

Nome: _____ Nº de estudante: _____

Atenção: Este teste tem 13 questões em 8 páginas, num total de 200 pontos.

Parte I — Questões de Escolha Múltipla

Cada questão tem uma resposta certa. Respostas erradas não descontam.

As respostas às questões de escolha múltipla devem ser assinaladas com × na grelha seguinte.

Apenas as respostas indicadas na grelha são consideradas para efeitos de avaliação.

Opção	Questão									
	1	2	3	4	5	6	7	8	9	10
A	×				×		×			
B		×	×							
C								×		
D				×		×			×	×

Pontos: _____ / 100

- [10] 1. Assumindo que os valores iniciais dos registos R0 e R1 são respetivamente 0x45450000 e 0x0000ABAB, qual é o valor do registo R2 após a execução do seguinte fragmento de código:

```
ror    R2, R1, #4
orr    R2, R2, R0
```

A. 0xF5450ABA B. 0xABAB4545 C. 0xB5450ABA D. 0xB0000ABA

- [10] 2. Considerar que as *flags* estão a zero e se executa o seguinte fragmento de código *assembly*.

```
sub    R0, R1, R2
blt    L1
```

Para que valores de R1 e R2 é que o salto condicional é tomado?

A. R1 = -17 e R1 = 16

B. Nenhuma das restantes opções está correta

C. R1 = -5 e R2 = -3

D. R1 = 8 e R2 = 16

- [10] 3. Assuma as seguintes condições iniciais:

R0 = 0xACAC0101 R1 = 0x0000A0A0 R2 = 0x101 R3 = 0x0000A0A0

Qual é o valor do registo R3 após a execução do seguinte fragmento de código:

```
eor    R4, R1, R0
str    R4, [R2]
ldrb   R3, [R2]
```

A. 0xACACA1A1 B. 0x000000A1 C. 0xA1A1A1A1 D. 0x0x0000A0A1

- [10] 4. Assuma que a saída `Read data 2` do banco de registos é sempre 0. Que instrução ARMv8 não é afetada por esta anomalia?
- A. STUR B. ADD C. CBZ D. LDUR
- [10] 5. Relativamente a sub-rotinas, qual das seguintes afirmações é falsa?
- A. Sub-rotinas terminais devem preservar o valor de LR antes de invocarem outras sub-rotinas.**
- B. Uma sub-rotina do tipo função devolve um valor como resultado.
- C. Uma sub-rotina do tipo procedimento não devolve resultados.
- D. Na invocação de uma sub-rotina o endereço da instrução seguinte é guardado no registo LR.
- [10] 6. 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. 2,62 D. **1,56**
- [10] 7. Uma memória cache com 64 B/bloco contém 32 KiB de dados. Quantos blocos tem esta memória?
- A. 512** B. 2048 C. 256 D. 1024
- [10] 8. Que instrução ARMv8 poderá ser executada se `MemtoReg=0`, `Reg2Loc=1` e `ALUSrc=1`?
- A. ORR B. CBZ C. STUR D. LDUR
- [10] 9. Um dado programa foi compilado com um novo compilador. Numa tarefa, a nova versão é 20 % mais rápida (no mesmo computador). O aumento de desempenho pode dever-se a:
- A. redução da frequência e do CPI médio
- B. redução do período e do número de instruções executadas
- C. aumento da frequência e do CPI médio
- D. redução do número de instruções executadas e/ou do CPI médio**
- [10] 10. Que instrução ARMv8 tem o código `0xCB0201E8`?
- A. ADD X15,X8,X2 B. SUB X2,X15,X8 C. SUB X15,X2,X8 D. SUB X8,X15,X2

Fim da parte I

Nome: _____ Nº de estudante: _____

Parte II — Questões de Resposta Aberta**Atenção:** Responder diretamente no enunciado. **Justificar** todas as respostas.

11. O programa abaixo procura um determinado elemento EP numa sequência L1num recorrendo para isso à sub-rotina *pesquisa*.

[20] (a) Completar o programa.

```

EP      DCD  -3
L1num   DCD  11, -1, 23, -3, -3
Tam     DCD  5

      ldr    R12, =L1num
      ldr    R11, =Tam
      ldr    R10, =EP
      mov    R0, R12
      ldr    R1, [R11]      ; carrega tamanho da sequência
      ldr    R2, [R10]      ; carrega elemento a pesquisar
      bl     pesquisa      ; invoca sub-rotina
      END      ; termina programa

pesquisa  mov    R6, #0      ; inicializa posição
pciclo    ldr    R5, [R0]
          cmp    R2, R5
          bne    prox      ; se não é igual
          b      terminarE
prox      subs   R1, R1, #1
          beq    terminarNE
          add    R0, R0, #4
          add    R6, R6, #1
          b      pciclo
terminarE  mov    R0, R6
          b      fim
terminarNE  mov    R0, #-1
fim        mov    PC, LR      ; fim da sub-rotina

```

- [10] (b) Indique qual é o valor do registo R0 no final da execução do programa. O que significa esse valor?

No final do programa o registo R0 terá o valor 3. Esse valor é a posição da primeira ocorrência do valor -3 na sequência.

