CentraleSupélec

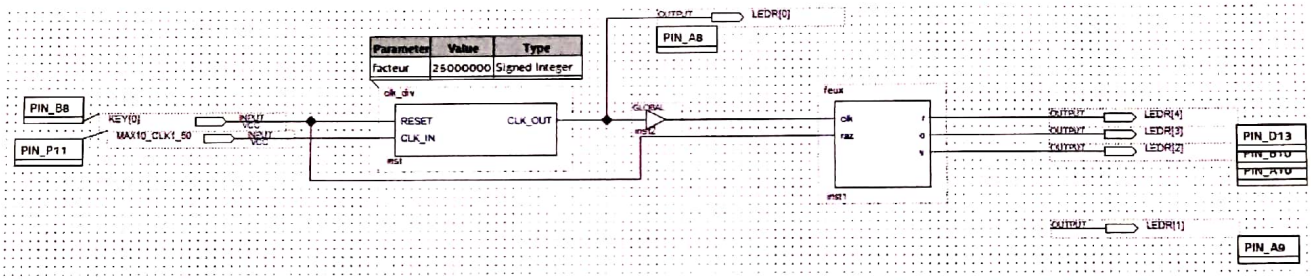
**ICS M2 Degree - 2021-2022 - Advanced Digital Electronics**

**Practical Session N°4**
**Realization of a finite state machine**

The aim of this tutorial is to describe the sequencing of a traffic light for pedestrians. We consider a single traffic light and not a crossroads. This has a red light request button.

1°) Project opening

We are currently working in BE_ALTERA in the TD4 subdirectory.
A project containing a traffic light finite state machine code is provided to you.
This project uses the code presented in the lecture with a slightly different approach.
You launch quartus and you open the feu_signalisation project. The project contains a frequency divider to provide a 1 Hz clock and the state machine module shown above.




Analyze the code of the lights module and identify the differences compared to the code provided during the lecture. Download the code to the board and check that it works

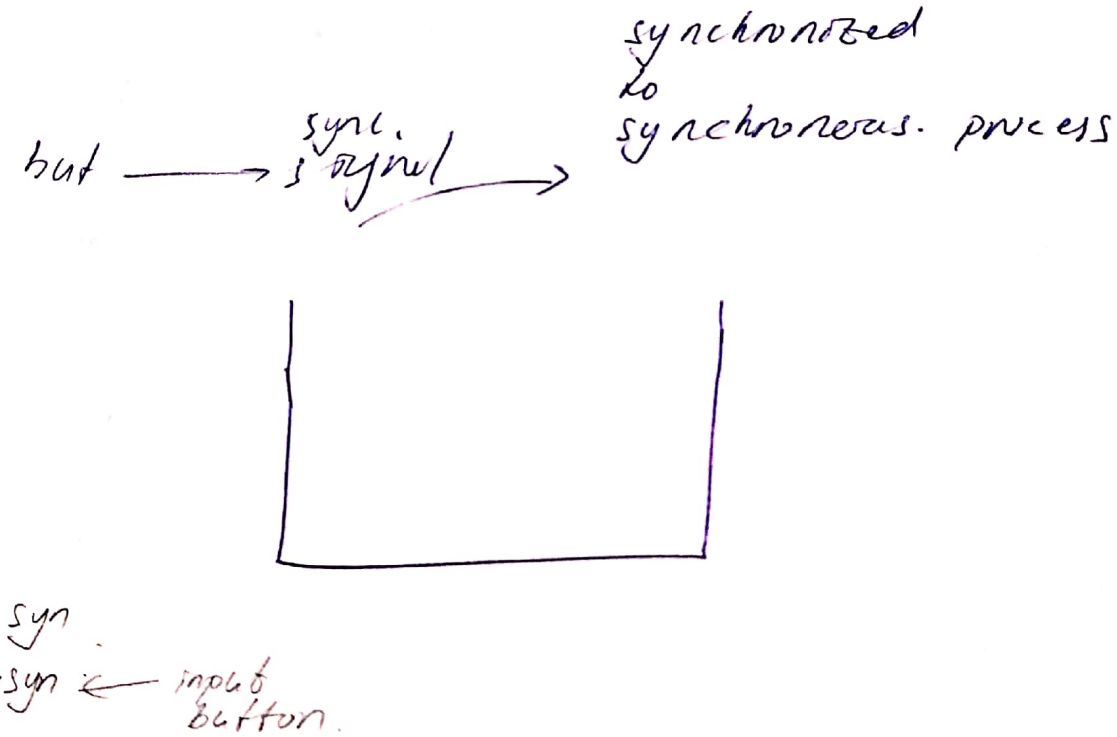
2 °) The goal is now to achieve a signaling system for which the holding time in each "rouge" "vert" or "orange" state can be configured.
We start by modifying the state machine so that there are only 3 states left. Each state will initially last only one clock cycle.
Rename the states to keep only 3 and code this second version.

3 °) We will now configure the duration of each state. To do this, we will introduce 3 generic parameters which indicate the number of clock cycles that each state lasts.
We introduce a counter, a signal of unsigned type, which is loaded at each change of state and is then decremented. The next change of state will be when the counter has reached the value 0.
Modify the feux.vhd code accordingly and compile it. (processing -> analyze current file)
The entity has changed because generic parameters have been added. You have to recreate the symbol (file -> create -> create symbol file for current file)
We will then "Update" the symbol of the feux module so as to display the generic parameters thus created.
We position ourselves with the mouse on the diagram above the "feux" block and with the right key "update symbol".
Compile the project and download the code. Check that the states last the expected number of cycles

4 °) We add a pedestrian call button operating as follows:

If the light is green and the user presses the button, the light turns orange. In the other cases nothing is changed.
Knowing that pressing the button will necessarily be asynchronous with the clock, what must be done to ensure correct operation of the system?
Add an additional input to the feux module and add the requested functionality. The added input will be connected to the KEY [1] signal. The latter is active low.
Compile, download and test.

