

→ Arquitetura de Computadores - Ficha 3

1-

Addr S=4, E=1, B=2, m=5

5	set 2 / linha 0	miss
14	set 3 / linha 0	miss
10	set 1 / linha 0	miss
29	set 2 / linha 0	miss
4	set 2 / linha 0	miss
21	set 2 / linha 0	miss
16	set 0 / linha 0 / miss	
5	set 2 / linha 0	miss

Miss rate: 100%

$$b = \log_2(B) = \log_2 2 = 1 \Rightarrow \text{block offset}$$

$$s = \log_2 S = \log_2 4 = 2 \Rightarrow \text{set}$$

$$14 = \underline{01} \underline{11} \underline{0}$$

$$10 = \underline{01} \underline{01} \underline{0}$$

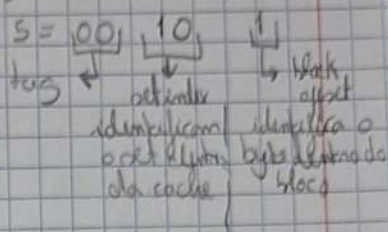
$$29 = \underline{11} \underline{10} \underline{1}$$

$$4 = \underline{00} \underline{10} \underline{0}$$

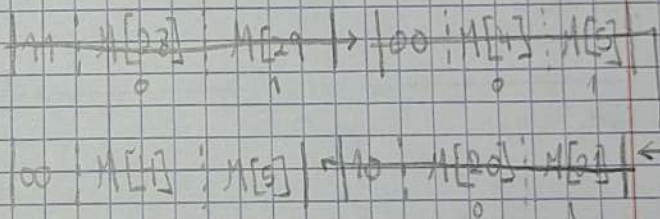
$$21 = \underline{10} \underline{10} \underline{1}$$

$$16 = \underline{10} \underline{00} \underline{0}$$

$$5 = \underline{00} \underline{10} \underline{1}$$



	tag			
S00	10	M[16]	M[17]	L0
S01	01	M[10]	M[11]	L0
S10	00	M[4]	M[5]	L0
S11	01	M[12]	M[13]	L0
	0	1		



Addr	S=1, E=4, B=2, m=5
5	set 0 / linha 0 cold miss
14	set 0 / linha 1 miss
10	set 0 / linha 2 miss
29	set 0 / linha 3 miss
4	set 0 / linha 0 hit
21	set 0 / linha 1 miss
16	set 0 / linha 2 miss
5	set 0 / linha 0 hit

Miss rate: 75%

• $b = \log_2 B = \log_2 2 = 1 \Rightarrow \text{offset}$

14 = 01110

10 = 01010

29 = 11101

4 = 00100

21 = 10101

16 = 10000

5 = 00100

• procura pela tag, se ela existir
o set vai automaticamente para essa
linha e depois é que vai o offset
pra confirmar qual o set da li

	0010	ME[3]	ME[5]	L0
50	0111	ME[14]	ME[10]	L1 →
	0101	ME[29]	ME[4]	L2 →
	1110	ME[21]	ME[16]	L3

* 1 → pos

//

24

Addr	S=2 E=2, B=2, m=5
5	set 0 / limba 0 miss
14	set 1 / limba 0 miss
10	set 1 / limba 1 miss
29	set 0 / limba 1 miss
4	set 0 / limba 0 hit
21	set 0 / limba 1 miss
16	set 0 / limba 0 miss
5	set 0 / limba 1 miss
Miss rate 87,5%	

$b = \log_2 B = \log_2 2 = 1$
 $s = \log_2 S = \log_2 2 = 1$
 $14 = 011 \textcircled{1} 0$
 $10 = 010 \textcircled{1} 0$
 $29 = 111 \textcircled{0} 1$
 $4 = 001 \textcircled{0} 0$
 $21 = 101 \textcircled{0} 1$
 $16 = 100 \textcircled{0} 0$
 $5 = 001 \textcircled{0} 1$

tag				
S0	001	M[4]	M[5]	L0
	11	M[20]	M[29]	L1
S1	011	M[4]	M[5]	L0
	010	M[20]	M[21]	L1
		0	1	

Diagram showing memory access sequence and cache state transitions:
 - Initial state: S0, tag 001, M[4], M[5], L0.
 - Access 14: L1 miss, 2R0, L1, 011, M[20], M[29].
 - Access 10: L1 miss, 2R0, L1, 010, M[20], M[21].
 - Access 29: L1 miss, 2R0, L1, 001, M[4], M[5].
 - Access 4: L0 hit, 001, M[4], M[5].
 - Access 21: L1 miss, 2R0, L1, 011, M[4], M[5].
 - Access 16: L1 miss, 2R0, L1, 010, M[20], M[21].
 - Access 5: L1 miss, 2R0, L1, 001, M[4], M[5].

2-

a)

$$m = 8$$

$$S = 8 \text{ porque } r = 3 = \log_2 S = \log_2 8$$

$$E = 1$$

$$B = 8$$

$$L = m - b - r \Leftrightarrow 2 = 8 - b - 3 \Rightarrow b = 3$$

$$\Rightarrow 1 \text{ B} = 2^3 = 8$$

$$b) \text{ 0x28} = \underbrace{00}_{\text{tag}} \underbrace{101}_{\text{offset}} 000 \Rightarrow \text{hit (valid bit = 1)}$$

$$\text{0x4a} = \underbrace{01}_{\text{tag}} \underbrace{001}_{\text{offset}} 010 \Rightarrow \text{miss (valid bit = 0 e porque o offset 111)}$$

$$\text{0x46} = \underbrace{01}_{\text{tag}} \underbrace{001}_{\text{offset}} 011 \Rightarrow \text{hit porque o anterior colocou o valid bit = 1}$$

$$\text{0x68} = \underbrace{01}_{\text{tag}} \underbrace{101}_{\text{offset}} 000 \Rightarrow \text{miss (colisão)}$$