**Lab assignment 4 / Week 4**

To be done during week 04/03/19 - 10/03/19.

**Preparation for the lab**

Please update your code with the repository. In your “intro_hdl” folder do:

```
git pull
```

**Task 1**.

A. Write a structural version of the 2 bit counter.

B. In the file counter.v, modify “counter” module to produce a modulo “M” counter, where M is a parameter.

**Task 2**.

? A. Write a test bench to test the module ‘reduce’. As discussed in class this module replaces first 1 by a 0, in a continuous string of 1s. Test both mealy and moore versions.

? B. Write a test bench for module named ‘edge_detect’ to test both mealy and moore versions. Suppose that clock has a period of 10 units and the signal named ‘level’ is 0 to begin with. After few clock cycles, ‘level’ become 1, 1 time unit before a rising edge of clock and maintain that value for 24 time units. Compare the output of the mealy and moore versions.

**Task 3**.

Write a module called stack on lines of the module “fifo”. Stack is a “lifo” queue, i.e. the last element stored in the stack is the first element to come out. These operations are called ‘push’ and ‘pop’.

**Task 4**.

A programmable square-wave generator is a circuit that can generate a square wave with variable on (i.e., logic 1) and off (i.e., logic 0) intervals. The durations of the intervals are specified by two 4-bit control signals, m and n, which are interpreted as unsigned integers. The on and off intervals are m*100 time units and n*100 time units, respectively. Design a programmable square-wave generator circuit.