Sétif 1 University
Faculty of Sciences
Common Core Mathematics and Computer Science

1st semester (year 2025/26)
Machine Structures 1 course
Kara Abdelaziz professor

Machine Structures 1 exam solution

Exercise 1 : (4 points)

1. Examples of output devices: Monitor (Display) and Printer. (Others include Speakers and Headphones). **(1 point)**

2. Bits for a hexadecimal digit: 4 bits. **(1 point)**

3. Definition of a bit: The absolute smallest unit of data in a computer, representing one of two states: **0** ("off" or low voltage) or **1** ("on" or high voltage). **(1 point)**

4. Definition of a "Minterm": A product (AND term) consisting of all variables of a function, in either true or complemented form. **(1 point)**

Exercise 2 : (5 points)

1. Convert **$(1024)_{10}$** to binary using Successive Euclidean Division by 2 : **(1 point)**

Division	Remainder
1024 ÷ 2 = 512	0
512 ÷ 2 = 256	0
256 ÷ 2 = 128	0
128 ÷ 2 = 64	0
64 ÷ 2 = 32	0
32 ÷ 2 = 16	0
16 ÷ 2 = 8	0
8 ÷ 2 = 4	0
4 ÷ 2 = 2	0
2 ÷ 2 = 1	0
1 ÷ 2 = 0	1

$(1024)_{10}$ = $(10\ 000\ 000\ 000)_2$

2. Convert **$(1100111010)_2$** to hexadecimal using 4-bits equivalence method: **(1 point)**

$(0011 \mid 0011 \mid 1010)_2 = (3 \mid 3 \mid A)_{16}$

$(1100111010)_2 = (33A)_{16}$

3. Convert **$(A.C)_{16}$** to decimal.

First, convert the integer part $(A)_{16}$ to decimal using Polynomial Formula with base-16: **(0.5 point)**

$$\begin{aligned}(A)_{16} &= 10 \times 16^0 \\ &= 10\end{aligned}$$

$$\mathbf{(A)_{16} = (10)_{10}}$$

Next, convert the fractional part $(0.C)_{16}$ to decimal using Fractional Polynomial Formula with base-16: **(0.5 point)**

$$\begin{aligned}(0.C)_{16} &= 12 \times 16^{-1} \\ &= 12 \times (1/16) \\ &= 12 / 16 \\ &= 3 / 4 \\ &= 0.75\end{aligned}$$

$$\mathbf{(0.C)_{16} = (0.75)_{10}}$$

Combine the integer and fractional parts: **$(A.C)_{16} = (10.75)_{10}$**

4. Determining the unknown base *x* solving the following equation **$(22)_x = (16)_{10}$**

To solve this, we need to convert $(22)_x$ to base-10. **(1 point)**

$$\begin{aligned}(22)_x &= 2 \cdot x^1 + 2 \cdot x^0 \\ (22)_x &= 2x + 2\end{aligned}$$

Now we set this equal to $(16)_{10}$:

$$\begin{aligned}2x + 2 &= 16 \\ 2x &= 16 - 2 \\ x &= 14 / 2 \\ x &= 7\end{aligned}$$

So, the unknown base **x = 7**.

5. Binary Multiplication: **(1 point)**

$(101)_2 \times (110)_2 = (11110)_2$

$$\begin{array}{r} 1\,0\,1 \\ \times\ 1\,1\,0 \\ \hline =\ 0\,0\,0 \\ +\ \ 1\,0\,1\,\cdot \\ +\,1\,0\,1\,\cdot\,\cdot \\ \hline =1\,1\,1\,1\,0 \end{array}$$

Exercise 3 : **(5 points)**

1. Decoding the 4-bits binary value $[1111]_{4\text{-bits}}$:

- Decode $\mathbf{[1111]_{UI\text{-}4}}$ to Decimal is to convert $(1111)_2$ to Decimal using PF: **(0.25 point)**
  $(1x2^3) + (1x2^2) + (1x2^1) + (1x2^0) = 8 + 4 + 2 + 1 = (15)_{10}$

$\mathbf{[1111]_{UI\text{-}4} = (15)_{10}}$

- Decode $\mathbf{[1111]_{SM\text{-}4}}$ to Decimal is to convert the magnitude $(111)_2$ to Decimal using PF then adding sign: **(0.25 point)**

  sign = 1 → negative number **(-)**
  $(1x2^2) + (1x2^1) + (1x2^0) = 4 + 2 + 1 = \mathbf{(7)_{10}}$

