UPC. EETAC bachelor's degrees in telecommunications systems and in network engineering. Semester 2A. Digital Circuits and Systems (CSD). Grades will be available online by June 21st. Questions about the exam at office time.

Second exam. **17th June 2024**

Solve [Problem 1 or Problem 2 or Problem 3] and [Problem 4 or Problem 5 or Problem 6]

[2.5p] ---------- [7.5p]

Problem 1.

a) Analyse the circuit in Fig. 1 using a timing diagram, naming all the signals of interest and indicating sampled values.

b) Find the binary codes (numbers) generated at the output vector Q(2..0).

c) Discuss what will be the circuit's CLK maximum frequency of operation if *flip-flop* components are implemented internally as standard FSM. Typical CLK to output propagation time (t_{CO}) of a *D_FF* is 3.7 ns. Typical propagation delay (t_P) of a generic logic gate is 3.9 ns.

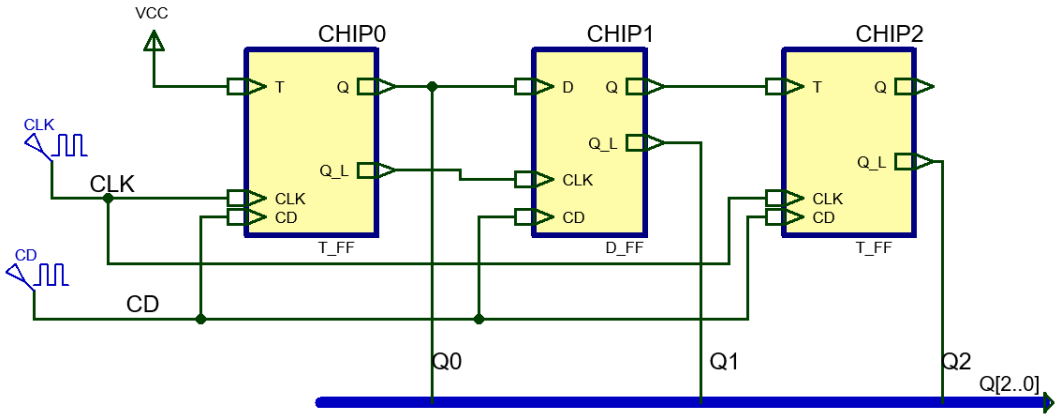
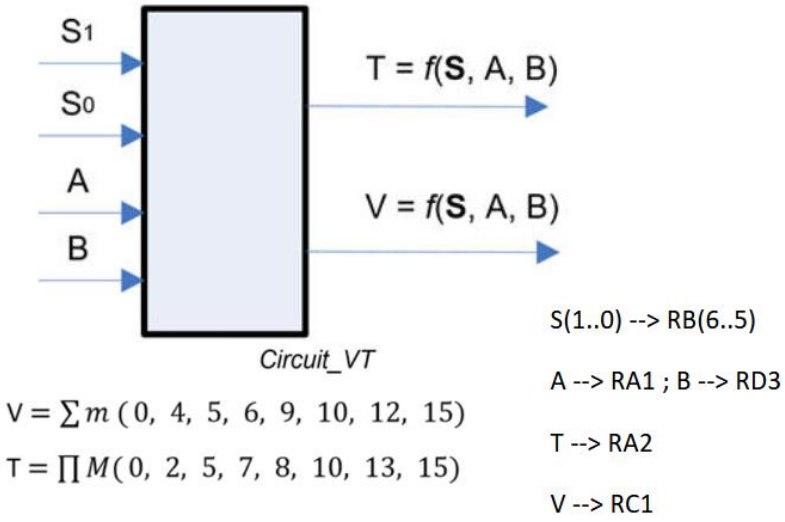


Fig. 1. Circuit based on flip-flops.

Problem 2.

Solve the combinational circuit presented in Fig. 2 using a PIC18F4520 microcontroller.



$V = \sum m\,(0,\ 4,\ 5,\ 6,\ 9,\ 10,\ 12,\ 15)$

$T = \prod M(0,\ 2,\ 5,\ 7,\ 8,\ 10,\ 13,\ 15)$

Fig. 2. Combinational circuit.

a) Draw the hardware schematic including microcontroller reset and oscillator circuits.

b) Organise the main program in our CSD way. Propose a hardware-software diagram naming all the electrical signals, RAM variables and software functions. Explain how to configure the µC in *init_system().*

c) Organise and explain using a flowchart the interface function *read_inputs()* and its bitwise operations.

d) Organise using a flowchart the interface function *write_outputs()* and its bitwise operations.

