

2.1.

A sequência de acessos à memória que tem um impacto maior na performance do sistema é a de fazer load de uma imagem, recorrendo tantas vezes quanto ao seu tamanho às funções: *frame_memory_read()* e *frame_memory_write()*.

2.3.1.

1.

Compulsory: No 1º acesso a um bloco de memória este tem de ser sempre trazido para cache;

Capacity: A cache não tem tamanho suficiente para todos os blocos que usamos;

Conflict: Vários blocos são mapeados para a mesma localização na cache e esta não tem espaço suficiente para todos.

2.

Write-Through: Os dados quando modificados são alterados na cache e na memória;

Write-Back: Os dados quando modificados são apenas reescritos na cache. Só é escrito em memória quando esse bloco for substituído na cache ou passe uma determinada porção de tempo;

Write-Allocate: Os dados são carregados na cache aquando um Write-Miss;

Write-No-Allocate: Os dados são apenas modificados na memória e não são carregados na cache aquando um Write-Miss;

2.3.2.

a)

Max Price = 0.013€

L1 Price = $9/1000 = 0.009\text{€/KB}$

Frame Memory = 128KB

Frame Memory Price = $0.01\text{€/MB} = 0.00001\text{€/KB}$

Max L1 Memory = $(0.013 - 128 * 0.00001) / 0.009 = 1.3\text{KB} = 1.3 * 2^{10} \text{ B}$

MAX = 10

b)

Miss-Rate	L1 Size = 1024B	L1 Size = 512B	L1 Size = 256B
Block Size = 8B	3.05%	12.47%	19.60%
Block Size = 16B	3.63%	11.84%	18.29%
Block Size = 32B	7.70%	14.92%	22.88%
Block Size = 64B	11.81%	20.21%	33.40%

d)

L1 Config 1

Cache Size	1024B
Block Size	8B
Miss-Rate	3.05%
Cost = Price * Miss-Rate	$3.13 * 10^{(-4)} \text{ ns€}$

L1 Config 2

Cache Size	1024B
Block Size	16B
Miss-Rate	3.63%
Cost = Price * Miss-Rate	$3.73 * 10^{(-4)} \text{ ns€}$

2.3.3.

a)

L1 Config 1

Miss-Rate	1-Way	2-Ways	4-Ways	8-Ways
Compulsory	9504	9504	9504	9504
Capacity	21010	21436	23465	23401
Conflict	344230	408141	272	145
TOTAL	374744	439081	33241	33050

L1 Config 2

Miss-Rate	1-Way	2-Ways	4-Ways	8-Ways
Compulsory	4752	4752	4752	4752
Capacity	13540	14715	14695	14939
Conflict	428645	428356	0	0
TOTAL	446937	447823	19447	19691

c)

Dentro de uma configuração de cache o miss capacity & compulsory tendem em manter-se constantes, variando o número de ways. O miss conflict diminui com o aumento de ways. A configuração com maior tamanho de bloco (configuração 2) tem menor número de miss capacity & compulsory em todas as ways e tem menor miss conflict a partir das 4 ways.

d)

AMAT: $140 * (\text{prob miss L1}) + 2 * (0.7 + 0.35 * \log(\#ways, 2)) \text{ ns}$

e)

L1 Config 1

	1-Way	2-Ways	4-Ways	8-Ways
Miss-Rate	3.0455%	3.5683%	0.2701%	0.2686%
Access Time	5.7191	6.6353	2.1317	2.2123
Price	$10.28 * 10^{(-3)}$	$10.28 * 10^{(-3)}$	$10.28 * 10^{(-3)}$	$10.28 * 10^{(-3)}$
Cost Function	0.0588	0.0682	0.0219	0.0227

L1 Config 2

	1-Way	2-Ways	4-Ways	8-Ways
Miss-Rate	3.6322%	3.6394%	0.1580%	0.1600%
Access Time	6.5405	6.7349	1.9748	2.0602
Price	$10.28 * 10^{(-3)}$	$10.28 * 10^{(-3)}$	$10.28 * 10^{(-3)}$	$10.28 * 10^{(-3)}$
Cost Function	0.0672	0.0692	0.0203	0.0212

f)

Sabendo que o Miss-Rate é utilizado para calcular o access time e que o preço da cache é igual nas duas configurações, podemos apenas ter em conta o access time para avaliar o que tem menor cost function.

Dentro de cada configuração o access time tende a diminuir significativamente a partir das 4 ways. Entre configurações a partir das 4-ways o access time da configuração 2 é ligeiramente mais baixo.

2.3.4.

a)

Neste algoritmo, a melhor solução seria usar Write-Through com Write-No-Allocate uma vez que é mais simples de implementar e o número de Write-Hits é bastante reduzido.

2.3.5.

a)

L1 Config

Cache Dimension	1KB
Block Size	16B
Associativity	4-Ways
Write-Policy	Write-Through com Write-No-Allocate
Miss-Rate	0.1580%
Access Time	1.9748ns
Price	$10.28 * 10^{(-3)} \text{ €}$
Cost Function	0.0203ns€

2.4.1.

a)

L1 Cache Price + Frame Memory Price = $10.28 * 10^{(-3)} \text{ €}$

L2 Price = $0.4/1000 = 0.0004\text{€}/\text{KB}$

Max L2 Memory = $(0.013 - 10.28 * 10^{(-3)}) / 0.0004 = 6.8\text{KB} = 1.7 * 2^{12} \text{ B}$

MAX = 12

b)

	Block-Size	Miss-Rate
Block-Size = $(1 * L1_Block)$	16B	52.33%
Block-Size = $(2 * L1_Block)$	32B	26.77%
Block-Size = $(4 * L1_Block)$	64B	17.53%
Block-Size = $(8 * L1_Block)$	128B	20.47%

c)

L2 Block-Size = 64B

Pois tem a menor Miss-Rate, e tendo em conta que a diferença entre o Miss-Rate deste bloco e o Miss-Rate mais baixo para um bloco com dimensão menor é ainda bastante significativo (9.24%), não compensaria escolher o de dimensão menor, mesmo sendo o tempo de carregamento do bloco mais rápido que o de 64B.

2.4.2.

a)

	Miss-Rate
Compulsory	25.50%
Capacity	53.93%
Conflict	20.57%
TOTAL	17.53%

c)

$$\text{AMAT: } 2 * (0.7 + 0.35 * \log(4, 2)) + 0.1580 * (10 * 0.7 + 0.1753 * 140) = 7.78\text{ns}$$

d)

Miss-Rate	2.7697%
Access Time	2.8498ns
Price	$1.6010 * 10^{-3} \text{ €}$
Cost Function	$4.5625 * 10^{-3} \text{ ns€}$

2.5.

a)

	Cache L1	Cache L2	Frame Memory
Dimension [Bytes]	1024	$4 * 1024$	$128 * 1024$
Block-Size [Bytes]	16	64	-
Associativity	4-Ways	1-Way	-
Write-Policy	Write-Through com Write-No-Allocate	Write-Through com Write-No-Allocate	-
Local Miss-Rate [%]	15.80%	17.53%	-
Price [€]	0.009	$16 * 10^{-4}$	0.00128
Glocal Miss-Rate [%]	2.7697		
Glocal Access Time [ns]	2.8498		
Total Price [€]	$1.6010 * 10^{-3}$		
COST FUNCTION [ns€]	$4.5625 * 10^{-3}$		