UPC. EETAC. Bachelor Degree. 2A. Digital Circuits and Systems (CSD). F. J. Robert, J. Jordana. Grades will be available by June 7. Questions about the exam: office time.

Exam 2 **May 31, 2019**

Problem 1. **2.5p**

The aim of the circuit in Fig. 1 is to activate a LED lamp during a time *T*, following the operating mode represented in Fig. 2.

a) Draw a possible state diagram, indicating state transitions and outputs. Name states as S1, S2, etc.
b) Draw and adapt the standard FSM architecture indicating signals and ports. How many D_FF (D-type flip flop) are required to implement this sequential circuit if the states are coded in binary sequential? Justify your response.
c) Draw the CC1 truth table and its algorithmic state machine chart or flow chart.
d) Draw the CC2 truth table and its algorithmic state machine chart or flow chart.

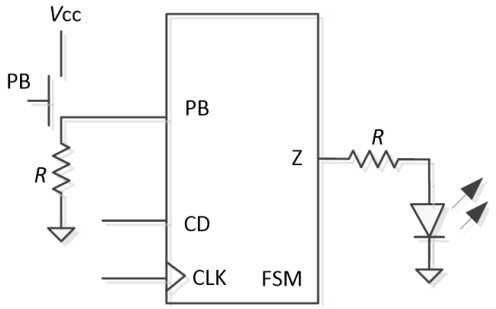

Fig. 1 Sequential circuit.

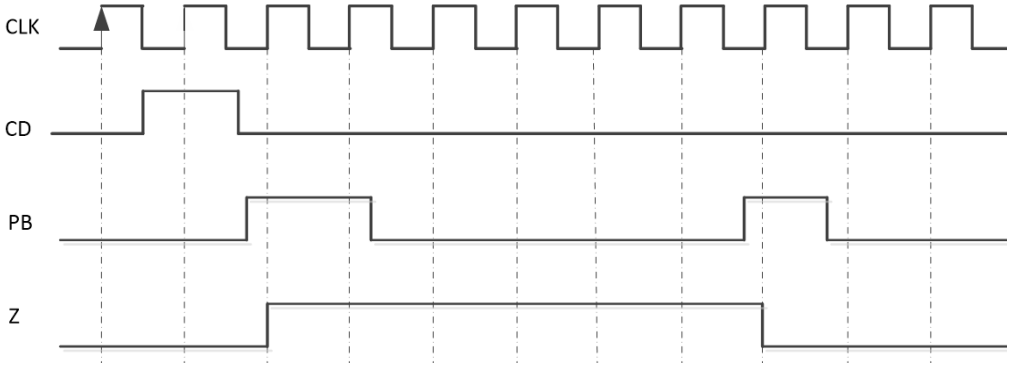

Fig. 2 Timing diagram that shows the circuit's mode of operation.

Problem 2. **2.5p**

We want to implement a *Counter_mod14* (Fig. 3a) using a hierarchical strategy based on two components: a *Quad_MUX2* and a *Counter_mod16* (Fig. 3b).

a) Draw an example timing diagram for the operating mode: *CE* = '1' and *UD_L* = '0'.
b) Plan the schematic of the *Counter_mod14* indicating the name of the signals and components involved. Explain how it works and the use of the multiplexer. How many *D_FF* the *Counter_mod14* will include?
c) Explain and justify the truth table of the circuit to drive the *Quad_MUX2* selection input. Explain and justify the truth table for driving the *LD* input of the *Counter_mod16*.
d) How many VHDL files are involved in this design? How the *Counter_mod16* component is organised internally: using the plan X (state enumeration) or using the plan Y (STD_LOGIC_VECTOR state signals)?

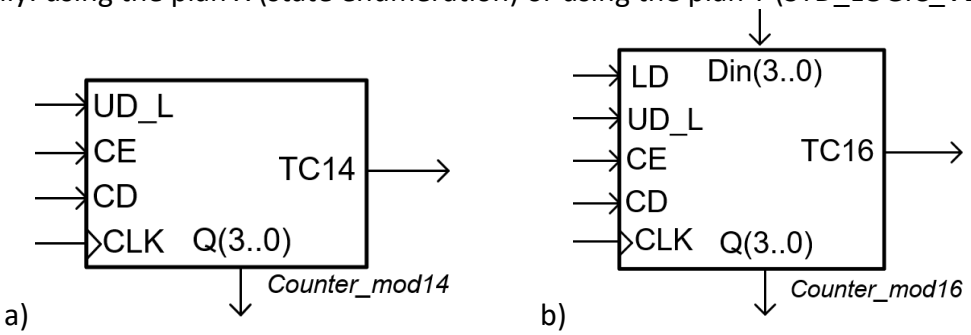

Fig. 3 Symbols for the top entity to be designed and one of the components.

Problem 3. **5p**

Let's design a simple 2-wire asynchronous data transmitter based on a µC for sending to another computer a nibble (4-bits) of data. It is basically a right-shift register. The application symbol and pinning of the PIC18F4520 is represented in Fig. 4. We'll use the FSM style of programming in C language. The format for the serial output once the start-transmission *ST* rising edge is detected by means of an interrupt is: Start-bit ('0'), *Data_in* (0), *Data_in* (1), *Data_in* (2), *Data_in* (3); and then the end-of-transmission *EoT* pulse is generated to indicate that the transmitter has ended the process (see the Fig. 5). *Serial_out* is held high when idle.

