Assignment 3A (50 marks) – Lab Week Nine (Due: End of your week Ten lab period).

This lab exercise may be optionally performed by TWO students working as a group (students pick their own partners from the SAME lab section). If you are in a multi-student group, ensure all student names/numbers are on all program listings and documentation in order for all students to receive credit for the work (No name, no credit).

Additionally, you must be present to receive the marks for the demonstration of your solution.

Switches/LEDs, Pointers and Flowcharts

The ***AsmIDE*** assembler, ***Simulator – Dragon12 & Student Mode*** and the Dragon12 hardware board will all be used in this lab assignment.

Additional Software required for this lab.

Download and install Microsoft Visio Professional 2013 32/64-bit (English) from DreamSpark. As advised in class, I recommend that you install the 32-bit version even if you are using a 64-bit Windows operating system.

Notes:

i. If you already have an earlier version of Visio on your computer use that version
ii. The **Resources** folder on Blackboard contains a file that you can use as a template for Black and White Visio diagrams (the default stencils are BLUE and very difficult to see when printed if you do not use a colour printer.

LAB TASKS:

a. Task One – Implement an Assembly Language solution to a given problem involving Switches/LEDs (20 marks)
b. Task Two – Working with Pointers and Arrays (20 marks)
c. Task Three – Counter Flowchart (10 marks)

Assembly Language Source Code Directory Structure

All of the Assembly Language files in this lab should be stored on your N: drive in ASM\Lab9
(Note: The folder can also be a subfolder of CST8216 or however you organize your files).

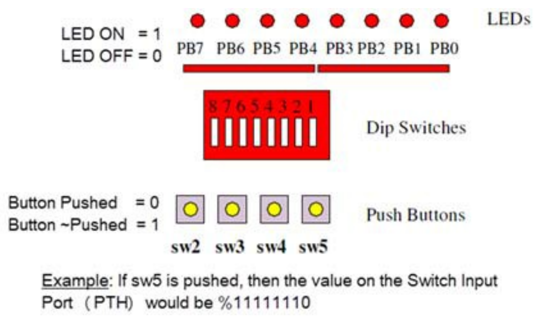
Task One – Implement an Assembly Language solution to a given problem (20 marks)

Using a previous lab's software listing (Lab Week Eight – ***SWs_To_LEDs.asm***), you should have discovered that pressing one of the four Push Buttons (SW2 to SW5) resulted an LED being momentarily turned off – e.g. pressing SW2 turned off the PB3 LED or pressing SW5 turned off the PB0 LED.

In that lab you were also asked to click on any of the SW1-1 to SW1-8 check boxes, which simulate the physical DIP switches on Port H of the hardware board, and to observe the resulting changes to the LEDs on Port B. You should have also discovered that there was a certain relationship with SW1-4 to SW1-1 and the Switches, SW2 to SW5.

In this task you are to build upon the previous lab assignment to realize the requirements as detailed in the video **Switch_Behaviour.mp4** found on Blackboard.

The following diagram further explains the operation of the LEDs connected to their Output Port (PORTB) and the Switch Input Port (PTH).

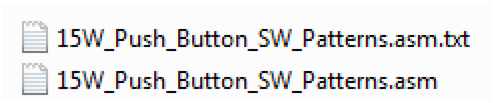


Example: If sw5 is pushed, then the value on the Switch Input Port (PTH) would be %11111110

To complete this task, you must write an Assembly Language program (***15W_Push_Button_SW_Patterns*.asm)** from the flowchart I created to solve the given problem.

Visio-15W_Flowchart for Push_Button_SW_Patterns.pdf

To further assist you, I have provided skeleton code (partial code listings) to get you started with the solution in a compressed file that contains the following two source code listings:


15W_Push_Button_SW_Patterns.asm.txt
15W_Push_Button_SW_Patterns.asm

To complete this task, you are to add additional code to the given listing; however, do **not** change any of the code that I have supplied to you. Note that you will be reusing the subprograms ***Config_SWs_and_LEDs.asm*** and ***Delay_ms.asm*** from a previous lab.

Hints:

i. Since the **equ** table is not complete for the various values of the Switches SW2 – SW5, you should first determine what values are read into Accumulator A from PTH (Port H switch input) using the software from a previous lab and fill in the table values. You can do this by using the previous lab's software as follows:
   a. discover the relationship with SW1-4 to SW1-1 and the Switches, SW2 to SW5.
   b. single-step through the code using the simulator; vary the switch selection and observe the values placed into Accumulator A
   c. complete the table of SW2 to SW3 values in the skeleton code

ii. The LED2 to LED5 bit patterns are simply the binary values for the base 10 numbers 2 to 5
iii. There is a "pattern" to the code. Compare the given code to the flowchart to discover it.

