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 January 18th. Questions about the exam at office time.

Second exam. **11th January 2024**

Chapter 2

1. Analyse the circuit in Fig. 1 using a timing diagram and find the binary codes available in the output vector Q(2..0).

Discuss what will be the circuit's CLK maximum frequency of operation if flip-flop components are implemented internally as standard FSM.

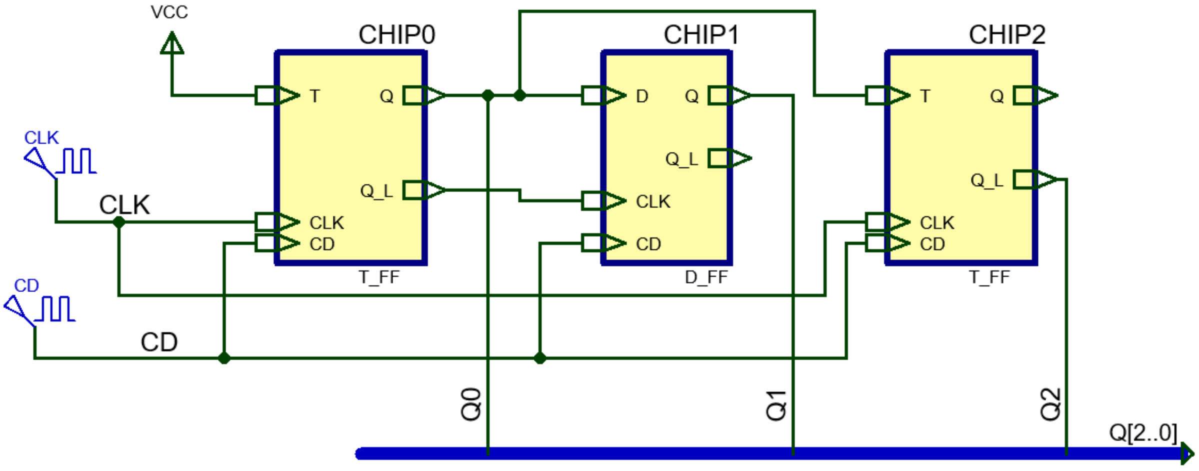


Fig. 1. Circuit based on flip-flops.

2. Design a synchronous 6-bit one-hot code rotator with *CE_L* and reversibility (SR_*L*) using a FSM architecture and our systematic procedure to be used as a bike lamp. The output codes generated (when SR_L = '1') are: 000001, 000010, 000100, 001000, 010000, 100000, 000001 …

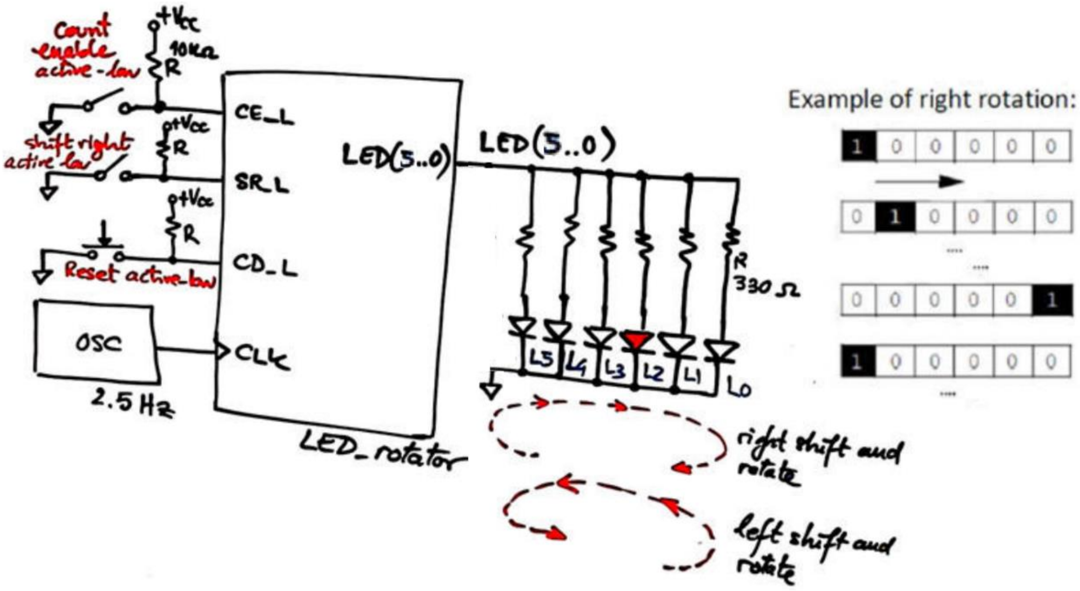


Fig. 2. *LED_rotator* symbol.

