Caches

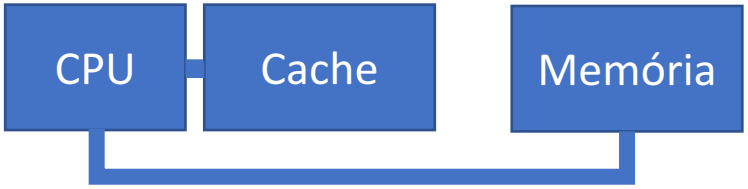
Cache

- Acesso à memória é lento
  - Tempo de processador desperdiçado a fazer nada

Memory technology	Typical access time	$ per GiB in 2012
Processor register	1 clock cycle (0.5 ns @ 2 GHz)	$???
DRAM semiconductor memory	50–70 ns	$10–$20
Flash semiconductor memory	5,000–50,000 ns	$0.75–$1.00
Magnetic disk	5,000,000–20,000,000 ns	$0.05–$0.10

Cache

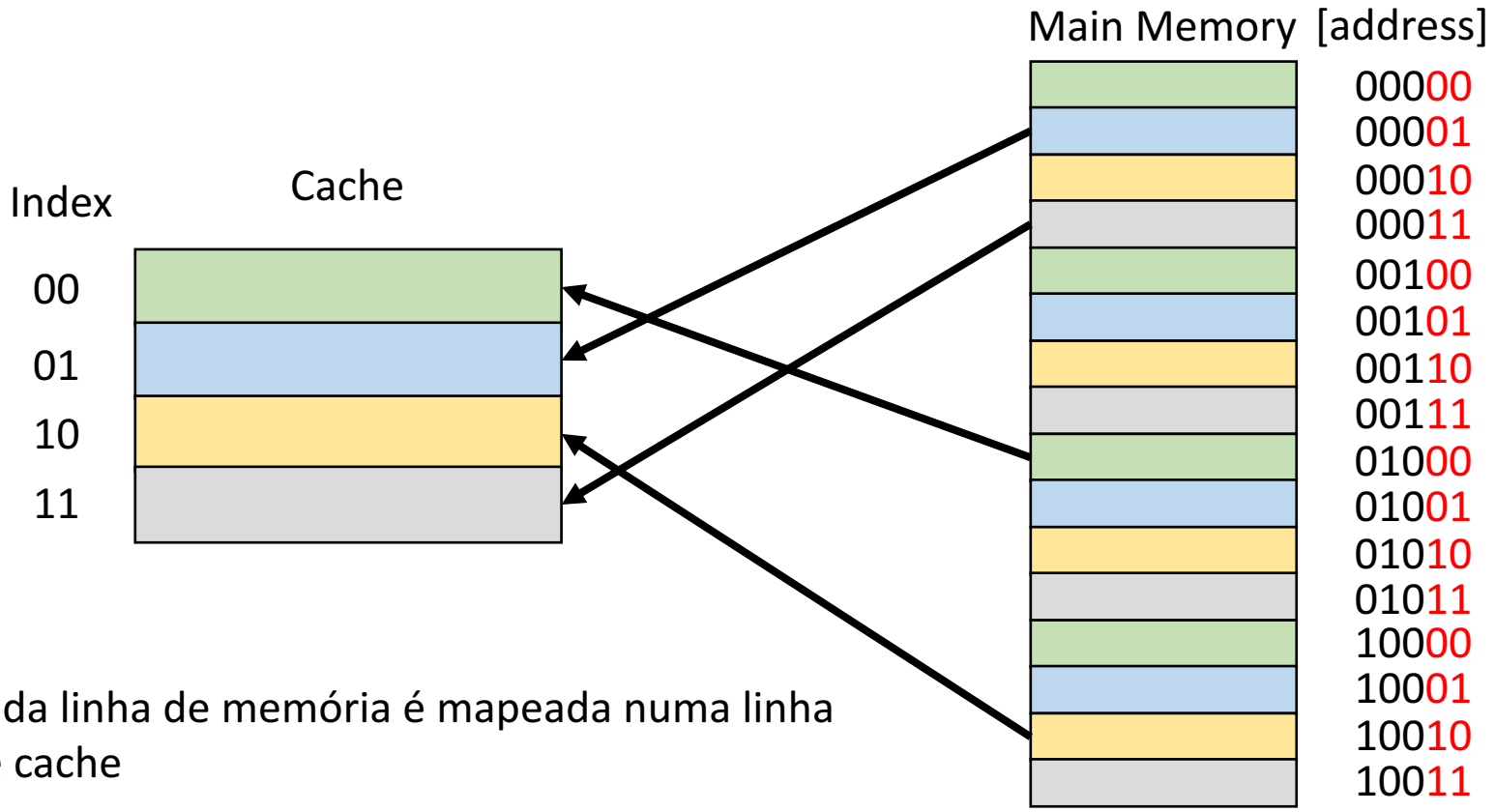
- Cache permite trazer dados para mais perto do processador e reduzir o tempo gasto a fazer nada



Memory technology	Typical access time	$ per GiB in 2012
Processor register	1 clock cycle	$???
SRAM semiconductor memory	**0.5–2.5 ns (5 a 10 clock cycles)**	**$500–$1000**
DRAM semiconductor memory	50–70 ns	$10–$20
Flash semiconductor memory	5,000–50,000 ns	$0.75–$1.00

Exemplo (simples)

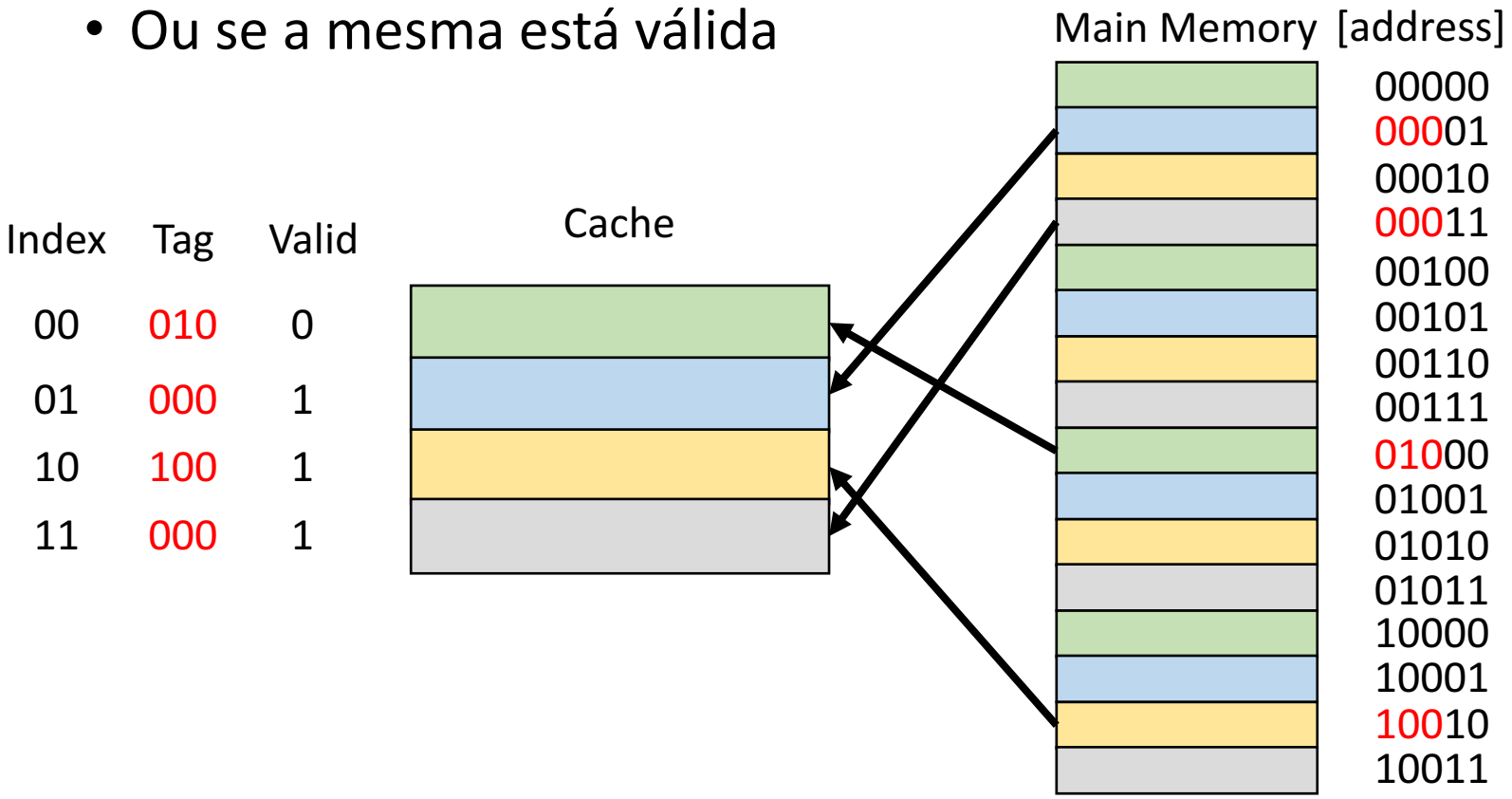
- Cache de **mapeamento direto**



Cada linha de memória é mapeada numa linha de cache

Exemplo (simples)

- Mas é preciso saber se uma dada linha de memória está presente em cache
  - Ou se a mesma está válida



Cache mapeamento direto

- Cada linha de memória é mapeada numa determinada linha de cache
  - O endereço de memória define a linha de cache que pode conter os dados
- Utiliza os endereços de memória para verificar a presença dos dados na cache
  - Comparando a “tag do endereço” com a tag na linha de cache correspondente
- Bit validade define se a linha é ou não válida
  - Se os dados lá contidos podem ser usados

- Quando é que se dá um hit na cache? E um miss?

Cache associativa pura

- **Qualquer linha** de memória pode ser mapeada em **qualquer linha** de cache
- Utiliza os endereços de memória para verificar a presença dos dados na cache
  - Comparando a “tag do endereço” com a tag na linha de cache correspondente
- Bit validade define se a linha é ou não válida
  - Se os dados lá contidos podem ser usados

- Qual o problemas de caches associativas puras?

Cache associativa por conjuntos (compromisso)

- Cada linha de memória é mapeada num determinado **conjunto** da cache
  - Podendo ser mapeada em qualquer linha do conjunto
- Utiliza os endereços de memória para verificar a presença dos dados na cache
  - Comparando a "tag do endereço" com a tag na linha de cache correspondente
- Bit validade define se a linha é ou não válida
  - Se os dados lá contidos podem ser usados

- Vantagens face a uma cache associativa pura? E face a uma cache de mapeamento direto?

Nomenclatura

- Bloco ou linha
  - Unidade base de armazenamento da cache
  - Associado a uma mesma **tag**
  - Pode conter múltiplos(as) bytes/words

- Conjunto
  - Linha física da cache
  - Associado a um mesmo **index**
  - Pode conter **múltiplas tags**
  - Depende da configuração da cache

Configurações de cache

Mapeamento Directo
(ou Associativo de 1 via)

