Lab 3: Latch, Flip-Flop, and Counter

Today, we will dive into sequential logic circuits by implementing memory cells and clocked circuit components.

14:15-14:35: Discussion – Lab 2

1. Any issues?
2. Solutions

Task 1 – D-Type Latch

1. Implement a D-type latch in a new VHDL file, *d_latch.vhd*, following the port configuration in figure 1 and the characteristics in table 1
    a. Think about which coding style is the smartest; *sequential* or *concurrent – if/else*, *case*, *when*, or *with/select*
    b. Bonus: Make the width of the input *D* and outputs *Q* and *nQ* configurable with a single parameter *n_bits* using the *generic* keyword
2. Test and verify in simulation

D	en	Q	nQ
-	0	Q	nQ
0	1	0	1
1	1	1	0

Table 1: D-type latch

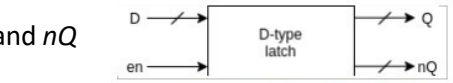

Figure 1: D-type latch block diagram

Task 2 – D-Type Flip-Flop

1. Implement a D-type flip-flop in a new VHDL file, *d_flip_flop.vhd*, following the port configuration in figure 2 and the characteristics in table 2
    a. Think about coding style…
    b. Bonus: Make the width of the input *D* and outputs *Q* and *nQ* configurable with a single parameter *n_bits* using the *generic* keyword
2. Test and verify in simulation

D	en	Q	nQ
-	0	Q	nQ
-	1	Q	nQ
-	falling_edge	Q	nQ
0	rising_edge	0	1
1	rising_edge	1	0

Table 2: D-type flip-flop characteristics table

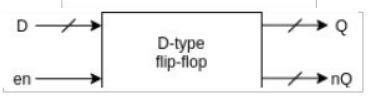

Figure 2: D-type flip-flop block diagram

Task 3 – Counter

1. Implement a 4-bit counter in a new VHDL file, *counter.vhd*, following the port configuration in figure 3
    a. Let the counter count up on each rising edge of *clk* when *rst* is low and *en* is high
    b. Let the counter pause on it's current value (stop counting) when *en* is low; Let the counter value reset to 0 when *rst* is high
    c. Bonus: Let the number of bits in the counter be configurable with a single parameter *n_bits* using the *generic* keyword
2. Test and verify in simulation

Figure 3: Counter block diagram