - 10001101	11000101	**ERROR**	**ERROR**
00110111 + 10111000	11101111	**239**	**-17**
10000001 - 10000010	11111111	**ERROR**	**-1**
11000111 - 01101100	01011011	**91**	**ERROR**

Exercise 3 (5 points)

Amongst the great variety of binary encoding techniques, there is the 2421 code. In this code, the weights of the binary digits are 2, 4, 2, 1, instead of 8, 4, 2, 1. Therefore, several binary patterns are possible for some decimal numbers. For instance, the encoded value of 5_{10} can be either 0101 or 1011. Furthermore, the encoded value of 9_{10} is made up of four ones: 1111. It means that, with four bits, no value greater than 9_{10} can be encoded in 2421 code (unlike the 8421 natural binary form, where values from 0_{10} to 15_{10} can be encoded).

The Aiken code is a kind of 2421 code:

- **The encoded values from 0 to 4 in Aiken code are identical to the encoded values from 0 to 4 in BCD code.**
- **The encoded values from 5 to 9 in Aiken code are identical to the encoded values from 11 to 15 in natural binary code.**

We want to design a circuit that converts a 4-bit natural binary code (DCBA) into its 4-bit Aiken code (D'C'B'A'). Complete the following truth table and the Karnaugh maps below (**draw also the circles**). Then, give the most simplified expression for each output. When a solution is obvious, you do not have to complete its associated Karnaugh map. As a reminder, an obvious solution does not have any logical operations apart from the complement (for instance: A' = 1, A' = $\overline{A}$).

D	C	B	A	D'	C'	B'	A'
0	0	0	0	0	0	0	0
0	0	0	1	**0**	**0**	**0**	**1**
0	0	1	0	**0**	**0**	**1**	**0**
0	0	1	1	**0**	**0**	**1**	**1**
0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	**1**	**1**	**0**	**0**
0	1	1	1	**1**	**1**	**0**	**1**
1	0	0	0	**1**	**1**	**1**	**0**
1	0	0	1	1	1	1	1

BA

DC

D'	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	Φ	Φ	Φ	Φ
10	1	1	Φ	Φ

D' = D + C.A + C.B

BA

DC

C'	00	01	11	10
00	0	0	0	0
01	1	0	1	1
11	Φ	Φ	Φ	Φ
10	1	1	Φ	Φ

C' = D + C.$\overline{A}$ + C.B

BA

DC

B'	00	01	11	10
00	0	0	1	1
01	0	1	0	0
11	Φ	Φ	Φ	Φ
10	1	1	Φ	Φ

B' = D + $\overline{C}$.B + C.$\overline{B}$.A

BA

DC

A'	00	01	11	10
00				
01				
11				
10				

A' = A

Exercise 4 (5 points)

For the whole exercise, write down the result only (do not show any calculation).

Let us consider the two following expressions:

$S1 = A.B.C + \overline{A}.B.\overline{C} + A.\overline{B}.\overline{C} + A.\overline{B}.C$

$S2 = \overline{\overline{A}.B + \overline{A}.\overline{C}}$

1. Give the most simplified expressions of *S1* and *S2*. **The result must be given as a sum of products (without parentheses)**. Do not simplify by using the EXCLUSIVE-OR operator.

$S1 = A.C + A.\overline{B} + \overline{A}.B.\overline{C}$

$S2 = A + \overline{B}.C$

2. Simplify *S1* by using the EXCLUSIVE-OR operator.

$S1 = A \oplus (B.\overline{C})$

3. Write down the maxterm canonical form of *S1*.

$S1 = (A + B + C).(A + B + \overline{C}).(\overline{A} + \overline{B} + C).(A + \overline{B} + \overline{C})$

4. Write down the minterm canonical form of *S2*.

$S2 = A.B.C + A.B.\overline{C} + A.\overline{B}.C + A.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C$

Exercise 5 (3 points)

Perform the operations below. **Show all calculations.**

Base 2

		1	0	0	1	1	1	0	0	0
	–		1	0	1	0	0	0	1	1
			1	**0**	**0**	**1**	**0**	**1**	**0**	**1**

Base 16

		4	8	A	6	
+		E	5	C	7	
	1	**2**	**E**	**6**	**D**	

Base 2: **two digits after the point**

		1	1	1	0	1	0	1	0	1	0	0	0					
	–	1	0	0	0					**1**	**1**	**1**	**0**	**1**	**.**	**0**	**1**	
			1	1	0	1												
		–	1	0	0	0												
				1	0	1	0											
			–	1	0	0	0											
						1	0	1	0									
					–	1	0	0	0									
								1	0	0	0							
							–	1	0	0	0							
											0							

Feel free to use the blank space below if you need to: