

# AOCO: Questões e exercícios adicionais

## Parte II

As questões de escolha múltipla (secção 1) e os problemas de resposta aberta (secção 2) foram retirados ou adaptados de testes de AOCO de anos anteriores.

### Informação de referência

|               |        |       |       |     |     |
|---------------|--------|-------|-------|-----|-----|
| Field         | opcode | Rm    | shamt | Rn  | Rd  |
| Bit positions | 31:21  | 20:16 | 15:10 | 9:5 | 4:0 |

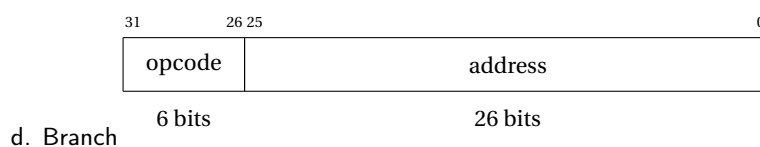
a. R-type instruction

|               |              |         |       |     |     |
|---------------|--------------|---------|-------|-----|-----|
| Field         | 1986 or 1984 | address | 0     | Rn  | Rt  |
| Bit positions | 31:21        | 20:12   | 11:10 | 9:5 | 4:0 |

b. Load or store instruction

|               |       |         |     |
|---------------|-------|---------|-----|
| Field         | 180   | address | Rt  |
| Bit positions | 31:24 | 23:5    | 4:0 |

c. Conditional branch instruction



| Instrução | Opcode        |
|-----------|---------------|
| ADD       | 100 0101 1000 |
| SUB       | 110 0101 1000 |
| AND       | 100 0101 0000 |
| ORR       | 101 0101 0000 |
| LDUR      | 111 1100 0010 |
| STUR      | 111 1100 0000 |
| CBZ       | 101 1010 0    |
| B         | 000 101       |

ALU trabalha em 3 contextos diferentes.

1. instruções lógico-aritméticas:  $ALUOp[1:0]=10$
2. cálculo de endereços:  $ALUOp[1:0]=00$
3. comparação:  $ALUOp[1:0]=01$

Para programação, usar também **a folha de consulta** com as instruções mais comuns.

