

# SIMULAÇÃO DE UMA MEMÓRIA CACHE

## *Split, 2 way set associative com LRU*

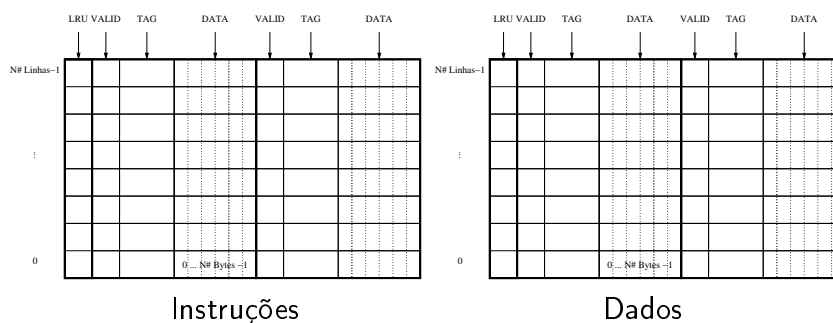
*Nuno Nunes & Paulo Bluemel*

### Tipo de Cache

**Split** - Cache separada para instruções e dados

**2 way set associative** - 2 entradas para o mesmo endereço da cache

**LRU** - indicador da última entrada usada (*least recently used*)



### Programa em "C"

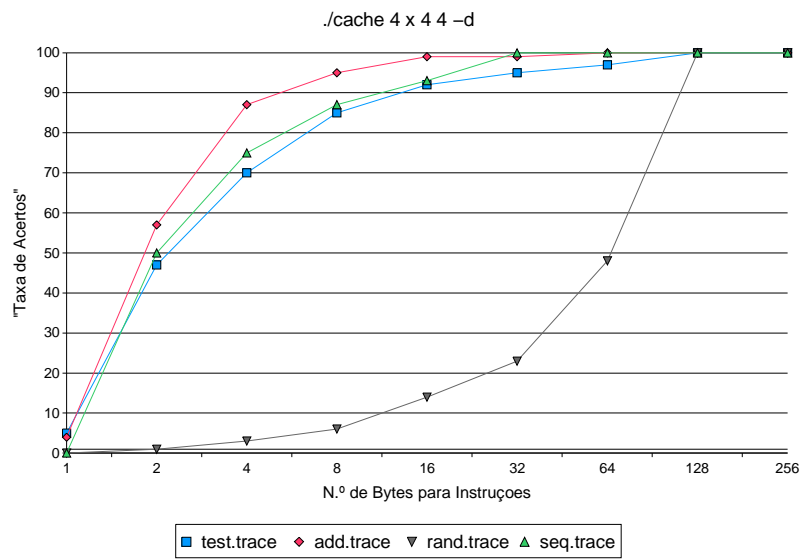
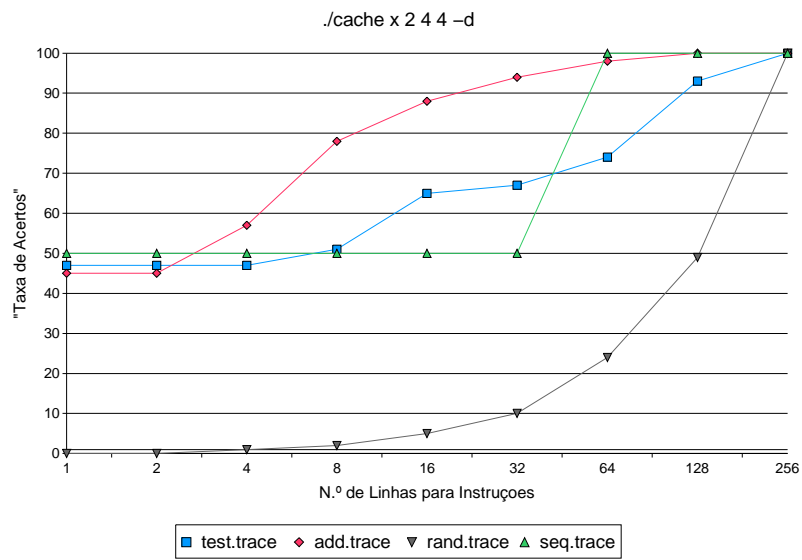
- Cálculo da linha da cache e *Tag*
- *Se* (Valid = 1) && (Tag= *Tag*)  
→ *Acerto de Cache* (cachehit++)  
(Actualiza-se LRU)
- *Senão*  
→ *Falha de Cache* (cachemiss++)  
(Tag[!*LRU*]= *Tag*; Valid[!*LRU*]=1; Data[!*LRU*]= *Data*; LRU=!LRU;

