

## Exercício Prático 06

A partir de agora, nos exercícios com programação você deverá usar o MARS. Você deverá apresentar um print da tela contendo o programa e a sua execução.

Exemplo:

Seja o exercício de implementar o seguinte programa:

$x=1$ ;

$x=x+1$ ;

A solução é a seguinte:

# Associações:  $x \rightarrow \$s0$

# inicio

addi \$s0, \$zero, 1 #  $x = 1$

addi \$s0, \$s0, 1 #  $x = x + 1$

#fim

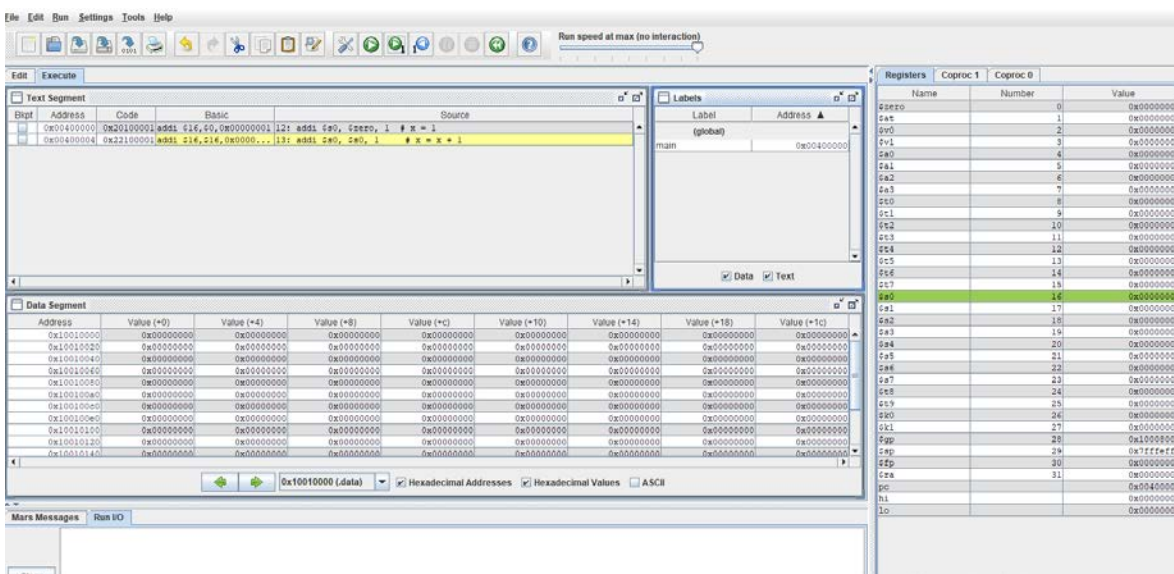
O que deverá ser então apresentado no relatório:

O programa implementado:



```
1 # Exemplo 01
2 # x=1;
3 # x=x+1;
4 # Associações:
5 # x -> $s0
6
7 # inicio
8
9 .text
10 .globl main
11 main:
12 addi $s0, $zero, 1 # x = 1
13 addi $s0, $s0, 1 # x = x + 1
14
15 #fim
16
```

A sua execução:



The screenshot displays the MARS IDE interface during program execution. The 'Registers' panel on the right lists registers \$s0 through \$s31, with \$s0 highlighted. The 'Data Segment' panel at the bottom shows memory addresses and their corresponding values. The 'Registers' panel shows the following values:

Name	Number	Value
\$s0	0	0x00000001
\$s1	1	0x00000000
\$s2	2	0x00000000
\$s3	3	0x00000000
\$s4	4	0x00000000
\$s5	5	0x00000000
\$s6	6	0x00000000
\$s7	7	0x00000000
\$s8	8	0x00000000
\$s9	9	0x00000000
\$s10	10	0x00000000
\$s11	11	0x00000000
\$s12	12	0x00000000
\$s13	13	0x00000000
\$s14	14	0x00000000
\$s15	15	0x00000000
\$s16	16	0x00000000
\$s17	17	0x00000000
\$s18	18	0x00000000
\$s19	19	0x00000000
\$s20	20	0x00000000
\$s21	21	0x00000000
\$s22	22	0x00000000
\$s23	23	0x00000000
\$s24	24	0x00000000
\$s25	25	0x00000000
\$s26	26	0x00000000
\$s27	27	0x00000000
\$s28	28	0x00000000
\$s29	29	0x00000000
\$s30	30	0x00000000
\$s31	31	0x00000000

Apresente estas telas para cada exercício de programação

## Parte1 - Resposta

1. O que é um arquivo fonte?
  - A. um arquivo de texto que contém instruções de linguagem de programação.
  - B. um subdiretório que contém os programas.
  - C. um arquivo que contém dados para um programa.
  - D. um documento que contém os requisitos para um projeto.
2. O que é um registrador?
  - A. parte do sistema de computador que mantém o controle dos parâmetros do sistema.
  - B. uma parte do processador que possui um padrão de bits.
  - C. parte do processador que contém o seu número de série único.
  - D. parte do bus de sistema que contém dados.
3. Qual o caracter que, na linguagem assembly do SPIM, inicia um comentário?
  - A. #
  - B. \$
  - C. //
  - D. \*
4. Quantos bits há em cada instrução de máquina MIPS?
  - A. 8
  - B. 16
  - C. 32
  - D. instruções diferentes possuem diferentes comprimentos.
5. O que é o contador de programa?
  - A. um registrador que mantém a conta do número de erros durante a execução de um programa.
  - B. uma parte do processador que contém o endereço da primeira palavra de dados.
  - C. uma variável na montadora que os números das linhas do arquivo de origem.
  - D. parte do processador que contém o endereço da próxima instrução de máquina para ser obtida.
6. Ao executarmos uma instrução, quanto será adicionado ao contador de programa?
  - A. 1
  - B. 2
  - C. 4
  - D. 8
7. O que é uma diretiva, tal como a diretiva .text?
  - A. uma instrução em linguagem assembly que resulta em uma instrução em linguagem de máquina.
  - B. uma das opções de menu do sistema SPIM.
  - C. uma instrução em linguagem de máquina que faz com que uma operação sobre os dados ocorra.
  - D. uma declaração que diz o montador algo sobre o que o programador quer, mas não corresponde diretamente a uma instrução de máquina.
8. O que é um endereço simbólico?
  - A. um local de memória que contém dados simbólicos.
  - B. um byte na memória que contém o endereço de dados.
  - C. símbolo dado como argumento para uma directiva.
  - D. um nome usado no código-fonte em linguagem assembly para um local na memória.

