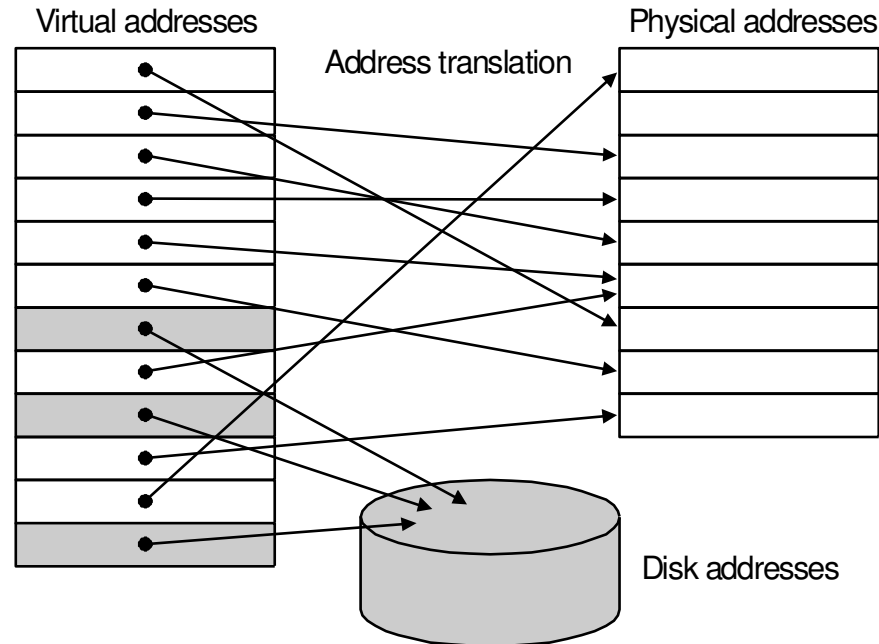

Capítulo

Sistemas de Memória

- Memória Virtual

Memória Virtual

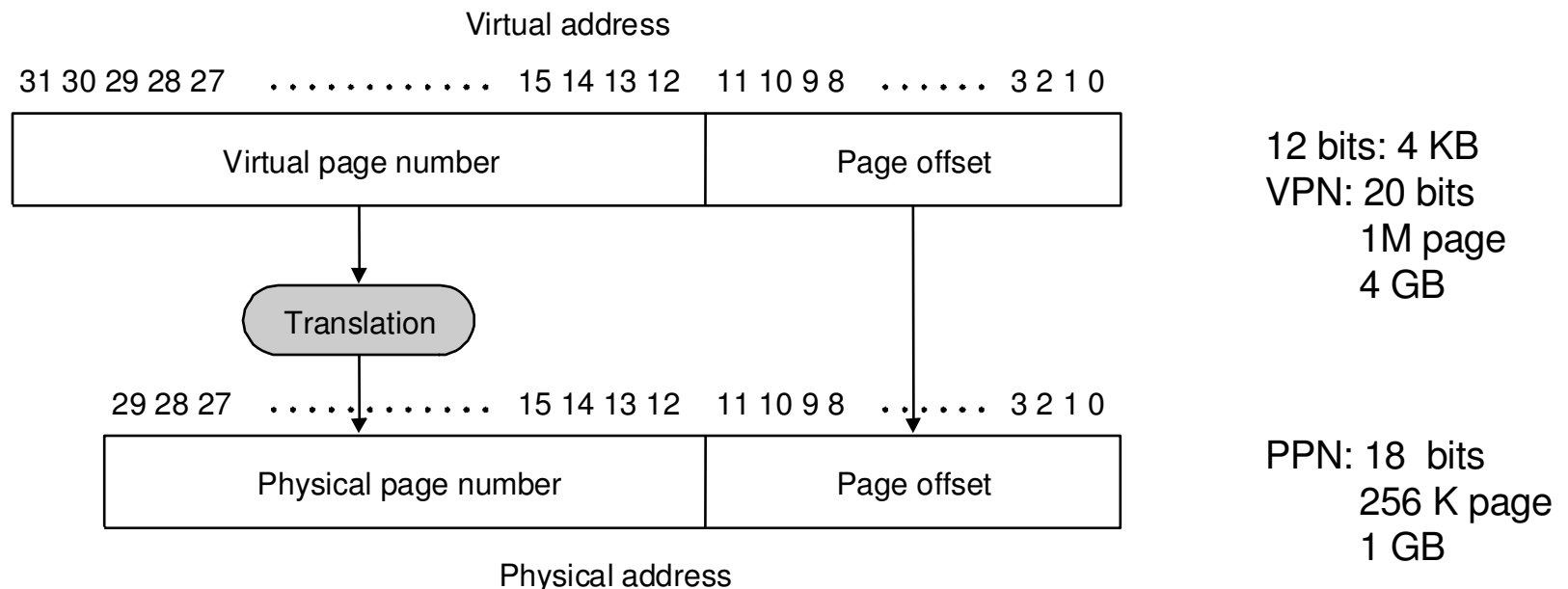
- **Memória principal funciona como uma cache para o armazenamento secundário (disco)**



