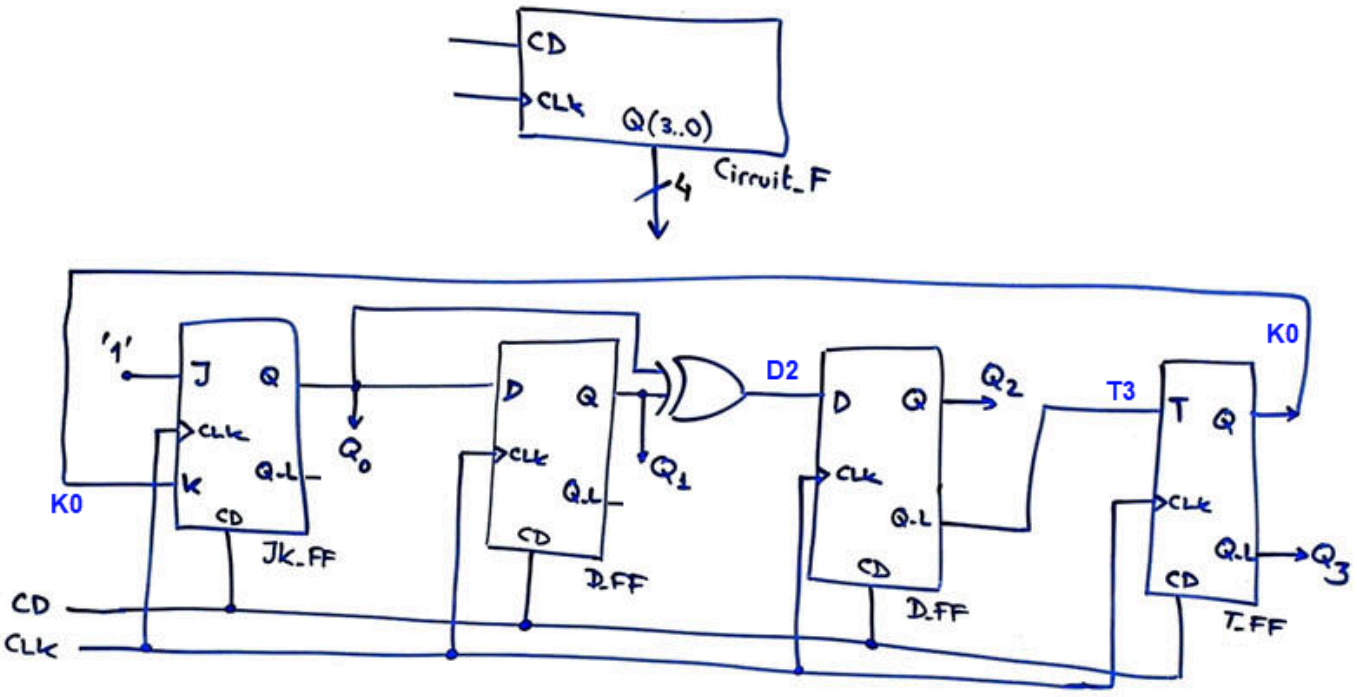
UPC. EETAC. Bachelor's degree. 2A. CSD Digital Circuits and Systems. **Exam 2.** Grades available by January 19th. Questions and revision: January 19th, 17:00 – 19:00, office C4-103P. **January 16th**

