Lab02 – Conditional assignments, Registers & Counters

1 Conditional assignments

1.1 Simple multiplexer circuit

Write a VHDL model of a 4:1 multiplexer (MUX) with 4-bits input vector `D`, 1-bit output `Q` and select signal `SEL`. How many bits are required to encode `SEL`? You are free to chose the approach, but think of the options. Then write the VHDL of the test-benches and validate their operation using waveforms.

1.2 Single D-FF

Write the model of a single D Flip-Flop: the input `D` is assigned to output `Q` when a control signal `Clk` changes from 0 to 1 (rising edge). Otherwise any change on the input is ignored. Then write the VHDL of the test-benches and validate their operation using waveforms.

2 Registers

For all the specifications below:

- Write the VHDL models.
- Use `generics` to make registers configurable at instantiation time.
- Write test-benches assuming that the number of bits N=8.
- Simulate designs and validate their operation using waveforms.
- For all of the above assume an asynchronous reset signal `RST`. If `RST=1`, the the output is set to 0 no matter the value of input `D`.

2.1 Serial-in / Serial-out (shift) registers

Write a VHDL model of an N-bit LSB shift register with enable signal EN. When $EN = 1$, input bit D is pushed to LSB. All subsequent bits are shifted right (assuming that LSB is on the left and MSB on the right). The old MSB is lost. The output `Q` is the old MSB.

2.2 Parallel-in / Parallel-out registers

Write a VHDL model of an N-bit register with parallel input/output. When signal `LOAD=1` then `Q<=D`. Otherwise the old value of `Q` is retained.

2.3 Parallel or Serial-in / Parallel-out registers

In this model you will add a choice from parallel or serial input using `SERIAL` signal. When set to 1 the register operates as serial input shift register, otherwise operates as parallel input. The output is parallel.

3 Counters

Write an N-bit up/down counter using arithmetic operators. A signal `UP` determines the counting direction: when it is set to 1 we count up, otherwise we count down. Then write the VHDL of the test-benches and validate their operation using waveforms.