

Problema 1 - Fatorial

A exemplificação abaixo será para o cálculo do fatorial de 3.

Código-fonte no simulador:

```
1 # Tarso Jabbes Lima de Oliveira
2 # Problema 1 - Roteiro 7
3
4     addi a1, zero, 3    # Carrega o valor que queremos descobrir o fatorial em a1
5     addi a2, zero, 1    # Carrega 1 em a2 para ser a condição de parada
6     addi a5, zero, 1    # Carrega 1 em a5 para multiplicação efetiva entre termos
7
8 loop: beq a1, a2, fim    # Condição de parada do fatorial que atinge 1
9     beq a1, zero, fim    # Condição de parada para N = 0
10    addi a3, a1, -1      # Carrega em a3 o antecedente de a1
11    mul a4, a1, a3       # Multiplica a1 com o seu antecedente e guarda em a4
12    mul a5, a5, a4       # Multiplica o último valor de a5 com o valor de a4
13    addi a1, a1, -2      # Decrementa a1 em 2 unidades
14    j loop              # Repete a execução do loop
15
16 fim: nop               # Finaliza a execução
```

Ambiente antes da execução das instruções:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00958c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff58693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4

console output

a0 (x10)0

a1 (x11)0

a2 (x12)0

a3 (x13)0

a4 (x14)0

a5 (x15)0

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da primeira instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00958c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff58693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4

console output

a0 (x10)0

a1 (x11)3

a2 (x12)0

a3 (x13)0

a4 (x14)0

a5 (x15)0

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da segunda instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff50693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o anterior de a4

console output

a0 (x10)0

a1 (x11)3

a2 (x12)1

a3 (x13)0

a4 (x14)0

a5 (x15)0

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da terceira instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff50693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o anterior de a4

console output

a0 (x10)0

a1 (x11)3

a2 (x12)1

a3 (x13)0

a4 (x14)0

a5 (x15)1

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da quarta instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff50693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o anterior de a4

console output

a0 (x10)0

a1 (x11)3

a2 (x12)1

a3 (x13)0

a4 (x14)0

a5 (x15)1

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da quinta instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff50693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o anterior de a4

console output

a0 (x10)0

a1 (x11)3

a2 (x12)1

a3 (x13)0

a4 (x14)0

a5 (x15)1

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

Após a execução da sexta instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff58693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4

console output

a0 (x10)0

a1 (x11)3

a2 (x12)1

a3 (x13)2

a4 (x14)0

a5 (x15)1

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da sétima instrução:

RunStepPrevResetDump

0x00300593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff58693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4

console output

a0 (x10)0

a1 (x11)3

a2 (x12)1

a3 (x13)2

a4 (x14)6

a5 (x15)1

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da oitava instrução:

RunStepPrevResetDump

0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff58693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4
0xffe58593	addi x11 x11 -2	addi a1, a1, -2 # Decrementa a1 em 2 unidades

console output

a0 (x10)0

a1 (x11)3

a2 (x12)1

a3 (x13)2

a4 (x14)6

a5 (x15)6

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

s7 (x23)0

s8 (x24)0

Após a execução da nona instrução:

RunStepPrevResetDump

0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff58693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4
0xffe58593	addi x11 x11 -2	addi a1, a1, -2 # Decrementa a1 em 2 unidades
0xfe9ff06f	jal x0 -24	j loop # Repete a execução do loop

console output

a0 (x10)0

a1 (x11)1

a2 (x12)1

a3 (x13)2

a4 (x14)6

a5 (x15)6

a6 (x16)0

a7 (x17)0

s2 (x18)0

s3 (x19)0

s4 (x20)0

s5 (x21)0

s6 (x22)0

Após a execução da décima instrução (há o retorno para a quarta instrução):

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x00309593	addi x11 x0 3	addi a1, zero, 3 # Carrega o valor que queremos descobrir o fatorial em a1
0x00100613	addi x12 x0 1	addi a2, zero, 1 # Carrega 1 em a2 para ser a condição de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff50693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4

console output

a0 (x10)0a1 (x11)1a2 (x12)1a3 (x13)2a4 (x14)6a5 (x15)6a6 (x16)0a7 (x17)0s2 (x18)0s3 (x19)0s4 (x20)0s5 (x21)0s6 (x22)0s7 (x23)0s8 (x24)0

