P_Ch3. Designing a microcontroller-based stopwatch

Specifications

Design a stopwatch with the features described in this commercial product represented in Fig. 1 using a PIC18F4520 microcontroller and applying technical know-how from our CSD company. Study and discuss the product features from this website.

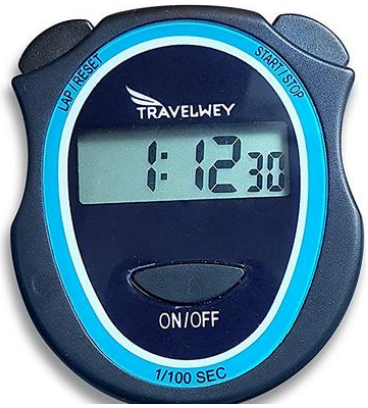

Fig. 1. Picture of the commercial product stopwatch.

The main ideas on symbol, input/outputs, timing diagram and counting capacity are shown in Fig. 2.

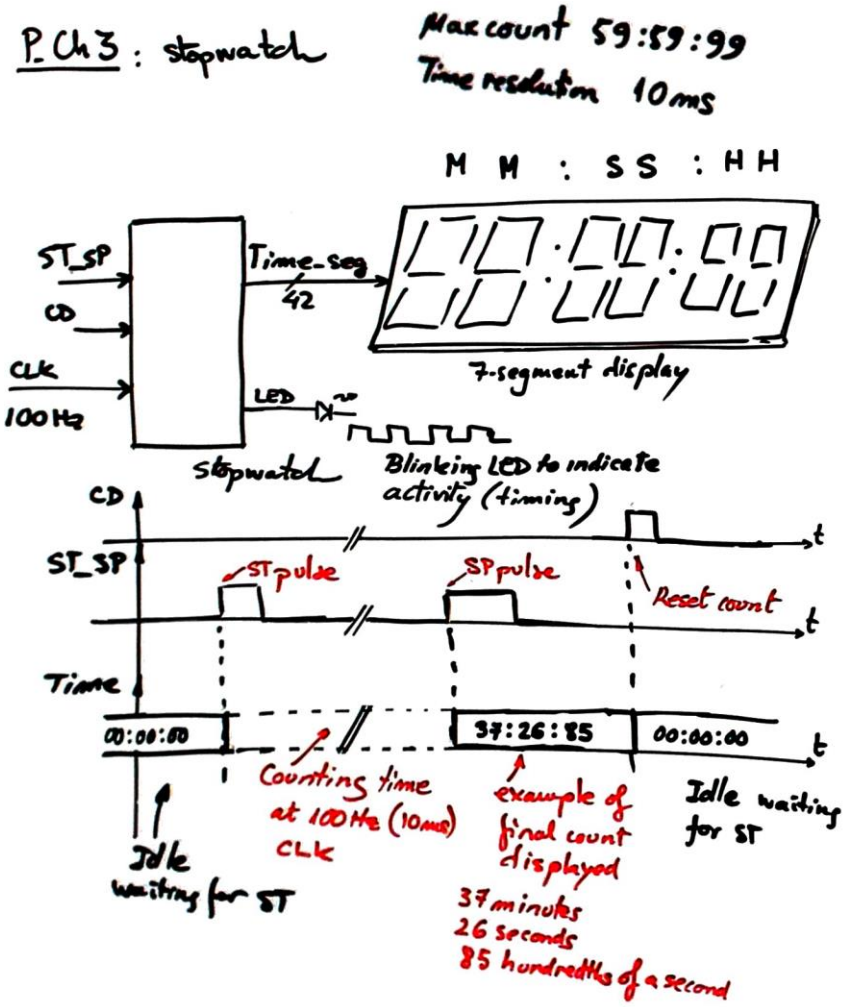

Fig. 2. Sketch of the symbol and example timing diagram.

Planning

To conceive the circuit efficiently, we consider it a standard dedicated processor (datapath + control unit + CLK generator) as if it were proposed as an exercise in P8, even if the final technology for developing the product is a microcontroller.

A. General architecture of a dedicated processor.
B. Plan the datapath using *Counter_mod16* as basic component.
C. Plan the CLK generator from a 16 MHz quartz oscillator.
D. Plan the control unit as a canonical FSM.
E. Plan the complete schematic for this product.

