

Ficha 5 – Cache

Exercícios resolvidos

1. Considere uma memória *cache* de mapeamento direto, com 8 blocos e uma palavra por bloco. Complete a seguinte tabela, que representa a evolução do estado da memória *cache* para os 10 acessos consecutivos indicados. Assuma que, inicialmente, a memória *cache* está vazia e que os endereços são endereços de *palavras*.

Apresente também o conteúdo da memória *cache* após o último acesso.

Acesso	Endereço	Hit/Miss	Conteúdo do bloco	
			inicial	final
1	22			
2	26			
3	22			
4	26			
5	16			
6	3			
7	16			
8	8			
9	27			
10	10			

1. Como a memória tem 8 blocos, o índice do bloco é definido pelo resto da divisão do endereço por 8. O efeito da sequência de acessos é descrito a seguir. A operação de obtenção do resto é representada por %:

$$a \% b = \text{resto da divisão de } a \text{ por } b.$$

- Endereço 22 → posição $22 \% 8 = 6$; falta (*miss*); $\text{cache}[6] \leftarrow \text{mem}[22]$.
- Endereço 26 → posição $26 \% 8 = 2$; falta (*miss*); $\text{cache}[2] \leftarrow \text{mem}[26]$.
- Endereço 22 → posição $22 \% 8 = 6$; acerto (*hit*); $\text{cache}[6]$ tem conteúdo de $\text{mem}[22]$.
- Endereço 26 → posição $26 \% 8 = 2$; acerto (*hit*); $\text{cache}[2]$ tem conteúdo de $\text{mem}[26]$.
- Endereço 16 → posição $16 \% 8 = 0$; falta (*miss*); $\text{cache}[0] \leftarrow \text{mem}[16]$.
- Endereço 3 → posição $3 \% 8 = 3$; falta (*miss*); $\text{cache}[3] \leftarrow \text{mem}[3]$.
- Endereço 16 → posição $16 \% 8 = 0$; acerto (*hit*); $\text{cache}[0]$ tem conteúdo de $\text{mem}[16]$.
- Endereço 8 → posição $8 \% 8 = 0$; falta (*miss*); $\text{cache}[0]$ tinha conteúdo de $\text{mem}[16]$; $\text{cache}[0] \leftarrow \text{mem}[8]$.
- Endereço 27 → posição $27 \% 8 = 3$; falta (*miss*); $\text{cache}[3]$ tinha conteúdo de $\text{mem}[3]$; $\text{cache}[3] \leftarrow \text{mem}[27]$.

10. Endereço 10 \rightarrow posição $10 \% 8 = 2$; falta (*miss*); $\text{cache}[2]$ tinha conteúdo de $\text{mem}[26]$; $\text{cache}[2] \leftarrow \text{mem}[10]$.

A tabela resultante é:

Acesso	Endereço	Hit/Miss	Conteúdo do bloco	
			inicial	final
1	22	M	-	$\text{mem}[22]$
2	26	M	-	$\text{mem}[26]$
3	22	H	$\text{mem}[22]$	$\text{mem}[22]$
4	26	H	$\text{mem}[26]$	$\text{mem}[26]$
5	16	M	-	$\text{mem}[16]$
6	3	M	-	$\text{mem}[3]$
7	16	H	$\text{mem}[16]$	$\text{mem}[16]$
8	8	M	$\text{mem}[16]$	$\text{mem}[8]$
9	27	M	$\text{mem}[3]$	$\text{mem}[27]$
10	10	M	$\text{mem}[26]$	$\text{mem}[10]$

O conteúdo final da memória *cache* é:

Bloco	Conteúdo
0	$\text{mem}[8]$
1	-
2	$\text{mem}[10]$
3	$\text{mem}[27]$
4	-
5	-
6	$\text{mem}[22]$
7	-

2. Um CPU com endereços de 24 bits está equipado com uma memória *cache* de dados, de mapeamento direto, 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 realização de sucessivas 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 de relógio. 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?

2.

- a) O índice tem 6 bits, pelo que 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.
- b) A primeira leitura (endereço $0x3B7C94$) afeta o bloco 9 (001001_2):

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00111011011111														001001				0100					
etiqueta														índice				desl.					

Seja *hit* ou *miss*, a etiqueta do bloco 9 é 00111011011111_2 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, 00111011011011_2 .

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 um bloco da memória principal.

- c) O número de ciclos de protelamento (por operação de acesso a dados), C_p , é dado pelo 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, o valor máximo da taxa de faltas é $m_d = 12,5\%$.