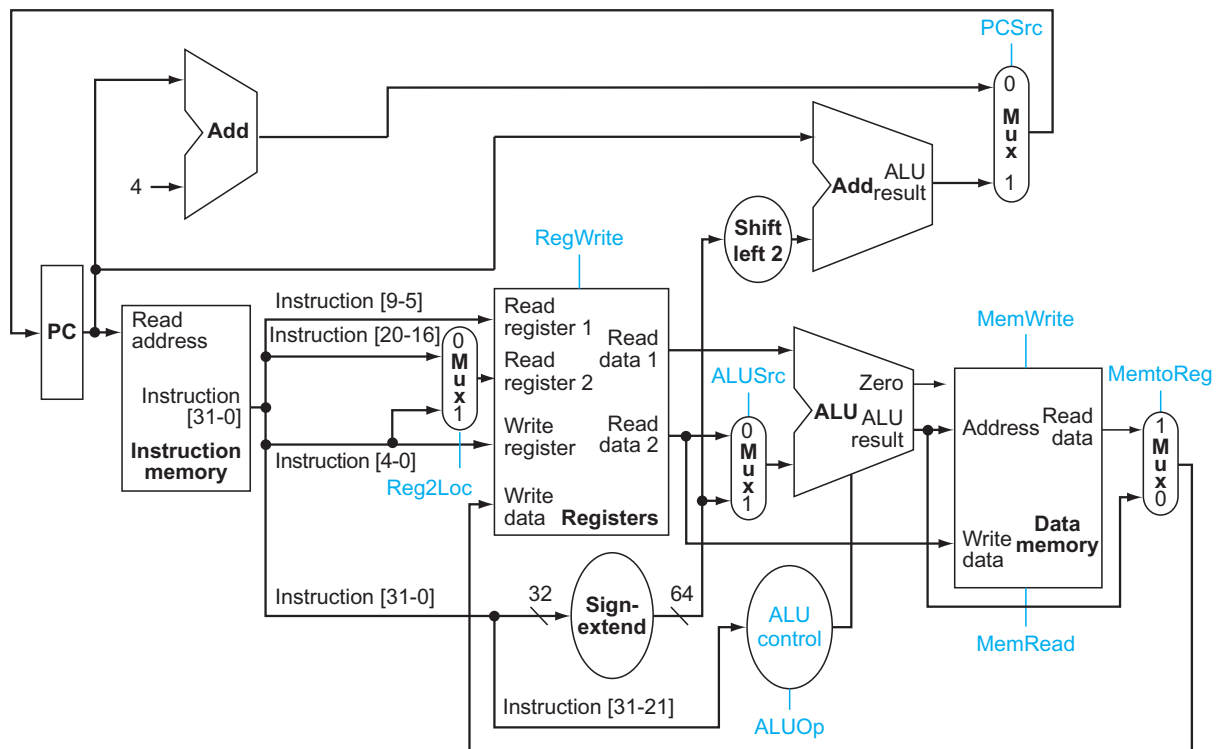


Figura 1: CPU com “multiplexers” e sinais de controlo

## 1 Questões de escolha múltipla

- Assuma que a saída Read data 2 do banco de registos é sempre 0. Que instrução ARMv8 não é afetada por esta anomalia?
  - STUR
  - ADD
  - CBZ
  - LDUR
- Que instrução ARMv8 poderá ser executada se MemtoReg=0, Reg2Loc=1 e ALUSrc=1?
  - ORR
  - CBZ
  - STUR
  - LDUR
- Que instrução ARMv8 tem o código 0xCB0201E8?
  - ADD X15,X8,X2
  - SUB X2,X15,X8
  - SUB X15,X2,X8
  - SUB X8,X15,X2
- Relativamente a sub-rotinas, qual das seguintes afirmações é falsa?
  - Sub-rotinas terminais devem preservar o valor de X30 antes de invocarem outras sub-rotinas.
  - Uma sub-rotina do tipo função devolve um valor como resultado.
  - Uma sub-rotina do tipo procedimento não devolve resultados.
  - Na invocação de uma sub-rotina, o endereço da instrução seguinte é guardado no registo X30.
- Um programa gasta 75 % do tempo em transferências de dados para outro computador via rede sem fios. Quantas vezes é preciso aumentar a velocidade de transferência para obter uma redução do tempo de execução do programa (*speedup*) de duas vezes?
  - 4
  - 1,5
  - 3
  - 2

6. Para um dado programa, o processador P1 com  $F_1 = 1$  GHz apresenta o mesmo tempo de execução que o processador P2 com  $F_2 = 1,25$  GHz. O tempo de execução de P1 fica maior que o de P2 se:
- passar a usar  $F_2 = 1$  GHz;
  - aumentar o valor do CPI médio de P2;
  - reduzir o valor do CPI médio de P1;
  - aumentar 1,3 vezes o período do relógio de P1.
7. O tempo de execução de um programa está repartido entre a execução de instruções da classe A (60 % do tempo) e da classe B (40 % do tempo). Qual das seguintes alterações leva ao melhor desempenho?
- diminuir para metade o tempo de execução das instruções de classe A;
  - diminuir o tempo de execução das instruções de classe B para um quarto do tempo original;
  - reduzir o tempo de execução das instruções de classe A para um terço e aumentar o tempo de execução das instruções de classe B para o dobro;
  - reduzir 1,5 vezes o tempo de execução das instruções de classe A e reduzir o tempo de execução das instruções de classe B para metade.
8. Um programa de cálculo científico gasta 80 % do seu tempo de execução em operações numéricas. Este tempo está repartido da seguinte forma:
- operações aritméticas: 40 %
  - operações trigonométricas: 60 %
- Um novo método de cálculo das funções trigonométricas reduzirá o respetivo tempo de execução em 4×. Qual dos valores indicados se aproxima mais da melhoria de desempenho (*speedup*) global que esta medida produzirá?
- A. 1,82   B. 2,40   C. 1,56   D. 2,62
9. Uma memória *cache* com 64 B/bloco contém 32 KiB de dados. Quantos blocos tem?
- A. 512   B. 2048   C. 256   D. 1024
10. Qual das seguintes afirmações sobre uma memória *cache* do tipo *write-through* é falsa?
- O conteúdo da memória principal está sempre atualizado.
  - No caso de uma falta num acesso de leitura (*read miss*) o valor é lido da memória principal e colocado na *cache* atualizando a etiqueta e alterando o valor de *v* para 1.
  - No caso de uma falta num acesso de escrita (*write miss*) o valor é escrito na memória principal e na memória *cache* atualizando a etiqueta e alterando o valor de *v* para 1.
  - No caso de um acerto num acesso de leitura (*read hit*), o valor é lido da memória *cache*.
11. Um CPU está equipado com uma memória *cache* unificada com taxa de faltas  $t_f = 0,1$ . Em média, 20 % das instruções de um programa acedem a dados. Qual das seguintes alternativas de *split cache* apresenta o mesmo desempenho?
- I-*cache* com  $t_f = 20\%$  e D-*cache* com  $t_f = 8\%$ ;
  - I-*cache* com  $t_f = 8\%$  e D-*cache* com  $t_f = 20\%$ ;
  - I-*cache* com  $t_f = 5\%$  e D-*cache* com  $t_f = 5\%$ ;