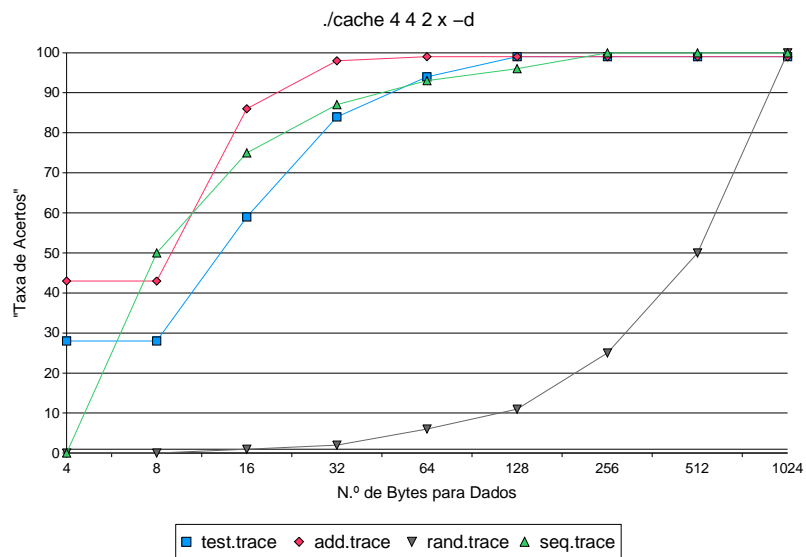
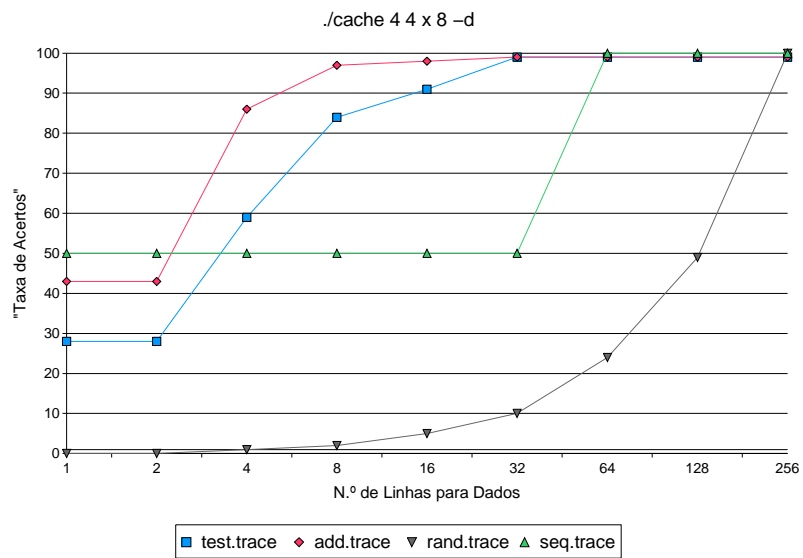
Simulação com os ficheiros:

*test.trace* e *add.trace*

Adicionais :

*seq.trace* (endereço 256 bytes) e *rand.trace* (aleatório) (endereço 1024 bytes)





## CONCLUSÕES FINAIS

- $\uparrow$  Linhas *e/ou*  $\uparrow$  N.<sup>o</sup> bytes p. linha  $\Rightarrow$   $\uparrow$  Taxa de acertos
- Cache *pequena* (taxa de acertos *baixa*)  
 $\uparrow$  tamanho  $\Rightarrow$   $\uparrow\uparrow\uparrow$  Taxa de acertos  
 Cache *grande* (taxa de acertos *alta*)  
 $\uparrow$  tamanho  $\Rightarrow$   $\uparrow$  Taxa de acertos
- Melhor relação tamanho / performance ?

Ficheiro	Config. da Cache Linhas $\times$ Bytes	Tamanho mínimo	Taxa de Acertos
test.trace	2 $\times$ 32	128	93%
	2 $\times$ 128	512	98%
	128 $\times$ 4	1024	99%
add.trace	8 $\times$ 4	64	93%
	4 $\times$ 16	128	99%
	64 $\times$ 4	1024	100%

- No caso geral :

$$test.trace, add.trace = \sum \left\{ \begin{array}{l} \% \quad seq.trace \\ \% \quad rand.trace \\ \% \quad constante.trace \end{array} \right\}$$

- Para  $seq.trace \approx test.trace \approx add.trace$  ( $n \rightarrow n.$ <sup>o</sup> de bytes por linha):

$$Taxa \ de \ Acertos = \frac{2^n - 1}{2^n} \quad ; n \geq 0$$

- $-d$  (2x)  $\approx$  Taxa de acertos
- *cacheprof* (confirmação de resultados)