3. Um CPU tem endereços de 16 bits e uma memória *cache* de mapeamento direto para dados (palavras, 32 bits). A memória *cache* é do tipo *write-back*, possui uma palavra por bloco e a respetiva indexação é formada por 4 bits.

A tabela seguinte mostra parcialmente o conteúdo da *cache* (conteúdo e etiqueta em hexadecimal).

	conteúdo	etiqueta	v	d
...
5	7BCDBCD7	241	0	0
6	76543210	1F3	1	0
7	FFFFFFFF	241	1	1
8	80000000	240	1	0
9	26E111A4	241	1	1
10	13012020	199	1	0
...

- Determine quantos blocos tem a *cache* e quantas palavras em memória podem ser mapeadas em cada entrada da *cache*.
- Determine o conteúdo de memória nos endereços 0x9020 e 0x9024.
- Mostre quantos acessos a memória ocorrem como consequência das seguintes operações:
 - ler Mem[0x7CDC]
 - escrever 0x12345678 em Mem[0xE020]
- Assuma que a taxa de faltas da *cache* é 1/9 da taxa de acertos, a penalidade de falta é 80 ciclos e o CPI devido a protelamento no acesso a dados é 2. Nestas condições, determine a percentagem de instruções executadas que acedem a memória.

3.

- Os endereços decompõem-se nos seguintes campos:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
etiqueta										índice				offset	
10 bits										4 bits				2 bits	

- byte *offset* – formado pelos 2 bits menos significativos, permite designar um dos 4 bytes de cada palavra;
- índice – formado pelos 4 bits que sucedem ao byte *offset*, permite representar os índices, de 0 a 15, dos blocos da *cache*;
- etiqueta – formada pelos 10 bits restantes, permite identificar que palavra de memória está em cada bloco da *cache*.

Portanto, o número de blocos é $2^4 = 16$.

O número de palavras em memória que podem ser mapeadas em cada entrada da *cache* é dado pelo quociente entre o número de palavras e o número de blocos da *cache*: $\frac{2^{16}/4}{2^4} = 2^{10} = 1024$. Alternativamente, este número corresponde ao total de etiquetas possíveis com 10 bits: $2^{10} = 1024$.

b)

$$0x9020 = \underbrace{1001\ 0000\ 00}_{240_H} \underbrace{10\ 00}_{8}$$

O conteúdo presente na entrada 8 da *cache* é válido ($v=1$) e a etiqueta (240_H) é igual à etiqueta do acesso atual. Além disso, como $d=0$ significa que o conteúdo em memória é igual ao conteúdo atual em *cache*. Assim, conclui-se que o conteúdo de memória no endereço $0x9020$ é $0x80000000$.

$$0x9024 = \underbrace{1001\ 0000\ 00}_{240_H} \underbrace{10\ 01}_{9}$$

O conteúdo presente na entrada 9 da *cache* é válido ($v=1$) mas a etiqueta (241_H) é diferente da etiqueta do acesso atual. Conclui-se assim que é impossível determinar o conteúdo de memória no endereço $0x9024$.

c) ler Mem[0x7CDC]:

$$0x7CDC = \underbrace{0111\ 1100\ 11}_{1F3_H} \underbrace{01\ 11}_{7}$$

O conteúdo presente na entrada 7 da *cache* é válido ($v=1$) mas a etiqueta (241_H) é diferente da etiqueta do acesso atual. Perante esta situação de *read-miss*, como $d=1$ é necessário fazer *write-back* (escrever conteúdo do bloco 7 da *cache* em memória) antes de concretizar a leitura de memória pretendida. Portanto, conclui-se que ocorrem dois acessos a memória.

escrever $0x12345678$ em Mem[0xE020]:

$$0xE020 = \underbrace{1110\ 0000\ 00}_{380_H} \underbrace{10\ 00}_{8}$$

O conteúdo presente na entrada 8 da *cache* é válido ($v=1$) mas a etiqueta (240_H) é diferente da etiqueta do acesso atual. Perante esta situação de *write-miss*, como $d=0$ não é necessário fazer *write-back* antes de atualizar o conteúdo da *cache*. É de notar que a escrita ocorre apenas na *cache*. Portanto, neste caso não há qualquer acesso a memória.

