NOMBRE:	GRUPO:

REMARKS:

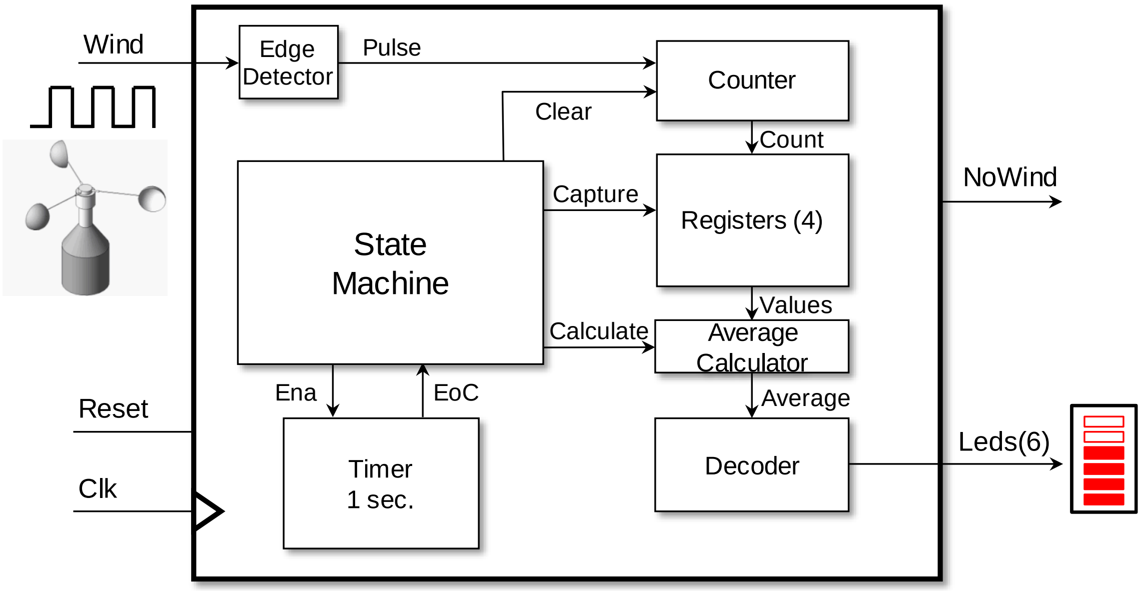
- Time: 3h
- Answers to different parts must be returned separately. If you do not answer a part, you must return a blank sheet with your name.
- Please make sure your NAME and GROUP appears in every sheet you return.
- It is not required to write entities unless explicitly asked for
- The statement does not specify the circuits requiring clock and reset. The student must include them when necessary

PART 1: DIGITAL CIRCUITS

We want to design a digital circuit to generate a visual representation using LEDs of the wind speed, measured with an anemometer. For this exercise, we will assume wind speed is proportional to the anemometer spinning speed. Spinning speed can be determined by the number of pulses per time unit provided to the *Wind* circuit input, which will have a maximum of 150 pulses per second for maximum wind speed.

The circuit will measure wind speed by counting the number of pulses received during a second. In order to minimize the effect of wind strokes in measurements, the circuit will calculate the average of the last four measurements.

The block diagram of the circuit is shown in the Figure below.



Circuit inputs and outputs are:

- Wind (input, 1 bit): square signal with a frequency proportional to wind speed.
- NoWind (output, 1 bit): it will activate ('1') when no wind is detected, that is, when last four measurements are null.
- Leds (output, 6 bit): representation of the average wind speed in thermometric code (the number of active LEDs is proportional to wind speed).
- Reset: asynchronous initialization, active high.
- Clk: clock, rising edge active (frequency is irrelevant).

Wind input is connected to an edge detector, which produces single cycle pulses. *Counter* counts the number of received pulses during a second, measured with *Timer*. Register stores the last four *Counter*

NOMBRE:	GRUPO:

values. Average Calculator calculates and stores the average of the values stored in *Register* and Decoder transforms the average value to a 6 bit thermometric code.