Block	Tag	Data
1		
2		
3		
4		
5		
6		
7		
8		

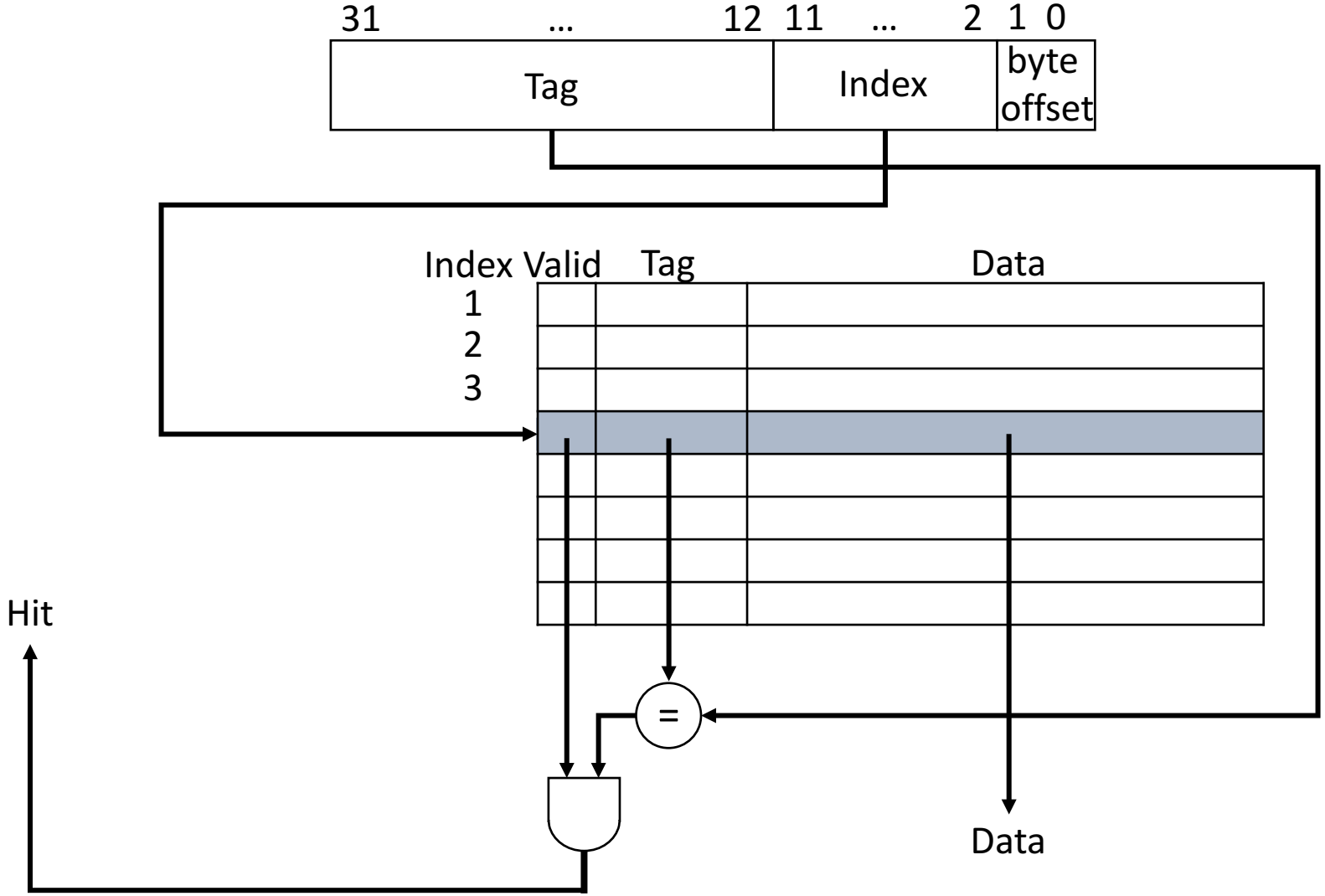
Mapeamento
Associativo de 2 vias

Set	Tag	Data	Tag	Data
1				
2				
3				
4				

Mapeamento
Associativo de 4 vias

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
1								
2								

Cache mapeamento directo (exemplo)