Conclui-se assim que após a realização das operações indicadas ocorreram 2 acessos a memória.

d) $\text{taxa}_{\text{falta}} = \frac{1}{9} \text{taxa}_{\text{acerto}} = \frac{1}{9}(100 - \text{taxa}_{\text{falta}})$ pelo que $\text{taxa}_{\text{falta}} = 10\%$

$$\text{CPI}_{\text{prot}} = \text{taxa}_{\text{falta}} \times N_{\text{acessos/inst}} \times \text{Pf}$$

$$2 = 0,1 \times N_{\text{acessos/inst}} \times 80$$

$$N_{\text{acessos/inst}} = 0,25$$

Percentagem de instruções que acedem a dados é 25 %.

4. Um processador funciona a 2 GHz e possui uma memória *cache* unificada. A taxa de faltas no acesso à *cache* é 9% e o tempo de acesso à memória principal é 50 ns.

a) Calcule a penalidade de falta em ciclos.

b) Medições efetuadas para um conjunto de *benchmarks* revelaram que o número de ciclos de protelamento em acessos a memória (por instrução) é $\text{CPI}_{\text{prot}} = 12$.

Determine a percentagem de instruções que acede a dados.

4.

- a) Para obter a penalidade de falta em ciclos, determina-se o número de ciclos que o processador executa durante o tempo de acesso a memória principal.

$$T = \frac{1}{F} = 0,5 \text{ ns} \quad \text{logo} \quad p_f = \frac{50}{0,5} = 100$$

- b) Seja n_a o número de acessos a memória por instrução e t_f a taxa de faltas.
O número médio de ciclos gastos em protelamento devido a faltas no acesso a *cache* é:

$$\text{CPI}_{\text{prot}} = t_f \times p_f \times n_a$$

Portanto:

$$n_a = \frac{\text{CPI}_{\text{prot}}}{t_f \times p_f},$$

o que implica

$$n_a = \frac{12}{0,09 \times 100} = \frac{4}{3}$$

Cada instrução requer obrigatoriamente um acesso para a sua obtenção. Acessos adicionais são necessariamente acessos a dados. Então, o número médio de acessos a dados é: $n_a - 1 = \frac{1}{3} = 33,3\%$.

Exercícios propostos

5. Uma memória *cache* de mapeamento direto usa a política de escrita *write-through*. A *cache* tem 4 blocos de duas *words* (8 bytes), sendo usada como *D-cache* num processador RISC-V.

O conteúdo da *cache* é o seguinte (em hexadecimal):

bloco	conteúdo								etiqueta	v
	7	6	5	4	3	2	1	0		
0	2a	5d	04	aa	78	00	9c	23	2900002	1
1	1b	b2	34	bb	7a	10	9f	a3	0000893	1
2	99	52	36	cc	88	b0	3c	2b	7bcd001	1
3	42	15	90	cd	71	ab	3f	6d	65abfff	0

Assuma endereços com 32 bits e que $t0 = 0x52000044$.

- Mostre como é decomposto o endereço para acesso à *cache* e indique o comprimento da etiqueta.
- Qual é o valor lido de memória por uma operação de 32 bits que use o endereço contido em $t0$?

6. A tabela seguinte apresenta o conteúdo de uma memória *cache* de mapeamento direto, 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

- Como é decomposto o endereço para acesso à memória *cache*? Justifique.
- Indique, justificando, que alterações ocorrem na memória *cache* após a execução da seguinte sequência de operações (se em algum caso não for possível conhecer o conteúdo escreva "indeterminado". Cada registo R_x tem 32 bits.

```

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

```

7. A tabela seguinte apresenta o conteúdo (em hexadecimal) de uma memória *cache* de mapeamento direto usada como *D-cache* num CPU com endereços de 16 bits. A *cache* tem 8 blocos de 8 bytes e a política de escrita usada é do tipo *write-back*.

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.

8. Um CPU tem endereços de 20 bits e usa uma memória *cache* de mapeamento direto com uma palavra por bloco. A *cache* é do tipo *write-back*, usa 6 bits para cada índice e 12 bits para cada etiqueta. Uma parte do conteúdo da *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, o endereço de memória do valor 0x6198FA34.
- c) Determinar, se possível, qual o valor atualmente armazenado na posição de memória 0x5670C.

