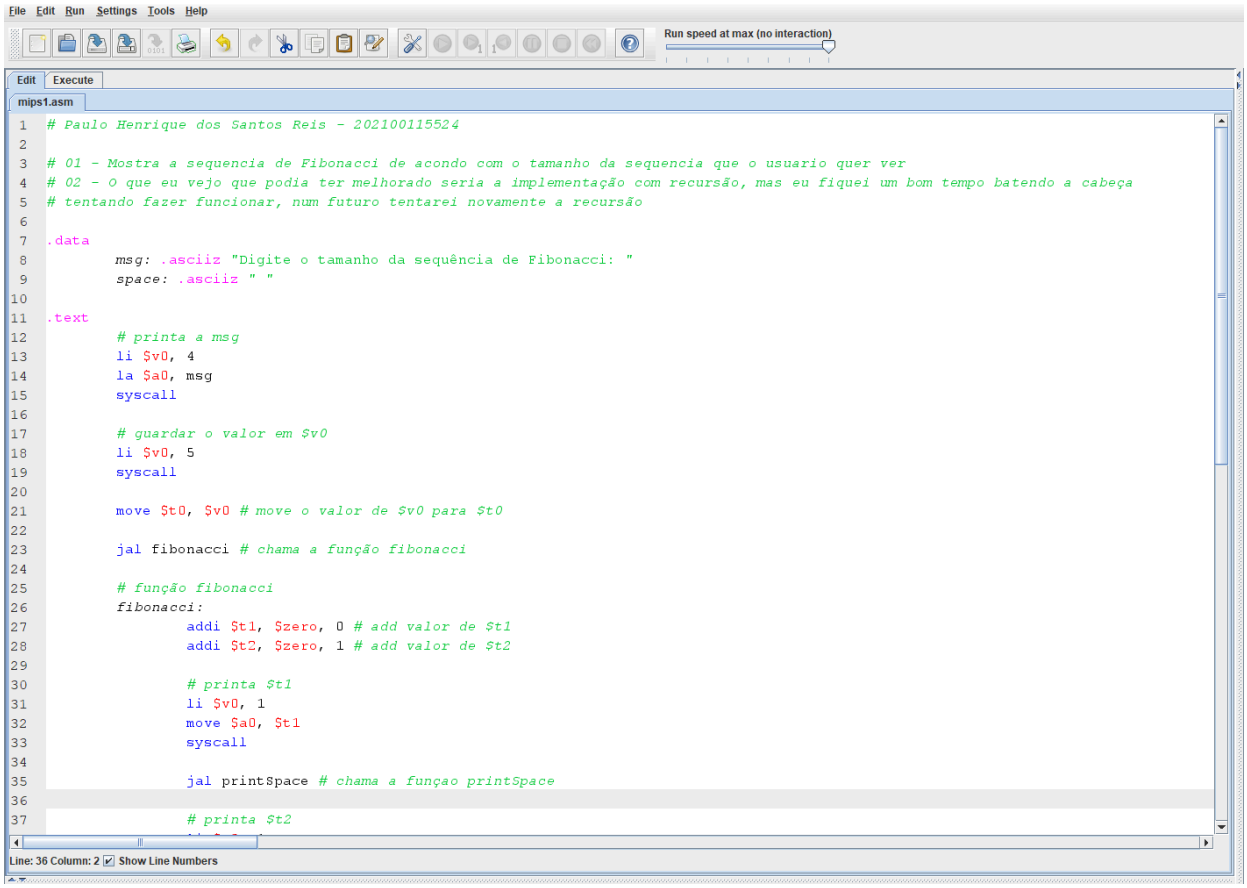
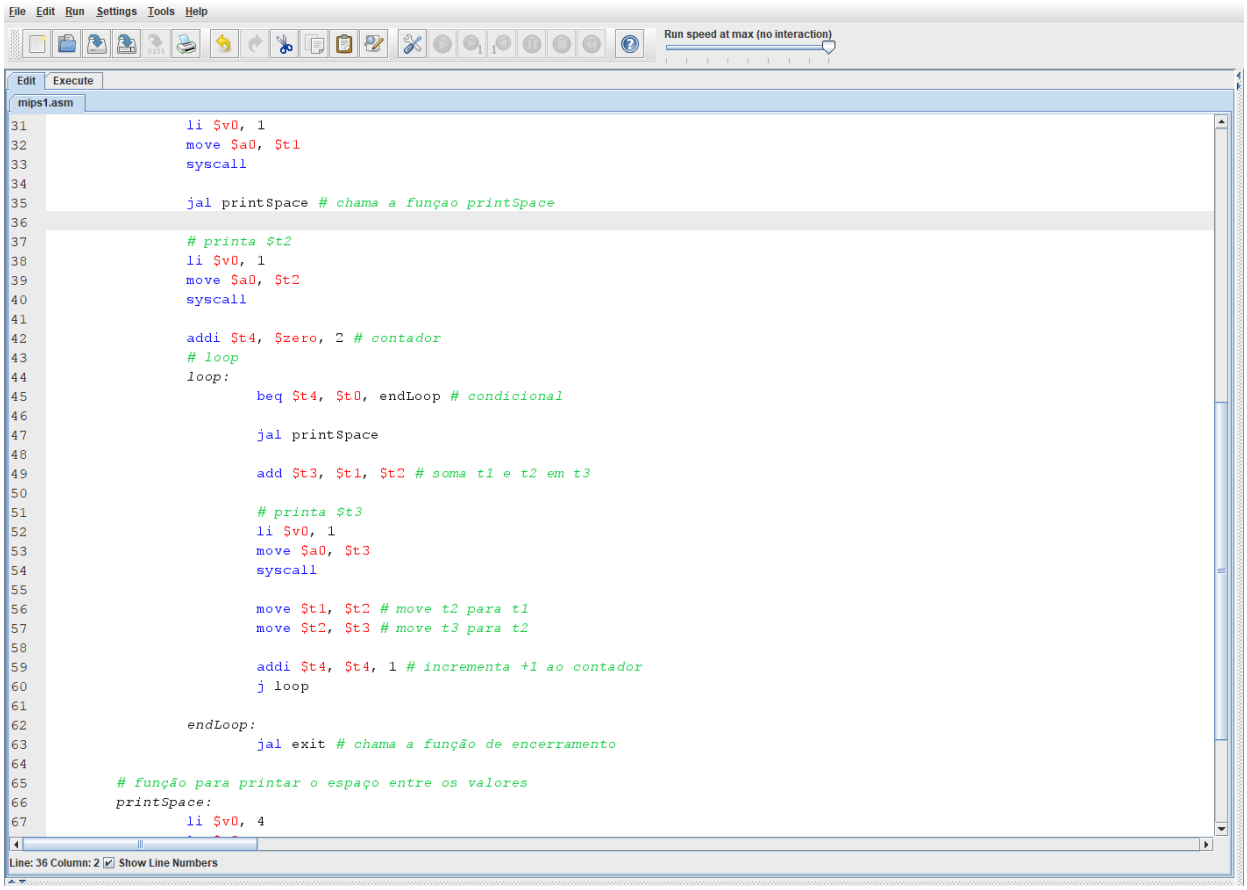
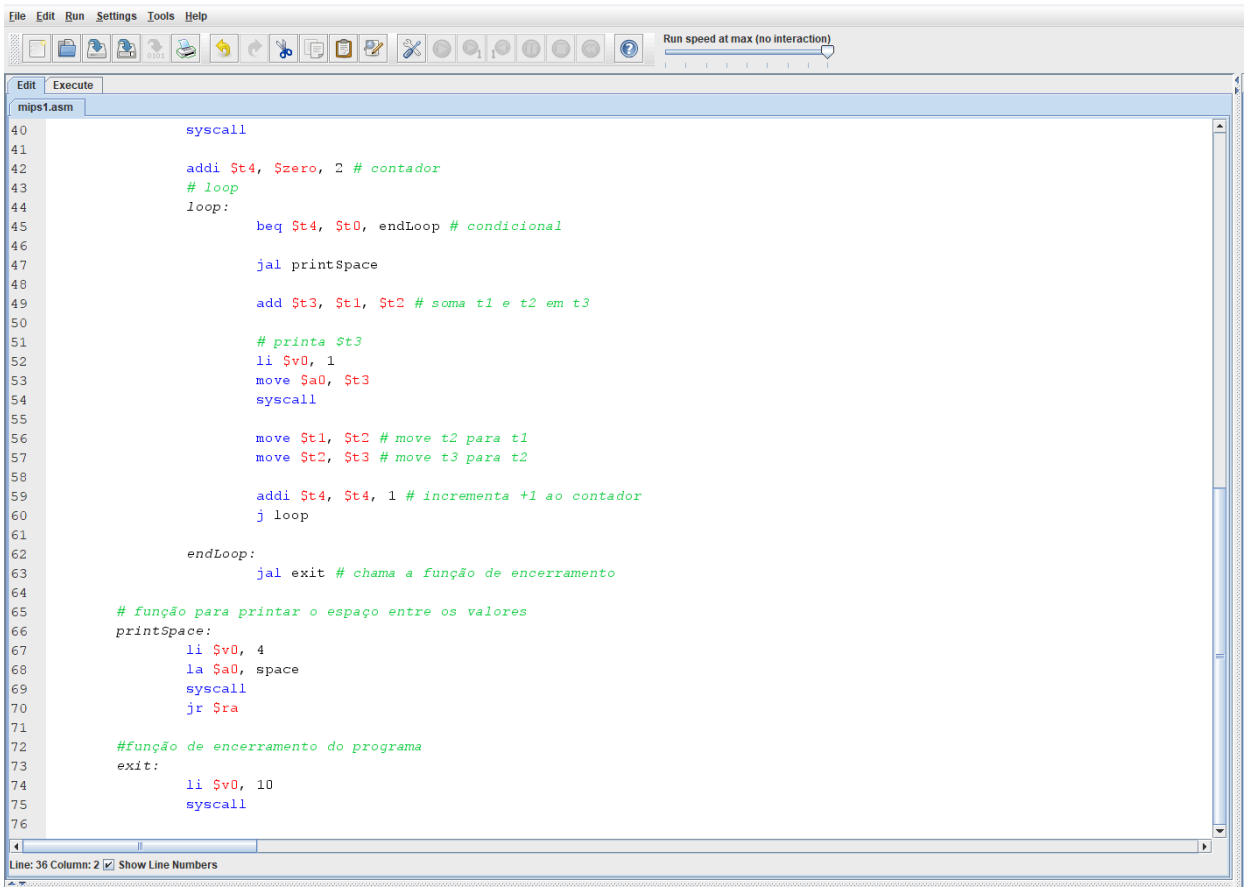

File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips1.asm
1 # Paulo Henrique dos Santos Reis - 202100115524
2
3 # 01 - Mostra a sequencia de Fibonacci de acondo com o tamanho da sequencia que o usuario quer ver
4 # 02 - O que eu vejo que podia ter melhorado seria a implementação com recursão, mas eu fiquei um bom tempo batendo a cabeça
5 # tentando fazer funcionar, num futuro tentarei novamente a recursão
6
7 .data
8 msg: .asciiz "Digite o tamanho da sequência de Fibonacci: "
9 space: .asciiz " "
10
11 .text
12 # printa a msg
13 li $v0, 4
14 la $a0, msg
15 syscall
16
17 # guardar o valor em $v0
18 li $v0, 5
19 syscall
20
21 move $t0, $v0 # move o valor de $v0 para $t0
22
23 jal fibonacci # chama a função fibonacci
24
25 # função fibonacci
26 fibonacci:
27 addi $t1, $zero, 0 # add valor de $t1
28 addi $t2, $zero, 1 # add valor de $t2
29
30 # printa $t1
31 li $v0, 1
32 move $a0, $t1
33 syscall
34
35 jal printSpace # chama a funçao printSpace
36
37 # printa $t2
Line: 36 Column: 2 Show Line Numbers


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips1.asm
31 li $v0, 1
32 move $a0, $t1
33 syscall
34
35 jal printSpace # chama a funçao printSpace
36
37 # printa $t2
38 li $v0, 1
39 move $a0, $t2
40 syscall
41
42 addi $t4, $zero, 2 # contador
43 # loop
44 loop:
45 beq $t4, $t0, endLoop # condicional
46
47 jal printSpace
48
49 add $t3, $t1, $t2 # soma t1 e t2 em t3
50
51 # printa $t3
52 li $v0, 1
53 move $a0, $t3
54 syscall
55
56 move $t1, $t2 # move t2 para t1
57 move $t2, $t3 # move t3 para t2
58
59 addi $t4, $t4, 1 # incrementa +1 ao contador
60 j loop
61
62 endLoop:
63 jal exit # chama a função de encerramento
64
65 # função para printar o espaço entre os valores
66 printSpace:
67 li $v0, 4
Line: 36 Column: 2 Show Line Numbers


File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
mips1.asm
40 syscall
41
42 addi $t4, $zero, 2 # contador
43 # loop
44 loop:
45 beq $t4, $t0, endLoop # condicional
46
47 jal printSpace
48
49 add $t3, $t1, $t2 # soma t1 e t2 em t3
50
51 # printa $t3
52 li $v0, 1
53 move $a0, $t3
54 syscall
55
56 move $t1, $t2 # move t2 para t1
57 move $t2, $t3 # move t3 para t2
58
59 addi $t4, $t4, 1 # incrementa +1 ao contador
60 j loop
61
62 endLoop:
63 jal exit # chama a função de encerramento
64
65 # função para printar o espaço entre os valores
66 printSpace:
67 li $v0, 4
68 la $a0, space
69 syscall
70 jr $ra
71
72 #função de encerramento do programa
73 exit:
74 li $v0, 10
75 syscall
76
Line: 36 Column: 2 Show Line Numbers

Edit | Execute

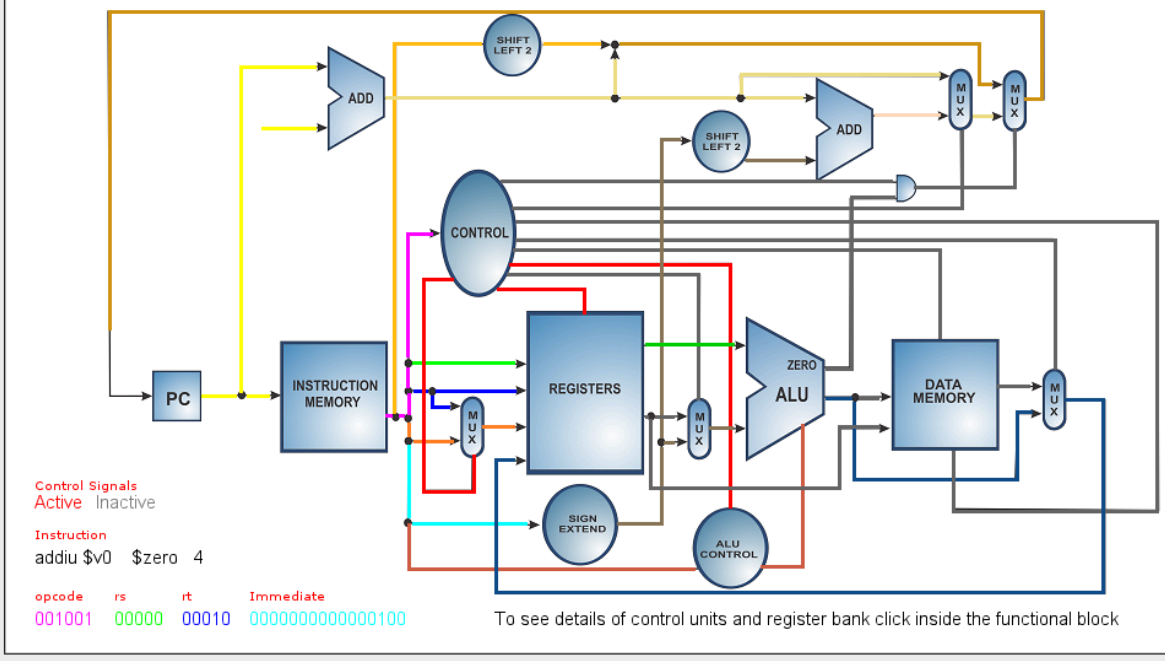
Text Segment

| Bkpt | Address | Code | Basic | | Source |
|---|---|---|---|---|---|
| | 0x00400000 | 0x24020004 | addiu $2,$0,4 | 8: | li $v0, 4 |
| | 0x00400004 | 0x3c011001 | lui $1,4097 | 9: | la $a0, msg |
| | 0x00400008 | 0x34240000 | ori $4,$1,0 | | |
| | 0x0040000c | 0x0000000c | syscall | 10: | syscall |
| | 0x00400010 | 0x24020005 | addiu $2,$0,5 | 13: | li $v0, 5 |
| | 0x00400014 | 0x0000000c | syscall | 14: | syscall |
| | 0x00400018 | 0x00024021 | addu $8,$0,$2 | 16: | move $t0, $v0 # move o valor de $v0 para $t0 |
| | 0x0040001c | 0x0c100008 | jal 0x00400020 | 18: | jal fibonacci # chama a função fibonacci |
| | 0x00400020 | 0x20090000 | addi $9,$0,0 | 22: | addi $t1, $zero, 0 # add valor de $t1 |
| | 0x00400024 | 0x200a0001 | addi $10,$0,1 | 23: | addi $t2, $zero, 1 # add valor de $t2 |
| | 0x00400028 | 0x24020001 | addiu $2,$0,1 | 26: | li $v0, 1 |
| | 0x0040002c | 0x00092021 | addu $4,$0,$9 | 27: | move $a0, $t1 |
| | 0x00400030 | 0x0000000c | syscall | 28: | syscall |
| | 0x00400034 | 0x0c10001d | jal 0x00400074 | 30: | jal printSpace # chama a funçao printSpace |
| | 0x00400038 | 0x24020001 | addiu $2,$0,1 | 33: | li $v0, 1 |