Function table, state diagram, timing diagram, FSM structure, state register schematic, CC1 and CC2 truth tables and flow charts. Code the internal states in binary Gray.

If the t_{CO} (CLK to output propagation time) of a *D_FF* is 3.7 ns, and the propagation delay t_P of a generic logic gate is 3.1 ns, estimate the maximum speed at which the rotator can work.

PROBLEM OPTIONS: 3 – 4 – 5 **(choose only one)**

3. A dedicated processor application requires several square signals for clocking purposes. Invent the synchronous *CLK_generator* represented in Fig. 3 applying **plan C2**.

Discuss the number of *D_FF* used. Discuss the number and names of the VHDL files included in the project. Which is the way to obtain more accurate output signals?

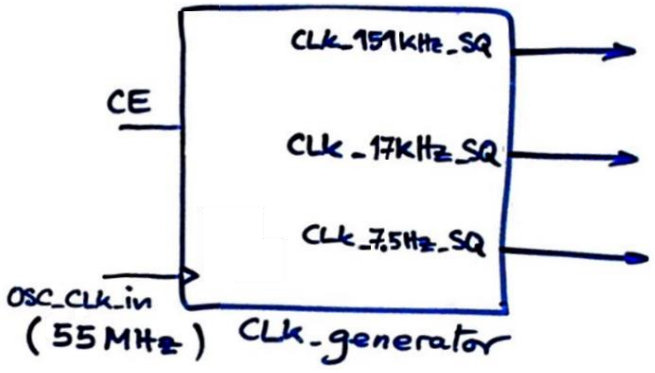
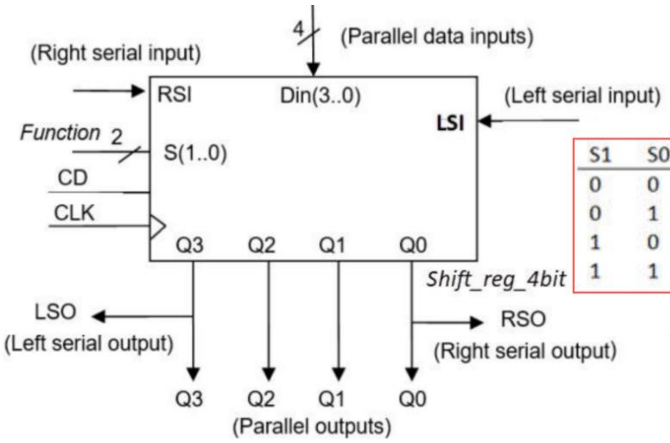


Fig. 3. *CLK_Generator*.

4. Design a synchronous 6-bit binary sequencer with *CE* and reversibility (*UD_L*) using **plan C2** and the basic shift register proposed in Fig. 4 and other components and combinational circuits if necessary. The output codes generated are: 010100, 101000, 010001, 100010, 000101, 001010, 010100 …

Specifications (symbol, timing diagram, state diagram) and plan (schematics and components). Explain and justify your circuits.

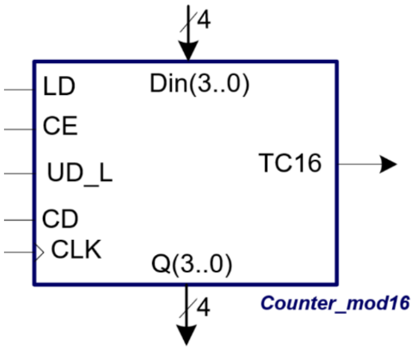


S1	S0	Function
0	0	Do nothing (inhibit)
0	1	Shift right (Q3 ← RSI, Q2← Q3, Q1← Q2, Q0← Q1)
1	0	Shift left (Q0 ← LSI, Q1← Q0, Q2← Q1, Q3← Q2)
1	1	Parallel load data (Q ← Din)

Fig. 4. Symbol of the 4-bit standard binary shift register component (*Shift_reg-4bit*).

5. Invent a synchronous BCD up counter module 75 (*Counter_BCD_mod75*) using **plan C2** and components *Counter_mod16* (Fig. 5) and other combinational circuits if necessary.

Specifications (symbol, timing diagram, state diagram) and plan (schematics and components).



