



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

THAYANE STHEFFANY SILVA BARROS

ROTEIRO 6

OBS: As imagens também estão disponíveis no outro arquivo enviado no arquivo compactado

Problema 1 - Código em Assembly

Editor Simulator

```
1 .text
2 main:
3     addi a0, zero, 1
4     add a1, a0, a0
5     addi a2, zero, 5
6 loop:
7     beq a0, a2, fim
8     slli a1, a1, 1
9     addi a0, a0, 1
10    j loop
11 fim:
12    nop
```

a) Qual é a operação realizada pelo código acima?

A operação realizada pelo código é um loop que multiplica o valor em a1 por 2 (slli a1, a1, 1) enquanto o valor em a0 é incrementado (addi a0, a0, 1) até que a0 seja igual a 5 (beq a0, a2, fim).

b) Qual é o conteúdo (em decimal) dos registradores a0, a1 e a2 ao final da execução do programa?

Ao final da execução do programa, o conteúdo dos registradores é o seguinte:

- a0: 5 (decimal)
- a1: 32 (decimal)
- a2: 5 (decimal)

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10) 5

a1 (x11) 32

a2 (x12) 5

a3 (x13) 0

a4 (x14) 0

a5 (x15) 0

a6 (x16) 0

a7 (x17) 0

Display Settings Decimal

c) Quais são os valores de loop (na instrução j loop) e fim (na instrução beq a0, a2, fim)

Os valores de "loop" e "fim" são rótulos (endereços de memória) para as instruções correspondentes. Portanto, eles não possuem valores numéricos específicos.

No entanto, em relação à posição de memória podemos determinar que o valor de "loop" é -12 (ou 0xff5ff06f em hexadecimal). Isso pode ser visto na instrução "jal x0 -12", onde o deslocamento é um valor negativo que representa a diferença entre o endereço atual da instrução e o endereço da instrução rotulada "loop". Nesse caso, o deslocamento de -12 indica que o endereço da instrução "loop" é 12 bytes (3 instruções) atrás do endereço atual.

O valor de "fim" é 16 (ou 0x00c50863 em hexadecimal). Isso pode ser visto na instrução "beq x10 x12 16", onde o deslocamento é um valor positivo que representa a diferença entre o endereço atual da instrução e o endereço da instrução rotulada "fim". Nesse caso, o deslocamento de 16 indica que o endereço da instrução "fim" é 16 bytes (4 instruções) à frente do endereço atual.

Portanto:

- O valor de "loop" é -12.
- O valor de "fim" é 16.

d) Adicione "prints" da tela do simulador ao arquivo PDF. Esses "prints" deverão mostrar todo o ambiente, incluindo o código em Assembly que foi executado e os valores exibidos na interface após a execução de cada instrução (código de máquina armazenado na memória, valores do pc e dos demais registradores envolvidos).

Editor

Simulator

Run

Step

Prev

Reset

Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Valores em hexadecimal

Valores em decimal

a0 (x10)

0x00000000

0

a1 (x11)

0x00000000

0

a2 (x12)

0x00000000

0

a3 (x13)

0x00000000

0

a4 (x14)

0x00000000

0

a5 (x15)

0x00000000

0

a6 (x16)

0x00000000

0

a7 (x17)

0x00000000

0

Display Settings

Hex

Display Settings

Decimal

- Registrador a0 é carregado com o valor 1 (addi a0, zero, 1).

Valores em hexadecimal
Valores em decimal

Editor Simulator

Run
Step
Prev
Reset
Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10)

a1 (x11)

a2 (x12)

a3 (x13)

a4 (x14)

a5 (x15)

a6 (x16)

a7 (x17)

Display Settings Hex

a0 (x10)

a1 (x11)

a2 (x12)

a3 (x13)

a4 (x14)

a5 (x15)

a6 (x16)

a7 (x17)

Display Settings Decimal

- Registrador a1 é carregado com o valor em a0 adicionado a a0, resultando em 2 (add a1, a0, a0).

Valores em hexadecimal
Valores em decimal

Editor Simulator

Run
Step
Prev
Reset
Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10)

a1 (x11)

a2 (x12)

a3 (x13)

a4 (x14)

a5 (x15)

a6 (x16)

a7 (x17)

Display Settings Hex

a0 (x10)

a1 (x11)

a2 (x12)

a3 (x13)

a4 (x14)

a5 (x15)

a6 (x16)

a7 (x17)

Display Settings Decimal

- Registrador a2 é carregado com o valor 5 (addi a2, zero, 5).