Após a segunda execução da quarta instrução:

RunStepPrevResetDump

		de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff50693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4
0xffe58593	addi x11 x11 -2	addi a1, a1, -2 # Decrementa a1 em 2 unidades
0xfe9ff06f	jal x0 -24	j loop # Repete a execução do loop
0x00000013	addi x0 x0 0	fim: nop # Finaliza a execução

console output

a0 (x10)0a1 (x11)1a2 (x12)1a3 (x13)2a4 (x14)6a5 (x15)6a6 (x16)0a7 (x17)0s2 (x18)0s3 (x19)0s4 (x20)0s5 (x21)0s6 (x22)0s7 (x23)0s8 (x24)0

Após a execução da décima primeira execução (finaliza a execução das instruções):

RunStepPrevResetDump

		de parada
0x00100793	addi x15 x0 1	addi a5, zero, 1 # Carrega 1 em a5 para multiplicação efetiva entre termos
0x00c58e63	beq x11 x12 28	loop: beq a1, a2, fim # Condição de parada do fatorial que atinge 1
0x00058c63	beq x11 x0 24	beq a1, zero, fim # Condição de parada para N = 0
0xffff50693	addi x13 x11 -1	addi a3, a1, -1 # Carrega em a3 o antecedente de a1
0x02d58733	mul x14 x11 x13	mul a4, a1, a3 # Multiplica a1 com o seu antecedente e guarda em a4
0x02e787b3	mul x15 x15 x14	mul a5, a5, a4 # Multiplica o último valor de a5 com o valor de a4
0xffe58593	addi x11 x11 -2	addi a1, a1, -2 # Decrementa a1 em 2 unidades
0xfe9ff06f	jal x0 -24	j loop # Repete a execução do loop
0x00000013	addi x0 x0 0	fim: nop # Finaliza a execução

console output

a0 (x10)0a1 (x11)1a2 (x12)1a3 (x13)2a4 (x14)6a5 (x15)6a6 (x16)0a7 (x17)0s2 (x18)0s3 (x19)0s4 (x20)0s5 (x21)0s6 (x22)0s7 (x23)0s8 (x24)0

Problema 2 - Leitura e Escrita na Memória de Dados

Código-fonte no simulador:

123456789

# Tarso Jabbes Lima de Oliveira

# Problema 2 - Roteiro 7

lui x5, 0x10000 # Carrega o valor 0x10000 nos 20 bits mais significativos de x5

addi x5, x5, 0x000 # Carrega o valor 0x000 nos 12 bits menos significativos de x5

addi x6, x0, 0xFE # x6 = 0xFE ou x6 = 254

sw x6, 12(x5) # Armazena o valor de x6 no endereço de memória de x5 aponta somado em 12

lw x10, 12(x5) # Carrega em x10 o valor armazenado no endereço de memória de (x5 + 12)

Simulador antes da execução das instruções:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x10002b7	lui x5 65536	lui x5, 0x10000 # Carrega o valor 0x10000 nos 20 bits mais significativos de x5
0x00028293	addi x5 x5 0	addi x5, x5, 0x000 # Carrega o valor 0x000 nos 12 bits menos significativos de x5
0x0fe00313	addi x6 x0 254	addi x6, x0, 0xFE # x6 = 0xFE ou x6 = 254
0x0062a623	sw x6 12(x5)	sw x6, 12(x5) # Armazena o valor de x6 no endereço de memória de x5 aponta somado em 12
0x00c2a503	lw x10 12(x5)	lw x10, 12(x5) # Carrega em x10 o valor armazenado no endereço de memória de (x5 + 12)

console output

RegistersMemory

zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0x7fffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000000
a1	0x00000000

Após a execução da primeira instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x10002b7	lui x5 65536	lui x5, 0x10000 # Carrega o valor 0x10000 nos 20 bits mais significativos de x5
0x00028293	addi x5 x5 0	addi x5, x5, 0x000 # Carrega o valor 0x000 nos 12 bits menos significativos de x5
0x0fe00313	addi x6 x0 254	addi x6, x0, 0xFE # x6 = 0xFE ou x6 = 254
0x0062a623	sw x6 12(x5)	sw x6, 12(x5) # Armazena o valor de x6 no endereço de memória de x5 aponta somado em 12
0x00c2a503	lw x10 12(x5)	lw x10, 12(x5) # Carrega em x10 o valor armazenado no endereço de memória de (x5 + 12)

console output

RegistersMemory

zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0x7fffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x10000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000000
a1	0x00000000

