Introduction

This problem set provides you a set of small yet stimulating digital design challenges. It is intended a) to make you comfortable with the digital design process and b) to provide recap for the material of the previous courses dedicated to the digital circuit design.

Submission

1. The **submission process** of the problem set is done via ORTUS. You have **2 weeks** from the announcement of the problem set to submit your work for the soft deadline and **1 additional week** for the hard deadline.
2. If you submit your work until the **soft deadline**, it entitles you for a feedback from lecturers, which you can use to improve your final submission.
3. The **hard deadline** is final, after which maximum grading for the problem set is reduced by a factor of 2.
4. The submission is expected to be an **archive** following this naming convention: *id_name_surname.zip*.
5. The submission itself can be done in a relatively free form, nevertheless, the drawings should be at the **level of RTL**, VHDL code should be put in the **template files** (if provided) and notably VHDL description design challenges are accomponied with previously prepared **simulations**.

Challenge 1 (1 point)

Derive the conceptual diagram (draw it) for the following code segment:

```
if (a > b and op = "00") then
    y <= a - b;
    z <= a - 1;
    status <= '0';
else
    y <= b - a;
    z <= b - 1;
    status <= '1';
end if;
```

Challenge 2 (1 point)

Assume that a and b are 16-bit inputs interpreted as unsigned numbers. Compare 5 VHDL schematics for the following operations. Synthesize the components and compare their area.

```
a + b                     -- 1.
a + "0000000000000001"    -- 2.
a + "0000000010000000"    -- 3.
a + "1000000000000000"    -- 4.
a + "1010101010101010"    -- 5.
```

Please fill the following table:

Table 1. Table to be filled.

Nr.	b signal	ALMs needed	Combinational ALUTs	Dedicated Logic Resources
1	signal (not constant)			
2	"0000000000000001"			
3	"0000000010000000"			
4	"1000000000000000"			
5	"1010101010101010"			

Challenge 3 (1 point)

Consider the following code segment and you can assume unsigned data type for all the signals:

```
if (a > b) then
    y <= a - b;
else
    if (a > c) then
        y <= a - c;
    else
            y <= a + 1;
    end if;
end if;
```

a) Draw the conceptual RTL diagram. b) Rewrite the code using two concurrent conditional signal assignments (when else construct).

Challenge 4 (1 point)

For each range answer the following question: How many bits are necessary to represent the following number ranges? Assume unsigned data type where possible, otherwise use two's complement signed data type.

- 0..31
- 0..57
- -1..57
- -1986..2020
- 0..68719476736

Challenge 5 (2 points)

We wish to design a circular shift-left (overshifted bits are mapped to the other side) circuit manually. The circuit has the following VHDL interface:

```
entity circular_shifter_left is
    port(
        input  : in  std_logic_vector(15 downto 0);
        shift  : in  std_logic_vector(3  downto 0);
        output : out std_logic_vector(15 downto 0)
    )
end entity;
```

Here:
- *a* is an 16-bit signal to be shifted;
- *ctrl* is a 4-bit signal specifying the amount to be shifted;
- *y* is an 16-bit signal output.

Draw the conceptual diagram (provide it in submission) and use concurrent (when-else) signal assignment statements to describe the circuit.

Challenge 6 (4 points)

Derive VHDL description of a trivial divider which has the following interface

```
entity trivial_divider is
    port(
        clk      : in  std_logic;
        start    : in  std_logic;
        divident : in  std_logic_vector(15 downto 0);
        divisor  : in  std_logic_vector(15 downto 0);
        done     : out std_logic;
        quotient : out std_logic_vector(15 downto 0);
        reminder : out std_logic_vector(15 downto 0)
    )
end entity;
```

NACIONĀLAIS ATTĪSTĪBAS PLĀNS 2020
EIROPAS SAVIENĪBA
Eiropas Sociālais fonds
IEGULDĪJUMS TAVĀ NĀKOTNĒ

The circuit assumes the following usage and functionality:

- *start* signal is active for a single period and it controls the divider to save *divident* and *divisor* signals in internal registers and start the trivial division process.
- The division process is done sequentially by continiously subtracting the divisor from the divident until the reminder is smaller then divident. When this occurs, the *done* signal is set up high indefinately until the next start pulse is received.
- With the assertion of the *done* signal, *quotient* and *reminder* signals should hold the results respectively.
- You can ignore the case when the divisor is 0, nevertheless if you feel obligated to fix it, make the circuit produce quotient 0 and reminder 0 (this might emulate a requirement by the rest of the circuit).