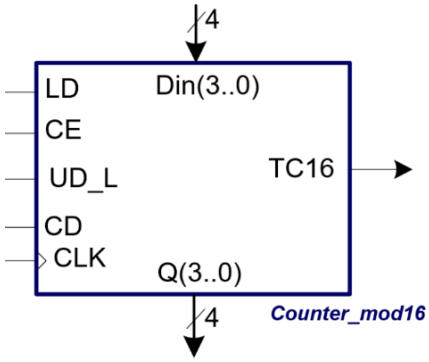
e) Infer the *truth_table()* software function using a behavioural interpretation and the corresponding flowchart.

Problem 3.

Invent a synchronous BCD down counter module 72 (*Counter_BCD_mod72*) with count enable (CE) and terminal count (TC72) using **plan C2** and standard components such *Counter_mod16* (Fig. 3) and other circuits if necessary.

a) Specifications: symbol, function table, timing diagram, state diagram.

b) Planning schematics and components. How many *D_FF* will include the *Counter_BCD_mod72* design?



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. 3. Symbol of the 4-bit standard universal binary radix-2 counter component (*Counter_mod16*).

Problem 4. LED lamp dimmer

The luminaries that we use today work with high efficiency LED lamps and the light generated can be dimmed using a control circuit. Let us propose two alternative designs.

Section 1. Hardware design (FPGA, VHDL).

To modulate the light intensity we can use a simple pushbutton that produces a PWM waveform. In this example, let us fix five intensity levels as shown in Fig. 4. Every time that we click the up-button (UB) we generate a new intensity level from 0% (Lamp OFF) to 100% (Lamp ON, max light intensity).

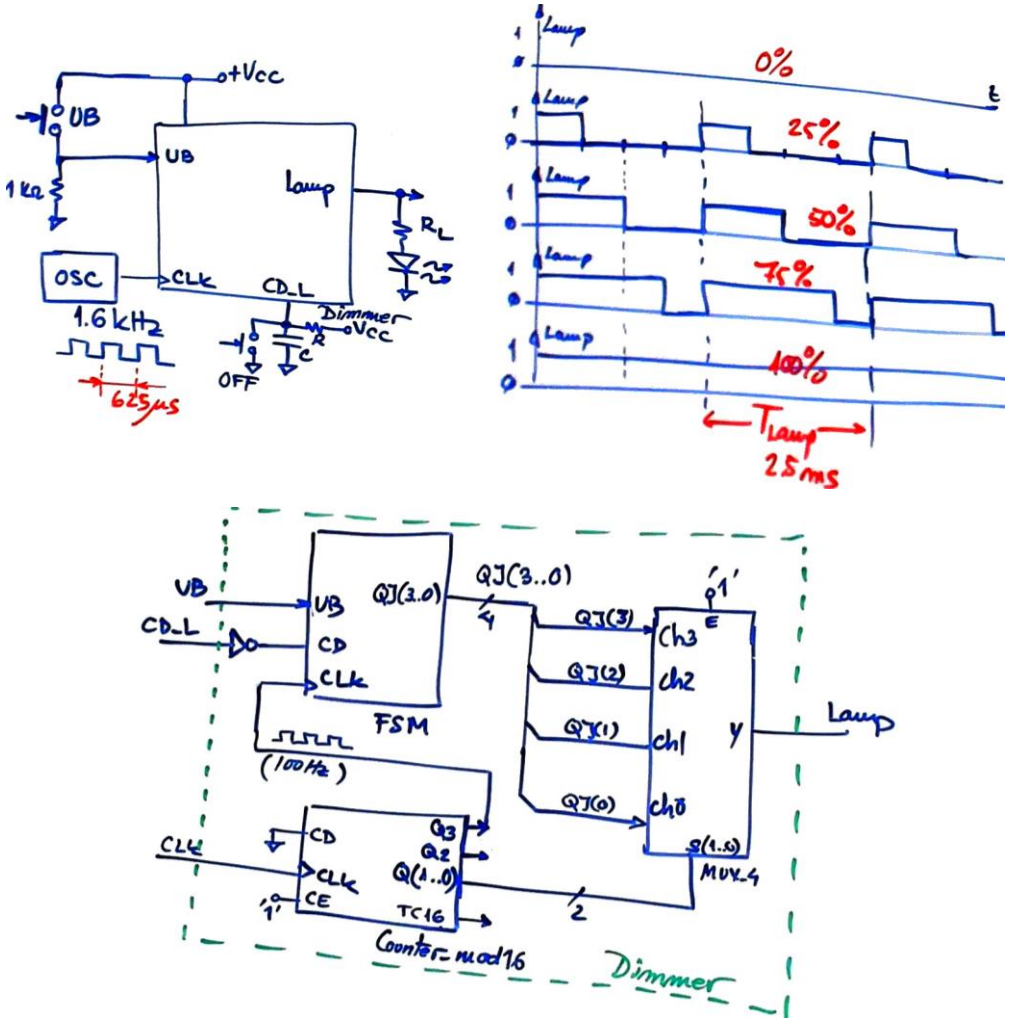
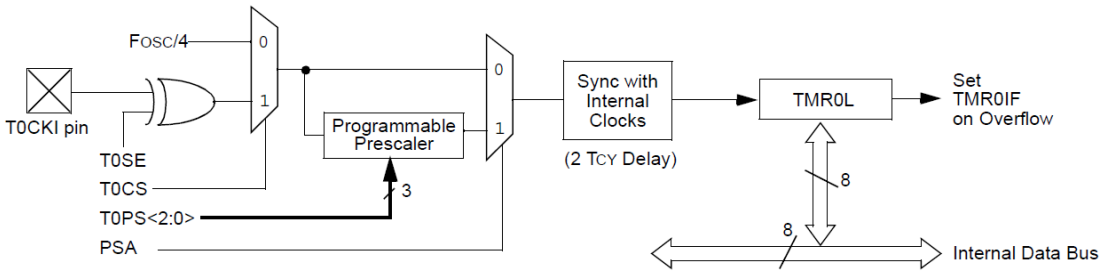