9. Um CPU tem endereços de 18 bits e uma memória *cache* de mapeamento direto para palavras (32 bits). A memória *cache* tem 16 posições. A política de escrita em memória é *write-through*. O conteúdo inicial da memória *cache* está indicado na tabela (em hexadecimal).

	conteúdo	etiqueta	v
0	12345678	abc	1
1	6548feab	123	1
2	3c1f56fd	678	1
3	afd12498	567	0
4	6198fa34	b7c	1
5	1929aaaa	8d1	0
6	bbabeedd	cd3	1
7	1123aa56	456	1
8	7611432a	001	1
9	ffffeffe	877	1
10	ddedd556	777	0
11	4444cccc	198	1
12	7627abed	fdf	1
13	8768888a	479	1
14	71672912	655	0
15	22256733	111	1

- a) Indique qual é o espaço de endereçamento de um sistema baseado neste CPU e determine a sua capacidade máxima de memória.
- b) Determine o número de bits da etiqueta e do índice, assim como o número total de bits que a *cache* armazena.
- c) Indique as alterações da memória *cache* para a seguinte sequência de acessos (L=leitura, E=escrita):

tipo	endereço	valor
L	2df10	--
L	23454	--
L	3f7b0	--
E	048c4	1212abab
E	3f7d0	00001111
E	1dde8	aaaabbbb

10. Assuma que o CPU do exercício anterior é usado com uma memória *cache* também semelhante à anterior, mas que usa a política de escrita *write-back*. O conteúdo inicial da memória *cache* está indicado na tabela (em hexadecimal).

	conteúdo	etiqueta	v	d
0	12345678	abc	1	1
1	6548feab	123	0	0
2	3c1f56fd	678	1	0
3	afd12498	567	0	0
4	6198fa34	b7c	1	0
5	1929aaaa	8d1	0	1
6	bbabeedd	cd3	1	1
7	1123aa56	456	1	0
8	7611432a	001	1	1
9	ffffeffe	877	1	1
10	ddedd556	777	0	0
11	4444cccc	198	1	1
12	7627abed	fdf	0	1
13	8768888a	479	1	0
14	71672912	655	0	0
15	22256733	111	1	0

- a) Explique a finalidade do campo d.
- b) Indique as alterações da memória *cache* para a seguinte sequência de acessos (L=leitura, E=escrita):

tipo	endereço	valor
L	2df10	--
E	2df10	33334444
E	10c64	9999aaaa
L	23454	--
L	3f7b0	--
E	21de4	bbbb7777
E	2afdc	1212abab
E	3f7b0	00001111

11. Para cada um dos sistemas indicados a seguir considere que o tempo de acesso a memória principal é de 70 ns e que 36% das instruções acedem a dados em memória. O acesso a memória principal tem início após falta de acesso à memória *cache*.

Processador	Cache	Taxa de faltas	Tempo de acerto
P1	1 KiB	11,4%	0,62 ns
P2	2 KiB	8,0%	0,66 ns

- a) Assumindo que é o tempo de acerto que determina o período de relógio, determine as frequências de operação dos dois sistemas.
- b) Determine o tempo médio de acesso à memória para os dois casos.
- c) Assumindo um CPI básico de 1, determine o CPI de cada um dos processadores. Qual é o processador mais rápido?

12. Um CPU funciona a 1 GHz e está equipado com memórias *cache* para instruções e para dados, cujas taxas de faltas são, respetivamente, 5 % e 10 %. O tempo de acesso a memória principal é 80 ns (a acrescentar ao tempo de acesso a memória *cache*). Em média, 40 % das instruções de um programa acedem a dados (i.e., são *load* ou *store*).

- Determine a taxa de faltas global da memória *cache* em número de faltas por 1000 instruções.
- Suponha que se pretendia equipar o CPU com uma memória *cache* unificada. Determine a máxima taxa de faltas desta alternativa para que ela apresente o mesmo desempenho que a versão *split cache*.
- Assuma que, na ausência de faltas de *cache*, o CPU tem $CPI_{ideal}=1,2$. Determine o CPI efetivo para os seguintes casos:
 - sistema sem memória *cache*;
 - sistema com memória *cache*.

13. Considere um sistema em que o endereçamento de memória é de 32 bits. A *cache* do sistema possui 4096 blocos com 4 palavras por bloco. Determine o número de bits necessários para armazenar as etiquetas na *cache* para cada forma de organização.

- Mapeamento direto.
- Associatividade com 2 vias.
- Associatividade de 4 vias.
- Associatividade total.

