

Alunos:

Gabriel Filipe Manso Araújo - 16/0121019

Guilherme Antonio Deusdará Banci - 16/0122996

**Fundamentos de Arquitetura de Computadores - Trabalho
01**

Sistema Operacional Utilizado: macOS

Ambiente de Desenvolvimento:

MIPS/MARS 4.5

Instruções de Uso:

1. Abra o arquivo trabalho01.spim com o MARS;
2. Execute o programa;
3. Será solicitado a inserção de dois valores inteiros;
4. Insira o primeiro valor e aperte ENTER;
5. Insira o segundo valor e aperte ENTER;
6. O resultado da execução do programa será impresso na tela no seguinte formato:

ADD: [resultado da adição entre os dois valores inseridos]

SUB: [resultado da subtração entre os dois valores inseridos]

AND: [resultado da operação and entre os dois valores inseridos]

OR: [resultadoda operação or entre os dois valores inseridos]

XOR: [resultado da operação xor entre os dois valores inseridos]

SLL: [resultado da operação sll com o primeiro valor inserido]

SRL: [resultado da operação srl com o segundo valor inserido]

Exemplo: Com as entradas 9 e 2, respectivamente, obtemos a seguinte saída:

ADD: 11

SUB: 7

AND: 0

OR: 11

XOR: 11

SLL(3): 72

SRL(1): 1

Limitações Conhecidas:

O programa funciona perfeitamente com módulos de 16 bits, acima disso não é garantido o bom funcionamento.

Printscreens:

O programa foi dividido em três partes: dados, main e função de prints.

.data:

```
1  # Gabriel Filipe Manso Araujo - 16/0121019
2  # Guilherme Antonio Deusdará Banci - 16/0122996
3
4  # FAC: TRABALHO 01
5
6      .data
7
8  space:      .asciiiz  " "          # space
9  new_line:   .asciiiz  "\n"        # newline
10 string_ADD: .asciiiz  "ADD: "
11 string_SUB: .asciiiz  "\nSUB: "
12 string_AND: .asciiiz  "\nAND: "
13 string_OR:  .asciiiz  "\nOR: "
14 string_XOR: .asciiiz  "\nXOR: "
15 string_SLL: .asciiiz  "\nSLL(3): "
16 string_SRL: .asciiiz  "\nSRL(1): "
17
```

.main:

```
18      .text
19
20 main:
21     li $v0, 5      # load appropriate system call code into register $v0;
22                     # code for reading integer is 5
23     syscall         # call operating system to perform operation
24     move $a0, $v0   # O PRIMEIRO VALOR LIDO DO TECLADO ESTA DISPONIVEL EM $A0
25     li $v0, 5      # load appropriate system call code into register $v0;
26                     # code for reading integer is 5
27     syscall         # call operating system to perform operation
28     move $a1, $v0   # O SEGUNDO VALOR LIDO DO TECLADO ESTA DISPONIVEL EM $A1
29
30 # ===== IMPLEMENTE AQUI SUA SOLUCAO: INICIO
31
32     add $t0, $a0, $a1 # Adding the values into register $a0 and $a1 and putting result into $t0
33     sub $t1, $a0, $a1 # Subtracting the values into register $a0 and $a1 and putting result into $t1
34     and $t2, $a0, $a1 # Making AND operation with the values into register $a0 and $a1 and putting result into $t2
35     or  $t3, $a0, $a1 # Making OR operation with the values into register $a0 and $a1 and putting result into $t3
36     xor $t4, $a0, $a1 # Making XOR operation with the values into register $a0 and $a1 and putting result into $t4
37     sll $t5, $a0, 3
38     srl $t6, $a1, 1
39
40 # ===== IMPLEMENTE AQUI SUA SOLUCAO: FIM
41
42     jal print        # call print routine.
43     li $v0, 10       # system call for exit
44     syscall          # we are out of here.
45
46
```

Função print:

```

47 print:
48
49     la    $a0, string_ADD
50     li    $v0, 4                # specify Print String service
51     syscall                    # print heading
52     move  $a0, $t0             #
53     li    $v0, 1                # specify Print Integer service
54     syscall                    # print $t0: ADD
55
56
57     la    $a0, string_SUB
58     li    $v0, 4                # load address of print heading
59     syscall                    # specify Print String service
60     move  $a0, $t1             # print heading
61     li    $v0, 1                #
62     syscall                    # specify Print Integer service
63                                # print $t1: SUB
64
65     la    $a0, string_AND
66     li    $v0, 4                # load address of print heading
67     syscall                    # specify Print String service
68     move  $a0, $t2             # print heading
69     li    $v0, 1                #
70     syscall                    # specify Print Integer service
71                                # print $t2: AND
72
73     la    $a0, string_OR
74     li    $v0, 4                # load address of print heading
75     syscall                    # specify Print String service
76     move  $a0, $t3             # print heading
77     li    $v0, 1                #
78     syscall                    # specify Print Integer service
79                                # print $t3: OR
80
81     la    $a0, string_XOR
82     li    $v0, 4                # load address of print heading
83     syscall                    # specify Print String service
84     move  $a0, $t4             # print heading
85     li    $v0, 1                #
86     syscall                    # specify Print Integer service
87                                # print $t4: XOR
88
89     la    $a0, string_SLL
90     li    $v0, 4                # load address of print heading
91     syscall                    # specify Print String service
92     move  $a0, $t5             # print heading
93     li    $v0, 1                #
94     syscall                    # specify Print Integer service
95                                # print $t5: SLL

```

Para execução, siga a ordem destes dos ícones a seguir:



E agora, digite os dois valores no console e logo terá a resposta:

```
9
2
ADD: 11
SUB: 7
AND: 0
OR: 11
XOR: 11
SLL(3): 72
SRL(1): 1
```