Fig. 4. Symbol, proposed internal schematic and waveforms generated for driving the Lamp.

a) Explain how the proposed circuit in Fig. 4 works. Which Johnson codes are required to generate such PWM waveforms?
b) Propose the FSM state diagram.
c) Draw the FSM structure. Draw the state register schematic using *D_FF*. Code the internal states in binary Gray. How many VHDL files are required to solve this plan C2 project?
d) Propose the CC1 and CC2 truth tables and their flowcharts.
e) Design a *CLK_Generator* circuit to obtain the 1.6 kHz CLK from a 26 MHz crystal oscillator. How many *D_FF* will contain the full *Dimmer* design?

Section 2. Microcontroller design.

f) Draw a schematic to implement the *Dimmer* using a PIC18F4520 microcontroller. Draw the power-on reset (MCLR_L) circuit and explain how it works. Draw the oscillator circuit using a 12 MHz quartz crystal.
g) Adapt the state diagram so that it can be driven by interrupt flags.
h) Draw the hardware-software diagram indicating the required RAM variables and how the FSM and shift operations (datapath) are solved in software. Organise the main program as a FSM in our CSD way.
i) Explain using diagrams and flowcharts how to write the signal Lamp. Explain the ISR().
j) Replace the external CLK signal by the internal TMR0.



REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **TMR0ON**: Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS<2:0>**: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

Fig. 5. TMR0. Hardware block diagram and configuration bits from Microchip datasheet.

Problem 5. Earbuds touch-button controller

Design the button decoder (*Button_controller*) typical in many tactile or touch commercial Bluetooth earbuds. The idea is to assign a binary function code FC(1..0) depending on the clicks detected. Fig. 6 shows typical waveforms for commercial earbuds when the user is touching the button pad. A convenient sampling frequency is f_{CLK} = 8 Hz.

- When no clicks are detected FC = "00". It means that the system is Idle.
- When one short click is detected, it means audio volume up. The function code value assigned is FC = "01".
- When two consecutive short clicks are detected, it means alternatively play/pause. The function codes assigned are FC = "10" / "11".