- D. I-cache com  $t_f = 10\%$  e D-cache com  $t_f = 20\%$ .
12. A memória principal usada com um CPU de 2 GHz tem um tempo de acesso de 60 ns. O acesso à memória *cache* demora 0,5 ns. Qual deve ser o valor máximo da taxa de faltas para que o tempo médio de acesso a memória seja 10 ciclos?
- A. 5 %   B. 7,5 %   C. 10 %   D. 2,5 %
13. Um programa gasta 50 % do tempo a executar cálculos de vírgula flutuante. Qual é o ganho de rapidez (*speedup*) que se poderia obter se a unidade de vírgula flutuante fosse 5 vezes mais rápida?
- A. 2,5   B. 5/3   C. 10/3   D. 2
14. Qual das seguintes afirmações sobre uma memória *cache* do tipo *write-back* é verdadeira?
- A. Existe a possibilidade de serem feitos 2 acessos a memória principal para apenas um acesso a memória *cache*.
- B. O conteúdo da memória principal está sempre atualizado.
- C. Pode haver um acesso a memória principal mesmo que o acesso a memória *cache* seja um acerto (*hit*).
- D. Não pode ser usada como *D-cache*.
15. Considere duas versões do mesmo programa a correrem no mesmo processador. A versão A foi produzida pelo compilador  $C_A$  e executa  $2 \times 10^{10}$  instruções; a versão B foi produzida pelo compilador  $C_B$  e executa  $1,25 \times 10^{10}$  instruções. Para a versão A, o valor de  $CPI_A$  é 2. A versão A é 25 % mais rápida que a versão B. Qual é o valor de  $CPI_B$ ?
- A.  $CPI_B = 25/16$    B.  $CPI_B = 4$    C.  $CPI_B = 57/16$    D.  $CPI_B = 3$
16. Um programa científico despende 80 % do seu tempo a executar operações de vírgula flutuante. Para tornar o programa 4 vezes mais rápido pretende-se usar uma nova unidade de vírgula flutuante. Quanto mais rápida que a anterior deve ser essa unidade?
- A. 16;   B. 8;   C. 12;   D. É impossível.
17. Assuma as seguintes condições iniciais:  $X0=0x8abc7520$  e  $X2=0x11110000$ . Determine o valor do registo  $X0$  após ser executada a instrução `ORR X0,X0,X2`.
- A. 0x8abc0000   B. 0x9bbd5702   C. 0x9bbd7520   D. 0x8bbc5720
18. Uma memória *cache* tem 32 blocos com 8 palavras por bloco e etiquetas de 16 bits. Quantos bits tem cada endereço?
- A. 24   B. 21   C. 26   D. 32
19. Um processador funciona a 3 GHz. Nesse processador, o programa P1 apresenta um CPI médio de 2,4 e o programa P2 apresenta um CPI médio de 2. Sabendo que o tempo de execução de cada um dos programas é 2 s, indique a afirmação verdadeira.
- A. O programa P2 executa menos instruções que o programa P1.
- B. O programa P2 executa  $3 \times 10^9$  instruções.
- C. Se a frequência baixar para 2 GHz, o programa P2 passa a ser mais rápido que P1.
- D. Se o CPI de P1 aumentar, P2 fica mais lento que P1.
20. O desempenho de uma memória *cache* unificada usada com um CPU de 1 GHz foi considerado insuficiente. Sabendo que o CPI de base é 1, indique a alteração que leva à maior redução do CPI efetivo.

