(Individual) Take Home Exam-1

Revision 1.0

Due: Mar 4, 2020, 23:55 hrs

Submission: ***via ODTUCLASS***

The purpose of this assignment is to familiarize you with **basic input/output operations on MPLAB X IDE simulation** environment.

Any clarifications and revisions for the assignment will be posted to the discussion group on ODTUCLASS.

This is not a group exam; you must solve and submit it **individually.**

Your mission is to implement the below scenario using the MPLAB X IDE simulation environment.

Scenario

In this study, we will implement a simple calculator. This calculator will be able to realize two mathematical operations: Addition and subtraction. The selection of the operation to be performed (addition or subtraction) depends on the time you press and release an assigned button. Furthermore, the transitions between the PORTs[B-D] and the value assignments on them will be done by the button control. The LEDs assigned on the selected PORT will be TURNED ON, respectively in order to show the current value on the PORT by the number of times the buttons are pressed and released. Finally, the result of the selected operation will be shown on the LEDs connected to PORTD.

Specifications

Five ports will be used in this assignment: **PORTA**, **PORTB**, **PORTC, PORTD, and PORTE**. LEDs are connected to the following pins: **RB[0-3], RC [0-3],** and **RD [0-7].** The push buttons are **RA4, RE3, and RE4, which** must be configured as **digital inputs,** please look at the relevant section in the datasheet.

At the very beginning of the program, you must turn on **the RB[0-3], RC[0-3], RD[0-7] LEDs of all the ports** (16 pins in total) for 1 second. (please look at the Coding Rules and Hints sections for time delay implementation.)

After the period of 1-second, the **RA4** button assigns the operation to be performed. When this button is pressed and released once, addition operation, when pressed twice, subtraction operation will be selected. You have to press and release the RA4 button to start the process. If you press and release the button three times, the addition operation will be selected again. Each press and release of the RA4 button, the active operation will change (addition-subtraction). After the selection of the operation, the **RE3** button will be used for the **port selection**. In order to select PORTB, RE3 button should be pressed and released once. When the RE3 button is pressed and released again, PORTC is selected. When the RE3 button is pressed and released again, PORTD is selected. Each press and release of the RE3 button will move the active PORTs from PORT C to PORT D. When you select a PORT, you cannot change your operation, in other words, the RA4 button will not be pushed and released during the port selection.

When PORTB or PORTC, respectively, is selected, the **RE4** button will **be used to enter the value on these PORTs**. Each time the RE4 button is pressed and released, the LEDs on these PORTs will turn on respectively from PORT[B-C]0 to PORT[B-C]3. When the RE4 button is pressed more than four times, the LEDs will reset and can be turned on respectively again from PORT[B-C]0 to PORT[B-C]3.

After determining the values of PORTB and PORTC, the PORTD is selected with the RE3 button, the result of the operation will be displayed as follows. For the addition operation, the LEDs should be turned ON as much as the total of the LEDs turned on in PORTB and PORTC. If the subtraction operation is selected, the LEDs should be turned ON as much as the absolute value of the difference of the LEDs turned on in PORTB and PORTC.

For one second, the result (addition or subtraction) is displayed in PORTD, all LEDs will be reset, and the operation selection can be realized by the RA4 button again.

- An example of an addition operation of 2 and 3 is provided step-by-step.
- Addition operation is selected by press-and-release the **RA4 once**,
- **PORTB** is selected by press-and-release the **RE3 once,** and **the value of 2** is assigned on **PORTB** by press-and-release of the **RE4 once**.

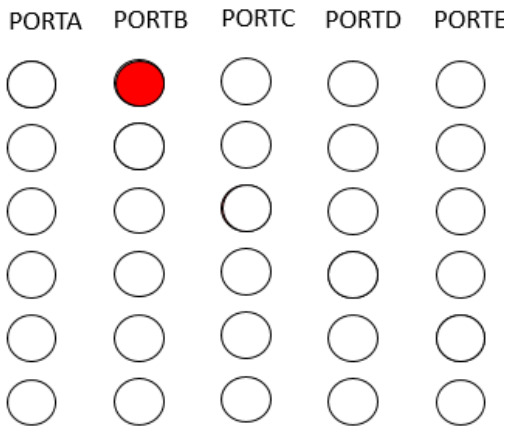

Figure 1: LED sample after press and released of the RE4 once.

- And then press and release of the RE4 again.

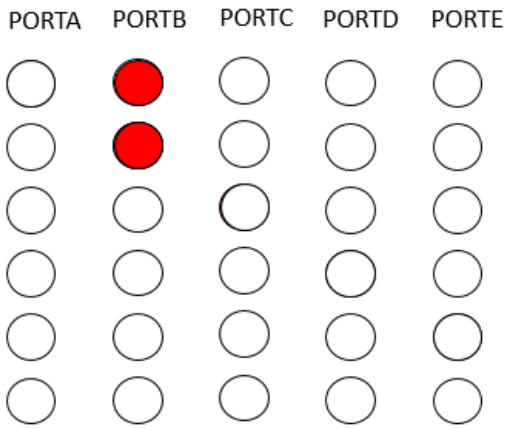

Figure 2: LED sample after press and release of RE4 again.