9. Em qual endereço o simulador SPIM coloca a primeira instrução de máquina quando ele está sendo executado?
- A. 0x00000000
  - B. 0x00400000
  - C. 0x10000000
  - D. 0xFFFFFFFF
10. Algumas instruções de máquina possuem uma constante como um dos operandos. Como é chamado tal operando?
- A. operando imediato
  - B. operando embutido
  - C. operando binário
  - D. operando de máquina
11. Como é chamada uma operação lógica executada entre bits de cada coluna dos operandos para produzir um bit de resultado para cada coluna?
- A. operação lógica
  - B. operação bitwise
  - C. operação binária
  - D. operação coluna
12. Quando uma operação é de fato executada, como estão os operandos na ALU?
- A. Pelo menos um operando deve ser de 32 bit.
  - B. Cada operando pode ser de qualquer tamanho.
  - C. Ambos operandos devem que vir de registros.
  - D. Cada um dos registradores deve possuir 32 bit.
13. Dezesesseis bits de dados de uma instrução de ori são usados como um operando imediato. Durante execução, o que deve ser feito primeiro?
- A. Os dados são estendidos em zero à direita por 16 bits.
  - B. Os dados são estendidos em zero à esquerda por 16 bits.
  - C. Nada precisa ser feito.
  - D. Apenas 16 bits são usados pelo outro operando.
14. Qual das instruções seguintes armazenam no registrador \$5 um padrão de bits que representa positivo 48?
- A. ori \$5,\$0,0x48
  - B. ori \$5,\$5,0x48
  - C. ori \$5,\$0,48
  - D. ori \$0,\$5,0x48
15. A instrução de ori pode armazenar o complemento de dois de um número em um registrador?
- A. Não.
  - B. Sim.
16. Qual das instruções seguintes limpa todos os bits no registrador \$8 com exceção do byte de baixa ordem que fica inalterado?
- A. ori \$8,\$8,0xFF
  - B. ori \$8,\$0,0x00FF
  - C. xori \$8,\$8,0xFF
  - D. andi \$8,\$8,0xFF
17. Qual é o resultado de um ou exclusivo de padrão sobre ele mesmo?

- A. Todos os bits em zero.
- B. Todos os bits em um.
- C. O padrão original utilizado.
- D. O resultado é o contrário do original.

18. Todas as instruções de máquina têm os mesmos campos?

- A. Não. Diferentes de instruções de máquina possuem campos diferentes.
- B. Não. Cada instrução de máquina é completamente diferente de qualquer outra.
- C. Sim. Todas as instruções de máquina têm os mesmos campos na mesma ordem.
- D. Sim. Todas as instruções de máquina têm os mesmos campos, mas eles podem estar em ordens diferentes.

## Parte2 - Implementar em MIPS/MARS os seguintes programas (usando apenas as instruções indicadas)

//programa 1 (add, addi, sub, lógicas)

```
{  
    a = 2;  
    b = 3;  
    c = 4;  
    d = 5;  
    x = (a+b) - (c+d);  
    y = a - b + x;  
    b = x - y;  
}
```

//programa 2 (add, addi, sub, lógicas)

```
{  
    x = 1;  
    y = 5*x + 15;  
}
```

// programa 3 (add, addi, sub, lógicas)

```
{  
    x = 3;  
    y = 4 ;  
    z = ( 15*x + 67*y)*4  
}
```

**Nos exercícios a seguir procure usar as inst. sll, srl e sra:**

**// programa 4**

```
{  
    x = 3;  
    y = 4;  
    z = ( 15*x + 67*y)*4  
}
```

**// programa 5**

```
{  
    x = 100000;  
    y = 200000;  
    z = x + y;  
}
```

**// programa 6**

```
{  
    x = o maior inteiro possível;  
    y = 300000;  
    z = x - 4y  
}
```

**// programa 7**

Considere a seguinte instrução iniciando um programa:

ori \$8, \$0, 0x01

Usando apenas instruções reg-reg lógicas e/ou instruções de deslocamento (sll, srl e sra), continuar o programa de forma que ao final, tenhamos o seguinte conteúdo no registrador \$8:

\$8 = 0xFFFFFFFF

**// programa 8**

Inicialmente escreva um programa que faça:

\$8 = 0x12345678.

A partir do registrador \$8 acima, usando apenas instruções lógicas (or, ori, and, andi, xor, xori) e instruções de deslocamento (sll, srl e sra), você deverá obter os seguintes valores nos respectivos registradores:

\$9 = 0x12

\$10 = 0x34

\$11 = 0x56

\$12 = 0x78