

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS

Organização de Computadores I  
Trabalho Prático III

**Alunos:** Eugênio Pacceli Reis da Fonseca,  
Jonatas Cavalcante,  
Victor Pires Diniz

## OBJETIVO

O objetivo do trabalho é adicionar duas novas instruções ao processador MIPS já trabalhado anteriormente. Para fazê-lo, é necessário alterar alguns componentes do processador para que ele seja capaz de decodificar as duas novas instruções e executá-las, como instruído. As instruções adicionadas, INC e MUL, não existem no conjunto de instruções do MIPS padrão.

## INSTRUÇÕES

**INC** \$regA, \$regB

Efetua uma adição de 1 com o valor inteiro em \$regB, e armazena o resultado da adição em \$regA.

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

**MUL** \$regA, \$regB, \$regC

Efetua uma multiplicação com os valores inteiros armazenados em \$regB e \$regC, armazenando o resultado em \$regA em caso de sucesso. Caso ocorra um overflow, ou seja, o resultado da operação for maior do que  $2^{32}-1$ , o flag de overflow deve ser sinalizado e o valor anterior de \$regA deve ser preservado.

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

## IMPLEMENTAÇÃO

Para adicionar INC:

- Control.v foi modificado para tratar o novo caso, que é uma copia dos sinais do comando ADDI.
- Foi adicionado um desvio condicional, baseado no opcode da instrução,

em Decode.v, para que, quando ele for equivalente a INC, 1 deve ser somado ao valor operando.

Para adicionar MUL:

- Control.v foi estendido para tratar o novo caso, que é similar ao do comando ADD, sendo a AluOp é 111, invés de 010.
- Alu.v foi modificada para adicionar o tratamento para o novo comando, cujo resultado é computado em um registrador intermediário de 64 bits. Esse registrador recebe o valor da multiplicação dos registradores-operandos, e, se algum dos seus 32 bits mais significativos for diferente de 0, quando os multiplicandos tiverem o mesmo sinal, ou diferente de 1, quando os multiplicandos tiverem sinais diferentes, ocorreu overflow na multiplicação, que é sinalizada. Caso contrário, os 32 bits menos significativos desse registrador são copiados para aluout\_reg.

## TESTBENCHES

Foram criados quatro testbenches: “tb0\_arith\_basic.a”, testa a soma de um valor com 2 e depois soma-o a outro, usando as operações padrão do MIPS; “tb1\_incr.a” efetua os mesmos passos do tb0, só que um dos valores é incrementado três vezes (com “INC”) invés de ser somado com 2, e o resultado deve ser uma unidade maior do que o resultado de tb0; “tb3\_fat.a” efetua a operação de fatorial de um número utilizando a instrução “MUL” adicionada; “tb2\_mul.a” testa o caso de overflow em “MUL”.

Os testes se encontram em “./tb0/”, e podem ser compilados, juntos ao projeto de processador, com o comando “make”, assumindo que o ambiente de execução possua o compilador iverilog.