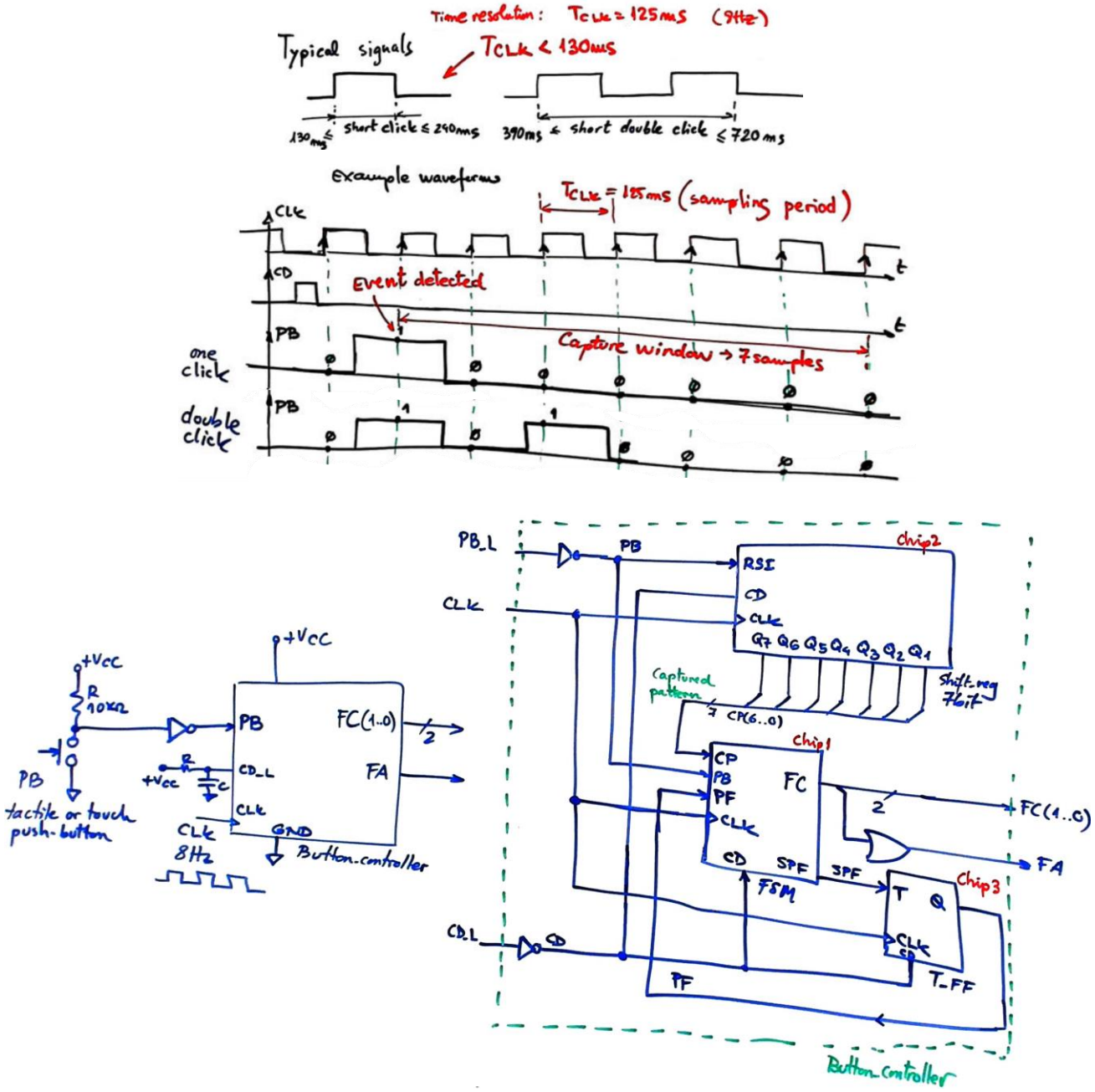


Fig. 6. Click waveforms, symbol and proposed internal schematic for the earbud controller.

Section 1. Hardware design (FPGA, VHDL).

a) Explain the circuit in Fig. 6. Why it may work for capturing click events and obtaining the function codes? Why a captured pattern CP(6..0) of seven push-button samples is ok to identify click events?

b) Propose the FSM state diagram.

c) Draw the FSM structure. Draw the state register schematic using *D_FF*. Code the internal states in one-hot. How many VHDL files are required to solve this plan C2 project?

d) Propose the CC1 and CC2 truth tables and their flowcharts.

e) Design a *CLK_Generator* circuit to obtain the 8 Hz sampling frequency from a 26 MHz crystal oscillator. How many *D_FF* will contain the full *Button_controller* design?