14. Para aumentar o desempenho geral, um dado computador utiliza uma *cache* associativa com duas vias (2-way) cuja estrutura e conteúdo, em hexadecimal, estão representados na tabela. Considerar que a memória endereçável do computador é de 64 KiB.

conjunto	dados								etiqueta		d		v	
	0	1	2	3	0	1	2	3						
0	57	C5	25	90	3F	21	42	10	000	7B1	1	0	1	1
1	10	24	E5	A9	11	A4	23	98	7A7	42B	0	0	0	0
2	FE	6A	34	1B	DE	76	03	77	27E	311	1	0	1	0
3	23	65	A5	65	65	AE	21	7C	42F	1FE	1	1	1	0
4	99	00	32	AC	7E	5E	60	77	675	777	1	0	1	1
5	A1	A3	6B	6D	DD	71	09	95	2EC	5EA	1	0	0	0
6	FF	4E	5D	A4	B8	BC	70	13	7BC	0D2	1	0	1	0
7	14	15	8C	55	AB	87	99	15	1AB	111	0	0	1	1

- Determine a capacidade de armazenamento da *cache* em bytes.
- Determine o número de comparações simultâneas efetuadas pelos circuitos da *cache* durante um acesso à memória.
- Indique o conteúdo da memória correspondente aos endereços 8567, F798 e 663E.
- Diga qual a posição de memória que contém o valor DE.
- Suponha que os endereços 01FF e 831C são utilizados um a seguir ao outro com frequência, isto é, correspondem a código num ciclo. Onde seriam eles colocados na memória *cache*? E se fossem os endereços 9A30, 9A50 e 9A71? Comente.
- Relacione o campo d (*dirty bit*) com o tipo de *cache*: *write-through* ou *write-back*.

15. Um processador de 32 bits (endereços, instruções e dados) possui dois níveis de *cache*. O tempo de acesso a memória principal é de 80 ns. As características das *caches* são as seguintes:

- Cache L1: 64 KiB, 64 palavras/bloco, mapeamento direto, $t_{\text{acerto}} = 1 \text{ ns}$, $t_{\text{falta}} = 10 \%$ e *write-back*.
 - Cache L2: 512 KiB, 128 palavras/bloco, 4 vias, $t_{\text{acerto}} = 4 \text{ ns}$, $t_{\text{falta}} = 25 \%$, *write-through* e LRU (critério de substituição *least recently used*).
- a) Determine o valor hexadecimal da etiqueta em cada uma das *caches*, quando o CPU acede ao endereço 0x25031968.
- b) Suponha que neste sistema são executadas 1000 instruções em que 10% são *stores*, 20% *loads*, 30% *branches* e 40% são instruções aritméticas ou lógicas. Determine o número de acessos à memória principal.
- c) Considere uma *cache* de nível 2 alternativa à acima indicada e que apresenta as seguintes características: 512 KiB, 128 palavras/bloco, 8 vias, $t_{\text{acerto}} = 8 \text{ ns}$, $t_{\text{faltas}} = 20 \%$, *write-through* e FIFO. O novo sistema veria o seu desempenho aumentar ou diminuir? Justifique.

Fim do enunciado

Soluções:

5.

- a) Byte *offset* ([1..0]): 2 bits
 Word *offset* ([2]): 1 bit
 Índice ([4..3]): 2 bits
 Etiqueta [31..5]: 27 bits
- b) 0x2a5d04aa

6.

- a) *Offset* ([1..0]): 2 bits
 Índice ([4..2]): 3 bits
 Etiqueta [31..5]: 27 bits
- b) Blocos 3 e 4 são alterados.

	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	fa fa fa fa	7fffd55	1
4	indeterminado	07f9910	1
5	78 00 9c 23	0000893	1
6	7a 10 9f a3	2900002	1
7	99 43 65 b4	7f01d12	0

7.

- a) *Offset* ([2..0]): 3 bits
 Índice ([5..3]): 3 bits
 Etiqueta [15..6]: 10 bits
- b) 0xEE
- c) O valor 0x45 é escrito no endereço 0xE44D da memória principal.

Lê de memória principal o bloco que inclui o valor no endereço 0xE48D de memória, escrevendo-o na posição 1 da *cache* (*offset* = 5). Atualiza a etiqueta desse bloco para 392, assim como os indicadores (*v*=1 e *d*=0).

8.

- a) N° blocos = 64; bits/bloco = 46; total de bits = 2944.
- b) 0xB7C10.
- c) Indeterminado.

9.

- a) Endereçamento: 0x00000 - 0x3ffff; capacidade máxima: 256 KiB.
- b) Etiqueta: 12 bits; índice: 4 bits; armazenamento: 720 bits.
- c) 1. Lê valor 0x6198fa34 de memória *cache*.