Valores em hexadecimal
Valores em decimal

Editor Simulator

Run
Step
Prev
Reset
Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10)

a1 (x11)

a2 (x12)

a3 (x13)

a4 (x14)

a5 (x15)

a6 (x16)

a7 (x17)

Display Settings Hex

a0 (x10)

a1 (x11)

a2 (x12)

a3 (x13)

a4 (x14)

a5 (x15)

a6 (x16)

a7 (x17)

Display Settings Decimal

- Verifica se o valor em a0 é igual ao valor em a2. Como não são (a0 = 1 e a2 = 5) o programa continua a execução (beq a0, a2, fim).

Valores em hexadecimal

Valores em decimal

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Display Settings: Hex

Display Settings: Decimal

Register values (Hexadecimal):

- a0 (x10): 0x00000001
- a1 (x11): 0x00000002
- a2 (x12): 0x00000005
- a3 (x13): 0x00000000
- a4 (x14): 0x00000000
- a5 (x15): 0x00000000
- a6 (x16): 0x00000000
- a7 (x17): 0x00000000

Register values (Decimal):

- a0 (x10): 1
- a1 (x11): 2
- a2 (x12): 5
- a3 (x13): 0
- a4 (x14): 0
- a5 (x15): 0
- a6 (x16): 0
- a7 (x17): 0

- O valor em a1 é deslocado uma posição à esquerda, multiplicando-o por 2 (slli a1, a1, 1).

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Display Settings: Hex

Display Settings: Decimal

Register values (Hexadecimal):

- a0 (x10): 0x00000001
- a1 (x11): 0x00000004
- a2 (x12): 0x00000005
- a3 (x13): 0x00000000
- a4 (x14): 0x00000000
- a5 (x15): 0x00000000
- a6 (x16): 0x00000000
- a7 (x17): 0x00000000

Register values (Decimal):

- a0 (x10): 1
- a1 (x11): 4
- a2 (x12): 5
- a3 (x13): 0
- a4 (x14): 0
- a5 (x15): 0
- a6 (x16): 0
- a7 (x17): 0

- O valor em a0 é incrementado em 1 (addi a0, a0, 1).

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Display Settings: Hex

Display Settings: Decimal

Register values (Hexadecimal):

- a0 (x10): 0x00000002
- a1 (x11): 0x00000004
- a2 (x12): 0x00000005
- a3 (x13): 0x00000000
- a4 (x14): 0x00000000
- a5 (x15): 0x00000000
- a6 (x16): 0x00000000
- a7 (x17): 0x00000000

Register values (Decimal):

- a0 (x10): 2
- a1 (x11): 4
- a2 (x12): 5
- a3 (x13): 0
- a4 (x14): 0
- a5 (x15): 0
- a6 (x16): 0
- a7 (x17): 0

- O programa salta de volta para a instrução loop para repetir o processo.

Valores em hexadecimal
Valores em decimal

Editor
Simulador

Run
Step
Prev
Reset
Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output:

Display Settings: Hex

a0 (x10)	0x00000002
a1 (x11)	0x00000004
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Decimal

a0 (x10)	2
a1 (x11)	4
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

- Verifica-se novamente se o valor em a0 é igual ao valor em a2. Como não são (a0 = 2 e a2 = 5) o programa continua a execução (beq a0, a2, fim).

Valores em hexadecimal
Valores em decimal

Editor
Simulador

Run
Step
Prev
Reset
Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output:

Display Settings: Hex

a0 (x10)	0x00000002
a1 (x11)	0x00000004
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Decimal

a0 (x10)	2
a1 (x11)	4
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

- O valor em a1 é deslocado uma posição à esquerda, multiplicando-o por 2 (slli a1, a1, 1).

Valores em hexadecimal
Valores em decimal

Editor
Simulador

Run
Step
Prev
Reset
Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output:

Display Settings: Hex

a0 (x10)	0x00000002
a1 (x11)	0x00000008
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Decimal

a0 (x10)	2
a1 (x11)	8
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

- O valor em a0 é incrementado em 1 (addi a0, a0, 1). a0 agora tem o valor 3.

Valores em hexadecimal
Valores em decimal

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50063	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Hex

a0 (x10)	3
a1 (x11)	0
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

Display Settings: Decimal

- O programa salta de volta para a instrução loop para repetir o processo.