Section 2. Microcontroller design.

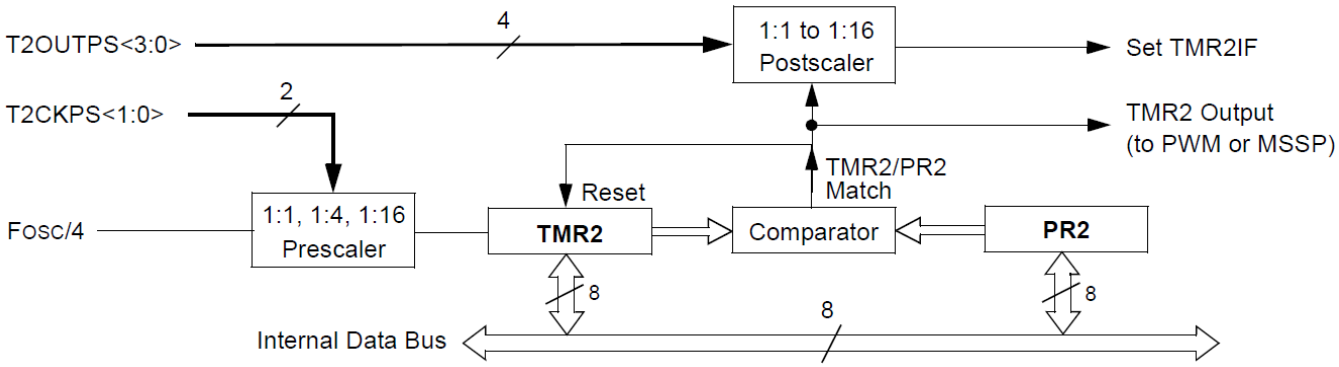
f) Draw a schematic to implement the *Button_controller* using a PIC18F4520 microcontroller. Represent the power-on reset (MCLR_L) circuit and explain how it works. Draw the oscillator circuit using a 12 MHz quartz crystal.

g) Adapt the state diagram so that it can be driven by interrupt flags. Do the interrupts have to be enabled all the time?

h) Draw the hardware-software diagram indicating the required RAM variables and how the FSM and sampling operations for capturing the PB signal is solved in software. Organise the main program as a FSM in our CSD way.

i) Explain using diagrams and flowcharts how to write FC(1..0). Explain the ISR().

j) Replace the external CLK signal by the internal TMR2 so that the PB signal is sampled at the same 8 Hz frequency. Configure the TMR2 parameters and explain the idea why such peripheral can be for this purpose.



REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

Fig. 7. TMR2 block diagram from Microchip datasheet.

Problem 6. Sequencer for a 7-segment display

We want to design a driver to show a LED sequence that indicates a kind of clockwise movement in a single 7-segment display. Fig. 8 represents the symbol, the 7-segment sequence and the proposed schematic diagram of the application. CLK signal to move segments is a 5 Hz rectangular wave.

- When clicking the active-low push-button start/stop (ST_L) the system start running continuously generating the indicated LED sequence.
- When running, if ST_L is clicked again, the system stops **after** completing the sequence.

