CASE WESTERN RESERVE UNIVERSITY
Case School of Engineering
Department of Electrical, Computer and Systems Engineering
ECSE 281. Logic Design and Computer Organization (4)

Assignment #11 **Due: May 4, 2023**

This is a required homework. It also has some extra credits. Please upload your homework solutions to Canvas. You need to upload your .asm file and a single pdf file containing your screenshots.

50 pts + 50 pts Extra Credit

In order to get the extra credit, your code should be working properly, and you need to upload both your asm code and the pdf containing the screenshots.

In this assignment, we will use MPLabX to develop and simulate an assembly language program for the PIC microcontroller that plays the guessing game, as described in Section 12.7 of the Wakerly textbook, 5th ed. The assembly language program is to be tested using *MPLab* to apply the sequence of inputs given below. The state diagram for this homework is modified such that there is a WIN output connected to a green light showing that the guess was right (WIN is asserted in the SOK state and the green light turns on if the guess was correct).

This state machine has 4 inputs and 6 outputs. Students should use the PIC16F84A. The required pin assignments are given in the following table.

Required Pin Assignments

Inputs		Right Outputs	
Function	PIC	Function	PIC
G1	RA0	L1	RB0
G2	RA1	L2	RB1
G3	RA2	L3	RB2
G4	RA3	L4	RB3
		ERR	RB4
		WIN	RB5

A high output is used to turn on a light while a low output signifies that the light is off.

Programs should initialize to state S1 and include a delay of ~1.0 second from one state to the next in the rotating light pattern. This delay could be created in different ways, for example, using a simple program loop, or using a timer with or without interrupts. It need not be precise, but should be accurate to ±0.05 seconds. You may choose your processor clock frequency to adjust the state delay.

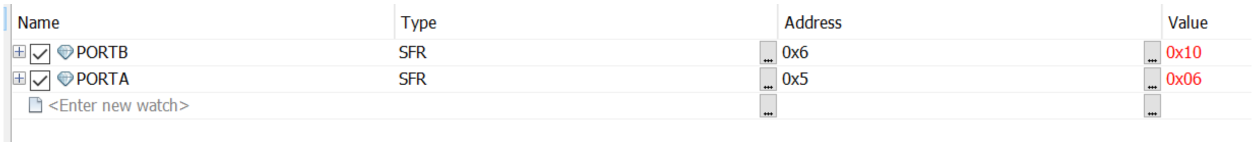
The program should be tested using a stimulus workbook ("guessing_game_stimuli_Spring2023.sbs") that has been posted on Canvas with this assignment. This external stimulus is applied using

Stimulus → open a stimulus workbook (from icons to the left of the Stimulus window)

being sure to "Apply" the stimulus before you start debugging your code.

Your solution to this homework should include your documented assembly language code and watch window to demonstrate that your game responds properly to stimulus. This example shows a watch window following the first wrong guess. Your solution should show this watch window following each state change. This can be done most easily by setting breakpoints immediately following each state change in your code, then running the processor, which will automatically stop at each state change. You need to use a processor frequency of **100kHz** (instruction frequency Fcyc = 25 kHz).

Sample Display window following the first wrong guess



Name	Type	Address	Value
⊞ ☑ PORTB	SFR	0x6	0x10
⊞ ☑ PORTA	SFR	0x5	0x06
<Enter new watch>			

Hints and Suggested Approach

Computation of the next state could potentially follow the approach used in the textbook which is geared for implementation using Verilog and logic hardware. There are other approaches, however, which are more sensible for a byte oriented, general purpose machine such as the PIC. For example, the state could be "one-hot encoded" to match the outputs, and the next-state could be computed using the table look-up or by if-then statements that consider the present state and the inputs.