UPC. EETAC. Bachelor degree, 2A. Digital Circuits and Systems (CSD). Dr F. J. Robert. Grades will be available online by January 23rd. Questions about the exam at office time.

Exam 2. **January 16th, 2023**

Problem 1. (2 p)

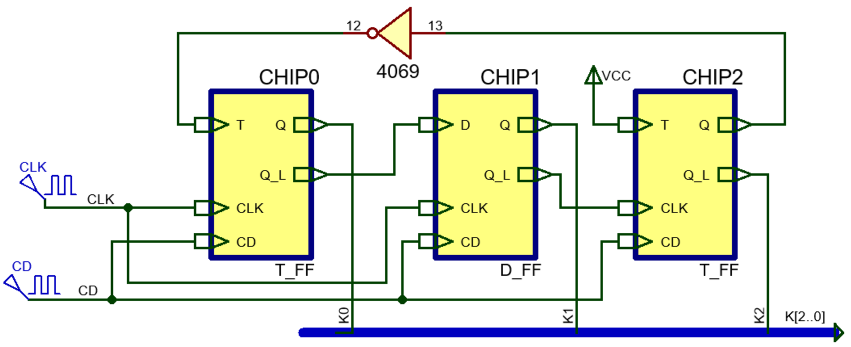


Fig. 1. Circuit based on 1-bit memory cells and combinational circuits.

1. Analyse the circuit in Fig. 1 to determine the output **K(2..0)** completing an example timing diagram.
2. Explain why asynchronous circuits have disadvantages respecting synchronous circuits. Explain how many VHDL files are necessary to develop and simulate the circuit using EDA tools. If the propagation time of a logic gate is t_P = 2.1 ns, discuss and estimate the circuit's maximum speed.

Problem 2. (4.5p)

Let us design an electronic dice as shown in Fig. 2. Seven LED's are used to simulate a real dice face. The *Counter_mod6* is enabled (E = '1') rolling continuously at high speed CLK_SYS and all LEDs are illuminated uniformly. When the player clicks the stop 'ST' button the dice shows a random[1] number from 1 ("001") to 6 ("110") for a timing period of T_P = 10 s because the timer disables the counter. Let us propose the architecture shown in Fig. 3.

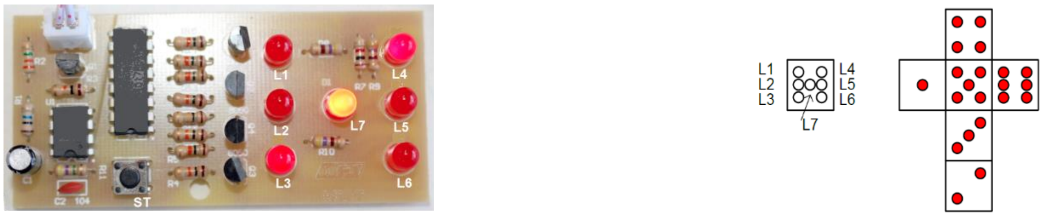


Fig. 2. Example of a commercial electronic dice kit.

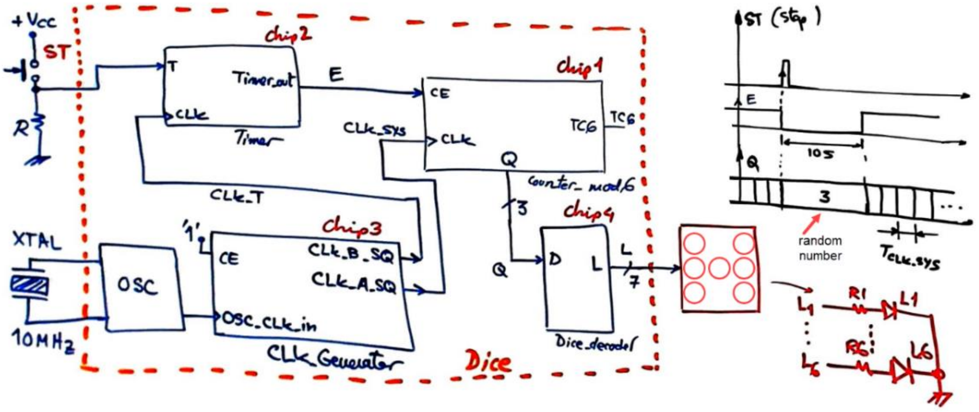
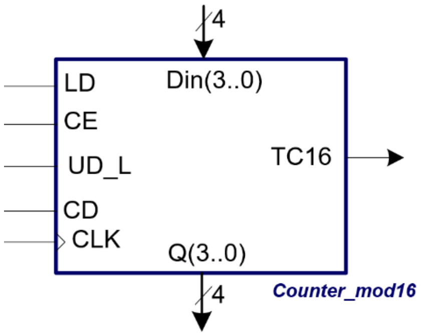


Fig. 3. Dice architecture.

3. Design the Chip3 *CLK_Generator* circuit. An external quartz crystal oscillator of 10 MHz is used as the OSC_CLK_in. Two synchronous squared CLK signals are generated: CLK_A_SQ = 116279 Hz, CLK_B_SQ = 5 Hz.
4. Design the Chip1 *Counter_mod6* version A. Plan X: FSM enumerating states: state diagram, architecture, state register, CC2 and CC1 truth table and flowcharts.
5. Design the Chip1 *Counter_mod6* version B. Plan C2: use counter truncation techniques and our standard component *Counter_mod16* represented in Fig. 4 and other component if necessary.