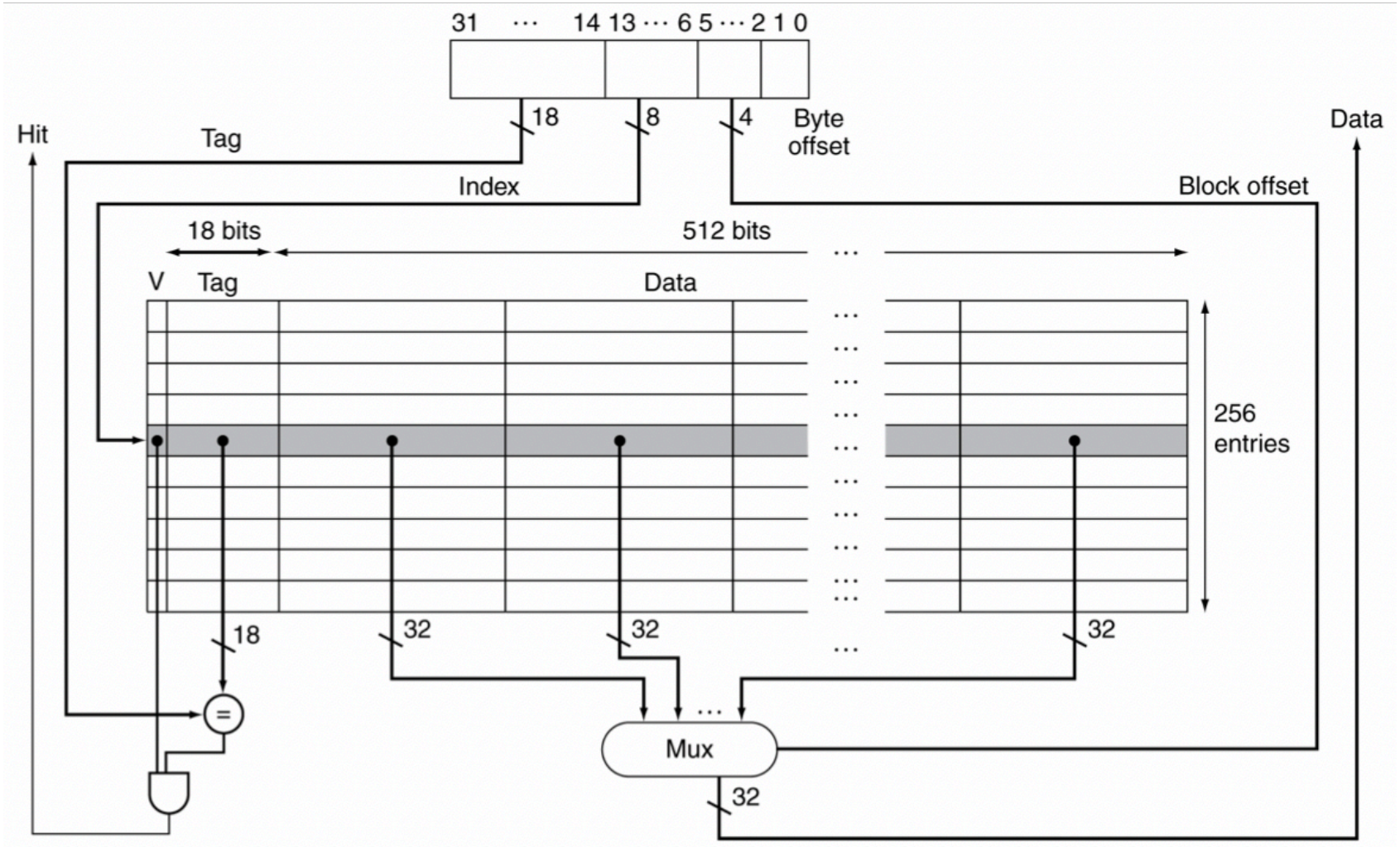


Cache mapeamento directo (exemplo)

- Endereços de 32 bits
  - 20 bits para tag
  - 10 bits para index
  - 2 bits para byte offset

- Nº de linhas da cache:2^{10} linhas ($2^{\text{bits de index}}$)
- Dimensão da linha = 2^0 x 2^2 bytes ($2^{\text{dim block offset}}$ x $2^{\text{dim byte offset}}$)
- Dimensão da cache = nº de linhas x dimensão da linha = 2^{10} x 2^2 bytes = 2^{12} bytes

Cache mapeamento directo (exemplo 2)

