Prof. Bernd Finkbeiner, Ph.D.
Dipl.-Inf. Rüdiger Ehlers
Markus Rabe, M.Sc.
Sebastian Hahn, B.Sc.

Summer term 2012
Problem Set 6

Embedded Systems

Please indicate your **name**, **matriculation number**, **email address**, and which **discussion session** you have been allocated to. We encourage you to collaborate in **groups** of up to **three** students. Only one submission per group is necessary.

Problem 1 (VHDL: 4-Bit Register)

In hardware design, one usually uses a global clock signal to establish a discrete model of time. Figure 1 shows such a clock signal where each period between two rising edges represents a certain (discrete) moment of time. Using this abstraction, it is possible to describe the behaviour of circuits in terms of clock cycle indexes. Here, SIG^i denotes the stabilized value of signal *SIG* during the *first* half (`clk` $= 1$) of cycle i.

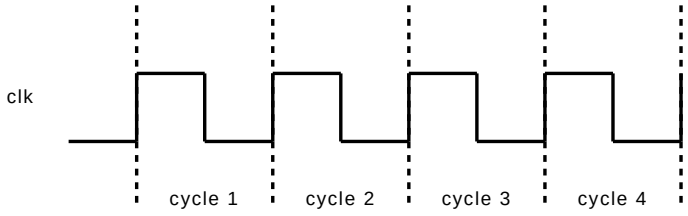

Figure 1: Timing diagram of a typical clock signal

An n-bit register is a standard circuit in hardware design that stores n-bit values. A register has an n-bit input IN, an 1-bit input EN, and an n-bit output OUT. It works as follows: whenever $EN = 1$ the register stores IN, OUT always gives the last value that has been stored in a previous cycle, even if $EN = 1$ in the current cycle. More formally, let IN^i, OUT^i, and EN^i be the input, output, and enabled signals of the i^{th} cycle, respectively, then the behaviour is defined as follows:

$$OUT^{i+1} = \begin{cases} IN^i, & \text{if } EN^i = 1 \\ OUT^i, & \text{otherwise} \end{cases}$$

The input and output interface of a 4-bit register is described by the following VHDL declaration:

```
entity register is
   port(
     in3, in2, in1, in0, en, clk: in bit;
     out3, out2, out1, out0: out bit);
end register;
```

Note that `in0`...`in3` describe the individual input bits in this interface, and `out0`...`out3` describe the output bits. Start with this declaration to develop a 4-bit register in VHDL. [1] When writing the structural architectures, you can use flip-flops and basic gates (NOT, AND, OR, NAND, XOR, NOR) as your starting subentities.

1. Write a VHDL *behavioral* architecture implementing a 4-bit register.
2. Write a VHDL *structural* architecture implementing a 4-bit register.

[1] Note that `clk` is the clock signal whose rising edges define the cycles.