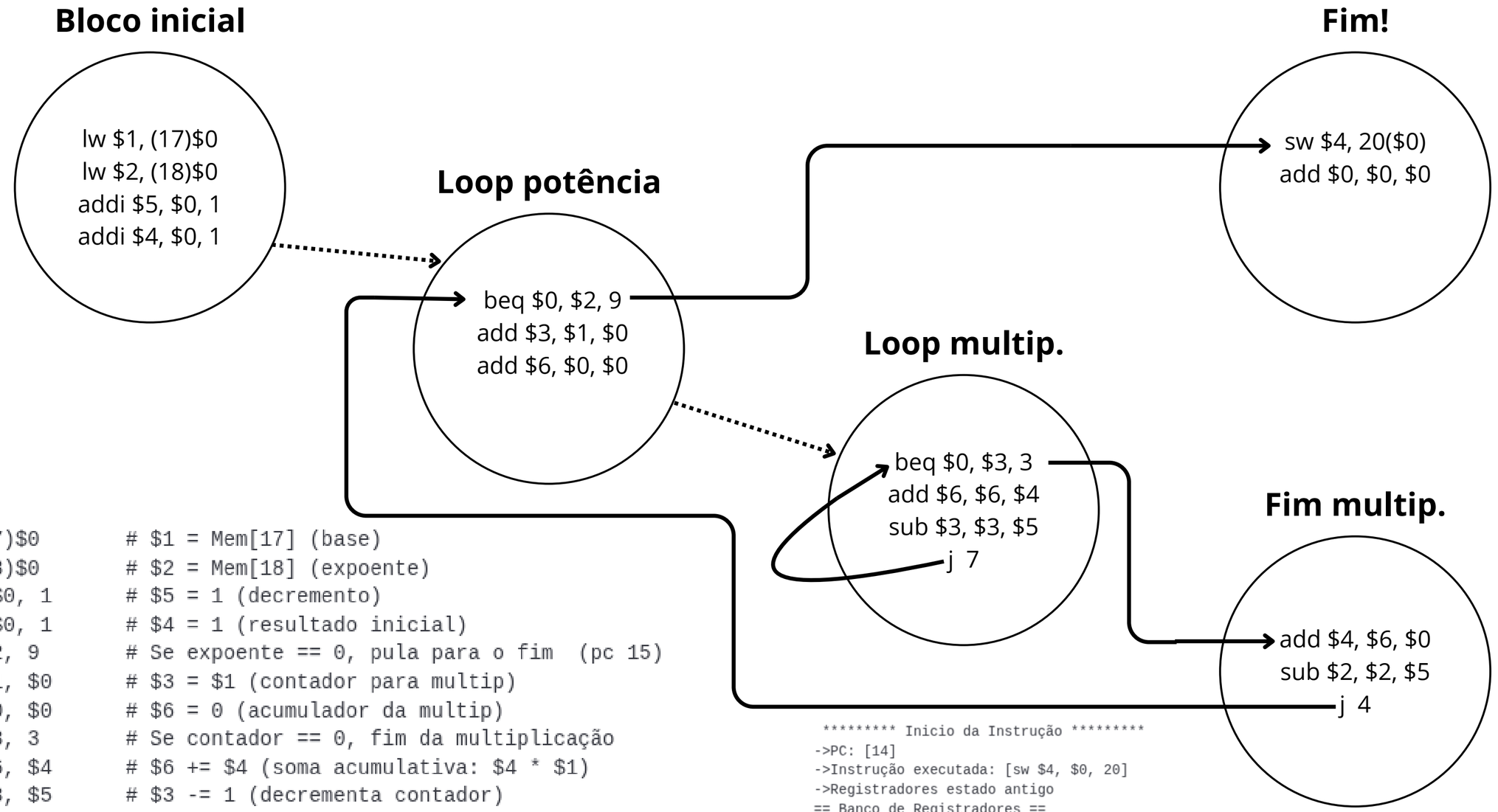


```
0:  lw $1, (17)$0       # $1 = Mem[17] (base)
1:  lw $2, (18)$0       # $2 = Mem[18] (expoente)
2:  addi $5, $0, 1      # $5 = 1 (decremento)
3:  addi $4, $0, 1      # $4 = 1 (resultado inicial)
4:  beq $0, $2, 9       # Se expoente == 0, pula para o fim  (pc 15)
5:  add $3, $1, $0      # $3 = $1 (contador para multip)
6:  add $6, $0, $0      # $6 = 0 (acumulador da multip)
7:  beq $0, $3, 3       # Se contador == 0, fim da multiplicação
8:  add $6, $6, $4      # $6 += $4 (soma acumulativa: $4 * $1)
8:  sub $3, $3, $5      # $3 -= 1 (decrementa contador)
10: j 7                 # Volta para loop multip.
11: add $4, $6, $0      # $4 = $6 (atualiza resultado)
12: sub $2, $2, $5      # $2 -= 1 (decrementa expoente)
13: j 4                 # Repete multiplicação com novo expoente
14: sw $4, $0, 20       # Salva resultado final em Mem[20]
15: add $0, $0, $0      # parada
```

```
 ********* Inicio da Instrução *********
->PC: [14]
->Instrução executada: [sw $4, $0, 20]
->Registradores estado antigo
== Banco de Registradores ==
$0 : 0     $1 : 4     $2 : 0     $3 : 0
$4 : 64    $5 : 1     $6 : 64    $7 : 0
============================
->Registradores estado novo
== Banco de Registradores ==
$0 : 0     $1 : 4     $2 : 0     $3 : 0
$4 : 64    $5 : 1     $6 : 64    $7 : 0
============================
 ********* FIM da Instrução *********

Programa finalizado com sucesso!!!
```