- A. Baixar a taxa de instruções *load/store* de 50 % para 20 %.
- B. Baixar o tempo de acesso à memória externa de 100 ns para 80 ns.
- C. Baixar a taxa de faltas de 15 % para 10 %.
- D. Nenhuma das outras alterações indicadas reduz o CPI efetivo.

## 2 Problemas de resposta aberta

*Nota: Justificar todas as respostas e apresentar todos os cálculos.*

1. O fragmento de código ARMv8 abaixo aplica a sub-rotina `calc` aos elementos de uma sequência de “double words” e acumula os resultados das invocações em X21. O endereço-base da sequência está inicialmente em X19 e o número de elementos em X20.

(a) Completar o fragmento.

```

        mov     X21, _____
ciclo:  cbz     X20, L1           // terminar ciclo
        ldur   X0, [X19]
        _____ calc         // invocar sub-rotina
        add    X21, X21, _____ // usar o resultado
        add    X19, X19, _____
        add    X20, X20, _____
        _____ ciclo
L1:     ....    // fim da execução do fragmento

calc:   eor    X1, X1, _____ // inicializar X1 com zero
LC1:    cbz    X0, _____     // terminar?
        and    X2, X0, 1
        add    X1, X1, X2
        lsr    X0, X0, 1
        b      LC1
LC2:    mov    X0, _____
        ret                                // fim da sub-rotina

```

► Considerar que a sequência processada tem 3 valores: {170, 42, 450}.

- (b) Determinar o número de instruções executadas pela sub-rotina `calc` quando é chamada pela primeira vez.

Assumindo que as instruções de alteração do fluxo de execução (condicional ou incondicional) têm  $CPI=2$  e todas as outras têm  $CPI=1$ , determinar também o valor de CPI médio para este fragmento ao processar a sequência indicada. Mostrar todos os cálculos.

- (c) Considerar agora o funcionamento da sub-rotina `calc` quando recebe argumentos de valor  $2^k$  ( $k$  inteiro,  $0 \leq k \leq 63$ ). Explicar o valor do resultado da sub-rotina e determinar o número de instruções executadas (em função de  $k$ ).

2. A sub-rotina `substitui` procura a primeira ocorrência de um número  $N$  numa sequência de “double words”, substituindo esse número por 0 (zero). Os parâmetros da sub-rotina são, por ordem, os seguintes: 1) endereço-base da sequência; 2) número de elementos da sequência; 3) valor de  $N$ .

- (a) Completar o código da sub-rotina tendo em atenção as convenções relacionadas com o uso de registos.

```

substitui:  cbz    ____, final      // terminar?
            ldur   X5, [X0]        // obter um valor da sequência
            cmp    ____, X5        // é o valor procurado?
            ____   LS1
            add    X0, ____, 8      // preparar próxima iteração
            sub    X1, X1, ____
            b      substitui
L1:         stur   ____, [X0]      // substituir valor na sequência
final:     ret                      // retornar

```

- (b) Supondo que a sequência é {12, 56, 17, 21, 72, 7} e que  $N=21$ , determinar quantas instruções são executadas pela sub-rotina `substitui`.
- (c) Suponha que o programa a que pertence a sub-rotina anterior é executado em dois computadores A e B. O período do sinal do relógio dos computadores A e B é 300 ps e 400 ps respetivamente. O número de ciclos de relógio consumidos por instrução (CPI) é 4 no computador A e 2,5 no computador B. Determinar qual dos computadores é o mais rápido a executar o programa.
3. A sub-rotina `sumsel` retorna a soma dos elementos de uma sequência (de  $N$  “double words”) que pertencem ao intervalo  $[a; b]$ . Os parâmetros da sub-rotina são, por ordem, os seguintes: 1) endereço-base da sequência; 2) número de elementos da sequência; 3) valor de  $a$ ; 4) valor de  $b$ .