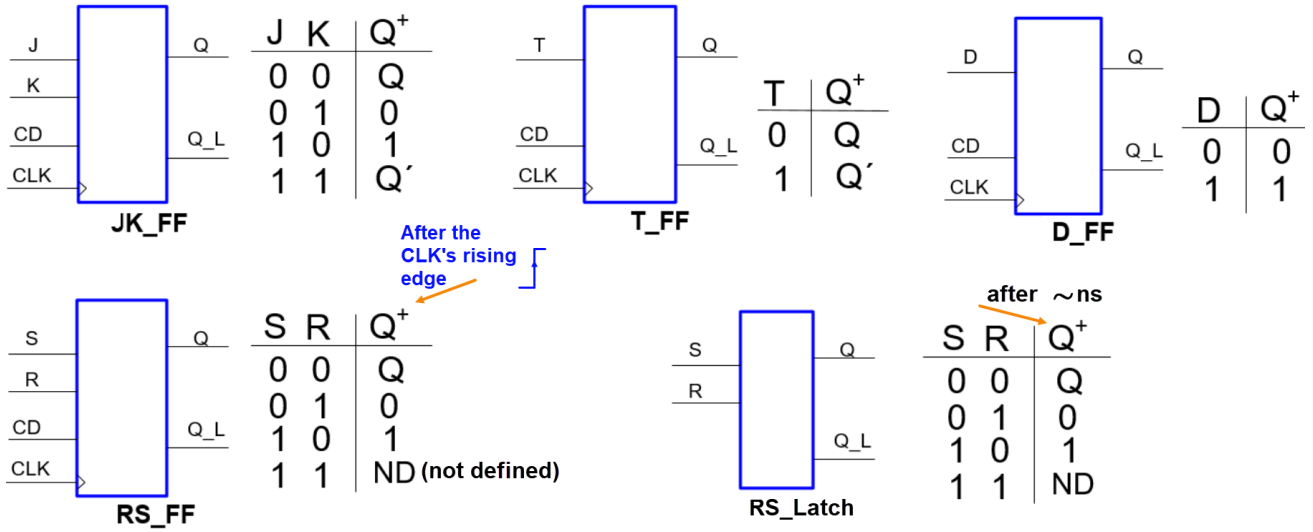
NOTE for all problems and questions: firstly, draw your circuits, sketches and diagrams; secondly, explain your plan and what you do. Justify your results.

Problem 1 (2p)

a) Analyse the circuit in Fig. 1a by means of a timing diagram, this is analysis method I, naming all the signals and indicating sampled values on the CLKs' rising edges of interest. Find the binary codes (numbers) generated at the output vector Q(3..0). In this circuit, due to **K0** feedback, you have to consider the evolution of all the flip-flops every **CLK** period.



a)



b)

Fig. 1a) *Circuit_F* based on flip-flops. b) *Flip-flop* and latch symbols and their functions tables.

b) Draw the circuit again ready for VHDL translation, as if you liked to obtain the solution performing VHDL ModelSim simulations. How many VHDL files are required in this plan C2 project?

Problem 2 (4p)

We want to design in VHDL a synchronous 5-bit one-hot code rotator (*rotator_onehot_5bit*) with **CE** and left-right reversibility (**LR_L**) using one of these two strategies: (**A**) plan C1 FSM plan or (**B**) plan C2 based on counter truncation.

These are the output codes generated when rotating left (**LR_L** = '1'): "00001", "00010", "00100", "01000", "10000", "00001", …

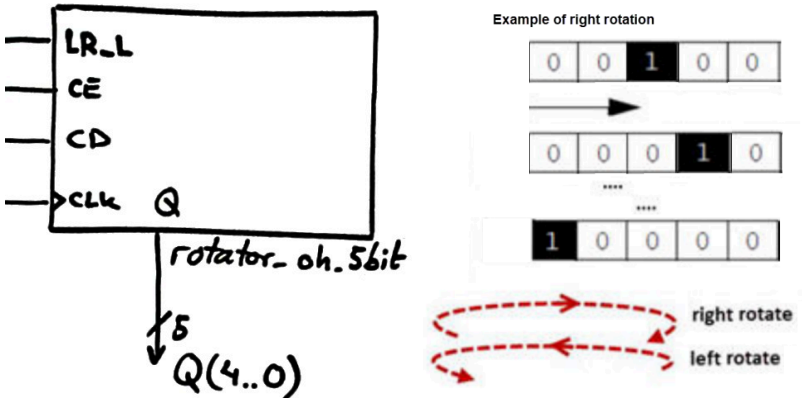


Fig. 2a. Symbol and example of bit rotation when **LR_L** = '0'

a) Draw the circuit's function table. Plot an example of a timing diagram. Draw its **state diagram**.