[1] Stopping any time a counter running at a *weird* high frequency should generate a random number; thus, this is a good experiment for the lab once the device is developed and running. For example, play the dice for 15 minutes and annotate the numbers generated, calculate probability and verify independence (a fair 6-sided die will give you 1/6 (or 16.7%) probability of rolling any of its numbers).



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. 4. *Counter_mod16* symbol and function table.

6. Design the T_P = 10 s timer circuit Chip2 using our plan C2 techniques and components such FSM, counters or shift registers. The circuit is driven by CLK_T = 5 Hz and is triggered when detecting 'ST' rising edge.

7. How many VHDL files and *D_FF* are used in this dice project?

Problem 3. (3.5p)

As an alternative to the hardware circuit in Problem 2, and to be able to compare which design is better, we will solve a similar electronic dice project using a single chip PIC18F4520 microcontroller as shown in Fig. 5. In this application, the user clicks the button ROLL and the dice counts continuously from '1' to '6' at high speed (rolling frequency is 777 Hz). When the button is released, counting stops and a random number is displayed.

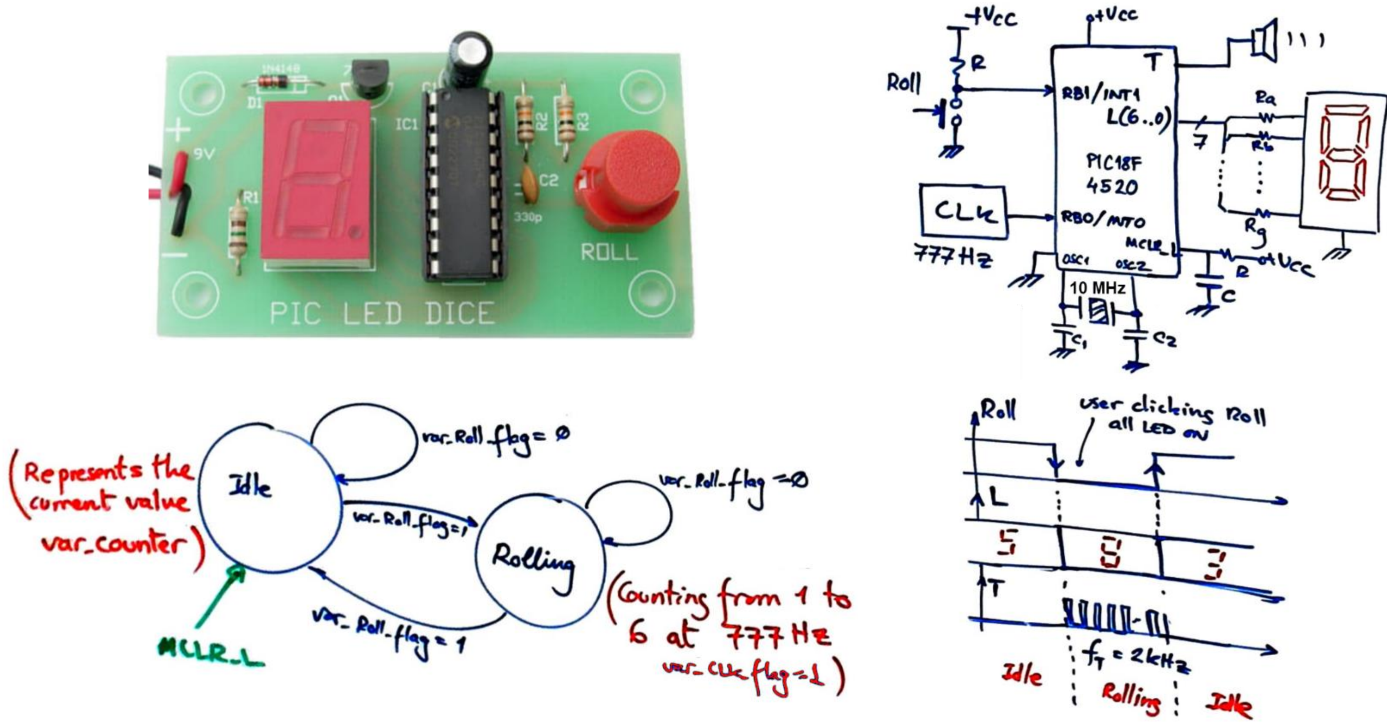


Fig. 5. Electronic dice using a PIC chip and a 7-segment display to represent dice numbers from 1 to 6.

8. Draw and explain the hardware-software diagram. Fig. 5 shows an idea of FSM state diagram to run the application. Explain the difference between variables and flag variables.

9. Explain how to generate and drive the dice outputs in *output_logic()* and *write_outputs().*

10. Explain the external interrupt mechanism for connecting Roll button at RB1/INT1. Explain the general software flowchart for this application and how the *ISR()* is used.

11. Draw the *state_logic()* truth table and its equivalent flowchart ready for translation to C language.

12. If we like to replace the 7-segment digit by an LCD screen, explain how to connect and program the LCD display to show ASCII messages and numbers on the screen.

13. Configure and program TMR2 to replace the external CLK allowing the dice rolling at the same 777 Hz.

14. Generate a tone of 2 kHz audible while the player is clicking the Roll button using the TMR0.