Valores em hexadecimal
Valores em decimal

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50063	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Hex

a0 (x10)	3
a1 (x11)	0
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

Display Settings: Decimal

- Verifica-se novamente se o valor em a0 é igual ao valor em a2. Como não são (a0 = 3 e a2 = 5) o programa continua a execução (beq a0, a2, fim).

Valores em hexadecimal
Valores em decimal

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50063	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10)	0x00000003
a1 (x11)	0x00000000
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Hex

a0 (x10)	3
a1 (x11)	0
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

Display Settings: Decimal

- O valor em a1 é deslocado uma posição à esquerda, multiplicando-o por 2 (slli a1, a1, 1).

Valores em hexadecimal Valores em decimal

Editor **Simulador**

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10) 0x00000003

a1 (x11) 0x00000010

a2 (x12) 0x00000005

a3 (x13) 0x00000000

a4 (x14) 0x00000000

a5 (x15) 0x00000000

a6 (x16) 0x00000000

a7 (x17) 0x00000000

Display Settings Hex

a0 (x10) 3

a1 (x11) 16

a2 (x12) 5

a3 (x13) 0

a4 (x14) 0

a5 (x15) 0

a6 (x16) 0

a7 (x17) 0

Display Settings Decimal

- O valor em a0 é incrementado em 1 (addi a0, a0, 1). a0 agora tem o valor 4.

Editor **Simulador**

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10) 0x00000004

a1 (x11) 0x00000016

a2 (x12) 0x00000005

a3 (x13) 0x00000000

a4 (x14) 0x00000000

a5 (x15) 0x00000000

a6 (x16) 0x00000000

a7 (x17) 0x00000000

Display Settings Hex

a0 (x10) 4

a1 (x11) 16

a2 (x12) 5

a3 (x13) 0

a4 (x14) 0

a5 (x15) 0

a6 (x16) 0

a7 (x17) 0

Display Settings Decimal

- O programa salta de volta para a instrução loop para repetir o processo.

Editor **Simulador**

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10) 0x00000004

a1 (x11) 0x00000016

a2 (x12) 0x00000005

a3 (x13) 0x00000000

a4 (x14) 0x00000000

a5 (x15) 0x00000000

a6 (x16) 0x00000000

a7 (x17) 0x00000000

Display Settings Hex

a0 (x10) 4

a1 (x11) 16

a2 (x12) 5

a3 (x13) 0

a4 (x14) 0

a5 (x15) 0

a6 (x16) 0

a7 (x17) 0

Display Settings Decimal

- Verifica-se novamente se o valor em a0 é igual ao valor em a2. Como não são (a0 = 4 e a2 = 5) o programa continua a execução (beq a0, a2, fim).

Editor **Simulator**

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xfffff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Valores em hexadecimal

a0 (x10)	0x00000004
a1 (x11)	0x00000010
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Hex

Valores em decimal

a0 (x10)	4
a1 (x11)	16
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

Display Settings: Decimal

- O valor em a1 é deslocado uma posição à esquerda, multiplicando-o por 2 (slli a1, a1, 1).

Editor **Simulator**

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xfffff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Valores em hexadecimal

a0 (x10)	0x00000004
a1 (x11)	0x00000020
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Hex

Valores em decimal

a0 (x10)	4
a1 (x11)	32
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

Display Settings: Decimal

- O valor em a0 é incrementado em 1 (addi a0, a0, 1). a0 agora tem o valor 5.

Editor **Simulator**

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00150593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xfffff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Valores em hexadecimal

a0 (x10)	0x00000005
a1 (x11)	0x00000020
a2 (x12)	0x00000005
a3 (x13)	0x00000000
a4 (x14)	0x00000000
a5 (x15)	0x00000000
a6 (x16)	0x00000000
a7 (x17)	0x00000000

Display Settings: Hex

Valores em decimal

a0 (x10)	5
a1 (x11)	32
a2 (x12)	5
a3 (x13)	0
a4 (x14)	0
a5 (x15)	0
a6 (x16)	0
a7 (x17)	0

Display Settings: Decimal

- O programa salta de volta para a instrução loop para repetir o processo. Verifica se o valor em a0 é igual ao valor em a2. Como são iguais (a0 = 5 e a2 = 5), o programa salta para a instrução fim.

