

P3 - ORG

1) a) $I(m_i + m_d L_s)$

$$\frac{m_i + m_d L_s}{1 + L_s}$$

b) $(I) \cdot m_i \cdot p$
 $h + m_i \cdot p$

2) $\begin{cases} 16\text{-way} \\ \text{bloco: } 64 \text{ bytes} \\ 1024 \cdot 2^{10} \text{ bytes} \end{cases}$

bloco $\begin{cases} +1 : \text{valido} \\ +1 : \text{reference} \\ +1 : \text{dirty} \end{cases}$ $\{ \text{endereço} = 48 \text{ bits} \}$

a) index?

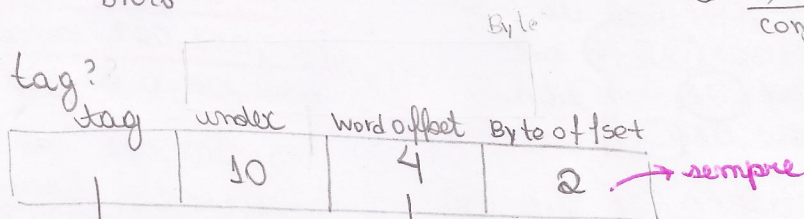
$\rightarrow 16\text{-way} = \frac{16 \text{ blocos}}{\text{conjunto}} \rightarrow \frac{64 \text{ bytes}}{\text{bloco}} = \frac{2^{20} \text{ bytes}}{2^6 \text{ bytes}} = 2^{14} \text{ blocos}$

$\rightarrow \text{Size} = 2^{20} \text{ bytes}$

$\rightarrow \frac{64 \text{ bytes}}{\text{bloco}}$

$\rightarrow \text{conjuntos} = \frac{2^{14} \text{ blocos}}{2^4 \text{ blocos/conjunto}} = 2^{10} \text{ conjuntos}$
 $\downarrow \log_2(2^{10})$
 $10 \text{ bits} = \text{índice}$

b) tag?



$= 48 - 10 - 4 - 2$
 $= 32 \text{ bits}$

$\frac{64 \text{ bytes}}{\text{bloco}} = \frac{2^6 \text{ words}}{2^2 \text{ words/bloco}} = 16 \text{ words/bloco}$
 $\log_2 16 = 4$
 $\rightarrow \text{word} = 4 \text{ bytes}$

c) Tamanho em bits de um bloco?

Bloco i:

V	R	D	data
1	1	1	64 bytes

$\text{Bloco} = 1 + 1 + 1 + 64 \cdot 8 = 547 \text{ bits}$

Cache?

Conjunto de: 16 blocos

$\text{Conjuntos} = 2^4 \cdot 547 \text{ bits}$
 $= 8752 \text{ bits}$

Cache: 2^{10} conjuntos

cache = $2^{10} \cdot 8752 \text{ bits}$
 $= 8752 \text{ K bits}$

d) penalidade_{main} = 100 ciclos (leitura)

• penalidade_{main} = 80 ciclos (escrita) • I-C e I-D main

• $I = 10^6$

• % store = $10\% = 10^{-1}$

• write-through + write-buffer

• 4 palavras / bloco \rightarrow 4 escritas + 4 leituras

• write allocate

• miss rate_{I-C} $10\% = 10^{-1}$

penalidade buffer = 0 ciclos

• stalls escrita ??

• total de escritas = programa

$= 10^6 \cdot 10^{-1} \cdot 8 \cdot 10 \cdot 4 = 32 \cdot 10^5 \text{ ciclos}$

$I \cdot \%s$
 $10^6 \cdot 10^{-1} = 10^5$

$$stalls = 10^5 \cdot 10^{-1} (\underbrace{4 \cdot 80}_{4 \text{ escritas}} + \underbrace{4 \cdot 100}_{4 \text{ leituras}})$$

↓
str prog

- write-miss: tag ≠
- ↳ substituir e carregar os adjacentes no bloco
- ↳ buffer: evita a penalidade de escrita na MP (*)
- ↳ allocate: evita a penalidade de leitura do bloco antes de substituir
- ↳ write-through + buffer

$$stalls = 10^5 \cdot 10^{-1} \cdot 4 \cdot 10^2$$

$$stalls_{escrita} = 4 \cdot 10^6 \text{ ciclos}$$

3) a)

```

Sum += 1; add $v0, $0, $0
loop: lw $t0, 0($a0)
      add $v0, $t0, $v0
      addi $a0, $a0, -4
      bne $a0, $0, loop
      jr $ra
    
```

loop unrolling

①

```

lw $t0, 0($a0)
add $v0, $t0, $v0
addi $a0, $a0, -4
bne $a0, $0, loop
    
```

```

lw $t0, 0($a0)
add $v0, $t0, $v0
addi $a0, $a0, -4
bne $a0, $0, loop
    
```

→ 4 bytes

b) 8 instruções
8 × 4 = 32 bytes

c) Emissão dinâmica: as npos desaparecem, pois o HW vai emitir quando puder e irá parar quando necessário.

logo: 6 instruções · 6 × 4 = 24 bytes

d) $\frac{2 \cdot 10^8 \text{ iterações}}{2} = 10^8 \text{ iterações c/ loop unrolling}$
 ↳ $\frac{6 \text{ instruções iterações}}{\text{iterações}} \Rightarrow 6 \cdot 10^8 \text{ instruções}$

e) 6 instruções e 4 ciclos

$$CPI_{ideal(cache)} = \frac{4}{6} = \frac{2}{3} //$$

$$CPI = \frac{\text{ciclos}}{\text{instruções}}$$

→ 2-issue
 ↳ $0,6 < CPI = 0,5$
 ↳ ideal (cache + hazards)
 ↳ faz sentido.

