Name: Neptun code: Homework code:

Homework assignment 2 – Assembly programming – Front page

The work can be submitted at any time before the deadline to the lecturer. **Please give detailed information about the process of the design work. Final results without appropriate reasoning will be rejected**. You shall write the homework documentation on your computer, for the submission you have to send it to the lecturer: palfi@mit.bme.hu. As you are going to be an engineer, we require a clean, tidy work. **Untidy, illegible works will be rejected.**

DEADLINE *(no late submissions!):* 13th of December, 23:59.

1. Your task is to write an assembly program for the MiniRISC processor. Based on your homework code (the same as in homework #1) the task is the following:
a. (Preparation: no assembly code is needed, you just have to select your task) If the first digit of your code is <=4 if the digit is odd you have to find the second smallest value in an array stored in the data memory at a given address and a given length. If it is even, you have to find the *index* of the second smallest value.
If the first digit of your code is >4, then you subtract it from 8. If the resulting digit is odd you have to find the second largest value in an array stored in the data memory at a given address and a given length. If it is even, you have to find the *index* of the second largest value.
The code have to show the resulting value on the LEDs.
The starting address of the array is the sum of the 2nd, 3rd, 4th digit of your code. The length of the data is the sum of the remaining four bits.
For the homework assignment you can assume that every value stored in the array is different. You can use any algorithm to solve the task, only: it should work correctly on any array stored in the memory.

Based on the earlier rule and your code the task is: ____

b. In your homework submission you have to include:
- Front page with your name, neptun code and personal code
- The algorithm of the task written in English sentences (=how it works?)
- (Optional: If it helps you: a C-like code which solves the task)
- The assembly program code for the given task **with COMMENTS.**
- The results of a simulation – you have to include the initial contents of the memory and the final contents of the processor registers.