Valores em hexadecimal Valores em decimal

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10) 0x00000005
a1 (x11) 0x00000029
a2 (x12) 0x00000005
a3 (x13) 0x00000000
a4 (x14) 0x00000000
a5 (x15) 0x00000000
a6 (x16) 0x00000000
a7 (x17) 0x00000000

a0 (x10) 5
a1 (x11) 32
a2 (x12) 5
a3 (x13) 0
a4 (x14) 0
a5 (x15) 0
a6 (x16) 0
a7 (x17) 0

Display Settings: Hex Display Settings: Decimal

- A instrução nop não realiza nenhuma operação. É usada como uma instrução vazia para fins de controle do fluxo de execução.

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

a0 (x10) 0x00000005
a1 (x11) 0x00000029
a2 (x12) 0x00000005
a3 (x13) 0x00000000
a4 (x14) 0x00000000
a5 (x15) 0x00000000
a6 (x16) 0x00000000
a7 (x17) 0x00000000

a0 (x10) 5
a1 (x11) 32
a2 (x12) 5
a3 (x13) 0
a4 (x14) 0
a5 (x15) 0
a6 (x16) 0
a7 (x17) 0

Display Settings: Hex Display Settings: Decimal

- Print da memória

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00100513	addi x10 x0 1	addi a0, zero, 1
0x00a505b3	add x11 x10 x10	add a1, a0, a0
0x00500613	addi x12 x0 5	addi a2, zero, 5
0x00c50863	beq x10 x12 16	beq a0, a2, fim
0x00159593	slli x11 x11 1	slli a1, a1, 1
0x00150513	addi x10 x10 1	addi a0, a0, 1
0xff5ff06f	jal x0 -12	j loop
0x00000013	addi x0 x0 0	nop

console output

Registers Memory

Address	+0	+1	+2	+3
0x00000018	6f	f0	5f	ff
0x00000014	13	05	15	00
0x00000010	93	95	15	00
0x0000000c	63	08	c5	00
0x00000008	13	06	50	00
0x00000004	b3	05	a5	00
0x00000000	13	05	10	00
-----	--	--	--	--
-----	--	--	--	--
-----	--	--	--	--
-----	--	--	--	--

Display Settings: Hex

Problema 2 - Código de Máquina

a) Qual é o código em Assembly correspondente?

Instruções em hexadecimal:	Instruções em Binário	Instrução em RV32I
0x00200513 :	0000 0000 0010 0000 0000 0101 0001 0011 :	ADDI
0x00400593 :	0000 0000 0100 0000 0000 0101 1001 0011 :	ADDI
0x00b50663 :	0000 0000 1011 0101 0000 0110 0110 0011 :	BEQ
0x00a50533 :	0000 0000 1010 0101 0000 0101 0011 0011 :	ADD
0xff9ff06f :	1111 1111 1001 1111 1111 0000 0110 1111 :	JAL
0x00b58633 :	0000 0000 1011 0101 1000 0110 0011 0011 :	ADD
0x00000013 :	0000 0000 0000 0000 0000 0000 0001 0011 :	NOP

Montando o código através das instruções em binário

```
0000 0000 0010 0000 0000 0101 0001 0011
      2          0          10 (a0)          : ADDI X10, X0, 2

0000 0000 0100 0000 0000 0101 1001 0011
      4          0          11 (a1)          : ADDI X11, X0, 4

0000 0000 1011 0101 0000 0110 0110 0011
      11         10         - 0000 0000 0110 0 = 12 : BEQ X10, X11, 12

0000 0000 1010 0101 0000 0101 0011 0011
      10         10          10              : ADD X10, X10, X10

1111 1111 1001 1111 1111 0000 0110 1111
      . 1111 1111 1111 1111 1100 = - 4      : JAL X0, -4

0000 0000 1011 0101 1000 0110 0011 0011
      11         11          12              : ADD X12, X11, X11

0000 0000 0000 0000 0000 0000 0001 0011 : NOP
```

O código convertido para a linguagem assembly RISC-V:

```
.text
main:
    ADDI a0, zero, 2
    ADDI a1, zero, 4
loop:
    BEQ a0, a1, fim
    ADD a0, a0, a0
    JAL zero, loop
fim:
    ADD a2, a1, a1
    NOP
```

b) Qual é a operação realizada pelo código acima?

