**CS223 Laboratory Project**

**Game of Codes**

**Groups:** Each student will do the project individually. Group size = 1

**Important dates:**

Progress report:  27.11.2017-1.12.2017 during lab times

Demo presentation day and final report: 18.12.2017-22.12.2017 during lab times

In the course project of this semester, all students implement the same project topic. You need to use both the FPGA BASYS-3 board and the Beti board together. BASYS-3 board should be assembled on the Beti board properly using a connector board. You need to refer to documents on Unilica for more information. For the first time, you can also ask lab supervisor to give you a guide on this to prevent damaging your BASYS-3 board. Two pictures from project setup is inside the project files.

Project description: A sequence of random codes is generated by the hardware inside the FPGA. The stepper motor module on the Beti board plays this sequence by applying unique movements for each code. The player person watches the movements of stepper motor and enters the corresponding ASCII character sequence using the 4x4 keypad on the Beti board. For each correct/incorrect character, the player earns/loses 1 point. Total points are displayed on the 7-segment display module on the BASYS-3 board (use only one of digits).

For doing the project, some of the modules that you need are already given to you and you do not need to implement or even modify them. It is enough to just use them as black-box in your design and connect them to other part of the design by writing your own SystemVerilog codes. For doing this project you need to take the following steps:

1- Define your own mapping table: you need to define a mapping table to assign a unique code to each member of a character set. Character set contains 16 characters written on the 4x4 keypad, (‘0’, ‘1’…’#’…). The format of each code:

* A code is composed of two parts.
* Each part represents one of four movements: short right (SR), long right (LR), short left (SL), long left (LL).

For example, you can define the code of character ‘1’ to be SR-LL, meaning first short-right movement and then long left movement. Then, you can define the code for character ‘2’ to be SL-SL, meaning first short-left movement and then another short-left movement. You need to define a unique code for each of the 16 characters on the keypad. **We expect each student to come up with a different mapping table, so please determine your codes randomly**.

2- Pseudo random generator: you have to design a hardware pseudo random generator module which generates a pseudo-random sequence. You use this module to generate random sequence of codes. You can refer to internet to see how a random number generator can be implemented using a linear-feedback shift register (LFSR). However, you need to have your own design and implementation for it and cannot use ready code directly from internet. You need to prepare simulation and proper testbench for your random generator to show during your demo presentation.

3- Stepper-motor: the stepper motor module (available on the Beti board) should play each code by moving the motor according to the character mapping table you defined above. For example, for character ‘A’ with code SR-LL, motor first shortly rotates right, then takes a long rotate to left. Motor should wait for a few moments between each code (character), so the player person can differentiate successive codes. The driver SystemVerilog code for stepper motor is given to you.

4- Keypad: a 4x4 matrix keypad (available on Beti board) is used by player person for entering the recognized code into the FPGA. After watching carefully to the motor movements, the player person will press the corresponding key on the keyboard. 16 keys are available for 16 distinct characters. The driver SystemVerilog of keypad is also given to you.

5-The 7-segment display: if the player presses the right key, his/her score will be incremented, otherwise it will be decremented. The score is displayed on the 7-segment display of BASYS-3 FPGA board. We can limit the scores to be between 0 and 9 for simplicity. There should be a reset button to reset score to 0. The driver SystemVerilog of 7-segment is also given to you.

**NOTES**

* Project should have a ‘reset’ button to reset the game.
* In the progress report on first week, you need to prepare your block diagram for the project and **hand it to your TA in your assigned lab hour**.
* In the demo presentation day, you are asked to upload your code for MOSS testing. Do not upload the ready modules. You will submit your final report at the same time.
* You are not expected to modify or even read the ready modules of keypad, stepper motor and 7-segment. Files are given to you in a non-readable format. Please read the interface description word file and use them accordingly.
* Example\_project\_1 and example\_project\_2 show you how to use the ready modules for keypad, stepper motor and 7-segment display. Projects have been built by Vivado 2017.3 and tested on FPGA. If you use older versions of Vivado, you can copy source and constraint files from project folder to your own new project. Read the description file inside each project to see how to run them.
* Two pictures showing the project setup are inside the project files. You can setup your project like the pictures. Because both keypad and stepper-motor use the same pins of connector cables, you should NOT use two cables to connect them at the same time. Otherwise pin conflict and short-circuit can damage your FPGA pins. As you see in the pictures, one module is connected by cable, and the other is connected by wires.