Full-Forwarding

① IF ID EX ME WB \$t0
 ② IF ID EX ME WB

→ dar arquivo 3-a. eds!

IF ID EX ME WB
 IF ID EX ME WB

3) $mru = 0,01 = 10^{-2}$ $\%LS = \frac{1}{4} = 0,25$

$p = 100$ ciclos

$CPI_{total} = CPI_{ideal} + CPI_{stall} \cdot \frac{ciclos_{stall}}{I}$

$CPI_{total} = \frac{2}{3} + \frac{1}{4} = \frac{2^8 + 1 \cdot 3}{12}$

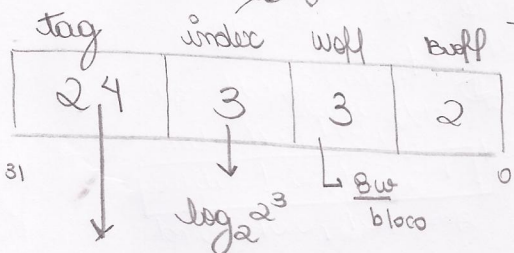
$CPI_{total} = \frac{11}{12}$

$ciclos_{stall} = (I \cdot \frac{1}{4}) \cdot 10^{-2} \cdot 10^2 = \frac{I}{4}$

4) • 2-way • 128 palabras • 8 palabras • LRU

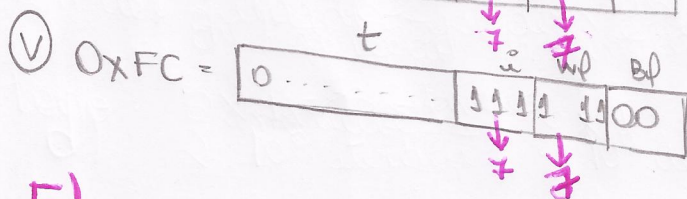
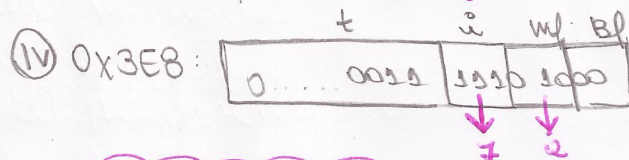
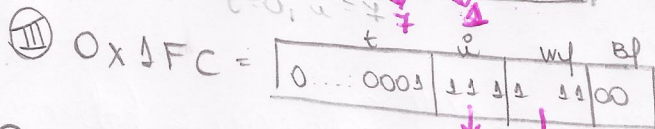
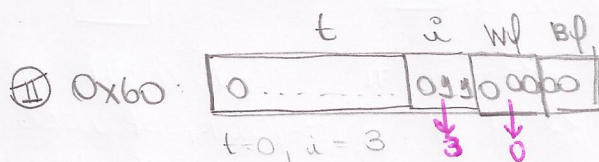
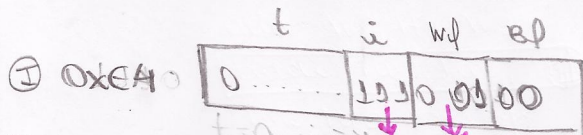
$\rightarrow \frac{128}{8} = \frac{2^7}{2^3} = 2^4$ bloques

2^4 bloques = 2^3 conjuntos



$32 - 3 - 3 - 2 = 24$

Endereços = ① 0xEA, ② 0x60, ③ 0x1FC, ④ 0x3E8, ⑤ 0xFC
6 byte address!



• char is requiring 4.eds!

5) `addi $s0, $0, 0`
`addi $s1, $0, 8` (?)

loop: `addi $s0, $s0, 1 # $s0++`
`bne $s1, $0, loop # se $s1 != 0 -> loop`
`addi $s1, -4 #`

6) $16\text{Hz} \rightarrow T = 3\text{ns}$
 $L1: 1\text{ns} \rightarrow 1 \text{ ciclo}$
 $L2: 16\text{ns} \rightarrow 16 \text{ ciclos}$
 $MP: 200\text{ns} \rightarrow 200 \text{ ciclos}$
 $\rightarrow 1/LIS = \frac{1}{9} \rightarrow L1\text{mr} = \frac{5}{100} \rightarrow gmr = \frac{2}{100}$
 $\rightarrow CPI_{base} = 2$

a) $a_{L2} = \left(I + \frac{I}{4} \right) \cdot 0,05 = \frac{I}{16}$
 (unificado!)

b) $a_{MP} = \left(I + \frac{I}{4} \right) \cdot 0,02 = \frac{I}{40}$
 (unificado)

c) $CPI = 2 + \left(\frac{I}{16} \cdot 16 + \frac{I}{40} \cdot 200 \right) / I$
 (ciclos/instrução)
 (ciclos!)

$CPI = 2 + 6 = 8$

7) olhar arquivo 7.ods!

