

## Associativa

→ posicionamento mais flexível na cache → taxa de falhas

→ totalmente associativa: um bloco pode ir para qualquer posição da cache.

→ consequências:  $\text{index} = 0$ ,  $\text{tag} = 30 - \text{woff}$  e  $\oplus$  comparadores  
cada vez maiores  
↑ custo ↑ potência

→ "n-way": um bloco pode ir para n posições da cache  
conjunto

→ consequências: pesquisa em conjunto → n comparadores

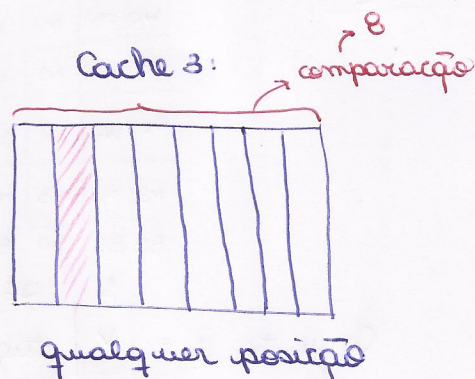
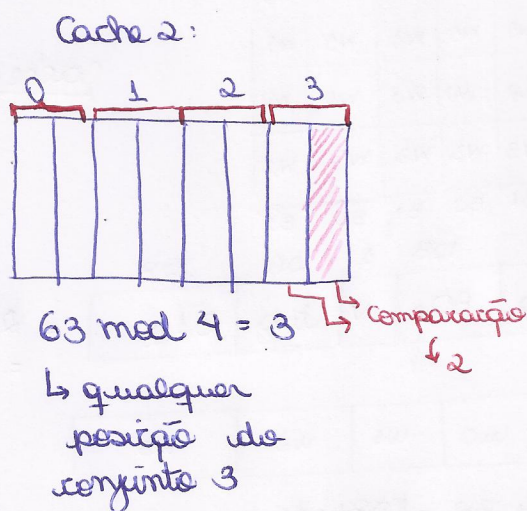
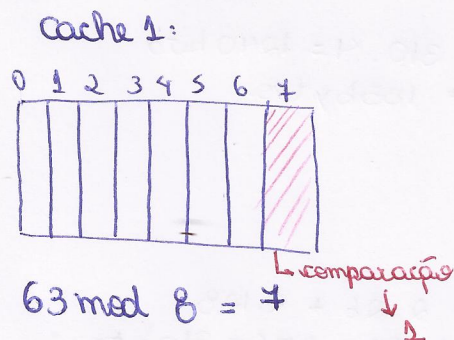
→ mapeamento:  $(\text{Block address}) \bmod (\text{número de conjuntos})$

ex: Cache 1: { 8 blocos  
1 w / bloco  
MD

Cache 2: { 8 blocos  $\Leftrightarrow$  4 conjuntos  
1 w / bloco  
2-way

Cache 3: { 8 blocos  
1 w / bloco  
Full-associative

block address: 63.

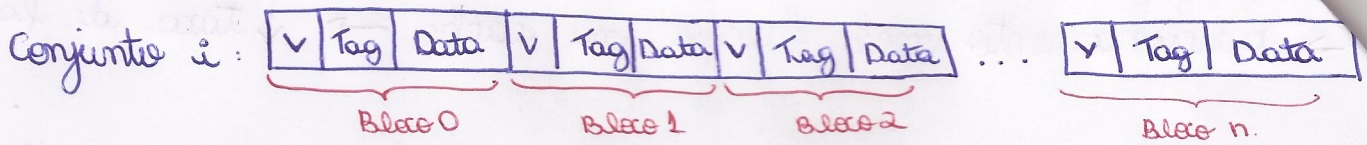


→ Conjunto Cheio, qual tirar? LRU: least recently used

→ o que não é referenciado o mais tempo.



# Estrutura



→ Quantidade de bits que uma cache precisaria

Cache 1 { 8 blocos  
4 w / bloco  
MD

Cache 2 { 8 blocos  $\Leftrightarrow$  4 conjuntos  
4 w / bloco  
2-way

Cache 3 { 8 blocos  $\Leftrightarrow$  2 conjuntos  
4 w / bloco  
4-way

Cache 1:

tag	index	W	B
25	3	2	2

Bloco i:

valido	tag	data 0	data 1	data 2	data 3
1	25	32	32	32	32

$$\begin{array}{r} 32 \\ \times 4 \\ \hline 128 \\ + 26 \\ \hline 154 \end{array}$$

$$= 1 + 25 + 4 \cdot 32 = 26 + 128 = 154$$

$$\text{Cache 1: } 154 \cdot 8 = 1232 \text{ bits} = 154 \text{ bytes}$$

tag	index	W	B
26	2	2	2

Cache 2:

0		1		2		3	
w0	w0	w0	w0	w0	w0	w0	w0
w1	w1	w1	w1	w1	w1	w1	w1
w2	w2	w2	w2	w2	w2	w2	w2
w3	w3	w3	w3	w3	w3	w3	w3
b0	b1	b0	b1	b0	b1	b0	b1

$$\text{Cache 2: } 310 \cdot 4 = 1240 \text{ bits} = 155 \text{ bytes}$$

Conjunto i = 

V	tag	B0 <sub>i</sub>
---	-----	-----------------

V	tag	B1 <sub>i</sub>
---	-----	-----------------

 ...

$\downarrow \quad \downarrow$

B0<sub>i</sub> / B1<sub>i</sub> = 

V	tag	w0	w1	w2	w3
---	-----	----	----	----	----

$= 2 + 2 \cdot 26 + 2 \cdot 128 = 2 + 52 + 256 = 310 \text{ bits}$

Cache 3:

tag	index	W	B
27	1	2	2

$$\text{Cache 3: } 2 \cdot 624 = 1248 \text{ bits} = 156 \text{ bytes}$$

Conjunto i = 

V	tag	B0
---	-----	----

V	tag	B1
---	-----	----

V	tag	B2
---	-----	----

V	tag	B3
---	-----	----

$= 4 + 4 \cdot 27 + 4 \cdot 128$

Bloco 0 / ... / Bloco 3:

V	tag	w0	w1	w2	w3

$$= 4 + 108 + 512 = 624 \text{ bits}$$

classificação: Supondo uma cache de tamanho fixo:

↑ associatividade    ↑ bloco  
conjunto    ↑ tamanho do tag    ↓ tamanho do índice

↓ número de conjuntos

Obs: Quanto mais associatividade, mais comparadores (e maiores) <sup>→ tag</sup>  
e isso causa um gasto maior e uma maior potência dinâmica  
↳ inviável para caches grandes!