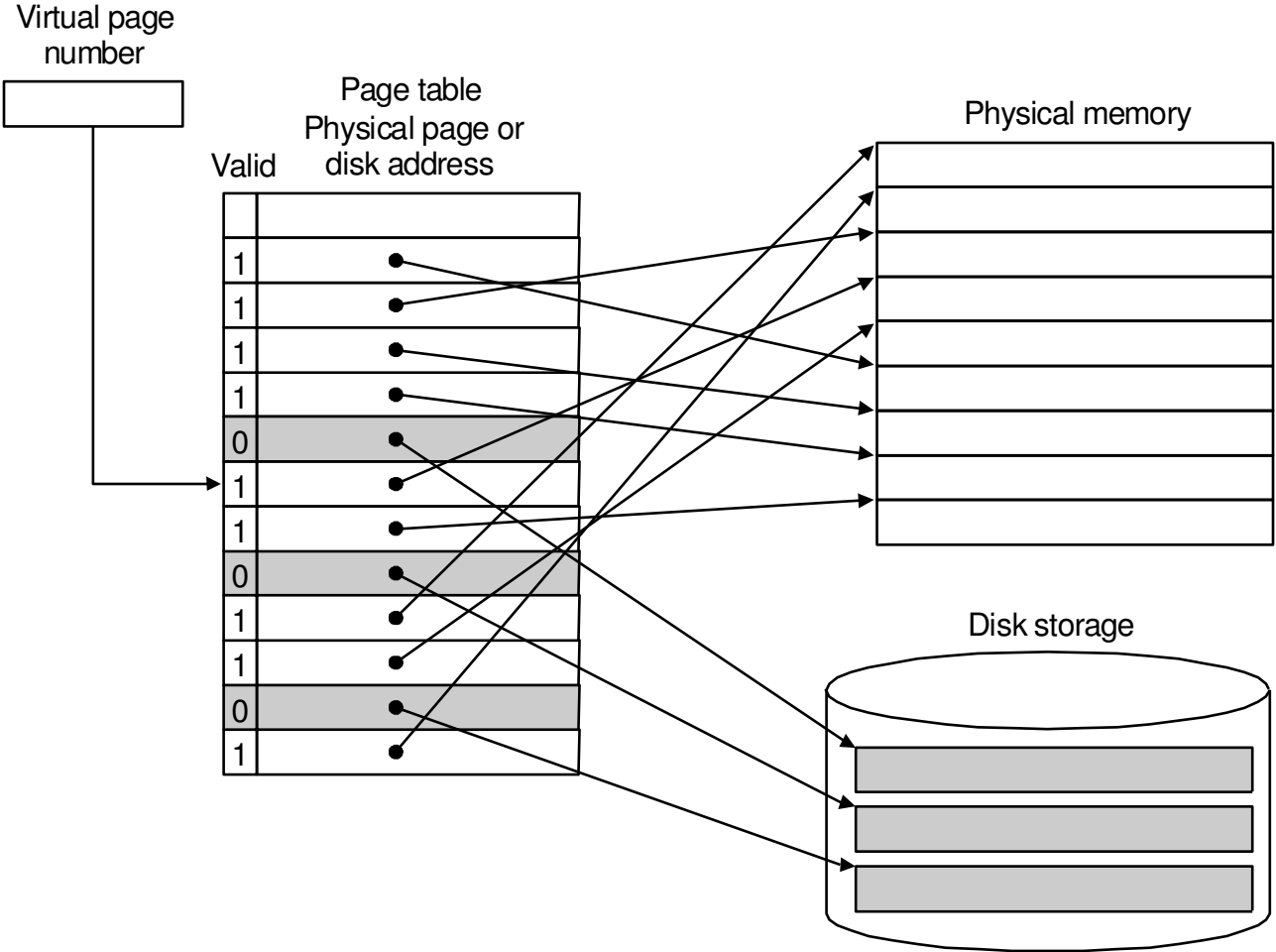
- **Vantagens:**
 - Ilusão de possuir mais memória do que realmente possui (programa independente da configuração do hardware)
 - Realocação do programa
 - Proteção (Espaço de endereços)