If you choose the strategy **A**)

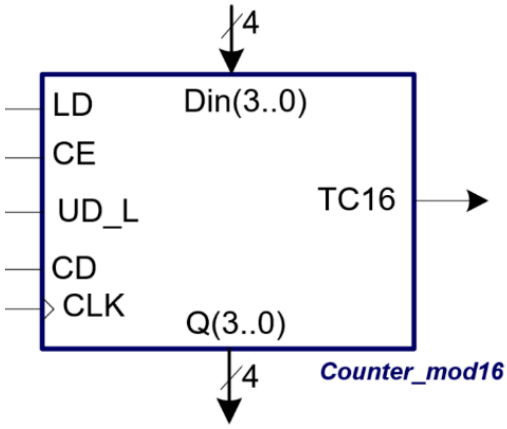
b) Draw the FSM structure consisting of CC1, CC2 and state register, indicating the name of the inputs and outputs. How many *D_FF* are necessary in this application if encoding internal states in one-hot? Draw the internal structure of the state register if the system is encoded in binary radix-2 (sequential).

c) Write the truth tables for CC1 and CC2. Draw the flowchart for CC1 and the main ideas on how it is translated into VHDL. How many VHDL files will include this project?

If you choose the strategy **B**)

b) Plan your circuit using the standard synchronous *Counter_mod16* represented in Fig. 2b and other components.

c) How many *D_FF* are necessary in this application? How many VHDL files will include this project?



LD	CE	UD_L	Q^+	Synchronous operation after the CLK's rising edge
1	x	x	Din	Parallel load (register data)
0	0	x	Q	Do nothing (inhibit)
0	1	1	$(Q+1)_{mod16}$	Up counting in binary
0	1	0	$(Q-1)_{mod16}$	Down counting in binary

TC16 = '**1**' when CE = '1' *and* [(Q = 15 *and* UD_L = '1') *or* (Q = 0 *and* UD_L = '0')]; '0' otherwise

Fig. 2. Symbol and function table of the 4-bit synchronous universal radix-2 counter component that is currently used as a building brick for many applications.

d) Draw a circuit diagram to answer this question: if the t_{CO} (CLK to output propagation time) of a *D_FF* is 3.4 ns, and the propagation delay t_P of a generic logic gate is 2.5 ns, estimate the maximum speed at which your designed rotator can work. Justify your calculations according to your circuit.

Problem 3 (4p)

We will redesign the lamp controller proposed in LAB6 for a microcontroller PIC18F46K22 using our FSM style of programming in C language. For the design phase #1: when clicking the push-button **PB** two times, the lamp **Z** turns ON. When clicking it again another time the lamp turns OFF. Use external interrupts from the push-button **PB**.

