

## Lab One: Introduction to PowerPC Instruction Set, MPC875 and CodeWarrior

In this lab you will start exploring the PowerPC architecture, its registers and instruction set. You will learn how to code and debug programs in Metrowerks Codewarrior, an integrated software development environment for many embedded, handheld, and desktop platforms. You will also become familiar with the laboratory report requirements.

### 1 Introductory PowerPC Exercises

In this exercise you will use PowerPC instructions to accomplish several basic tasks. Before you start to write your own program, you may want to get familiar with the tools. A sample project `lab1.zip` can be downloaded from the course website. Follow the instructions in the manual to compile and run the sample code first. Then, you can add your own code into the sample code to complete the following tasks.

1. Implement the bubble sort algorithm with C language to sort an array of unsigned integers into *descending* order. The interface of the function will look like:

```
void my_bubblesort(unsigned int *p, int n);
```

The function will sort the  $n$  unsigned integer stored in  $p$  in descending order.

2. Define an array `BUF` of eight (8) unsigned integers. Implement a function with embedded instructions to set the array to 0. For example, your code may look like:

```
void clear_array(unsigned int *p, int n) {
    asm {
        }
}
```

You can then call `clear_array(BUF, 8)` to set `BUF` to 0.

3. Implement a function and use embedded assembly code to deposit the following 32-bit words into `BUF`. Do not assume these values are stored somewhere in the memory. (Hint: each constant requires two instructions.)  
`0x01234567, 0x89ABCDEF, 0x55555555, 0xAAAAAAAA,`  
`0xFEDCBA98, 0x76543210, 0xD409ACAB, 0X3C7A9BC0.`

4. Consider the words in BUF as unsigned 32-bit words. Compute the sum of the eight words with embedded instructions. (Hint: you may need two registers to store the sum during the computation and store the result in an array.)
5. (Bonus tasks, 10% of the total points) Implement the bubble sort with embedded instructions.

You are free to use any instructions to accomplish the above tasks, but awkward choice of instructions will result in point deduction. You should try your best to optimize the code for performance.

Document your design, implementation, and results carefully. You may consider these tasks as several steps in a big task. So you can have only one objectives section, in which you describe all the tasks you have done (instead of having several individual objectives sections).

## **2 Deliverables**

Write a report adhering to the lab requirements. The report should include documentation (including results) and code for all the tasks.