Oblig 4
Structural code using component instantiation

IN3160 / IN4160

Version 1.4/10.02.2021

In this exercise, we will look at instantiation of components. The attached code in `dff.vhd` is a register with an asynchronous reset, where the register is given the value `'0'` when the signal `rst_n` is active low (i.e. has the value `'0'`). For this exercise, the dff component shall be used *without modifications*.

The shift registers implemented in this assignment should have *serial input and both parallel and serial output* (both "SIPO" and "SISO" functionality). The shift registers shall have single bit input, there is no need to construct parallel input[1].

a)

Create an 8-bit shift register by instantiating the component `dff` 8 times, and call this component `shift8`. Use named association in `port map`. Create a test bench that simulates the shift register.

b)

Create a 32-bit shift register by means of the component `dff` and generate the statement. Name this component `shift32`. Use named association in `port map`. Create a test bench that simulates the shift register.

c)

Create an n-bit shift register by means of a generic statement. Name this component `shiftn`. Create a test bench that simulates an instantiation of `shiftn` with the length set at 64 bit.

Approval:

VHDL source file and test bench for the individual questions.

[1] Shift registers are typically used for serial transceivers. Most known is "UART" - Universal Asynchronous Receiver Transmitter. Fast board-to-board communication are normally never parallel, although you may have several serial transmitters working in parallel, each transmitter works individually to perform fast transfer rates and error correction.