CPE 325: Embedded Systems Laboratory Laboratory Assignment #5

Assignment [100 pts]

1. Write an assembly program that performs the **sum of squares** on all elements of an integer array by calling two separate subroutines, SW_MUL and HW_MUL. The array should be initialized in the main program along with a constant to describe its length. The base address of the array and the length of the array will then be passed to the subroutines through the program stack. The result from SW_MUL and HW_MUL should be passed back by registers R12 and R13 respectively. Remember, the number of array elements should be no less than five and take both the signed and unsigned numbers.
2. SW_MUL uses Shift-and-Add multiplication algorithm and HW_MUL uses Hardware Multiplier to multiply elements. Please go through the Shift-and-Add multiplication algorithm pdf for reference before implement this into your assignment.
3. Measure the number of clock cycles used by each subroutine for a small range of values. In your report provide the screenshot for the clock cycles you get in your case. Comment on the efficiency of each subroutine. Which one do you prefer? Why?
4. **Bonus (up to 15pts)** Create a subroutine that converts a string of at least a five-digit number to its numerical value by using the Hardware Multiplier, and stores the result in a variable in the memory. The subroutine needs to receive the base address of the string and the address of the variable through the program stack. For full credit you need to use the accumulator. If the accumulator is not used only up to 10pts will be rewarded.

Topics for Theory

a. Subroutines - what are they, and how did you use them in the lab
b. Passing parameters - describe 3 different ways data can be input to a subroutine
c. Hardware Multiplier – What it is/does. Why it is useful.

Deliverables

1. Lab report which includes:
    a. Flowchart
    b. Answers to the questions from the tutorial if any
    c. Screenshots of final outputs
2. Source files (.asm files)

Notes:

1. Do not hard code numbers or arrays in the sub routines.
2. You must pass the required parameters by using the specified method.
3. Try different inputs before you conclude which method is efficient.
4. In your explanation, include the inputs as well.
5. Assume that any of the results does not exceed 16-bits.