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

Chapter 3

6. We have in mind to build a 1-bit output pattern generator using a PIC18F4520 chip. The circuit symbol is represented in Fig. 6. **D_in** input consist of a binary number of 12 switches to allow the generation of any 12-bit combination of ‘1’ and ‘0’.

Explain the RAM variables required.

Propose an example of port connections. **ST** (start signal) is an external interrupt.

Explain using diagrams and flowcharts how to read and write pins.

Represent the power-on reset (MCLR_L) circuit and explain how does it works.

Represent the crystal oscillator circuit.

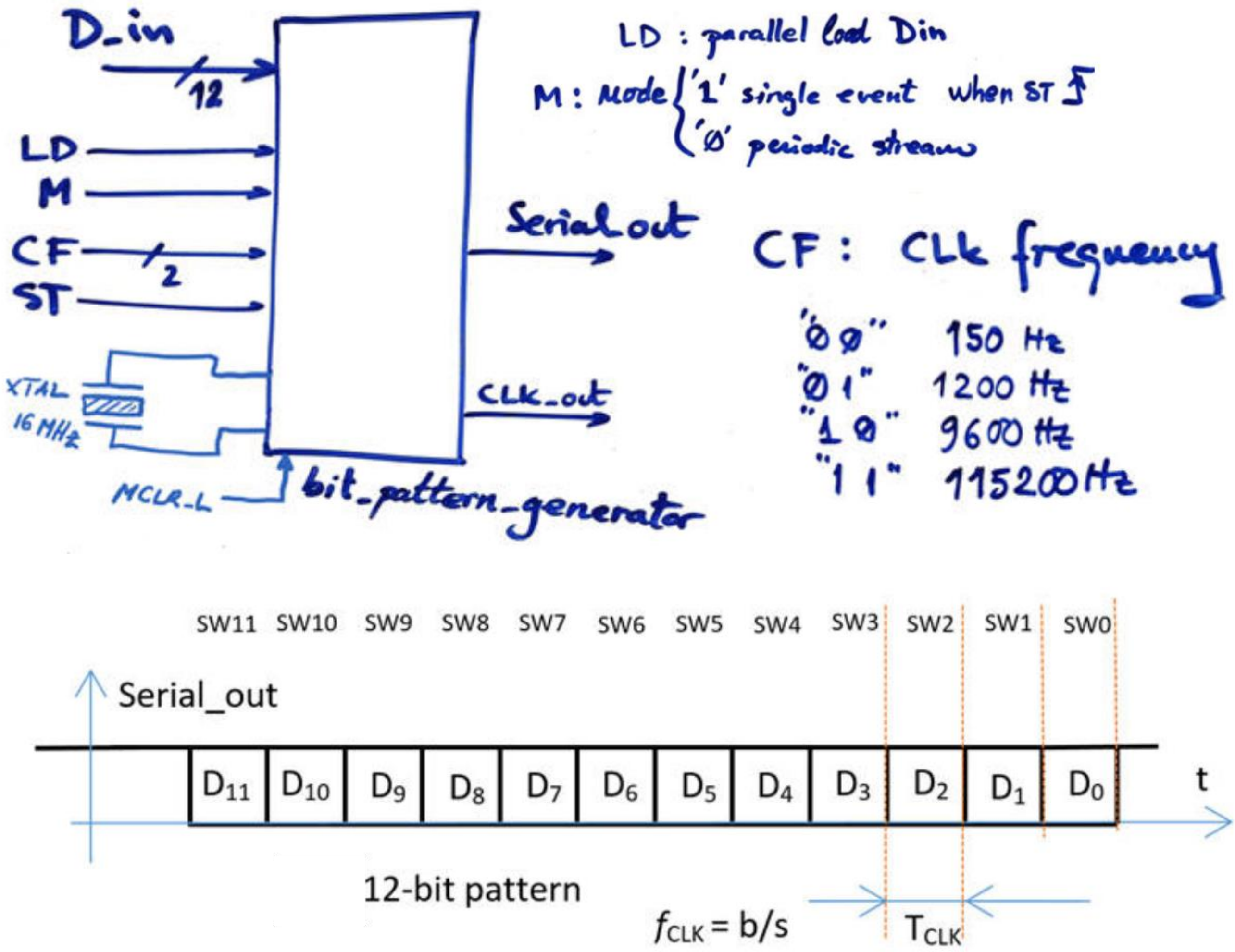


Fig. 6. Proposed symbol for the 1-bit serial pattern generator.

