

# Trabalho Prático 3 – Adicionando Instruções

Organização de Computadores I (DCC006)

Antônio Otávio Fernandes e Lucas Ferreira de Melo Diniz

## 0 Mips

No trabalho prático passado, foram disponibilizados os módulos necessários para a construção do MIPS, o qual já continha algumas instruções implementadas. No trabalho 3 novas instruções deverão ser implementadas.

As instruções propostas não fazem parte da lista de instruções comumente encontradas no MIPS e por isso serão exclusivas do processador de vocês. Para a realização do mesmo será necessário conhecer todas as etapas do processador, alterar o mesmo e realizar os testes necessários.

## Incremento

A instrução que adiciona 1 ao valor do registrador *rs* e armazena em *rt*. Lembrando que a soma deve considerar o sinal do valor em *rs*.

Sinal	Formato	op						fn					
bits		5	4	3	2	1	0	5	4	3	2	1	0
INC	rs, rt, offset (16)	0	0	1	0	1	0	X	X	X	X	X	X

Exemplo:

```
001010ss sssttttt -----
```

## Multiplicação

A instrução multiplica o valor do registrador de endereço *rs* ao valor do registrador de endereço *rt* e salva o resultado no registrador de endereço *rd* (32bits). Se o resultado ocupar mais de 32bits, o sinal de *overflow* deve ser ativado e nenhum valor salvo na memória.

Sinal	Formato	op						fn					
bits		5	4	3	2	1	0	5	4	3	2	1	0
MUL	rs, rt, rd, 0	0	0	0	0	0	0	1	0	1	0	0	0

Exemplo de instrução:

```
000000ss sssttttt ddddd--- --101000
```

## Testes

Para a implementação dos comandos de incremento vocês podem utilizar as instruções já disponibilizadas ou adicionar novos operadores à ALU. Deixem claro porém qual foi a escolha do grupo na documentação. A operação de multiplicação por outro lado deve necessariamente ser adicionada à ALU.

Na documentação deixem explícito todo código alterado da implementação atual, de forma que eu possa avaliar e encontrar as mudanças. Indiquem no código de vocês com comentários os lugares que tiveram que alterar os MIPs disponibilizado.

TODOS deverão fazer uso dos MIPs disponibilizado junto a esse trabalho.

Lembrem-se de realizar os testes e verificar as situações de exceção da execução dos novos comandos. Os testes realizados também devem ser incluídos na entrega.

## Entrega

A entrega deverá ser feita via moodle até o dia 24/06.

Deve ser entregue um arquivo com nome `trabalho3_grupo_<numero_do_grupo>.zip` contendo uma pasta chamada `trabalho3_grupo_<numero_do_grupo>`.

Dentro dessa pasta devem constar três diretórios chamados `src`, `tb` e `doc`, que devem conter, respectivamente, os arquivos referentes aos módulos implementados/alterados, os arquivos correspondentes aos testbenches e a documentação sobre cada um dos módulos alterados.

Testbenches devem ter o seguinte padrão de nomes: `tb_<nome_do_arquivo_sendo_testado>`. Por exemplo, o testbench de um arquivo chamado `Mips.v` deve chamar `tb_Mips.v`.

A documentação sobre os módulos alterados deve ser sucinta (menos de uma página deve ser suficiente) e pode ser entregue em qualquer formato, `.txt`, `.pdf`, `.doc`, mas preferencialmente com um nome que indique a qual módulo ela se refere. Esta documentação deve conter uma breve descrição das alterações realizadas no módulo, eventuais decisões de projeto.

Qualquer dúvida ou ambiguidade com relação à especificação, por favor, entrem em contato comigo. Porém estarei disponível para dúvidas somente até o dia **21/06**. Depois disso somente em casos excepcionais responderei aos e-mails. Por isso, comecem o trabalho o quanto antes.

Atrasos sofrerão penalizações e cópias não serão toleradas.