UPC. EETAC. Bachelor Degree. 2A. Digital Circuits and Systems (CSD). F. J. Robert, J. Jordana. Grades will be available on January 21st. Questions about the exam: office time.

Exam 2. Solve in separate sheets of paper the problems 1, 2, and 3, and then choose between 4 or 5. January 14th

Problem 1. **2.5p**

a) Analyse the circuit in Fig. 1 drawing a timing diagram of the outputs Q(3..0) indicating which signals are sampled and when.

b) How many states this circuit have? Which is the output value Q(3..0) for each state?

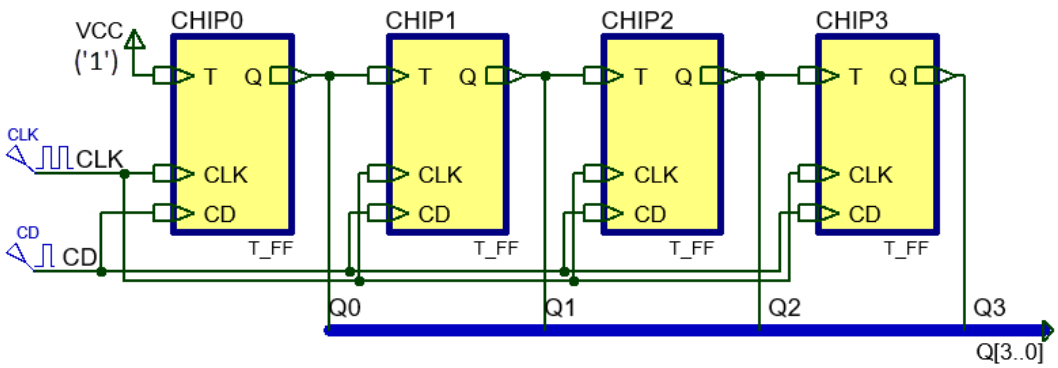


Fig. 1 Synchronous circuit based on *T_FF* having this function table:

T	Q^+
0	Q
1	Q´

Use this type of stimulus signals to determine the outputs.

Problem 2. **2.5p**

A synchronous application requires the *CLK_generator* component represented in the Fig. 2 for obtaining the three square waves at the given frequencies from a crystal oscillator of 25.175 MHz.

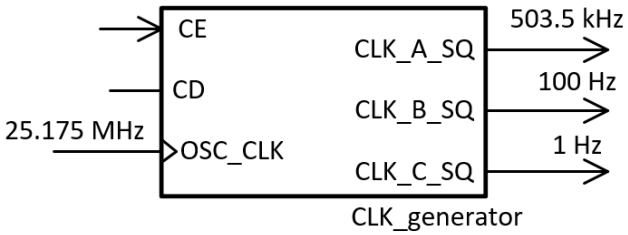


Fig. 2 CLK generator circuit

a) Invent an internal architecture following a plan C2 based on *T_FF* and *freq_div_N* components where *N* is an ajustable parameter.

b) How many VHDL files are required to synthesise the project? How many *D_FF* are required in the *CLK_generator*?

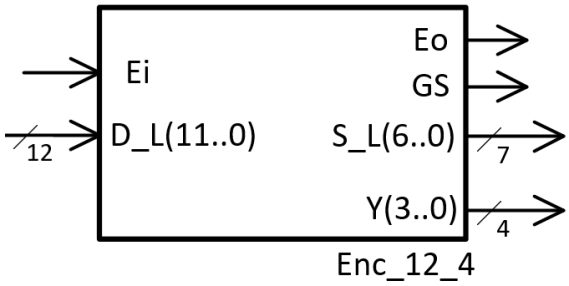
c) Explain the architecture and how the *freq_div_N* is designed following a plan Y for synchronous counters.

Problem 3. **2.5p**

In PLA#3.1 we studied the 12-key priority-high encoder represented in Fig. 3 that was based on a microcontroller PIC18F4520. Basically, to solve its truth table, we had to define and program using our CSD style the functions *init_system()*, *read_inputs()*, *truth_table()* and *write_outputs()*.

a) Plan a section of *read_inputs()* polling the pins corresponding the input vector **D_L(11..0)** accordingly to the given port assignment. This is: bitwise operations, flowchart and example C code to obtain the ***var_D*** type *int*.

b) Plan a section of *write_outputs()* driving the pins corresponding the output vector **Y(3..0)** accordingly to the given port assignment. This is: bitwise operations, flowchart and example C code to write from the ***var_Y*** type *char*.