PROBLEM OPTIONS: 7 – 8 – 9 – 10 (choose only one)

7. Adapt the 6-bit one-hot code rotator explained above in **problem 2** to a microcontroller PIC18F4520 using our software FSM architecture and procedures.

Use the problem 2: symbol, timing diagram and function table. Plan: hardware, general software flowchart, FSM adaptation, software-hardware diagram, interrupts, RAM variables, state diagram, how to write, how to read, state and output logic truth tables and flowcharts.

8. Let us design using a microcontroller PIC18F4520 a hobbyist servo motor kit driver for a remote control boat rudder. As shown in the waveforms in Fig. 7, the rudder angle can be continuously adjusted from -90° to +90°. **Angle** input vector is an 8-bit radix-2 number to be read in the main loop. Angle = 0 will drive the minimum pulse with a PW = 900 µs (rudder at -90°); Angle = 128 will drive the pulse PW = 1500 µs (0°, centre position); Angle = 255 will drive the max pulse duration PW = 2100 µs (rudder at +90°). In this design phase #1, as shown in the circuit symbol, an external CLK signal generates the 50 Hz (T = 20 ms) frequency required by the motor.

Explain how to generate the PWM waveforms. Plan: hardware, general software flowchart, FSM adaptation, software-hardware diagram, interrupts, RAM variables, state diagram, how to write, how to read, state and output logic truth tables and flowcharts.