É um loop que multiplica o valor em a0 por 2 repetidamente até que a0 seja igual a a1. Em cada iteração, o valor em a0 é duplicado pela instrução ADD a0, a0, a0, e então o programa salta de volta para o rótulo "loop" usando a instrução JAL zero, loop.

c) Quais são os registradores utilizados no código?

Os registradores utilizados no código são zero, a0, a1 e a2.

d) Quais são os conteúdos desses registradores ao final da execução do programa

- O conteúdo do registrador a0 será igual ao conteúdo inicial do resgistrador a1 ou seja 4.
- O conteúdo do registrador a1 permanecerá o mesmo (4).
- O conteúdo do registrador a2 será igual a 2*a1, ou seja, 8.

Editor

Simulator

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jal x0 -8	JAL zero, loop
0x00b50633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

console output

a0
(x10)

0x00000004

a1
(x11)

0x00000004

a2
(x12)

0x00000008

a3
(x13)

0x00000000

a4
(x14)

0x00000000

a5
(x15)

0x00000000

a6
(x16)

0x00000000

a7
(x17)

0x00000000

Display
Settings

Hex

e) Adicione "prints" da tela do simulador ao arquivo PDF. Esses "prints" deverão mostrar todo o ambiente, incluindo o código em Assembly que foi executado e os valores exibidos na interface após a execução de cada instrução (código de máquina armazenado na memória, valores do pc e dos demais registradores envolvidos).

Valores em hexadecimal

Valores em decimal

The screenshot shows the RISC-V simulator interface. The 'Editor' tab is active, displaying a list of instructions. The first instruction, 'addi x10 x0 2', is highlighted in green. The 'Run' button is also highlighted. The 'console output' area is empty. On the right, the register values are displayed in hexadecimal (0x00000000) and decimal (0).

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jal x0 -8	JAL zero, loop
0x00b50633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

- Registrador a0 é carregado com o valor 2 (ADDI a0, zero, 2).

The screenshot shows the RISC-V simulator interface. The second instruction, 'addi x11 x0 4', is highlighted in green. The 'Run' button is also highlighted. The 'console output' area is empty. On the right, the register values are displayed in hexadecimal (0x00000002 for a0 and 0x00000004 for a1) and decimal (2 for a0 and 4 for a1).

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jal x0 -8	JAL zero, loop
0x00b50633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

- Registrador a1 é carregado com o valor 4 (ADDI a1, zero, 4).

The screenshot shows the RISC-V simulator interface. The third instruction, 'beq x10 x11 12', is highlighted in green. The 'Run' button is also highlighted. The 'console output' area is empty. On the right, the register values are displayed in hexadecimal (0x00000002 for a0 and 0x00000004 for a1) and decimal (2 for a0 and 4 for a1).

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jal x0 -8	JAL zero, loop
0x00b50633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

- O programa entra em um loop. Verifica se o valor em a0 é igual ao valor em a1 usando a instrução BEQ a0, a1, fim.
- Como são diferentes (a0 = 2 e a1 = 4), o programa executa a instrução ADD a0, a0, a0 para multiplicar o valor em a0 por 2.

Valores em hexadecimal Valores em decimal

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jnl x0 -8	JAL zero, loop
0x00b58633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

console output

Display Settings Hex Display Settings Decimal

a0 (x10) 0x00000002 a0 (x10) 2

a1 (x11) 0x00000004 a1 (x11) 4

a2 (x12) 0x00000000 a2 (x12) 0

a3 (x13) 0x00000000 a3 (x13) 0

a4 (x14) 0x00000000 a4 (x14) 0

a5 (x15) 0x00000000 a5 (x15) 0

a6 (x16) 0x00000000 a6 (x16) 0

a7 (x17) 0x00000000 a7 (x17) 0

- Registrador a0 é carregado com o valor 4 (ADD a0, a0, a0).