As optional ideas, extra features that may easily be included in the specification of such products, for instance:

(1) If count overflow is detected the display may show and “E” symbol.
(2) When timing, 1 s - 1 kHz sound beep is generated every minute.

Microcontroller circuits here in CSD are adaptations of previous schematics conceived in P6-P7-P8. Now, you have a proposed sequence of intermediate projects to complete the design in smaller steps, so that every mini-project can be considered as another tutorial learning material.

1. Goal: Install a basic “operating system”. Learning to control a given operation by means of a push-button.

Design a circuit to blink a LED at 2 Hz when clicking the push-button to start/stop the operation.

Project location: *L:\CSD\P12\stopwatch\v1*

a. Planning (Propose and comment the general hardware architecture and software flowchart)
b. Hardware. Basic I/O and connections.
c. Software. Memory RAM variables, hardware-software diagram, state diagram, C functions: truth tables, flowcharts, etc.
d. ***Develop*** and ***test*** (watch window, RAM variables, breakpoints, etc.).

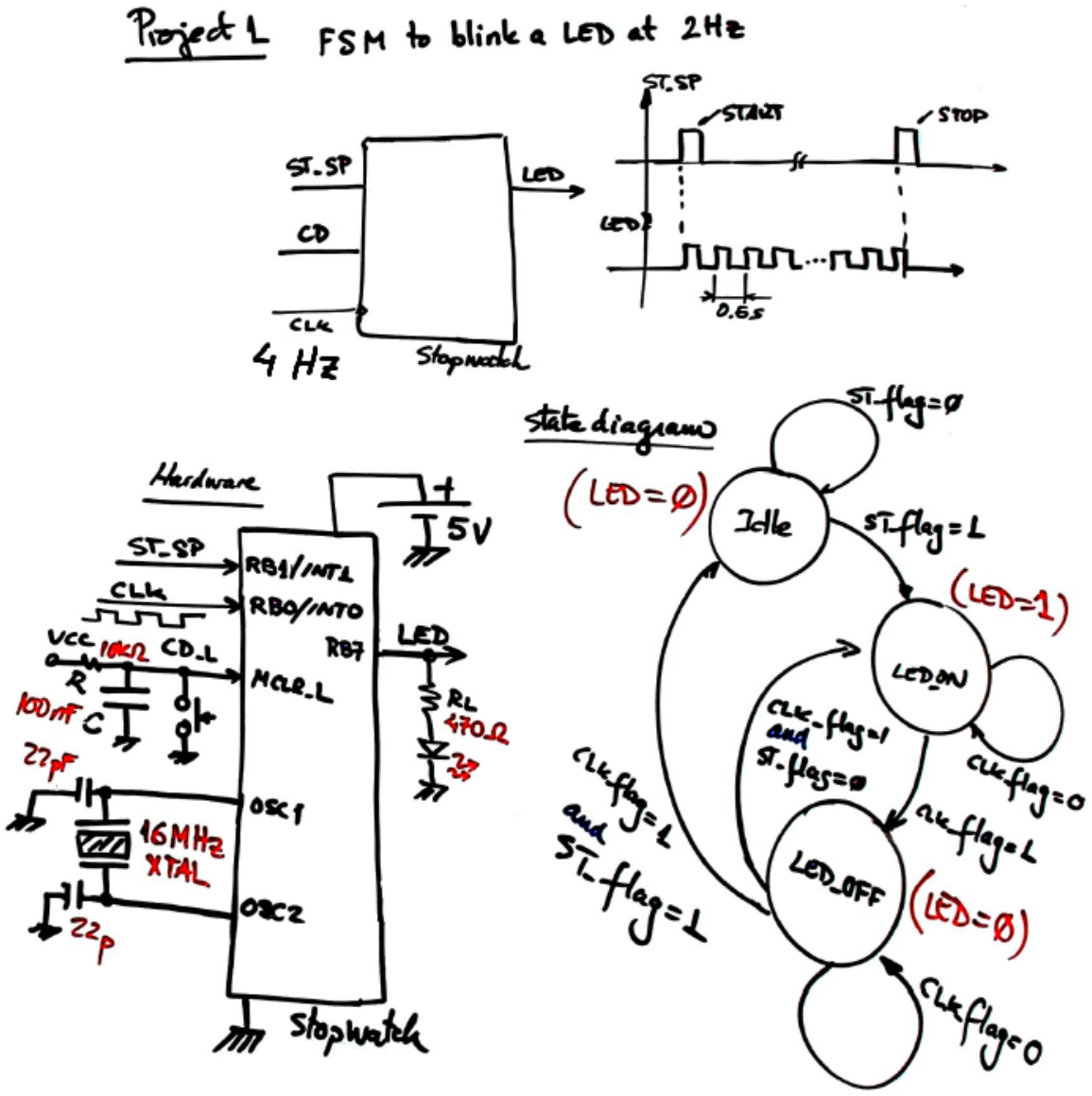

Fig. 3. Project 1 main ideas. Blinking a LED symbol, timing diagram, hardware and suggested state diagram.