- **PORTC** is selected by press-and-release of the **RE3 once again,** and **the value of 3** is

assigned on **PORTC by press-and-release of the RE4 three times**.

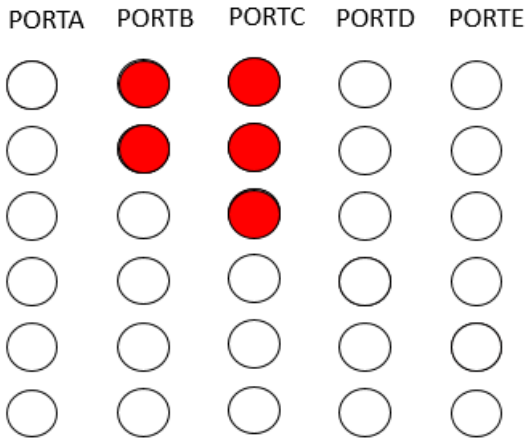

Figure 3: LED sample after select PORTC and press and release of the RE4 button three times.

- **Finally**, **PORTD** is selected by press-and-release of the **RE3 for the last time,** and the **addition** of the values assigned (already turned) on PORTB and PORTC are shown on **RD**[0-7] LEDs.

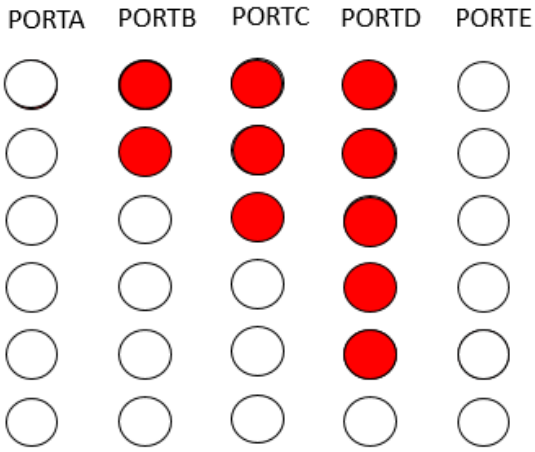

Figure 4: LED sample after select PORTD.

An example of a subtraction operation of 0 and 3 is provided step-by-step. **Subtraction** operation is selected by press-and-release **RA4 twice**, **PORTC** is selected by press-and-release the **RE3 twice,** and **the value of 3** is **assigned** on **PORTC** by press-and-release of the **RE4 three times**. **Finally**, **PORTD** is selected by press-and-release of the **RE3 for the last time,** and the **subtraction result** is obtained, and presented by using the first 3 LEDs by turning on PORTD. You can see the sample of subtraction example in Figure 5.

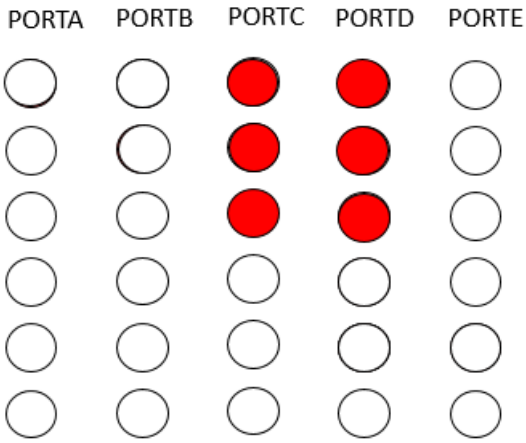

Figure 5: Subtraction example.

The naming of pins is <R>+"PORT NAME"+"BIT ORDER". R represents PORT. Here, ports names are A, B, C, and D. And the range of the BIT ORDER is between 0 and 7.

Implementation

You are expected to simulate the above scenario in the **MPLAB X IDE simulator**. You can track the status of the LEDs and buttons, by using **RB0, RB1, RB2, RB3, RC0, RC1, RC2, RC3, RD0, RD1, RD2, RD3, RD4, RD5, RD6, RD7, RA4, RE3 and RE4** pins of the simulator. The output of the pins can be seen by using the **I/O pins menu** which is available under the **Window → Simulator** menu. A screen shot from the I/O pins menu of the simulator is given in *Figure 1*. In this figure, you can find states of some pins that are representing the LEDs and buttons. You can also simulate push buttons by using the **Stimulus** menu. This feature is available under the **Window → Simulator** menu. You can control voltage levels of the buttons as High or Low by clicking the **RA4**, and **RE3-4** pin button which is can be seen in *Figure 2*. You can add the pin corresponded with push button by using **the I/O pins menu** to follow the changes on this pin. **Note that** in *Figure 1* and *Figure 2*, some pins are not relevant to this assignment and only given as an example.

