

# CST8134: In-lab Exercise 9

## Stack frame

Before the break, you used the stack to pass arguments from a calling to a called function by writing two functions placing two words onto the stack as function arguments and adding them together, returning their sum in register d0. Today you are to do the same thing (use a copy of your functions and modify the copies – keep the original functions) and also create and use a stack frame with saved registers and local variables.

First, in the calling function, place your arguments on the stack. Then call your Add function. Be sure to adjust the sp value correctly upon return from the adding function and before you display the sum returned in d0.

You will write the adding function (as a function, not as part of your main program) to have at least three words of local variable space in its stack frame. Call it from your main program like this:

```
Num1:      .WORD      47
Num2:      .WORD      53
...
|||      place Num1 and Num2 onto the stack
          jsr      Add      | call my function
|||      correct sp to remove the arguments
/* register d0 now contains the sum */
```

In your Add function, save the registers and create your frame pointer. Now copy the arguments from the stack into two of your three local words. Compute their sum from the local variables and store the result in the third local word. Display the two numbers from inside the Add function on the screen using OutHex with some meaningful text and suitable new lines. Then load d0 from variable three, and clean up the stack.

Show the source code, your stack frame map, and the correct output of your program to your lab instructor before the end of your lab period.