



### Implementação em *Assembly* MIPS de um somador de números IEEE754 usando apenas inteiros

O objetivo deste trabalho é implementar em *Assembly* MIPS um somador de números IEEE754 usando apenas a estrutura existente para números inteiros. **A soma implementada deve seguir a representação e os passos explicados em sala de aula** e considerar apenas a precisão simples (32 bits-float). Não é necessário implementar as demais operações aritméticas, apenas a soma. Deve considerar números no formato IEEE754 positivos e/ou negativos.

O algoritmo deve estar modularizado. Assim haverá uma função denominada ***add\_ieee754***, chamada pela ***main*** e cujos argumentos de entrada (em ***\$a0*** e ***\$a1***) são os dois números em binário, já no padrão IEEE754. O retorno dessa função ***add\_ieee754*** deve ser o resultado da soma (em ***\$v0***) a ser impresso pela função ***main***.

A função ***add\_ieee754*** não pode usar nenhum recurso para ponto-flutuante, apenas para inteiros. Isso inclui registradores, unidades funcionais e instruções. Obviamente, a soma a ser implementada neste trabalho não pode considerar as instruções de soma para ponto flutuante já existentes no *Assembly* MIPS.

O algoritmo deve executar apenas uma soma e finalizar após a mesma.

O algoritmo deve considerar o caso especial usado para o número 0 (zero) no padrão IEEE 754. Não precisam ser considerados infinito, números desnormalizados e demais casos de exceção.

As operações de arredondamento e normalização devem ser, obrigatoriamente, implementadas como funções distintas do restante do código, passando-lhes argumentos de entrada e obtendo-se os seus resultados, de acordo com o padrão *Assembly* MIPS. As funções devem tratar o uso da pilha corretamente, conforme visto em sala de aula e descrito no livro de Organização do Patterson & Hennessy.

Para simplificar, o arredondamento deve ser implementado da seguinte maneira: truncar a mantissa caso o primeiro bit não representável da mesma seja 0; incrementar a mantissa caso o primeiro bit não representável da mesma seja 1.

O *overflow* e o *underflow* devem ser considerados. Caso ocorram, o retorno da função ***add\_ieee754*** deve indicar tais fatos com códigos de erro seguindo este padrão: *overflow*=>expoente contendo 0xFF e bit de sinal 1; *underflow*=>expoente contendo 0xFF e bit de sinal 0.

As referências para este trabalho devem ser os livros texto e os livros constantes na bibliografia complementar, todos usados na disciplina.

O algoritmo deve implementar a soma de maneira correta e o mesmo deve possuir legibilidade, comentários, modularidade, indentação, nomes significativos para os rótulos e seguir o padrão do *Assembly* MIPS para o uso dos registradores.

O aluno deverá procurar o professor responsável pela disciplina ou os alunos PAE se houver qualquer dúvida para o desenvolvimento deste trabalho ou se forem encontrados eventuais problemas/omissões nesta especificação.

O trabalho deverá ser feito em grupo, conforme regras já discutidas para os grupos.

A data máxima de entrega do trabalho já está fixada para às 08:00h do dia 28/09/2009, uma segunda-feira.