- (a) Completar a sub-rotina tendo em atenção as convenções relacionadas com o uso de registos.

```

sumsel:     eor    X5, X5, ____    // inicializar acumulador
loop:      cbz    X1, ____        // terminar?
            ldur   X6, [X0]
            cmp    ____, X2        // limite inferior
            b.lt   cont
            cmp    X6, X3          // limite superior
            ____   cont
            add    X5, X5, ____
cont:      add    X0, ____, 8
            add    X1, X1, ____
            ____   loop
fim:       mov    ____, X5
            ret

```

- (b) Para a sequência  $\{-3, 3, 6, 5, 0, -5, 8, 2, -1\}$  e intervalo  $[-1; 6]$ , determinar quantas instruções são executadas pela sub-rotina `sumsel` e qual o resultado.
- (c) O modelo do processador usada para a execução da sub-rotina emprega um sinal de relógio com a frequência de 1 GHz. O tempo de execução da sub-rotina com os dados da alínea (b) é de 170 ns. Determinar o valor médio de ciclos por instrução (CPI).
4. Considere o CPU ARMv8 simplificado, apresentado na figura 1, e que o valor em cada registo  $X_i$  é  $i + 2$ . A latência de componentes usados no CPU é a seguinte (componentes não indicados têm latência nula):

| I-Mem | Add | Mux | ALU | Regs | D-Mem | Control | ALU control |      |
|-------|-----|-----|-----|------|-------|---------|-------------|------|
| 400   | 100 | 30  | 130 | 220  | 350   | 80      | 40          | (ps) |

- (a) Indique o valor dos seguintes sinais de entrada/saída de componentes e sinais de controlo para a execução da instrução `CBZ X1, fim`:
- Read register 2 = \_\_; Write register = \_\_; Write data de D-Mem = \_\_
- ALUSrc = \_\_; PCSrc = \_\_; MemtoReg = \_\_
- (b) Determine o caminho crítico da instrução `STUR X7, [X2, #-4]` e a respetiva latência.
- (c) Determine a partir de que valor da latência da unidade de controlo o sinal `Write data de D-Mem` pertence ao caminho crítico da instrução `STUR`.
5. A tabela seguinte apresenta o conteúdo (em hexadecimal) de uma memória *cache* do tipo *write-back* com 8 blocos de 8 bytes usada como *D-cache* num CPU com endereços de 16 bits.

| bloco | conteúdo |    |    |    |    |    |    |    | etiqueta | v | d |
|-------|----------|----|----|----|----|----|----|----|----------|---|---|
|       | 7        | 6  | 5  | 4  | 3  | 2  | 1  | 0  |          |   |   |
| 0     | aa       | cc | de | hf | 34 | 33 | 11 | 01 | 235      | 1 | 0 |
| 1     | bb       | ad | 45 | 4f | af | de | 21 | 99 | 391      | 1 | 1 |
| 2     | cc       | 34 | ab | 1f | 56 | cd | ff | ff | 023      | 1 | 1 |
| 3     | dd       | 67 | 22 | 2b | 32 | 56 | 32 | 21 | 198      | 0 | 1 |
| 4     | ee       | 32 | 11 | 9f | aa | ba | ab | bb | 311      | 1 | 0 |
| 5     | ff       | 10 | 00 | 04 | 01 | 02 | 03 | 04 | 278      | 0 | 0 |
| 6     | 11       | 03 | 41 | 32 | cc | dd | ee | ff | 212      | 1 | 1 |
| 7     | 22       | 01 | 65 | 01 | 05 | 06 | 07 | 08 | 387      | 0 | 1 |