Guidelines: Solve the circuit plan as if it were proposed in P6. Solve P10 sessions exercises and study how the 4-bit serial transmitter implemented in P10 works. It contains an ST button and an external CLK, both features are required in this project.

2. Goal: Add a simple datapath to the basic system.

Install a *datapath()* function so that the basic system as well as blinking a LED is capable now of counting in BCD modulo 100. This *Counter_BCD_2digit* is the subsystem HH (hundredths of a second) when running at 100 Hz, it counts UP when clicking the ST_SP button and stops freezing the count when clicking again.

Project location: *L:\CSD\P12\stopwatch\v2*

a. Planning (Propose and comment the general hardware architecture and software flowchart)
b. Hardware. Basic I/O and connections.
c. Software. Memory RAM variables, hardware-software diagram, state diagram, C functions: truth tables, flowcharts, etc.
d. ***Develop*** and ***test*** (watch window, RAM variables, breakpoints, etc.).

Guidelines: Study how in P7 two counters of the same kind were chained to obtain a larger counter. Chain a couple of *counter_BCD_1digit* in P10 to implement *counter_BCD_2digit*.

3. Goal: Interface an LCD display to the basic system.

Add an LCD display to represent BCD numbers in modulo 100 obtained in the previous project. The *var_LCD_flag* will allow writing to the LCD only when there is new information or numbers to show.

Project location: *L:\CSD\P12\stopwatch\v3*

a. Planning (Propose and comment the general hardware architecture and software flowchart)
b. Hardware. Basic I/O and connections.
c. Software. Memory RAM variables, hardware-software diagram, state diagram, C functions: truth tables, flowcharts, etc.
d. ***Develop*** and ***test*** (watch window, RAM variables, breakpoints, etc.).

Guidelines: Study how the LCD is interfaced in P11. Be aware that first, you have to learn how to print a static message such “hello world”, and later how to print a count that is dynamic. This idea is developed in the Adder_BCD_1digit_LCD in P9-P11 learning materials.

4. Goal: Implement the complete system

Complete the datapath to full time count capacity MM:SS:HH (minutes: seconds: hundredths of a second).

Project location: *L:\CSD\P12\stopwatch\v4*

a. Planning (Propose and comment the general hardware architecture and software flowchart)
b. Hardware. Basic I/O and connections.
c. Software. Memory RAM variables, hardware-software diagram, state diagram, C functions: truth tables, flowcharts, etc.
d. ***Develop*** and ***test*** (watch window, RAM variables, breakpoints, etc.).

Marking grid:

Project 1	Project 2	Project 3	Project 4	Video Presentation
2p	2p	2p	2p	2p

5. (optional 1) Goal: Replace the external 100 Hz CLK by the internal peripheral Timer0 or Timer2.

This means that now *var_CLK_flag* is generated when Timer0 (or Timer2) overflows at terminal count.

Project location: *L:\CSD\P12\stopwatch\v5 (2 extra points)*

a. Planning (Propose and comment the general hardware architecture and software flowchart)
b. Hardware. Timer 0 circuit and configuration bits.
c. Software. Hardware-software diagram. State diagram.
d. ***Develop*** and ***test*** (watch window, RAM variables, breakpoints, etc.).

6. (optional 2) Goal: Implement the design phase 2 replacing the reset button (MCLR_L in phase 1) by a LAP/RESET.

Project location: *L:\CSD\P12\stopwatch\v6 (2 extra points)*

a. Planning (Propose and comment the general hardware architecture and software flowchart)
b. Hardware. Basic I/O and connections.
c. Software. Memory RAM variables, hardware-software diagram, state diagram, C functions: truth tables, flowcharts, etc.
d. ***Develop*** and ***test*** (watch window, RAM variables, breakpoints, etc.).