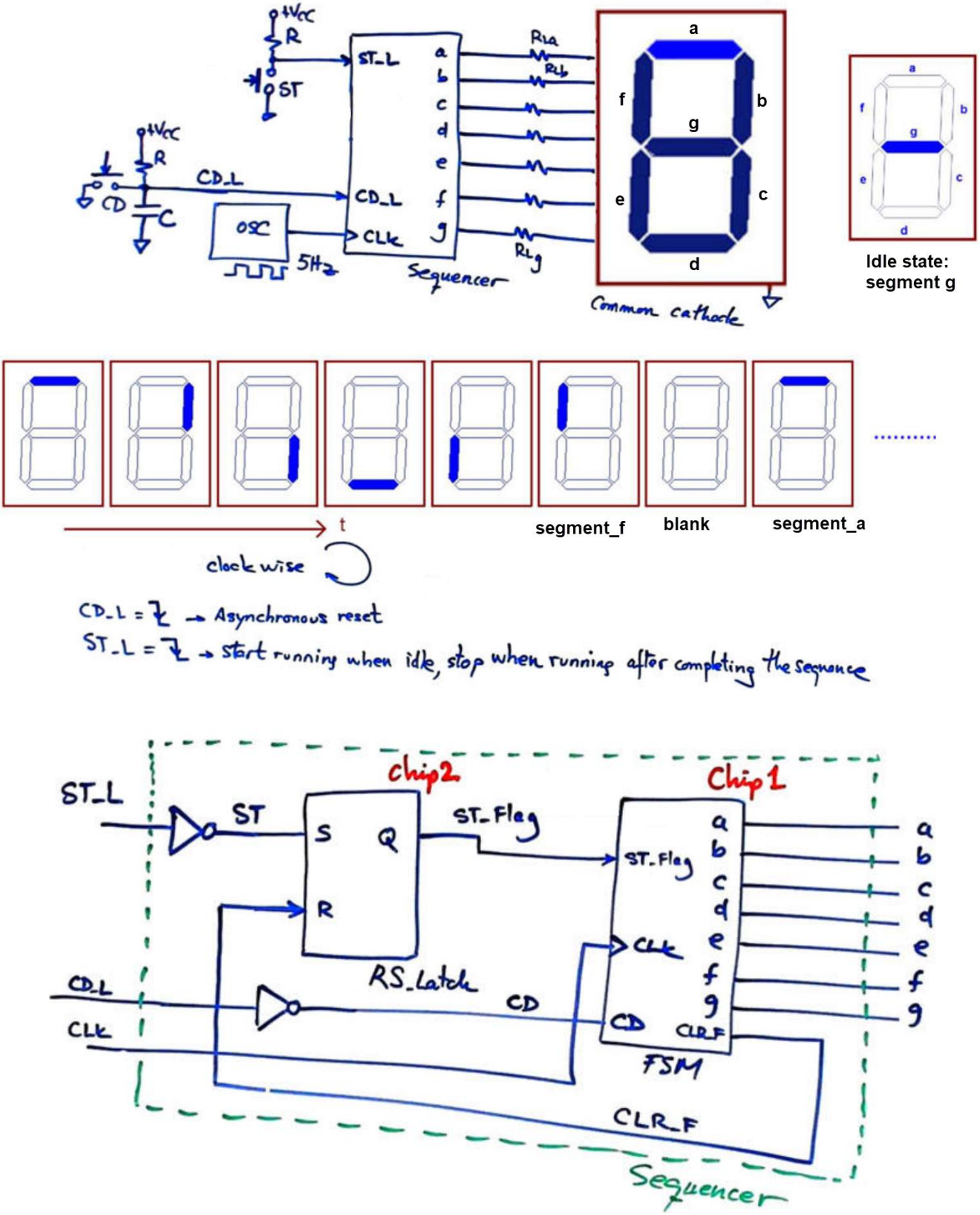


Fig. 8. *Sequencer* symbol, LED pattern generated in the 7-segment display and proposed schematic.

Section 1. Hardware design (FPGA, VHDL).

a) Explain the circuit in Fig. 8. Why it may work for sequencing LED and for controlling the push-button. How the stop flag is generated and memorised?
b) Propose the FSM state diagram.
c) Draw the FSM structure. Draw the state register schematic using *D_FF*. Code the internal states in binary sequential (radix-2). How many VHDL files are required to solve this plan C2 project?
d) Propose the CC1 and CC2 truth tables and their flowcharts.
e) Design a *CLK_Generator* circuit to obtain the 5 Hz sampling frequency from a 26 MHz crystal oscillator. How many *D_FF* will contain the full *Sequencer* design?

Section 2. Microcontroller design.

f) Draw a schematic to implement the *Sequencer* using a PIC18F4520 microcontroller. Draw the power-on reset (MCLR_L) circuit and explain how it works. Draw the oscillator circuit using a 12 MHz quartz crystal.
g) Adapt the state diagram so that it can be driven by interrupt flags. Do the interrupts have to enabled all the time?
h) Draw the hardware-software diagram indicating the required RAM variables and how the FSM is solved in software. Organise the main program as a FSM in our CSD way.
i) Explain using diagrams and flowcharts how to write the segment signals a, b, c, d, e, f and g. Explain the ISR().
j) Replace the external CLK signal by the internal TMR0 (represented above in Fig. 5). Configure the TMR0 parameters and explain the idea why such peripheral can be for this purpose.