# Fall 2020 CSE4117 Microprocessors Homework 1

Deadline: 18.12.2020 23:59
Demo schedule will be announced later.
You will upload your files to UES as a zip file that contains all answers and its name will be the name of group members. Any of group members can upload the file.

## Question 1 Logisim Part (30 Points)

1. Realize Reptile-8 in Logisim.
2.Connect register 0 to 4x7-segment display unit.
3. Write a program which will have a variable named TERMS in its .data area. Your code should evaluate the sum

$$1^2+2^2+3^2+4^2+....+ \text{TERMS}^2$$

4. When this sum is calculated, it must be loaded into register 0 and become visible in 4x7 segment display. After that, your program must enter into an infinite loop.
5. You must complete and use the assembler given in the lecture notes to assemble your code.
6. Your system must work with 4 KHz clock of Logisim.

Note: You must do the sum in a loop. You **cannot use** the established formula for the sum of squares $1^2+2^2+3^2+4^2+....+n^2 = \frac{n(n+1)(2n+1)}{6}$

## Question 1 Verilog Part (30 Points)

Realize the above system with Verilog and FPGA. In this case, you must connect 4x7 segment display to Register 0 physically via GPIO pins of DE0-nano card. Your system must work with 50 MHz system clock. But, for debugging purposes, you may find it more convenient to connect your clock to a pushbutton when developing the system.

## Bonus for Question 1

You may write your programs to display your result in hexadecimal. But if you display your result in decimal, you will get an additional 10 points. For this, you must write an additional piece of code (Assembly code not in Verilog) to convert your hexadecimal result into decimal (or binary coded decimal, BCD). You must do this in software and NOT in hardware. Some hints on how this can be done is described in the following link:
https://my.eng.utah.edu/~nmcdonal/Tutorials/BCDTutorial/BCDConversion.html
Disregard "BCD conversion in hardware" part.

**Question 2:** (40 Points) In this part of the project you will implement *Matchsticks-like* game in Verilog.

*Matchsticks* is a puzzle game for two players. Several matchsticks are placed in a line and players take turns removing one, two or three sticks from the table. You can't take any more than three sticks on any one turn, you have to take at least one, and you can never put any back. The object is to force your opponent to take the last stick.

In your Matchstick-like game, you have 100 (decimal) sticks initially. Each player can take at least 1, at most 10 sticks in a turn. Number of total sticks will be seen on 7 segment display (3 units on the right side will be used). On the leftmost 7-segment display unit, player number will be displayed (as 1 or 2). After that, first player starts to take sticks. The first player will enter how many sticks to be taken by using dipswitch placed on breadboard. Whenever you press the pushbutton-1 on FPGA, you will read the number of sticks. If dipswitch value that you read is 0 or more than 10 then "-" char will be seen on 7-segment display unit(3 units on the right side and leftmost unit will show player number). Then, you will read new value for first player. When accepted stick number is entered, you will subtract it from total and second player continue to play. The game will continue up to any player takes the last stick. The player who takes the last stick lost the game. The winner will be seen on 7-segment display like "--1-" or "--2-". Whenever pushbutton-2 is pressed the game will restart. (At any time pushbutton-2 can be pressed)
Near the end of the game, there may be less than 10 sticks, in this case it is not possible to enter any number that is more than remaining sticks, if so "-" char will be seen on 7-segment display unit and you will read new number.

The number of matchsticks will be seen on 7-segment display as decimal value. **Do not use hexadecimal values.** (Here you have to convert hexadecimal numbers to BCD in Verilog)