- (a) Como é decomposto o endereço para acesso à memória *cache*? Justifique.
- (b) Indique (se possível) o valor (byte) em memória principal no endereço `0xc467`. Justifique.
- (c) Explique quais as alterações que ocorrem na *cache* e na memória principal durante a leitura do valor (byte) residente no endereço `0xe48d`.
6. Um CPU com endereços de 24 bits está equipado com uma memória *cache* de dados do tipo *write-through*. Etiqueta e índice têm, respetivamente, 14 e 6 bits de comprimento.
- (a) Determinar o número de blocos e o número de bytes por bloco desta memória *cache*.
- (b) Considerar a seguinte situação. São realizadas sucessivamente leituras (de uma palavra) das seguintes posições de memória:

0x3B7C94, 0x3B6C90, 0x3B6C98, 0x3B6C94

Quantos blocos são transferidos de memória principal para memória *cache* por causa dos três últimos acessos?

- (c) A memória *cache* é usada num sistema em que a penalidade de faltas é de 80 ciclos. Qual deve ser o valor máximo da taxa de faltas desta *cache* para que o número médio de ciclos de protelamento *no acesso a dados* não exceda 10?

7. A tabela seguinte apresenta o conteúdo de uma memória *cache* do tipo *write-through* com 8 blocos de 4 bytes usada como *D-cache* num CPU com endereços de 32 bits.

|   | Conteúdo    | Etiqueta | v |
|---|-------------|----------|---|
| 0 | aa ff cc 33 | 123456a  | 0 |
| 1 | 12 34 56 78 | 7bcd001  | 1 |
| 2 | 88 b0 3c 2b | 7fffd55  | 1 |
| 3 | 71 ab 3f 6d | 7fffd55  | 1 |
| 4 | 34 ff 13 aa | 07f9910  | 0 |
| 5 | 78 00 9c 23 | 0000893  | 1 |
| 6 | 7a 10 9f a3 | 2900002  | 1 |
| 7 | 99 43 65 b4 | 7f01d12  | 0 |

- (a) Como é decomposto o endereço para acesso à memória *cache*? Justifique.
- (b) Apresente, justificando, o conteúdo da memória *cache* após a execução das seguintes operações (se em algum caso não for possível conhecer o conteúdo escreva "indeterminado"). Cada registo Rx tem 32 bits.

1: R1  $\leftarrow$  0xfafafafa  
 2: R4  $\leftarrow$  R3+R2  
 3: R4  $\leftarrow$  0x0ff32210  
 4: R5  $\leftarrow$  MEM[R4]  
 5: R5  $\leftarrow$  R4-R1  
 6: R6  $\leftarrow$  0xffffaaa8  
 7: MEM[R6+4]  $\leftarrow$  R1

|   | Conteúdo | Etiqueta | v |
|---|----------|----------|---|
| 0 |          |          |   |
| 1 |          |          |   |
| 2 |          |          |   |
| 3 |          |          |   |
| 4 |          |          |   |
| 5 |          |          |   |
| 6 |          |          |   |
| 7 |          |          |   |

8. Um CPU tem endereços de 20 bits e usa uma memória *cache* com 1 palavra/bloco. A memória *cache* do tipo *write-back* usa 6 bits para cada índice e 12 bits para cada etiqueta. Uma parte do conteúdo da memória *cache* está indicada (em hexadecimal) na tabela seguinte:

|     | conteúdo | etiqueta | v   | d   |
|-----|----------|----------|-----|-----|
| 0   | 12345678 | ABC      | 1   | 1   |
| 1   | 6548FEAB | 123      | 0   | 0   |
| 2   | 3C1F56FD | 678      | 1   | 0   |
| 3   | AFD12498 | 567      | 1   | 1   |
| 4   | 6198FA34 | B7C      | 1   | 0   |
| 5   | 1929AAAA | 8D1      | 0   | 1   |
| ... | ...      | ...      | ... | ... |



- (a) Determinar o número de blocos e o número total de bits usados na memória *cache*.
- (b) Determinar, se possível, a posição em memória do valor 0x6198FA34.
- (c) Determinar, se possível, qual o valor atualmente armazenado na posição de memória 0x5670C.

Fim.