Completing this assignment takes three steps:

1. Brainstorm about what the best solution would be for the problem in the assignments and *sketch the chosen solution* (top level and sublevel functional blocks that you think you will need).
2. Write the code, *verify* it with a *testbench* using ModelSim, and debug it.
3. Program the DE1-SoC board with your solution and *test* its functionality. Check the RTL-viewer to see if your design is implemented as you intended.

After completing *every* step, call your instructor to verify that you are done with that step.

Assigment 4: Using the clock

Implement a clock-divider to divide the 50 MHz input clock down to a frequency of 1 Hz with a duty cycle of 50%. Show this signal on LEDR0. Add a nibble-counter[1] going from 0...1...2...3...4...5...6...7...8...9...0...1...2...enz. every second over and over, and display the value of the counter on 7-segment display HEX0.

Create new components for a clock-divider and for a nibble-counter and save them in separate `.vhd` files. Include these components in your design with port-maps, in the same way as you did for the seven-segment display.

The clock-divider and the nibble-counter should both have a reset input. Pushbutton KEY0 should be used to reset the clock-divider and the nibble-counter.

The clock-divider should be a generic component with a generic parameter `divisor` that determines the division factor of the clock frequency.

The nibble-counter should be a generic component with a generic parameter `max_count` that determines the maximum value of the nibble-counter.

When you want to simulate, temporarily decrease the period in between counts, since in one second, using the 50 MHz clock, there are 50 million cycles. It will therefore take very long to simulate this.

[1] A nibble is a group of four bits. So a nibble-counter is a counter with four output bits.