# Floating-Point Arithmetic

Modifying the assembler and simulator to include arithmetic operations for floating-point numbers of the form (precision) given above.

Specifically, including the following functions:

| Opcode | Instruction | Semantics | Syntax | Type |
|---|---|---|---|---|
| 10000 | F_Addition | Performs `reg1 = reg2 + reg3`. If the computation overflows, then the overflow flag is set and `reg1` is set to 0. | `addf reg1 reg2 reg3` | A |
| 10001 | F_Subtraction | Performs `reg1 = reg2 - reg3`. In case `reg3 > reg2`, 0 is written to `reg1` and overflow flag is set. | `subf reg1 reg2 reg3` | A |
| 10010 | Move F_Immediate | Performs `reg1 = $Imm` where `Imm` is an 8-bit floating-point value. | `movf reg1 $Imm` | B |

**Note:**

- For moving 1.5 into `reg1`, the instruction (in assembly language) should be: `movf reg1 $1.5`.
- In floating-point multiplication, $Imm is 8 bits so you need to make a new Type B syntax with 8 bits.
- Only applying the operations for the floating-point numbers that can be represented in the choosen system (8 bits), else it will report it as an error.