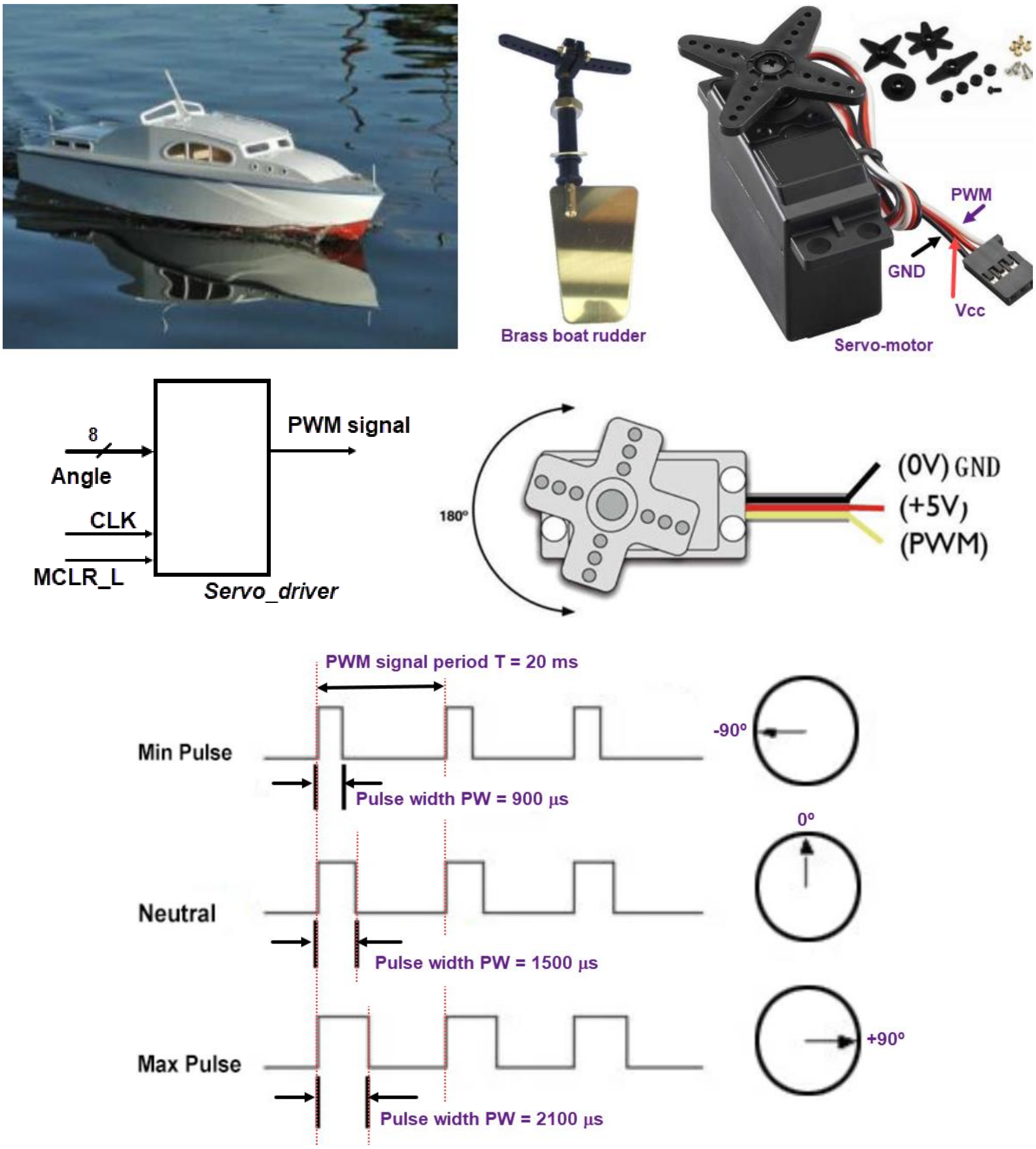
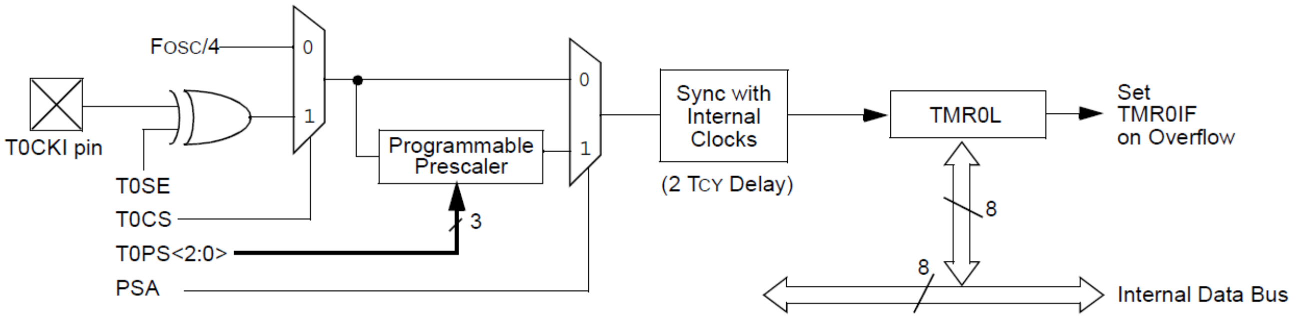
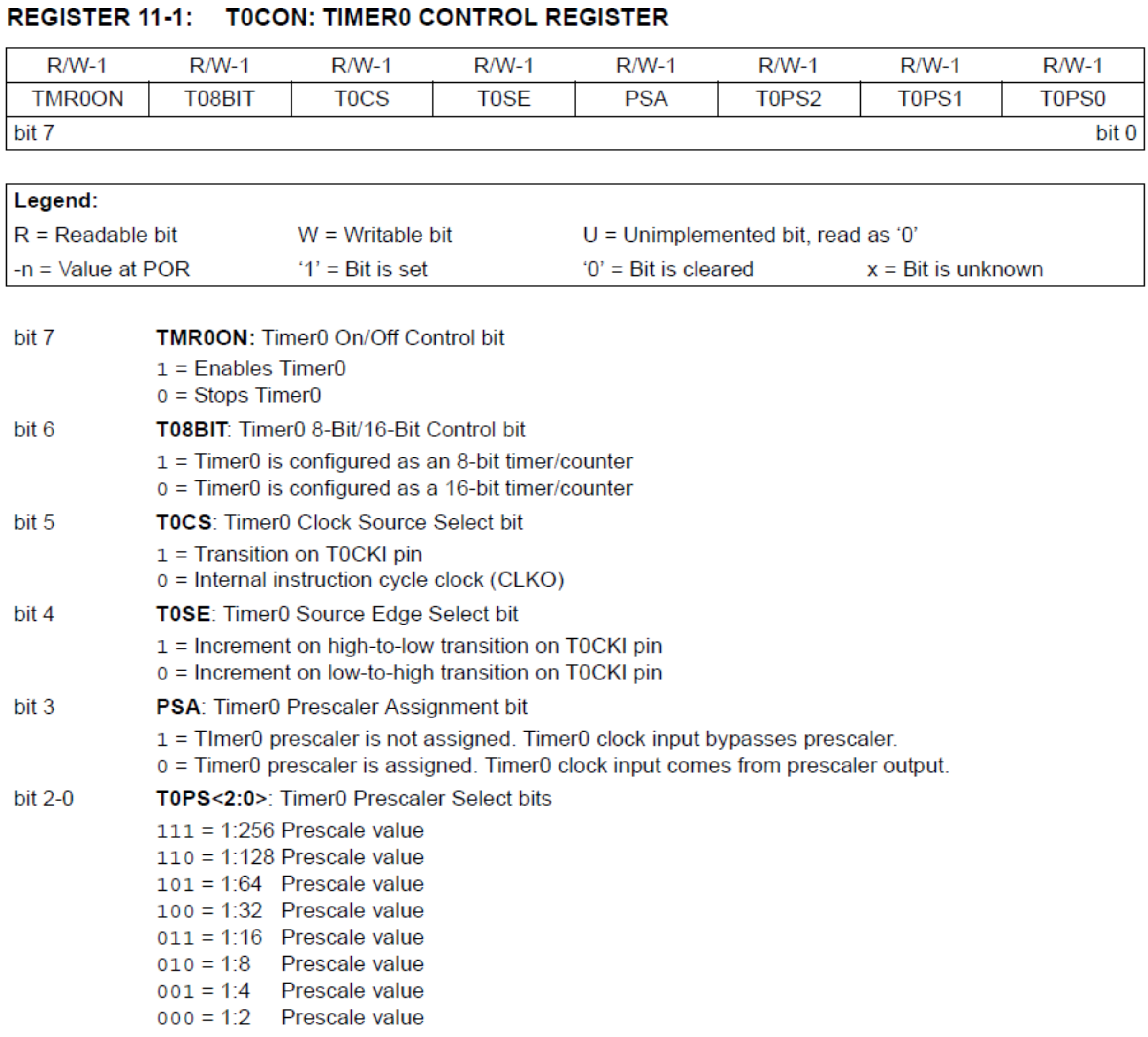


