

**UNIVERSIDADE FEDERAL DO CEARÁ****Campus de Quixadá**

Prof. Thiago Werlley Bandeira da Silva

QXD0288- ORGANIZAÇÃO DE COMPUTADORES E LINGUAGENS
DE MONTAGEM II

Lista 02

2025.2

1. Escrever o código abaixo em assembly ARM, em duas versões:

- Sem mnemônicos de condição (exceto instruções de Branch)
- Com mnemônicos de condição

```
1  X = 10
2  A = 10
3  B = 9
4  while ( X > 1 )
5  {
6      if (X > 5) {
7          B = 2 + X;
8      }
9      else if (X == 5) {
10         A = B = X;
11     }
12     else {
13         A = X - B;
14     }
15
16     B -= B*A;
17     X = X - 1;
18 }
```

2. Escreva rotinas assembly ARM para as seguintes situações:

- Soma de dois vetores de inteiros com 8 posições.
- Soma do maior e do menor número de um vetor de 8 posições.
- Qual o maior e o menor valor de um vetor de 10 posições.

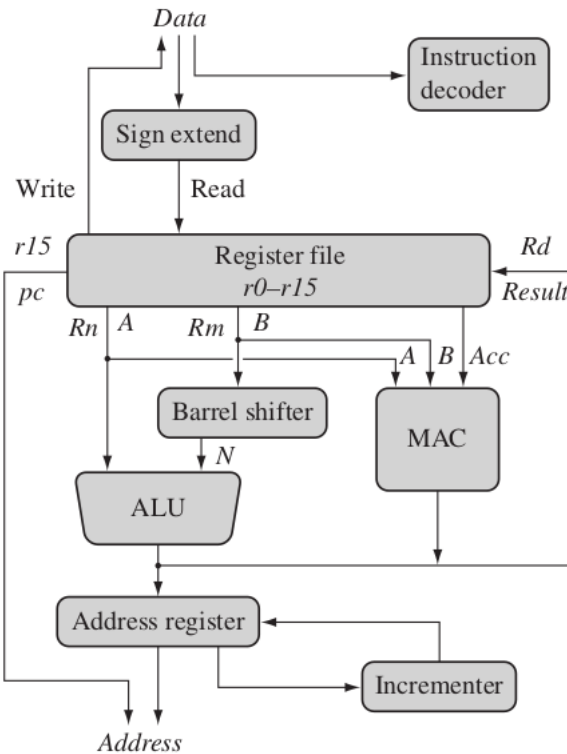
3. Considerando o conteúdo dos registradores mostrados abaixo, elabore uma tabela, para cada item, que represente o estado da memória (endereço/conteúdo) após a execução das seguintes instruções:

```
1  R10 = 0x55aabb44
2  R11 = 0x00001234
3  R12 = 0x00001000
4  R13 = 0x2000e000
```

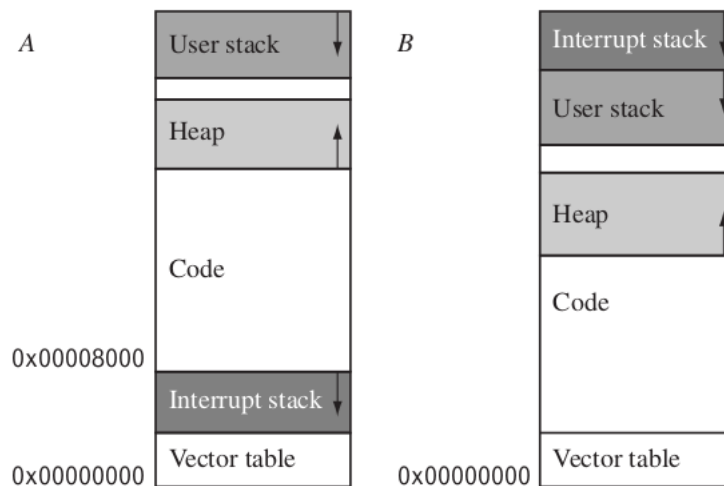
- stmfd sp!, r10-r12
- stmea sp!, r10
- stmib r10!, r11,r12
- stmia sp!, r11,r12
- str r10, [r12]
- str r11, [r10, r12]!
- str r10, [r1], #4

8. `stmed sp!, r10-r12`

4. Explique por que a arquitetura ARM não é puramente RISC.
5. Explique todos os blocos da figura abaixo.



6. Explique o propósito do SPSR, e dê uma justificativa do para o fato dele não estar disponível nos modos USER e SYSTEM.
7. Dado o status inicial `nzcqvift_SVC`, escreva um pequeno trecho de código que, após executado, resulte no status `NZCVqIFT_USER`.
8. Considere os dois layouts de memória mostrados abaixo, qual a principal vantagem do layout B sobre o layout A?



9. Explique o que é e como funciona o *Translation Lookaside Buffer* em uma MMU.

10. Sobre a filosofia de projeto das arquiteturas RISC e ARM, responda:
1. Quais as 4 principais regras de projeto para arquiteturas RISC?
 2. Quais as 5 principais características ARM que a diferem da arquitetura RISC pura?
 3. Por que a arquitetura ARM é considerada adequada para projetos de sistemas embarcados?
11. Explique as vantagens do Barrel Shifter e dê exemplos de uso.
12. Considerando cada pré-condição, indique uma instrução ARM que resulte na respectiva pós-condição, dada a memória abaixo.

Memória

1	mem32[0x80020] = 0x00000005
2	mem32[0x8001C] = 0x00000004
3	mem32[0x80018] = 0x00000003
4	mem32[0x80014] = 0x00000002
5	mem32[0x80010] = 0x00000001
6	mem32[0x8000C] = 0x00000000

Pré-condição

r0 = 0x00080010
 r1 = 0x00000000
 r2 = 0x00000000
 r3 = 0x00000000

Pré-condição

r0 = 0x00080010
 r1 = 0x00000000
 r2 = 0x00000000
 r3 = 0x00000000

Pré-condição

r0 = 0x0008001C
 r1 = 0x00000002
 r2 = 0x00000003
 r3 = 0x00000004

Pré-condição

r0 = 0x00080014
 r1 = 0x00000001
 r2 = 0x00000002
 r3 = 0x00000003

Instrução**Pós-condição**

r0 = 0x0008001C
 r1 = 0x00000001
 r2 = 0x00000002
 r3 = 0x00000003

Pós-condição

r0 = 0x0008001C
 r1 = 0x00000002
 r2 = 0x00000003
 r3 = 0x00000004

Pós-condição

r0 = 0x00080010
 mem32[0x80010] = 0x02
 mem32[0x80014] = 0x03
 mem32[0x80018] = 0x04

Pós-condição

r0 = 0x00080020
 mem32[0x8001C] = 0x03
 mem32[0x80018] = 0x02
 mem32[0x80014] = 0x01

13. Descreva os modos do processador ARM.
14. Explique o funcionamento do pipeline do ARM.
15. Diferencie os seguintes modos de endereçamento: *Preindex with writeback*, *Preindex* e *Postindex*.
16. Como funciona cada uma das seguintes instruções: *stmia*, *stmib*, *stmda*, *stmdb*, *ldmdb*, *ldmda*, *ldmib* e *ldmia*.
17. O que acontece quando ocorre uma interrupção no ARM?
18. Quais são as possíveis exceções nos processadores ARM? Qual modo é ativado na ocorrência de cada exceção?
19. Explique como funciona a interrupção IRQ e a FIQ.
20. Explique a diferença entre as políticas de cache *Writeback* e *Writethrough*.