<u>CPU Design Project: Part 1 – ISA, Report Due Wednesday, 9/25/2013</u>

You are to design an instruction set architecture (ISA) for a new 16-bit microprocessor (µP). The µP will be designed and modeled in VHDL in later parts. Your ISA is to be designed using RISC design principles, with the primary design goals being low cost and a minimal number of clock cycles per instruction. Following are the requirements for your ISA.

1. The ISA may contain no more than 16 unique instructions. However, you may have multiple formats for a given type of instruction, if necessary.
2. Of the 16 instructions, at least one instruction should make your processor **HALT.**
3. The ISA is to support 16-bit data words only. (No byte operands.)
    a. All operands are to be 16-bit signed integers (2's complement).
    b. Each instruction must be encoded using one 16-bit word.
4. The ISA is to support linear addressing of 1K, 16-bit words memory. The memory is to be word-addressable only - **not byte-addressable**.
5. The ISA should contain appropriate numbers and types of user-programmable registers to support it. Since this is a small processor, the hardware does not necessarily need to support dedicated registers for stack pointer, frame pointer, etc.
6. The ISA must "support" the following C Programming Language constructs:
    - Assignment operator: *variable = expression*;
        - Supported arithmetic operators in expressions must include: add (+) and subtract (-)
            - It is not necessary to support multiply (*) and divide (/)
        - Supported logical operators in expressions must include: and (&) and or (|)
        - Data are limited to:
            - 16-bit two's-complement integers (Example: int a;)
            - One-dimensional integer arrays (Example: int a[10];)
    - Control flow structures: "if-else" structures, "while" loops, "for" loops
        - These should support the six standard relational operators:
          ==, !=, >, <=, <, >=
    - Functions (call and return), with parameters able to be passed by value or by reference.

7. Provide the following information about your ISA:
    - List and describe the roles of the user-programmable registers.
    - List and describe the different instruction formats used.
    - For each instruction in your instruction set, list the following:
        - Assembly language for each form of the instruction - mnemonic and operands
        - Machine language for each form of the instruction: instruction code format, op-code, and operand encoding
        - *Justification for including each form of the instruction in your ISA*

8. For each C construct listed in item 6 above, provide an example showing how the construct would be "compiled", i.e. implemented with your instruction set, by writing an example of the C construct and the corresponding assembly language implementation.