Fig. 7. Pictures and servo motor details.

9. In a design phase #2 of the problem 8, add an LCD to print ASCII messages and the current riddle angular position on the screen. In a design phase #3 of the same problem, let us replace the external CLK by the internal peripheral TMR0 in 8-bit mode of operation as schematised in Fig. 8 obtaining the same servo driver functionality.

Discuss the general ideas on how the circuit works and is planned, as indicated in problem 8, and how to use the new resources in this application.





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

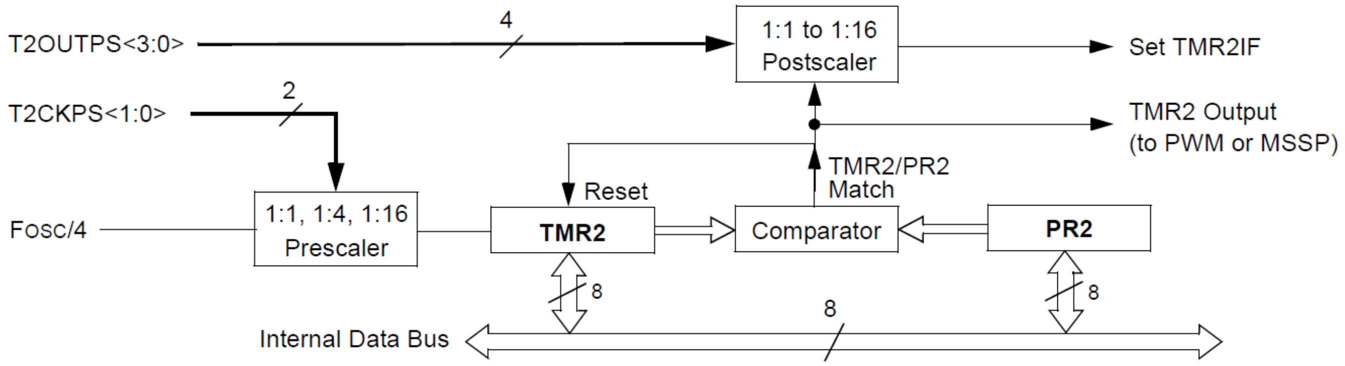
bit 3 **PSA**: Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

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. 8. TMR0 block diagram from Microchip datasheet.

10. In a design phase #4 of the problem 8, let us use the internal TMR2 peripheral represented in Fig. 9 to generate the required PWM signal to obtain the same servo driver functionality.

Discuss the general ideas on how the circuit is planned, as indicated in problem 8, and how to use the TMR2 in this application for timing pulse durations reading the Angle vector.



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. 9. TMR2 block diagram from Microchip datasheet.

Exam structure and grading:

- Solve problems: 1, 2, 6	(6p)
- Choose one: 3, 4, or 5	(2p)
- Choose one: 7, 8, 9, or 10	(2p)