12. Considere o CPU ARMv8 simplificado, apresentado na folha anexa, 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)

- [15] (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 = 1 ; Write register = 1 ; Write data de D-Mem = 3
ALUSrc = 0 ; PCSrc = 0 ; MemtoReg = x

(continua)

- [10] (b) Determine o caminho crítico da instrução **STUR X7, [X2, #-4]** e a respetiva latência.

A instrução **STUR X7, [X2, #-4]** guarda o conteúdo do registo **X7** no endereço de memória dado por **X2-4**.

A unidade de controlo do CPU recebe o código da instrução ao fim de 400 ps. A sua latência (80 ps) determina que os sinais de controlo ficam disponíveis aos 480 ps.

Para esta instrução há a considerar três caminhos constituídos por componentes importantes para a sua execução.

- **Atualização de PC:** O cálculo do endereço da próxima instrução utiliza **Add** e **Mux** (controlado por **PCSrc**). A entrada 0 de **Mux** fica disponível aos 100 ps, mas **PCSrc** (igual a 0 para esta instrução) só fica pronto aos 480 ps. Logo, o novo valor de **PC** fica disponível ao fim de $480 + 30 = 510$ ps.
- **Cálculo do endereço de acesso à memória:** Este caminho inclui a **ALU**, pelo que deve verificar-se ao fim de quanto tempo estão disponíveis as suas três entradas:
 - entrada superior (endereço base): **Read data 1** fica disponível aos $400 + 220 = 620$ ps;
 - entrada inferior (valor imediato): a entrada 1 de **Mux** controlado por **ALUSrc** está pronta aos 400 ps mas **ALUSrc**=1 só surge aos 480 ps, pelo que a entrada inferior de **ALU** fica pronta em $480 + 30 = 510$ ps;
 - entrada proveniente de **ALU control**: como **ALUOp** está disponível aos 480 ps então a saída de **ALU control** fica disponível aos $480 + 40 = 520$ ps.

Conclui-se desta análise que a entrada mais demorada da **ALU** é **Read data 1**, pelo que **ALU result** é obtido do caminho a que pertencem **I-Mem** → **Regs** → **ALU** e o tempo que demora a ser calculado é $400 + 220 + 130 = 750$ ps. Deste caminho resulta uma latência superior à do caminho que implementa a atualização de **PC**.

- **Obtenção do valor a escrever em memória:** O valor a escrever é obtido da saída **Read data 2**, demorando mais 220 ps que a saída de **Mux** controlado por **Reg2Loc**. Como este ocorre aos 480 ps então **Read register 2** fica pronto aos $480 + 30 = 510$ ps e **Read data 2** surge aos $510 + 220 = 730$ ps. Esta latência é inferior à de **ALU result**.

Da análise apresentada conclui-se que o caminho crítico para a instrução **STUR** é

I-Mem → **Regs** → **ALU** → **D-Mem**

e o valor da latência é $400 \text{ ps} + 220 \text{ ps} + 130 \text{ ps} + 350 \text{ ps} = 1100 \text{ ps}$

- [10] (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**.

Para que o sinal **Write data** de D-Mem pertença ao caminho crítico da instrução **STUR** é necessário que demore mais a ser obtido do que o endereço. Este surge ao fim de $400+220+130=750$ ps.

Portanto, **Read register 2** deve surgir depois de $750-220=530$ ps.

Logo, $400 + t_{control} + 30 > 530$, pelo que $t_{control} > 100$ ps.

Nome: _____ Nº de estudante: _____

13. 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.

- [10] (a) Determinar o número de blocos e o número de bytes por bloco desta memória *cache*.

Como o índice tem 6 bits, a memória *cache* tem $2^6=64$ blocos.
Como ficam $24-14-6=4$ bits para o deslocamento, cada bloco tem $2^4=16$ bytes por bloco.

- [15] (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?

A primeira leitura (endereço 0x3B7C94) afeta o bloco 9 (001001₂):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
00111011011111														001001				1000					
etiqueta														índice				desl.					

Seja *hit* ou *miss*, a etiqueta do bloco 9 é 00111011011111₂ após a execução do acesso. A segunda leitura (endereço 0x3B6C90) afeta o mesmo bloco (9):

00111011011011														001001				0000					
----------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--------	--	--	--	------	--	--	--	--	--

Como a etiqueta é diferente daquela que está em *cache*, trata-se de uma falta (*miss*) e é lido um bloco de memória principal. A etiqueta em memória *cache* é, agora, 00111011011011₂.

Os dois endereços seguintes levam sempre a *hit*, já que correspondem a partes do mesmo bloco (etiqueta e índices iguais aos do 2º acesso).

00111011011011														001001				1000					
00111011011011														001001				0100					

Logo, não existem mais acessos a memória principal: os últimos três acessos provocam a leitura de 1 bloco da memória principal.

- [10] (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?

O número de ciclos de protelamento (por operação de acesso a dados) C_p é dado pela número médio de acessos a dados que resultam em falta vezes a penalidade p_f (a dividir pelo número de acessos a dados N_d).

$$C_p = \frac{(N_d \times m_d) \times p_f}{N_d} = m_d \times p_f$$

Das condições do enunciado tem-se:

$$m_d \times 80 \leq 10 \quad \Rightarrow \quad m_d \leq \frac{10}{80} = 0,125$$

Portanto, $m_d \leq 12,5 \%$.