2. Lê valor de memória principal e coloca-o na posição 5, com $v=1$ e etiqueta $0x8d1$.
3. Lê valor de memória principal e coloca-o na posição 12, com $v=1$ e etiqueta $0xfde$.
4. Escreve valor $0x1212abab$ na posição 1, $v=1$, e atualiza a memória principal no endereço $0x048c4$.
5. Não altera memória *cache* (não existe informação em *cache* para este endereço, porque a etiqueta é diferente); coloca valor $0x00001111$ em memória principal no endereço $0x3f7d0$.
6. Não altera memória *cache* (não existe informação válida em *cache* para este endereço); coloca valor $0xaaabbbb$ em memória principal no endereço $0x1dde8$.

10.

- a) O campo d indica se o valor de conteúdo em memória *cache* é diferente do valor em memória principal.
- b)
1. Índice 4, etiqueta $0xb7c$ e $v=1$: *read hit*.
Ler valor $0x6198fa34$ de memória *cache* (bloco 4).
 2. Endereço igual ao da alínea anterior: *write hit*.
Como $d=0$, não é necessário fazer *write back*. O valor $0x33334444$ é escrito na memória *cache* (bloco 4), $v=1$, $d=1$, etiqueta $0xb7c$.
 3. Índice 9, etiqueta $0x431$, $v=1$: *write miss*.
Como $v=1$ e $d=1$, é necessário fazer *write back*: escrever o valor $0xffffeffe$ (posição 9) em memória (endereço $0x21de4$). O valor $0x9999aaaa$ é colocado na posição 9 da memória *cache*, com $v=1$, $d=1$, etiqueta $0x431$.
 4. Índice 5, etiqueta $0x8d1$ e $v=0$: *read miss*.
Como $v=0$, não faz *write back* (valor de d não interessa neste caso). Ler valor de memória principal e colocá-lo na posição 5, com $v=1$, $d=0$, etiqueta $0x8d1$.
 5. Índice 12, etiqueta $0xfde$, $v=0$: *read miss*.
Como $v=0$, não faz *write back*. Ler valor de memória principal e colocá-lo na posição 12, com $v=1$, $d=0$, etiqueta $0xfde$.
 6. Índice 9, etiqueta $0x877$. O bloco 9 foi alterado pelo terceiro acesso, pelo que a etiqueta é diferente (etiqueta atual do bloco $0x431$). Como $v=1$, ocorre um *write miss*.
Sendo $d=1$, é necessário efetuar *write back*: escrever o valor $0x9999aaaa$ na posição de endereço $0x00010c64$. Escrever valor $0xbbbb7777$ no bloco 9 de memória *cache*, $v=1$, $d=1$, etiqueta $0x877$.
 7. Índice 7, etiqueta $0xabf$, $v=1$: como as etiquetas não coincidem, ocorre *write miss*.
Como $d=0$, não existe necessidade de fazer *write-back*. Colocar o valor $0x1212abab$ em memória *cache* (bloco 7) com $v=1$ e $d=1$.
 8. Índice 12, etiqueta $0xfde$, $v=1$ (mesmo endereço que no quinto acesso, que alterou este bloco): *read hit*.
Basta atualizar valor em memória *cache*: escreve valor $0x00001111$ no bloco 12 com $v=1$, $d=1$, etiqueta $0xfde$ (mantém etiqueta).

11.

- a) P1: 1,61 GHz; P2: 1,52 GHz
- b) P1: $t_{\text{acesso}} = 8,6 \text{ ns}$; P2: $t_{\text{acesso}} = 6,26 \text{ ns}$

- c) P1: CPI=18,5; P2: CPI=12,5; P2 é 1,39 vezes mais rápido que P1.

12.

- a) 90
- b) 6,43 %
- c) i. 113,2
ii. 8,4

13.

- a) 64 Kib
- b) 68 Kib
- c) 72 Kib
- d) 112 Kib

14.

- a) 90
- b) Duas comparações entre etiquetas de 11 bits.
- c) Mem[8567] = indeterminado (v=0)
Mem[F798] = FF (após WB)
Mem[663E] = indeterminado (etiquetas diferentes).
- d) indeterminado (v=0)
- e) Conjunto 7, ocupando as duas vias.
- f) Os conteúdos nos endereços 9A30 e 9A50 ocupariam as duas vias do conjunto 4, mas o primeiro destes conteúdos seria substituído pelo valor de memória em 9A71.

15.

- a) L1: 0x2503; L2: 0x1281
- b) 40
- c) O desempenho mantém-se igual.