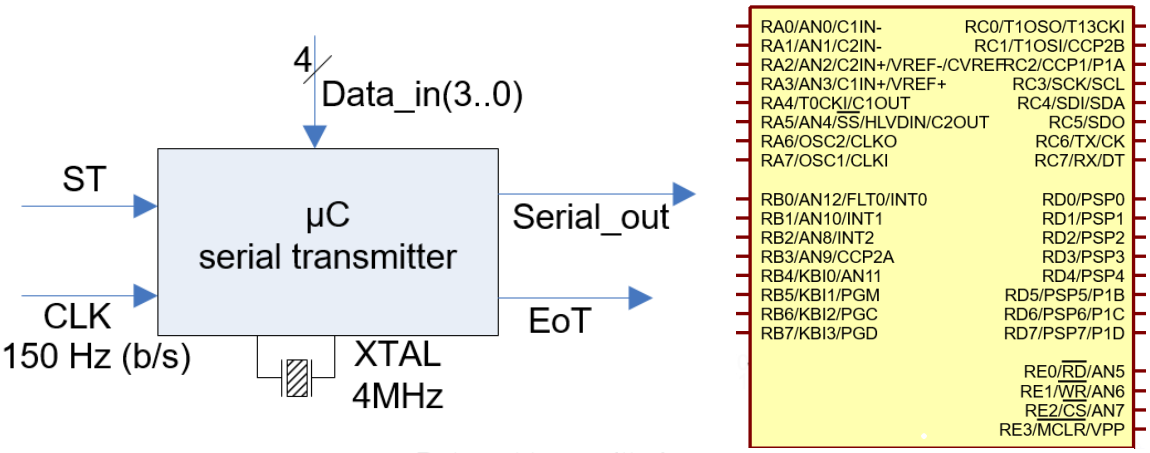

Fig. 4 Symbol of the data transmitter and the µC PIC18F4520 from Microchip.

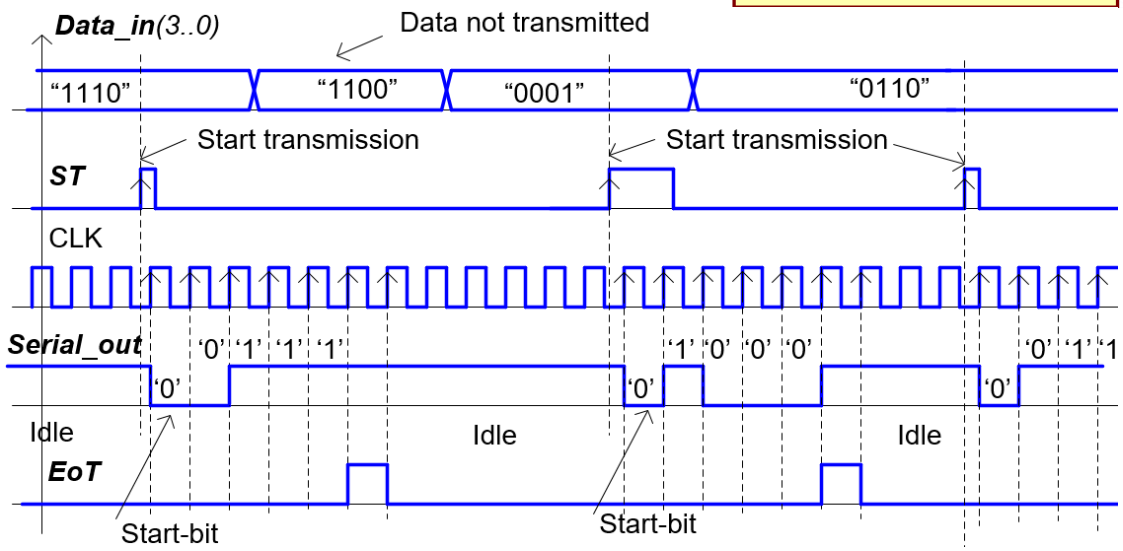

Fig. 5 Example of a section of a timing diagram where it can be seen how the data is read and right shifted in a single wire.

a) Draw the hardware schematic. Reset circuit, XTAL oscillator, *Data_in*(3..0) = (RA2, RA1, RD7, RD6), *CLK* (RB0), *ST* (RB1), *Serial_out* (RC5), *EoT* (RC2). Explain how to configure the inputs and outputs in the *init_system*().

b) Draw the hardware/software diagram indicating the required RAM variables and how the FSM is solved in software. The transmission sequence will start when a rising edge is detected at the start *ST* push button by means of the interrupt INT1IF. The CLK input will generate an interrupt INT0IF so that a new bit is transmitted at a time at the *Serial_out* as represented in Fig. 5. Transmission speed is 150 b/s.

c) How *read_input*() works to generate the *char* variable *var_Data_in*?

d) How the variables *var_Serial_out* and *var_EoT* are written to the corresponding pins using *write_outputs*() without interfering the other µC port pins?

e) Which is the *ISR*() used for? Propose the flow chart.

f) Draw an state diagram showing the state transitions and the outputs for each state. Name the states, for instance: Idle, Start_bit, Data_0, Data_1, etc.

g) Draw the truth tables and their equivalent flow charts for the *state_logic*() and *output_logic*().

h) How to use and program the TMR0 peripheral in 8-bit mode to replace completely the functionality of the external CLK as the baud-rate generator?