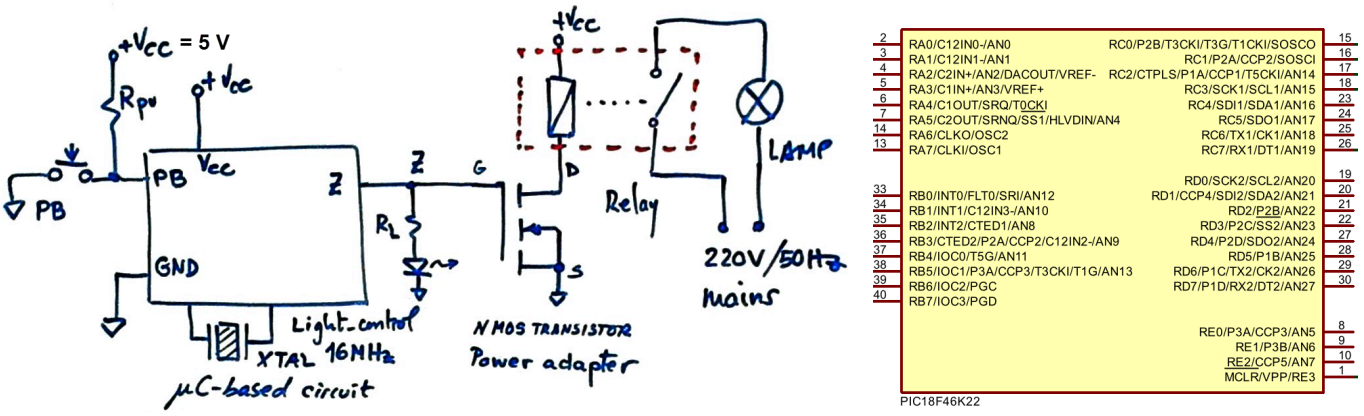


Fig. 3a. Symbol of the lamp controller and the µC chip from Microchip.

a) Draw the electronic schematic to implement the *Light_control* indicating what port pins to use and why. Draw the power-on reset (MCLR_L) circuit and explain how it works. Draw the oscillator circuit using a 16 MHz quartz crystal. Explain the *init_system*() function.

b) Draw the list of RAM variables that we will use in this application. Draw the **state diagram**. Organise the main program as a FSM in our CSD way. Explain the interrupt service routine *ISR*().

c) Draw the hardware-software diagram. Solve the function *write_outputs*() using the three column flowchart, memory bitwire operations and C code generated.

d) Solve the function *state_logic*() truth table and its flowchart.

For the next design phase #2: The classroom has installed in a ceiling corner a pyroelectric IR presence sensor type Kemet SS-430L-N. It generates a logic '1' when people are detected in the room.

e) Explain how the state diagram is modified and draw schematics on how to use the internal TMR0 so that the lamp **Z** will turn OFF automatically to save energy when for 3 minutes people are not detected in the room.

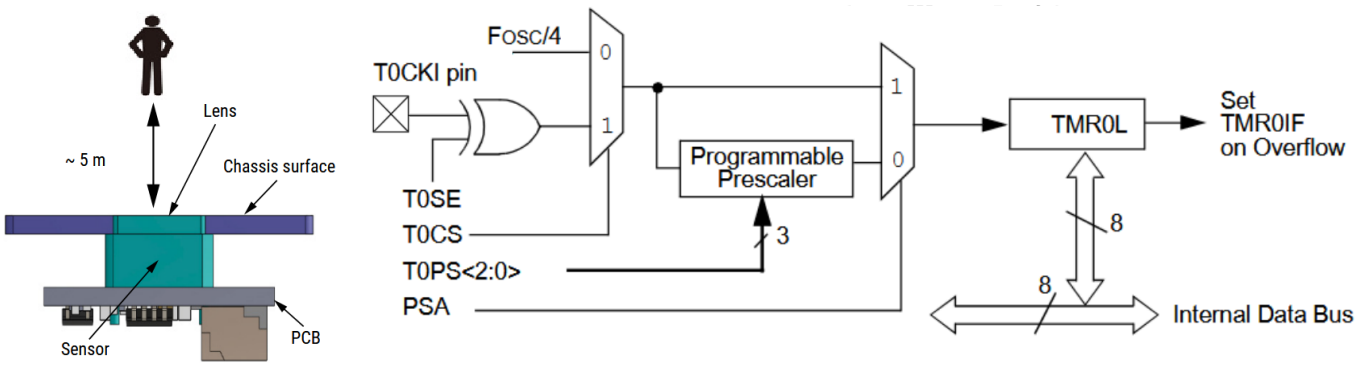


Fig. 3b. The idea of the presence detector and the internal simplified block diagram for the TMR0 peripheral.

⏩As usual, if you are interested, we invite you to develop, test and prototype these projects in our labs.