State Machine controls the process in the following way:

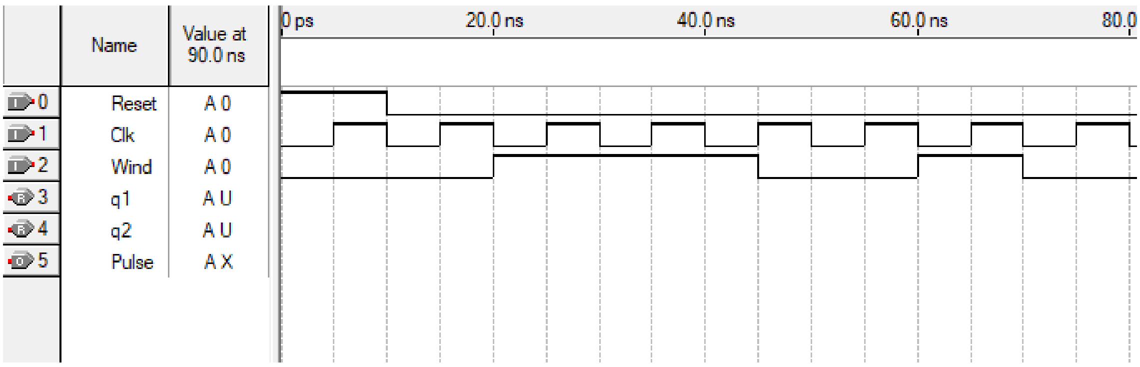
- *Counter* is reinitialized and *Timer* is enabled.
- *Counter* counts pulses during a second.
- After one second, *Register* is enabled to capture a new *Count* value and discard the oldest stored value.
- After that, *Average Calculator* calculates the average, including the new stored value and it is stored in the *Average* register.
- The process starts again

EXERCISE 1.1 (0.5 POINTS)

Edge Detector block generates single clock cycle pulses every time there is a rising edge at the input. The VHDL code for this circuit is:

```
signal q1,q2: std_logic;
………
Pulse <= q1 and not q2;
process(Clk, Reset)
begin
  if Reset='1' then
    q1 <= '0';
    q2 <= '0';
  elsif rising_edge(Clk) then
    q1 <= Wind;
    q2 <= q1;
  end if;
end process;
```

a) Draw an schematic of the circuit described by this code using logic gates and flip-flops
b) Fill in the following chronogram with the simulation of this circuit



NOMBRE:	GRUPO:

EXERCISE 1.2 (1 POINT)

Determine the bit size for *Counter* to be able to count a maximum of 150 pulses.

Write a VHDL description for this counter, considering *Pulse* signal increments the count value and *Clear* signal reinitializes the count value to zero. *Clear* takes precedence over *Pulse*.

Write declarations for involved signals.

EXERCISE 1.3 (1 POINT)

Write a VHDL description of four registers with a proper size to store four different count values (*Count*) values from *Counter*. They must work shifting values, so that when *Capture* signal is activated, *Count* value will be stored in the first register, which will copy its value to the second and so on, discarding the oldest value stored in the last register.

Write a VHDL description of the *Average Calculator* circuit tha calculates the average value of the four previous registers and store the result in other register (*Average*) when *Calculate* signal is activated ('1').

Write declarations for involved signals.

EXERCISE 1.4 (1,5 POINTS)

Draw a state diagram for *State Machine*, controlling the other circuits according to the description in the first page.

Write a VHDL description of the circuit implementing this state diagram.

Write declarations for involved signals.

Timer is enabled with *Ena* signal ('1'), counting during one second. After a second, *EoC* (End of Count) is activated ('1'), and it remains active until *Ena* is disabled. This way, it is required to disable *Ena* between consecutive timings, for at least one clock cycle.

EXERCISE 1.5 (1 POINT)

Write the VHDL description of *Decoder*. It generates *Leds* output according to the average calculated value (*Average)*, using a thermometric code. The number of turned on LEDs is proportional to *Average* value:

Average	LEDs on
0-24	1
25-49	2
50-74	3
75-99	4
100-124	5
125-150	6

Write the VHDL description of the circuit generating *NoWind* output.

Write declarations for involved signals.

NOMBRE:	GRUPO: