

Laboratory 2

(Due date: October 10th)

OBJECTIVES

- ✓ Learn how to use arithmetic, Boolean, shift, and other related Assembly Language instructions.
- ✓ Learn how to use branch instructions to implement loops
- ✓ Learn how to interface with pushbuttons and 7 segment display
- ✓ Learn how to call subroutines

FIRST ACTIVITY (50/100)

- Create a new project (**lab2a**). Add the given file lab2a.asm to the project.
- Default Connection: Select: *HCS12 Serial Monitor*.
- Debugger: When prompted to the Monitor Setup Window, select the proper HOST Serial Communication Port (go to Device Manager→Ports)
- In this project you will write a program in assembly to complete the following in order:
 1. Load an array of eight 8-bit unsigned numbers into RAM.
 2. Find the largest of these numbers (use a loop) and store it.
 3. Find the smallest of these numbers (use a loop) and store it.
 4. Wait in a loop for dip switches 1 or 2 to be set
 - If switch 1 is set display the smallest number on the 7 segment display
 - If switch 2 is set display the largest number on the 7 segment display
- Note: lab2a.asm has a subroutine for displaying hex values on the 7 segment display and one for reading the switches
- **Demonstrate that your code works to the TA and submit the working code to the Moodle Submission page**

TA signature: _____

Date: _____

SECOND ACTIVITY (50/100)

- Create a new project (**lab2b**). Try to make your own .asm file this time.
- In this project you will write a program in assembly that executes the following equation:
$$f[i] = s * (z^2[i] + z[i] + r)$$
where $0 \leq i < N-1$, choose $N < 64$, z is an input array of 16-bit unsigned N elements, $s = 0.25$ (use the shift operations to implement this), and r is an integer constant.
- Use loops and store the result array f (32-bit unsigned N elements) in a memory location different from z .
- **Demonstrate that your code works to the TA and submit the working code to the Moodle Submission page**

TA signature: _____

Date: _____

HINTS

- If you are using unsigned numbers in your program, make sure you are using the *unsigned* branch instructions in your ASM code.
- Your TA will be changing the numbers in your array to make sure it works for arbitrary 8-bit/16-bit values. Please make sure the array is somewhere near the top of your program file.
- In this lab you will use shift operations for division