$\mathbf{[1111]_{SM\text{-}4} = (-7)_{10}}$

- Decode $\mathbf{[1111]_{1C\text{-}4}}$ to Decimal, knowing sign is negative is to invert all bits in $(1111)_2$ then convert it to Decimal using PF: **(0.25 point)**

  sign = 1 → negative number **(-)**
  $(1111)_2 \xrightarrow{\text{invert}} (0000)_2$
  $(0x2^3) + (0x2^2) + (0x2^1) + (0x2^0) = 0 = \mathbf{(0)_{10}}$

$\mathbf{[1111]_{1C\text{-}4} = (-0)_{10}}$

- Decode $\mathbf{[1111]_{2C\text{-}4}}$ to Decimal, knowing sign is negative is to invert all bits in $(1111)_2$ to get 1C, then adding +1 to convert it after to Decimal using PF: **(0.25 point)**

  sign = 1 → negative number **(-)**
  $(1111)_2 \xrightarrow{1C} (0000)_2 \xrightarrow{+1} (0001)_2$
  $(0x2^3) + (0x2^2) + (0x2^1) + (1x2^0) = 1 = \mathbf{(1)_{10}}$

$\mathbf{[1111]_{2C\text{-}4} = (-1)_{10}}$

2. $(-64)_{10} - (65)_{10}$ in 2C is $(-64)_{10} + (-65)_{10}$:

- Convert $(-64)_{10}$ to 8-bits Two's Complement by first converting $(+64)_{10}$ to binary using SED by 2. Knowing the number is negative, all the bits in the binary needs to be inverted to get 1C, then adding +1: **(0.5 point)**

Division	Remainder
64 ÷ 2 = 32	0
32 ÷ 2 = 16	0
16 ÷ 2 = 8	0
8 ÷ 2 = 4	0
4 ÷ 2 = 2	0
2 ÷ 2 = 1	0
1 ÷ 2 = 0	1

$(+64)_{10} = (+1000000)_2$

negative number **(-)** → sign = 1
$(01000000)_2 \xrightarrow{1C} (10111111)_2 \xrightarrow{+1} (11000000)_2$
$(-64)_{10}$ = **$[11000000]_{2C\text{-}8}$**

- Convert $(-65)_{10}$ to 8-bits Two's Complement by first converting $(+65)_{10}$ to binary using SED by 2. Knowing the number is negative, all the bits in the binary needs to be inverted to get 1C, then adding +1: **(0.5 point)**

Division	Remainder
65 ÷ 2 = 32	1
32 ÷ 2 = 16	0
16 ÷ 2 = 8	0
8 ÷ 2 = 4	0
4 ÷ 2 = 2	0
2 ÷ 2 = 1	0
1 ÷ 2 = 0	1

$(+65)_{10} = (+1000001)_2$

negative number **(-)** → sign = 1
$(01000001)_2 \xrightarrow{1C} (10111110)_2 \xrightarrow{+1} (10111111)_2$
$(-65)_{10}$ = **$[10111111]_{2C\text{-}8}$**

- The addition $[11000000]_{2C\text{-}8}$ + $[10111111]_{2C\text{-}8}$: **(0.5 point)**

```
 1
   1 1 0 0 0 0 0 0
+  1 0 1 1 1 1 1 1
= 1|0 1 1 1 1 1 1 1
```

$[11000000]_{2C\text{-}8}$ + $[10111111]_{2C\text{-}8}$ = **$[01111111]_{2C\text{-}8}$**

- Decode $[01111111]_{2C\text{-}8}$ to Decimal, knowing sign is positive is to convert $(01111111)_2$ to Decimal using PF:

  sign = 0 → positive number **(+)**
  $(0x2^7) + (1x2^6) + (1x2^5) + (1x2^4) + (1x2^3) + (1x2^2) + (1x2^1) + (1x2^0) =$ **$(127)_{10}$**

$[01111111]_{2C\text{-}8}$ = **$(+127)_{10}$**

- Knowing the range of 2C 8-bits is [-128, +127], the subtraction should be $(-64)_{10}$ - $(65)_{10}$ = **$(-129)_{10}$** produces an **incorrect result**. The result don't belongs to the range, there is **an overflow**. **(0.5 point)**

3. Encoding the decimal $(\mathbf{+25.75})_{10}$ to Single Precision 32-bit binary number:

- Converting the Number to Binary: **(0.5 point)**

First, convert the integer part $(25)_{10}$ to binary using SED:

Division	Remainder
25 ÷ 2 = 12	1
12 ÷ 2 = 6	0
6 ÷ 2 = 3	0
3 ÷ 2 = 1	1
1 ÷ 2 = 0	1

$\mathbf{(25)_{10} = (11001)_2}$

Next, convert the fractional part $(0.75)_{10}$ to a binary fraction using SM:

Multiplication	Integer part
0.75 x 2 = 1.5	1
0.5 x 2 = 1.0	1

$\mathbf{(0.75)_{10} = (0.11)_2}$

Combine the integer and fractional parts: $\mathbf{(+25.75)_{10} = (+11001.11)_2}$

- Determining the Sign bit (S): **(0.25 point)**

The number +1111.01 is positive → **S=0**

- Normalizing the Binary Number: **(0.25 point)**

The binary number must be converted to the format $1.M\times2^{Ereal}$.

$(11001.11)_2 = 1.100111\times2^4$

Real Exponent: $\mathbf{E_r = 4}$

The Mantissa: **M = 100111**

- Calculating the Biased Exponent (E_b): **(0.5 point)**

We have: **Bias = 127**.

The Exponent formula:

$E_r = E_b$ - Bias → $E_b = E_r$ + Bias

$E_b = 4 + 127 = \mathbf{(131)_{10}}$

Now, we convert E_b to its 8-bit binary representation using SED:

Division	Remainder
131 ÷ 2 = 65	1
65 ÷ 2 = 32	1
32 ÷ 2 = 16	0
16 ÷ 2 = 8	0
8 ÷ 2 = 4	0
4 ÷ 2 = 2	0
2 ÷ 2 = 1	0
1 ÷ 2 = 0	1

$(130)_{10} = (10000011)_2$

$\mathbf{E_b = [10000011]_{8\text{-bits}}}$

- Assembling the Final 32-bit Representation: **(0.5 point)**

  We have the 3 fields in a 32-bits FP representation organized as follows:

  Sing bit (1-bit) | Biased Exponent (8-bits) | Mantissa (23-bits)

  Binary representation: **$[0|100,0001,1|100,1110,0000,0000,0000,0000]_{FP-32}$**

  In hexadecimal: **$[41CE0000]_{FP-32}$**

Exercise 4 : (6 points)

1. Reduction of the Boolean expression: $A + \overline{A}B$ **(1 point)**

$A + \overline{A}B$
$A\overline{B} + AB + \overline{A}B$
$(A\overline{B} + AB) + (AB + \overline{A}B)$
$A + B$

2. The Truth Table of the function: $F(A,B,C) = \overline{AB}+\overline{C}$ **(1 point)**

A	B	C	$\overline{AB}+\overline{C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

3. The PoS form of the previous Truth Table is: $F(A,B,C) = \overline{A}+\overline{B}+\overline{C}$ **(1 point)**

4. Quine-MacCluskey method on the Truth Table:

1. Stage 1: **(1.5 point)**

Minterms	Pass 1	Pass 2
(0) 000✓	(0,1) 00- ✓	(0,1,2,3) 0-- ✗
(1) 001✓	(0,2) 0-0 ✓	(0,1,4,5) -0- ✗
(2) 010✓	(0,4) -00 ✓	~~(0,2,1,3) 0--~~
(4) 100✓	(1,3) 0-1 ✓	(0,2,4,6) --0 ✗
(3) 011✓	(1,5) -01 ✓	~~(0,4,1,5) -0-~~
(5) 101✓	(2,3) 01- ✓	~~(0,4,2,6) --0~~
(6) 110✓	(2,6) -10 ✓	
	(4,5) 10- ✓	
	(4,6) 1-0 ✓	

Stage 2: **(0.5 point)**

Minterm / Implicant	(0) 000	(1) 001	(2) 010	(3) 011	(4) 100	(5) 101	(6) 110
(0,1,2,3) 0--	x	x	x	x			
(0,1,4,5) -0-	x	x			x	x	
(0,2,4,6) --0	x		x		x		x

EPI = {(0,1,2,3) 0--, (0,1,4,5) -0- , (0,2,4,6) --0 }.

$EPI_{Minterm}$ = {(0)000, (1)001, (2)010, (3)011, (4)100, (5)101, (6)110}.

all Minterms are covered. **(0.5 point)**

The final result : $F(A,B,C) = \overline{A} + \overline{B} + \overline{C}$ **(0.5 point)**