

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS  
GERAIS –  
CEFET/MG**

    Celso França Neto (20203018570)  
Diogo Emanuel Antunes Santos (20213002091)

**AULA 6 - Projeto de um Processador: implementação do programa-teste.**

**Laboratório de Arquitetura e Organização de Computadores**

**BELO HORIZONTE  
2022**

**1) Apresente o código do programa-teste em assembly do MIPS (utilize comentários para facilitar a revisão do seu código). Mostre seu correto funcionamento no MARS.**

```
.globl main

        .text

main:

    lw $t0,n($zero) #passando valor do inteiro n para o registrador $t0
    addi $t1, $zero, 1 #passando 1 pro registrador $t1
    sw $t1,fatorial($zero) #salvando o valor 1 ($t1) na memória

    while: #estrutura de repetição pra fatorial
        addi $t1, $zero, 1
        beq $t0, $t1, saida #comparação se n > 1, ($t0 > $t1). Condição de parada do
loop.

        lw $t1,fatorial($zero) #passando pro registrador $t1 o valor salvo na memoria

        mul $t1,$t1,$t0 #multiplicando o numero a ser fatorialado pelo seu antecessor, e
o resultado pelo antecessor do antecessor...

        sw $t1,fatorial($zero) # salvando resultado da multiplicação na memória

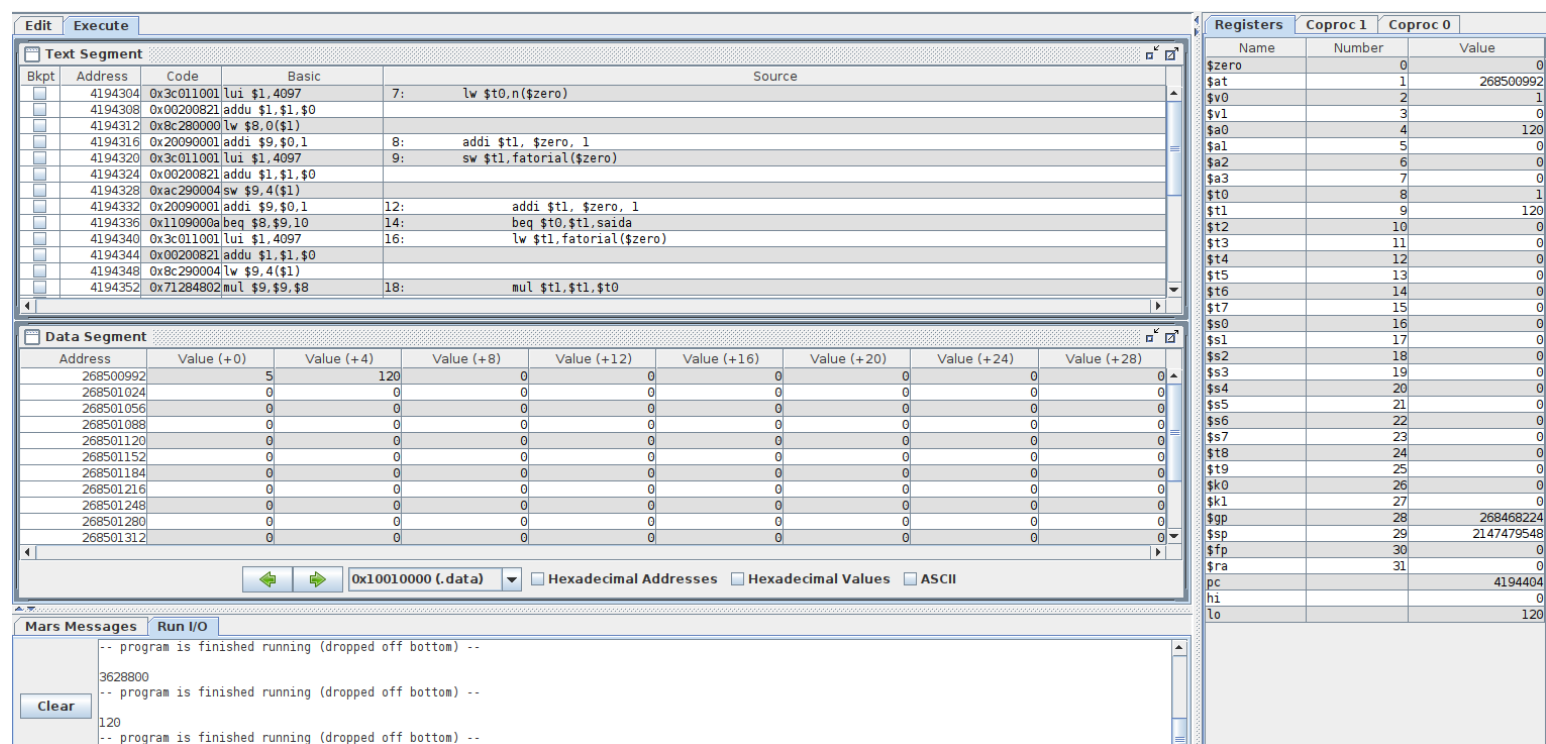
        subi $t0,$t0,1 # decrementando a interação n ou seja, jogando o antecessor do
atual n pro loop

        j while # retornando pro loop

saida:
    syscall #encerrando programa

.data

n:      .word 5
fatorial: .word
```