Demonstrate and submit your code according to the assignment Hand-In Sheet.

Task Two – Working with Pointers and Arrays (20 marks)

Write a complete program in Assembly Language (***A3A_Array.asm***) that effectively uses iteration and pointers to copy the values from **Source_Array** to **Destination_Array** and places the copied values backwards in memory. e.g. the last element from **Source_Array** is the first element in **Destination_Array** and so on.

Inside the same loop, you are to total the values in the **Source_Array** _as you iterate through the loop_, storing your additions in **Result**. Think carefully what registers to use in your solution. Zeroize the 16-bit value of **Result** before your loop and store intermediate additions at **Result** during each loop.

The following addresses are to be used for the given labels:

• **Source_Array**	**$1000**
• **Destination_Array**	address following end of **Source_Array**
• **Result**	address following end of **Destination_Array**

Your solution should realize the correct value in **Result** for any number of program runs – e.g. if **Result** = $0F2B the first program run, then if we run the program again, **Result** must = $0F2B again. Create a Test Plan and ensure you understand how the data is added by manually going through the data before you code the solution. Create a Memory Map that illustrates both the **BEFORE** and **AFTER** values of BOTH the **Source_Array** and **Destination_Array.**

The following data is to be used as your **Source_Array**

$13, $04, $4F, $74, $8B, $04

Your program code should commence at $2000, and the first line of code should initialize the stack to a value of $2000 (use an **equ** statement and an appropriate label to accomplish this, as per previous examples).

Hints:

i. To reserve two 8-bit bytes of memory for **Result**, use: **Result ds 2**
ii. Assembly Language instructions you may encounter in your solution: **lds, ldaa, staa, ldab, ldd, std, addd, ldx, ldy, cpx, cpy, bhs, swi**

Task Three – BCD Counter Flowchart (10 marks)
Using Microsoft Visio, create a flowchart that implements a counter that counts in BCD from $00 (BCD) - $99 (BCD), then starts over at $00 (BCD) - $99 (BCD). Each count will be displayed on PORT B LEDs as a binary number.

i.e. BCD $09 is displayed as: 0000 1001, BCD $10 is displayed as: 0001 0000, BCD $44 is displayed as: 0100 0100, where a 0 signifies that the LED is OFF and a 1 signifies that the LED is ON.

Ensure that your flowchart initializes the count to $00 (BCD) (all LEDs will be OFF for this value) and that it checks to see if you have reached the upper count of $99 (BCD) (the LED pattern will be 1001 1001 for this value).

You must also ensure that you do not have **any** invalid counts (e.g. in BCD the sequence is 0, 1, 2, … 9, 10, 11, … **not** 0, 1, 2, … 9, $A, $B, $C, $D, $E, $F, $10, $11…). Note that you **cannot** READ the values on the LEDs.

Lab Week Nine Hand-in Sheet – More on Assembly Language -- Hand in Sheet (by end of Lab Week Ten) /30 marks

Name: ____ Student Number: ________________________
Name: ____ Student Number: ________________________

Task One – Implement an Assembly Language solution to a given problem (20 marks)

In-lab Evaluation

- Demonstrate your completed Assembly Language program ***15W_Push_Button_SW_Patterns*.asm** functionality as per the supplied flowchart **using the Hardware Board** (**7 marks**)

Post-lab Evaluation – submit a hardcopy of your **Code** into your portfolio folder.

- Code matches supplied flowchart (**8 marks**)
- Code meets course standards – header, documentation, indentation, comments, printed from AsmIDE (**5 marks**)

Task Two – Working with Pointers and Arrays (20 marks)

Please staple all pages together!

In-lab Evaluation

- Demonstrate your completed Assembly Language program ***A3A_Array.asm*** functionality **using** Simulator - Dragon12 & Student Mode (**7 marks**)

Post-lab Evaluation – submit a hardcopy of your **Test Plan**, **Memory Map** and **Code** into your portfolio folder.

- Test Plan (**4 marks**)
- Memory Map (**4 marks**)
- Code meets course standards – header, documentation, indentation, comments, printed from AsmIDE (**5 marks**)

Task Three – Counter Flowchart (10 marks)

Post-lab Evaluation – submit a hardcopy of your BCD Counter Flowchart into your portfolio folder. It will be evaluated against the problem statement's required functionality (**10 marks**)