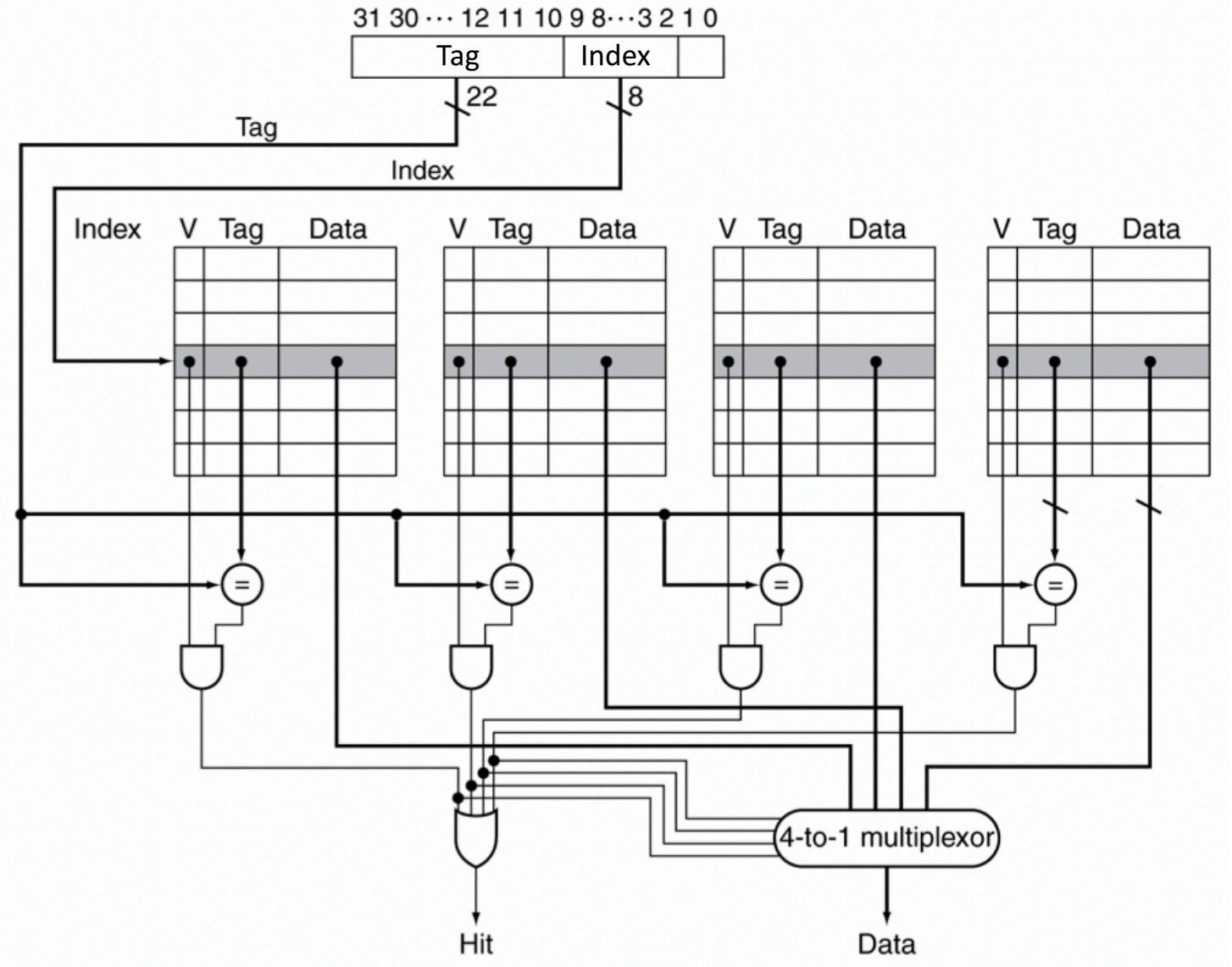


Cache mapeamento directo (exemplo 2)

- Endereços de 32 bits
  - 18 bits para tag
  - 8 bits para index
  - 4 bits para block offset
  - 2 bits para byte offset

- Nº de linhas da cache?
- Dimensão da linha?
- Dimensão da cache?

Cache associativa 4 vias (exemplo)



Cache associative 4 vias (exemplo)

- Endereços de 32 bits
  - 22 bits para tag
  - 8 bits para index
  - 2 bits para byte offset

- Nº de conjuntos da cache?
- Nº de linhas da cache?
- Dimensão do conjunto?
- Dimensão da cache?

Exercício

- Considere uma cache de mapeamento direto com dimensão 16 kbyte (2^{14} bits) e blocos de 4 words, endereçamento ao byte com endereços de 32 bits.

Quantos bits devem ser usados para index?

Quantos bits devem ser usados para tag?

Qual a dimensão (real) desta cache, em bits?