Após a execução da segunda instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original code
0x10002b7	lui x5 65536	lui x5, 0x10000 # Carrega o valor 0x10000 nos 20 bits mais significativos de x5
0x00028293	addi x5 x5 0	addi x5, x5, 0x000 # Carrega o valor 0x000 nos 12 bits menos significativos de x5
0x0fe00313	addi x6 x0 254	addi x6, x0, 0xFE # x6 = 0xFE ou x6 = 254
0x0062a623	sw x6 12(x5)	sw x6, 12(x5) # Armazena o valor de x6 no endereço de memória de x5 aponta somado em 12
0x00c2a503	lw x10 12(x5)	lw x10, 12(x5) # Carrega em x10 o valor armazenado no endereço de memória de (x5 + 12)

console output

RegistersMemory

zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0x7fffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x10000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000000
a1	0x00000000

Após a execução da terceira instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x10002b7	lui x5 65536	lui x5, 0x10000 # Carrega o valor 0x10000 nos 20 bits mais significativos de x5
0x00028293	addi x5 x5 0	addi x5, x5, 0x000 # Carrega o valor 0x000 nos 12 bits menos significativos de x5
0x0fe00313	addi x6 x0 254	addi x6, x0, 0xFE # x6 = 0xFE ou x6 = 254
0x0062a623	sw x6 12(x5)	sw x6, 12(x5) # Armazena o valor de x6 no endereço de memória de x5 aponta somado em 12
0x00c2a503	lw x10 12(x5)	lw x10, 12(x5) # Carrega em x10 o valor armazenado no endereço de memória de (x5 + 12)

console output

RegistersMemory

zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0x7fffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x10000000
t1 (x6)	0x000000fe
t2 (x7)	0x00000000
s0 (x8)	0x00000000

Após a execução da quarta instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x100002b7	lui x5 65536	lui x5, 0x10000 # Carrega o valor 0x10000 nos 20 bits mais significativos de x5
0x00028293	addi x5 x5 0	addi x5, x5, 0x000 # Carrega o valor 0x000 nos 12 bits menos significativos de x5
0x0fe00313	addi x6 x0 254	addi x6, x0, 0xFE # x6 = 0xFE ou x6 = 254
0x0062a623	sw x6 12(x5)	sw x6, 12(x5) # Armazena o valor de x6 no endereço de memória de x5 aponta somado em 12
0x00c2a503	lw x10 12(x5)	lw x10, 12(x5) # Carrega em x10 o valor armazenado no endereço de memória de (x5 + 12)

console output

RegistersMemory

Address	+0	+1	+2	+3
0x10000018	00	00	00	00
0x10000014	00	00	00	00
0x10000010	00	00	00	00
0x1000000c	fe	00	00	00
0x10000008	00	00	00	00
0x10000004	00	00	00	00
0x10000000	00	00	00	00
0xffffffc	00	00	00	00
0xffffff8	00	00	00	00
0xffffff4	00	00	00	00
0xfffffff0	00	00	00	00
0xfffffec	00	00	00	00

Após a execução da quinta instrução:

RunStepPrevResetDump

Machine Code	Basic Code	Original Code
0x100002b7	lui x5 65536	lui x5, 0x10000 # Carrega o valor 0x10000 nos 20 bits mais significativos de x5
0x00028293	addi x5 x5 0	addi x5, x5, 0x000 # Carrega o valor 0x000 nos 12 bits menos significativos de x5
0x0fe00313	addi x6 x0 254	addi x6, x0, 0xFE # x6 = 0xFE ou x6 = 254
0x0062a623	sw x6 12(x5)	sw x6, 12(x5) # Armazena o valor de x6 no endereço de memória de x5 aponta somado em 12
0x00c2a503	lw x10 12(x5)	lw x10, 12(x5) # Carrega em x10 o valor armazenado no endereço de memória de (x5 + 12)

console output

RegistersMemory

zero	0x00000000
ra (x1)	0x00000000
sp (x2)	0x7fffffff0
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x10000000
t1 (x6)	0x000000fe
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x000000fe
a1	0x00000000