

Systems I

FIRST AND LAST NAME:

1st midterm exam

ENROLLMENT NUMBER:

April 5, 2022

Theoretical questions

- a) What is a positional number system? Explain also its general form. [10 points]
- b) How do we represent negative integer numbers in binary form on a computer? Explain in details at least three different forms, how they differ, and what are advantages and disadvantages (if these exist) of each. [10 points]
- c) Minimize the Boolean function with Karnaugh maps (K-maps). [10 points]
- $$F = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}BC\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + ABC\overline{D} + ABCD$$

Tasks in assembly language

- a) Suppose there is a 16-bit variable **data** that stores values x , y , and *color* in the following format:

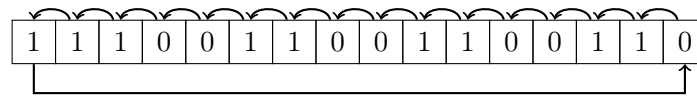
x	x	x	x	x	x	y	y	y	y	y	y	c	c	c	c
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Write a part of the program that unpacks these three values into separate 16-bit variables **x**, **y**, and **color**. To achieve this use AND-masking and bit shifting. [15 points]

- b) Suppose that there is no multiplication and that JC and JZ are the only available conditional jumps. Implement the following conditional statement: [15 points]

```
if (2*(x+y) <= 100)
    x++;
```

- c) A *circular left shift* operation is similar to the *left shift* operation, except that the discarded bit is moved to the first position (see the figure below). Define 16-bit integers **number** and **n** with initial values 0xE666 and 2 respectively. Write a program that performs left circular shift on the **number** variable n -times. [20 points]



At which memory address, according to your program code, will the result be found? [5 points]

- d) Translate the below program to assembly code. What are the final values of all three variables after running the program? [15 points].

```
int x = 0;
int y = 0;
int n = 10;
while (x <= n) {
    y = y + x;
    x++;
}
```

Assembly instructions

Arithmetics

ADD x,y $x \leftarrow x + y$
SUB x,y $x \leftarrow x - y$
MUL x $A \leftarrow A \cdot x$
DIV x $A \leftarrow A/x$
INC x $x \leftarrow x + 1$
DEC x $x \leftarrow x - 1$

Bitwise operations

AND x,y $x \leftarrow x \wedge y$
OR x,y $x \leftarrow x \vee y$
XOR x,y $x \leftarrow x \oplus y$
NOT x $x \leftarrow \bar{x}$
SHL x,n shift x left n -times
SHR x,n shift x right n -times

Jump instructions

JMP addr
JZ/JE addr $x = y$
JNZ/JNE addr $x \neq y$
JA/JNBE addr $x > y$
JNC/JAE/JNB addr $x \geq y$
JC/JB/JNAE addr $x < y$
JNA/JBE addr $x \leq y$

Pseudoinstructions

DB data define byte
DW data define word
ORG addr compile at address

Other CPU instructions

MOV x,y $x \leftarrow y$
CMP x,y $x - y$
HLT stop execution