Coding Rules:

- You will code your program using PIC assembly language.
- Your program should be written for **PIC18F8722** working at **40 MHz.**
- When the push button is pressed, then RA4/RE[3-4] pin goes high, and when we release the push button, the RA4/RE[3-4] pin goes low.
- When you are writing your codes, please look at the relevant lecture notes and recitation documents. In particular, you are expected to apply the **round robin approach** to your code, so your program will execute in an infinite loop, and there will be **your own tasks for your states** in this loop at the main scope.
- Be precise for constructing time delays (intervals). You should use loops for implementing the time delays. You may need to count the number of instructions for those sections. You must not use Timers for this assignment. +/- 5 ms deviation is acceptable. You can measure the time by using the stopwatch and break points, so please look at the Hint section.
- Each button action should be active after **the button pressed and released (not only pressed)**.

Hand in Instructions

- You should submit your code as a single file named as `the_1_##.asm` through ODTUCLASS where ## represents your seven-digit student number.

Hints

- You can use “Stopwatch” tool of the simulator in MPLAB X IDE to measure the time spent by a code segment. You can reach this tool from `“Windows -> Debugging -> Stopwatch”` menu. But, before starting simulator you must configure your clock speed to 10 MHz from `“Project properties -> Simulator -> Instruction Frequency”` (Since 40 MHz main clock frequency is divided by 4 inside microcontroller and one instruction cycle is equal to 4 cycle of 40 MHz clock signal, you are setting “Instruction Frequency” to 10 MHz). You can reach the project properties by right clicking to the project name in the left side “Projects” panel and selecting properties. By **putting breakpoints** on the lines between which you want to measure the amount of time spent by that code segment and running the code between these two breakpoints, you can see the time spent in stopwatch window.
- You will use **RA4 pin** as a **digital input** by configuring a **special register**, please check the data sheet for this configuration.
- You can also see the states of your ports, variables and pins by using the logical analyzer, SFR and variables menu under the window menu.

Grading

Your codes will be evaluated on the MPLAB X simulation environment. However, if you obey the specified rules, the program can directly be executed on the boards without any problem. You can test your code also in your MCDEV Kit. Please, look at the related switch configuration which is provide you in the exam files. This an individual exam but the group members can execute their codes with the dedicated MCDEV Kit to this group.

Resources

- PIC 18F8722 Datasheet
- ODTUCLASS course web page
- The **MPLAB X IDE** will be used. You can download by using the following link:

http://www.microchip.com/pagehandler/en-us/family/mplabx/home.html

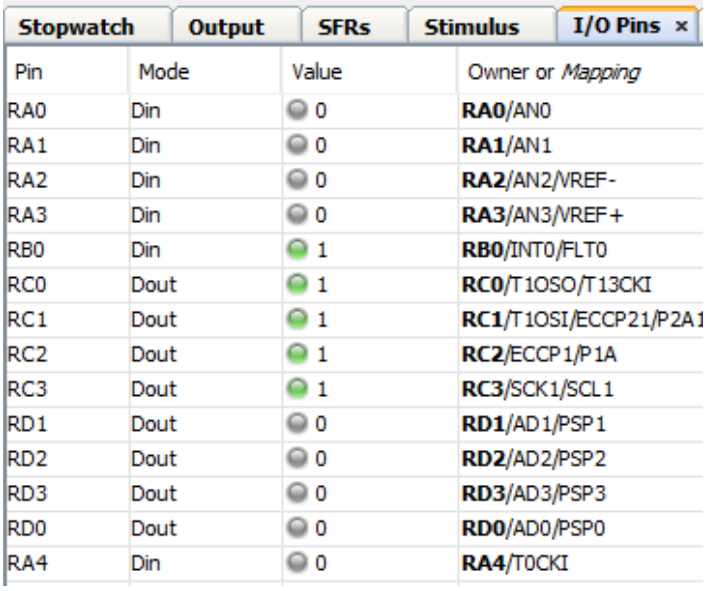

Figure 6: State of various LED and button pins shown in I/O pins window.

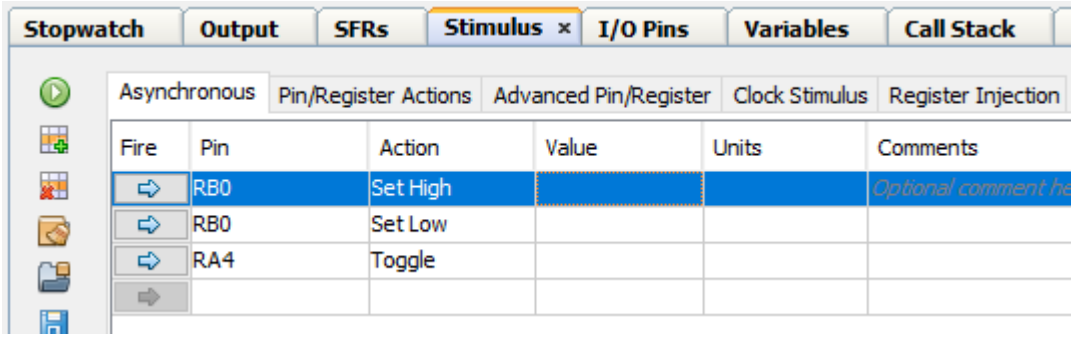

Figure 7: Example button (RA4 & RB0) control using the stimulus window.