# Lista de exercícios: ARM

1. Considere a seguinte configuração da memória em um dado instante:

r0=0x000001a0

r1=0x00000904

mem32[0x00000904]=0x10101020

mem32[0x00000908]=0x10101028

mem32[0x0000090c]=0x20202010

Mostre qual o valor dos registradores r0 e r1 após as seguintes instruções:

* 1. LDR r0,[r1,#4]! (preindex com writeback)
  2. LDR r0,[r1,#8] (preindex)
  3. LDR r0,[r1],#8 (postindex)

1. Considere a seguinte configuração da memória em um dado instante:

r1=0x00000904

r2=0x00000001

mem32[0x00000904]=0x00000001

Defina qual o valor de r4 ao final de:

Obs.: a arquitetura ARM permite que todas as instruções sejam executadas condicionalmente, para isso basta colocar um sufixo ao final do opcode da operação.

EQ → Equal to zero: executa se o bit Z da ULA está ativo

NE → Not Equal to zero: executa se o bit Z da ULA não está ativo

LDR r0,[r1,#0]

SUBS r3, r0,r2

ADD r4, r0,r2

MULNE r4,r3, r2

LDR r0,[r1,#0]

SUBS r3, r0,r2

ADDEQ r4, r0,r2

MULEQ r4,r4, r2

1. Considerando um pipeline de 5 estágios, mostre as etapas de execução do seguinte código:

Obs.: Considere que todos os estágios sejam executados em 1 ciclo de clock e que os condicionais “GT” e “LT” se referem a “Greater than” e “Lower than” e operam no segundo e terceiro operandos

1 LDR r0,[r1,#0]

2 LDR r2,[r1,#4]

3 ADD r3,r0,r2

4 SUBGT r4,r0,r2

5 SUBLT r4,r0,r2

Quando:

* 1. 0(r1)>4(r1)
  2. 0(r1)<4(r1)
  3. 0(r1)=4(r1)