Port pin	PORTA	PORTB	PORTC	PORTD	PORTE
7	-	D_L(10)	Ei	·	-
6	-	D_L(9)	·	D_L(5)	-
5	Eo	D_L(8)	D_L(7)	D_L(4)	-
4	·	Y(3)	D_L(6)	D_L(3)	-
3	S_L(2)	S_L(6)	·	D_L(2)	-
2	S_L(1)	S_L(5)	·	D_L(1)	Y(1)
1	S_L(0)	S_L(4)	Y(2)	D_L(0)	Y(0)
0	D_L(11)	S_L(3)	·	·	GS

Fig. 3
12-key encoder and pin assignment for a Microchip PIC18F µC.

----------(Solve problem 4 or problem 5) ----------

Problem 4. LED rotator using hardware and VHDL 2.5p

Our aim is to implement a LED rotator like the one represented in Fig. 4, which is basically a FSM acting as an 8-bit one-hot shift register that runs continuously. The system operates by means of two control signals:

CE_ L = 1 (do nothing), **CE_ L** = 0 (run rotating) **SR_L** = 0 (shift right), **SR_L** = 1 (shift left)

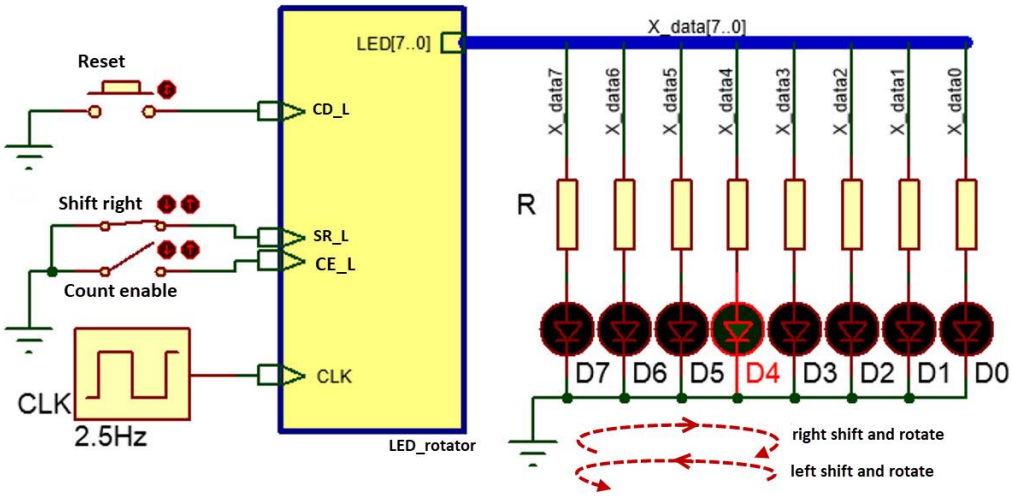
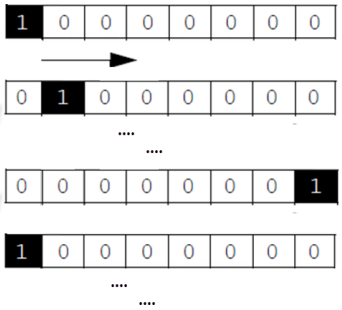


Fig. 4
An 8-bit LED rotator.

Example of right rotation:



a) Draw the function table and the state diagram indicating state transitions and outputs.
b) Particularise the FSM architecture to this problem and draw the state register based on D_FF. Deduce how many D_FF are required if coding states in binary sequential.
c) Write the truth table of CC1 and CC2 and their equivalent behavioural interpretations using flowcharts.
d) Translate the main details of the flowcharts to VHDL.
e) In Fig. 4 the circuit works at 2.5 Hz, but which is the maximun speed of operation if the target chip is a Xilinx CPLD with a CLK to output delay of t_{CO} = 4.5 ns and a propagation time for one gate t_P = 3.3 ns?

Problem 5. LED rotator using a microcontroller and C code 2.5p

Our aim is to implement the same LED rotator specified in Problem 4 using a microcontroller PIC18F4520, C language and our programming style. The external CLK in Problem 4 is replaced by the internal 8-bit **TMR0** peripheral to generate interrupts (TMR0IF) every 16 ms.

a) Draw the function table and the state diagram indicating state transitions and outputs.
b) Draw the hardware schematic. Buttons and switches, resistors, inputs SR_L, CE_L, outputs LED(7..0), reset circuit MCLR_L and quartz crystal oscillator of 8 MHz. Explain how to configure the inputs and outputs in the *init_system().*
c) Draw the hardware/software diagram indicating the required RAM variables and how the FSM is solved in software.
d) Draw the truth tables and their equivalent flowcharts for the *state_logic()* and *output_logic()* functions.
e) Which is the interrupt service routine *ISR()* used for in this application? Draw its flowchart.
f) Calculate the **N1** (prescaler), **N2**(TMR0) and **N3** (postscaler variable) values required to generate a CLK of 2.5 Hz (*var_CLK_flag* period = 400 ms) to run the machine.