6

Chronometer Design

6.1 Introduction

In the following experiment, you are going to design a chronometer with centisecond precision (not milliseconds). The chronometer is going to start/pause via a push button on Port 2 and represents its value through 4-digit 7-segment display.

6.2 Background Information

MSP430 family micro-controllers contain two 16-bit timers which could be utilized independently. Basic building blocks of the timer are represented in Figure 6.1. In this section, basic steps of the configuration and operation of the timer are briefly explained. For further information, you should read the "**Timer-A**" chapter in **MSP430 User Guide**.

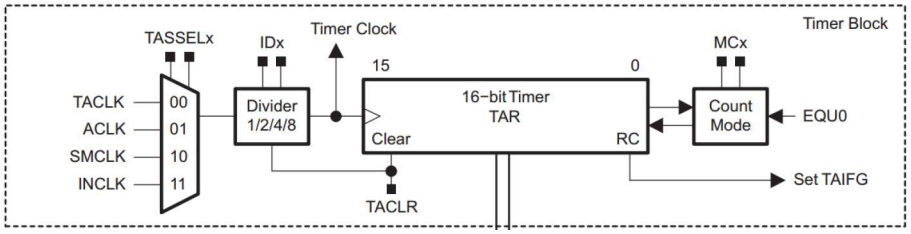

Fig. 6.1. Timer Circuit

As seen in Figure 6.1, timer could use four different signals as counting input. In this experiment, you should use SMCLK signal as counting input. SMCLK is a square wave with $1048576Hz$ frequency.

Timer contains two operation modes (capture and compare) and four different timer (counting) modes (Stop, Up, Continuous, UpAndDown). You could configure the timer via its control registers. List of registers you should use in the experiment is given below. You should decide which values to store on these registers in order to create timer interrupts with 10 millisecond periods (i.e., 1 centisecond). More detailed information about these registers could be found in **MSP430 User Guide** at pages *369-372*.

Register	Rep. in CCS	Description
TimerA Control	TA0CTL	Configuration of timer given in Fig. 6.1
TimerA Compare Capture	TA0CCR0	Holds the data to compare or stores the captured data
TimerA Comp. Cap. Control	TA0CCTL0	Configuration of Comp/Cap mechanism

6.3 Experiment

6.3.1 Experiment - Part 1

In the first part, you should write an infinite loop as your main program code in order to lit different digits of 7-segment display panel simultaneously. You should be able to accomplish following output given in Figure 6.2 as the result of the first part.

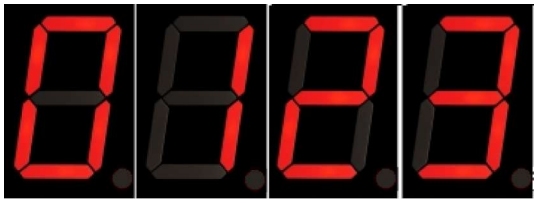

Fig. 6.2. Sample Output

6.3.2 Experiment - Part 2

There are three additional code blocks that you should include your code in order to build the chronometer. The list of these code blocks are given below.

- Timer Interrupt Subroutine
- Interrupt Subroutine
- Convert Subroutine

Additionally, you may include following code section in order to store *second* and *centisecond* values inside RAM.

```
1       .data
2 seconds          .byte 00h
3 centiseconds     .byte 00h
```

Timer Interrupt Subroutine

This subroutine should increase[1] *centiseconds* value whenever there is an interrupt from the timer. When *centiseconds* value reaches 100, it should clear the value of *centiseconds* and increase *seconds* value by one.

Lastly, a sample code block is given below in order to show how to handle interrupts from the timer with a subroutine called **TISR**.

```
1     .sect   ".int09"
2     .short  TISR
```

[1] Be sure that you have configured the timer correctly to generate interrupts with 10msec-period.

Interrupt Subroutine

Similar to previous experiment (Exp. 5), this subroutine should handle interrupts created by the 6th bit of Port 2 and it should start or pause the counting of chronometer. In order to do so, it should enable or disable interrupt mechanism of the timer by manipulating the bits within the control registers of the timer.

BCD Conversion Subroutine

Lastly, you should divide *centiseconds*, *seconds* values into BCD-digits in order to print them through 7-segment displays. This function should take its input value via R15 and return two output values through R14 and R15. Sample input and output combination are given below.

input	*output*
R15:0Ah	R14:01h, R15:00h
R15:12h	R14:01h, R15:02h
R15:36d	R14:03d, R15:06d

After BCD conversion, you should utilize 7-segment display array to display the value of chronometer.

```
1     ;Integer to 7-segment array
2 array    .byte 00111111b,00000110b,01011011b,01001111b
      ,01100110b,01101101b,01111101b,00000111b,01111111b
      ,01101111b
3 lastElement
```

6.4 Report

Prepare your report by using the guidelines and the report template which are posted on Ninova e-Learning System. During the experiment, please do not forget to take notes about the critical points of the implementations in order to write a proper report for the experiment. Additionally, if there were any complications which affect your performance during the experiment, please also indicate these difficulties in your report.