5 °) We now want the change to orange to take place only after 5 seconds of in the vert state (so as not to block the cars too often). If the user presses for the first 5 seconds the command must be memorized. Suggest a modification to implement this operation.



CentraleSupélec

**ICS M2 Degree - 2021-2022 - Advanced Digital Electronics**

**Practical Session N°5 & 6**

**Simulation using the modelsim tool**
**Use of processes**

# Realization of a data path and a state machine

We wish to realize an operator that allows to calculate the average power of a signal.
We will use the formula :

$$P = \frac{1}{N}\sum_{k=0}^{N-1} {x_k}^2$$

Where N is the number of samples used to make the calculation
The $x_k$ signal arrives synchronously to the clock signal, and the P signal must remain present until the next calculation.
The calculation is started only if the control signal C is at 1. If it is at 0, the operator eventually finishes his last calculation and then goes into standby mode..

1°) Propose a data path diagram allowing the realization of this operator

2°) Identify all the signals used to control this data path

3°) Propose a state machine diagram to control the previous signals

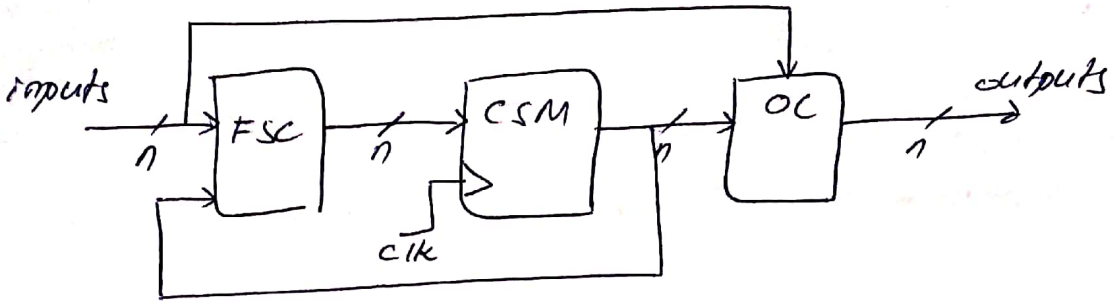
4°) Describe and compile the entity of the operator in the td5 directory. The number N must appear as a power of 2 : $N=2^l$ with l the generic parameter equal to 10 by default.

$N = 2^k$

5°) Write the testbench which will allow to check the correct operation of the operator. This testbench will generate a sinusoidal input signal with an amplitude equal to the maximum admissible amplitude. The amplitude of the input signal will be divided by 2 for every N samples. Compile and simulate the testbench.

6°) Describe the state machine part of the operator, then check its operation in simulation using the previous testbench.

7°) Describe the data path using integer operators and verify the operation of the whole circuit. In particular, check that the calculated power is divided by 4 at each new calculation. The size of the operators will be calculated from the generic parameter l.



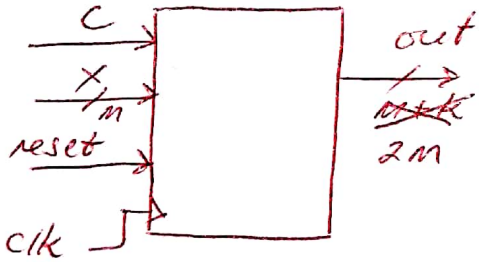

FSC = future state computation (combinatorial)

CSM = current state memorizing (sequential)

OC = output computation (combinatorial)

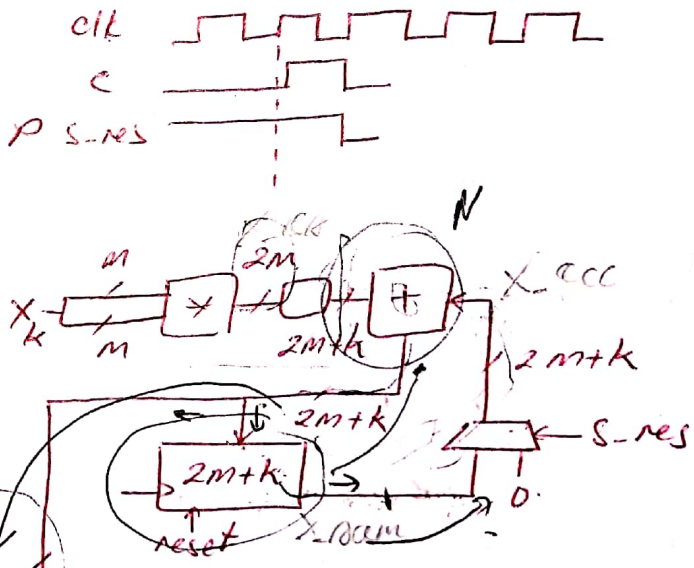
1. in one state block
2. input dependance - application dep.

latch → memorize @input

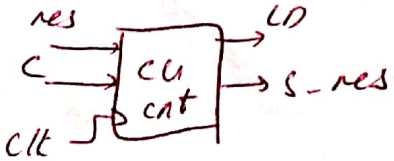



$$N = 2^k$$

$$\rho = \frac{1}{N} \sum_{k=0}^{N-1} x_k^2$$




block




r_ses : internal reset