| | 0x0040003c | 0x000a2021 | addu $4,$0,$10 | 34: | move $a0, $t2 |
| | 0x00400040 | 0x0000000c | syscall | 35: | syscall |
| | 0x00400044 | 0x200c0002 | addi $12,$0,2 | 37: | addi $t4, $zero, 2 # contador |
| | 0x00400048 | 0x11880009 | beq $12,$8,9 | 40: | beq $t4, $t0, endLoop # condicional |
| | 0x0040004c | 0x0c10001d | jal 0x00400074 | 42: | jal printSpace |
| | 0x00400050 | 0x012a5820 | add $11,$9,$10 | 44: | add $t3, $t1, $t2 # soma t1 e t2 em t3 |
| | 0x00400054 | 0x24020001 | addiu $2,$0,1 | 47: | li $v0, 1 |
| | 0x00400058 | 0x000b2021 | addu $4,$0,$11 | 48: | move $a0, $t3 |
| | 0x0040005c | 0x0000000c | syscall | 49: | syscall |
| | 0x00400060 | 0x000a4821 | addu $9,$0,$10 | 51: | move $t1, $t2 # move t2 para t1 |
| | 0x00400064 | 0x000b5021 | addu $10,$0,$11 | 52: | move $t2, $t3 # move t3 para t2 |
| | 0x00400068 | 0x218c0001 | addi $12,$12,1 | 54: | addi $t4, $t4, 1 # incrementa +1 ao contador |
| | 0x0040006c | 0x08100012 | j 0x00400048 | 55: | j loop |
| | 0x00400070 | 0x0c100022 | jal 0x00400088 | 58: | jal exit # chama a função de encerramento |
| | 0x00400074 | 0x24020004 | addiu $2,$0,4 | 62: | li $v0, 4 |
| | 0x00400078 | 0x3c011001 | lui $1,4097 | 63: | la $a0, space |
| | 0x0040007c | 0x3424002d | ori $4,$1,45 | | |
| | 0x00400080 | 0x0000000c | syscall | 64: | syscall |
| | 0x00400084 | 0x03e00008 | jr $31 | 65: | jr $ra |
| | 0x00400088 | 0x2402000a | addiu $2,$0,10 | 69: | li $v0, 10 |
| | 0x0040008c | 0x0000000c | syscall | 70: | syscall |

MIPS X-Ray - Animation of MIPS Datapath, Version 2.0

## MIPS X-Ray - Animation of MIPS Datapath

### IMMEDIATE TYPE INSTRUCTION




Control Signals
Active Inactive

Instruction
addiu $v0 $zero 4

| opcode | rs | rt | Immediate |
|---|---|---|---|
| 001001 | 00000 | 00010 | 0000000000000100 |

To see details of control units and register bank click inside the functional block

Tool Control

Disconnect from MIPS | Reset | Help | Close