**2) Apresente o código do programa-teste em assembly do seu nRisc (utilize comentários para facilitar a revisão do seu código).**

```
.globl main
```

```
.text
```

```
main:
```

```
lw reg0, n #passando valor do inteiro n pro registrador reg0.
```

```
defi reg1, 1 #passando 1 pro registrador reg1
```

```
sw reg1, fatorial #salvando o valor 1 (reg1) na memória
```

```
while: #estrutura de repetição para fatoraçaõ
```

```
defi reg1, 1
```

```
beq reg0, reg1, saida #comparaçaõ se n > 1, (reg0 > reg1). Condiçaõ de
parada do loop.
```

```
lw reg1, fatorial #passando pro registrador reg1 o valor salvo na memória
```

```
mul reg1, reg0 #multiplicando o número a ser fatorado pelo seu antecessor, e o
resultado pelo antecessor do antecessor...
```

sw reg1,fatorial # salvando resultado da multiplicação na memória

subi reg0, 1 # decrementando a interação n ou seja, jogando o antecessor do atual n pro loop

j while # retornando pro loop

saida:

encerra # encerrando programa

.data

n: .word 5

fatorial: .word

**3) Apresente o código do programa-teste em binário no formato de instruções definido para o seu nRisc (indique cada instrução em uma linha de texto separada). Mostre, no relatório, as etapas intermediárias do processo de conversão das instruções (exemplo: valores decimais dos campos das instruções, cálculos de endereços, etc).**

Tabelas de cada instrução em seu valor decimal e binário:

Instrução 0:

	op	rs	endereço
Decimais	2	0	1
Binários	0 1 0	0	0 0 0 1

**load word: carrega o valor de um local da memória para o registrador de destino**

Instrução 1:

	op	rs	valor
Decimais	0	1	1
Binários	0 0 0	1	0 0 0 1

**defi: move o valor imediato para o registrador destino**

Instrução 2:

	op	rs	endereço
Decimais	3	1	2
Binários	0 1 1	1	0 0 1 0

**store word: guarda o valor no registrador para um local da memória principal**

Instrução 3:

	op	rs	valor
Decimais	0	1	1
Binários	0 0 0	1	0 0 0 1

**defi: move o valor imediato para o registrador destino**

Instrução 4:

	op	rs	rt	endereço
Decimais	1	0	1	5
Binários	0 0 1	0	1	1 0 1

**beq: desvio condicional, se o primeiro registrador for igual ao segundo, salta PC + 1 + (Imediato)**

Instrução 5:

	op	rs	endereço
Decimais	2	1	2
Binários	0 1 0	1	0 0 1 0

**load word: carrega o valor de um local da memória para o registrador de destino**

Instrução 6:

	op	rs	rt	não utilizado
Decimais	4	1	0	0
Binários	1 0 0	1	0	0 0 0

**mul: multiplica o primeiro com o segundo registrador e guarda no primeiro**

Instrução 7:

	op	rs	endereço
Decimais	3	1	2
Binários	0 1 1	1	0 0 1 0

**store word: guarda o valor no registrador para um local da memória principal**

Instrução 8:

	op	rs	valor
Decimais	5	0	1
Binários	1 0 1	0	0 0 0 1

**subi: subtrai o valor no registrador pelo imediato e guarda no registrador**

Instrução 9:

	op	endereço
Decimais	6	3
Binários	1      1      0	0      0      0      1      1

**j: desvio incondicional, salta para o endereço: (Imediato)**

Instrução 10:

	op	não utilizado
Decimais	7	0
Binários	1      1      1	0      0      0      0      0

**end: finaliza o programa**

Cálculo de endereço de desvio utilizando um desvio condicional (beq):

Endereço de destino = PC + 1 +(Imediato).

Exemplo no nRisc utilizando o beq:

	op	rs	rt	end
4	1	0	1	5

Endereço de destino = 4 +1 + 5

Endereço de destino = 10

Cálculo de endereço de desvio utilizando um desvio condicional (beq):

Endereço de destino = Imediato

Exemplo no nRisc utilizando o j:

	op	end
9	6	3

Endereço de destino = 3

3	0	1	1
4	1	0	1   5
5	2	1	2
6	4	1	0   0
7	3	1	2
8	5	0	1
9	6		3
10	7		0

Código definido para o nRisc em Binário:

```

0 01000001
1 00010001
2 01110010
3 00010001
4 00101101
5 01010010
6 10010000
7 01110010
8 10100001
9 11000011
10 11100000

```