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jnl x0 -8	JAL zero, loop
0x00b58633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

console output

Display Settings Hex Display Settings Decimal

a0 (x10) 0x00000004 a0 (x10) 4

a1 (x11) 0x00000004 a1 (x11) 4

a2 (x12) 0x00000000 a2 (x12) 0

a3 (x13) 0x00000000 a3 (x13) 0

a4 (x14) 0x00000000 a4 (x14) 0

a5 (x15) 0x00000000 a5 (x15) 0

a6 (x16) 0x00000000 a6 (x16) 0

a7 (x17) 0x00000000 a7 (x17) 0

- O programa salta de volta para a instrução loop para repetir o processo usando a instrução JAL zero, loop

Editor Simulator

Run Step Prev Reset Dump

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jnl x0 -8	JAL zero, loop
0x00b58633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

console output

Display Settings Hex Display Settings Decimal

a0 (x10) 0x00000008 a0 (x10) 8

a1 (x11) 0x00000004 a1 (x11) 4

a2 (x12) 0x00000000 a2 (x12) 0

a3 (x13) 0x00000000 a3 (x13) 0

a4 (x14) 0x00000000 a4 (x14) 0

a5 (x15) 0x00000000 a5 (x15) 0

a6 (x16) 0x00000000 a6 (x16) 0

a7 (x17) 0x00000000 a7 (x17) 0

- O programa salta de volta para a instrução loop para repetir o processo. Verifica se o valor em a0 é igual ao valor em a1. Como são iguais (a0 = 4 e a1 = 4), o programa salta para a instrução fim.

Valores em hexadecimal
Valores em decimal

The screenshot shows the RISC-V simulator interface. The 'Editor' tab is active, displaying a table of machine code, basic code, and original code. The 'Simulator' tab is also visible. The registers are displayed in two columns: hexadecimal and decimal. The values for registers a0 through a7 are shown.

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jal x0 -8	JAL zero, loop
0x00b50633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

Registers (Hexadecimal):

- a0 (x10): 0x00000004
- a1 (x11): 0x00000004
- a2 (x12): 0x00000000
- a3 (x13): 0x00000000
- a4 (x14): 0x00000000
- a5 (x15): 0x00000000
- a6 (x16): 0x00000000
- a7 (x17): 0x00000000

Registers (Decimal):

- a0 (x10): 4
- a1 (x11): 4
- a2 (x12): 0
- a3 (x13): 0
- a4 (x14): 0
- a5 (x15): 0
- a6 (x16): 0
- a7 (x17): 0

- Quando o programa salta para o rótulo "fim", executa a instrução ADD a2, a1, a1 para duplicar o valor em a1 e armazena o resultado em a2.
- Registrador a2 é carregado com o valor 8.

The screenshot shows the RISC-V simulator interface. The 'Editor' tab is active, displaying a table of machine code, basic code, and original code. The 'Simulator' tab is also visible. The registers are displayed in two columns: hexadecimal and decimal. The values for registers a0 through a7 are shown.

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jal x0 -8	JAL zero, loop
0x00b50633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

Registers (Hexadecimal):

- a0 (x10): 0x00000004
- a1 (x11): 0x00000004
- a2 (x12): 0x00000008
- a3 (x13): 0x00000000
- a4 (x14): 0x00000000
- a5 (x15): 0x00000000
- a6 (x16): 0x00000000
- a7 (x17): 0x00000000

Registers (Decimal):

- a0 (x10): 4
- a1 (x11): 4
- a2 (x12): 8
- a3 (x13): 0
- a4 (x14): 0
- a5 (x15): 0
- a6 (x16): 0
- a7 (x17): 0

- Finalmente, o programa executa a instrução NOP que não realiza nenhuma operação. É usada como uma instrução vazia para fins de controle do fluxo de execução.
- Print da memória:

The screenshot shows the RISC-V simulator interface. The 'Editor' tab is active, displaying a table of machine code, basic code, and original code. The 'Simulator' tab is also visible. The memory dump is shown in the bottom right corner.

Machine Code	Basic Code	Original Code
0x00200513	addi x10 x0 2	ADDI a0, zero, 2
0x00400593	addi x11 x0 4	ADDI a1, zero, 4
0x00b50663	beq x10 x11 12	BEQ a0, a1, fim
0x00a50533	add x10 x10 x10	ADD a0, a0, a0
0xff9ff06f	jal x0 -8	JAL zero, loop
0x00b50633	add x12 x11 x11	ADD a2, a1, a1
0x00000013	addi x0 x0 0	NOP

Memory Dump:

Address	+0	+1	+2	+3
0x00000018	19	0	0	0
0x00000014	51	-122	-75	0
0x00000010	111	-16	-97	-1
0x0000000c	51	5	-91	0
0x00000008	99	6	-75	0
0x00000004	-109	5	64	0
0x00000000	19	5	32	0
-----	--	--	--	--
-----	--	--	--	--
-----	--	--	--	--
-----	--	--	--	--