Páginas: blocos de memória virtual

- Falhas de página: o dado não está na memória, recuperá-lo do disco
 - Grande miss penalty, páginas são grandes! (e.g., 4KB)
 - Reduzir misses (falhas) de páginas é importante
 - Pode manipular esses misses em software
 - Usar write-through é muito caro

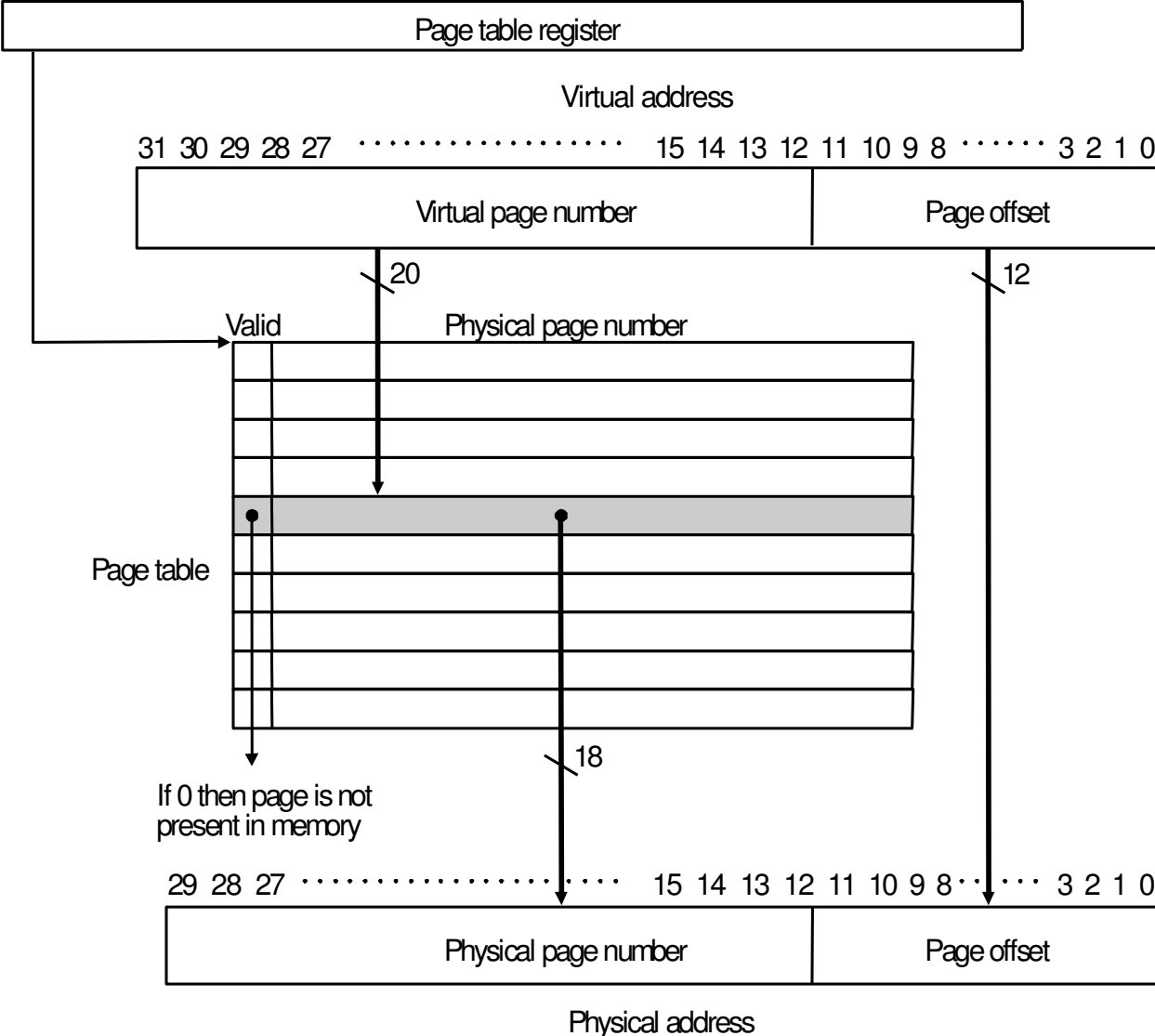


Tabelas de páginas



Tabelas de páginas

- uma TP por processo
- estado:
 - TB
 - PC
 - registradores



Política de substituição e tamanho da TP

- **Se page fault (bit válido= 0)**
 - sistema operacional executa a carga da página
- **Para minimizar page faults, política de substituição mais usada: LRU**

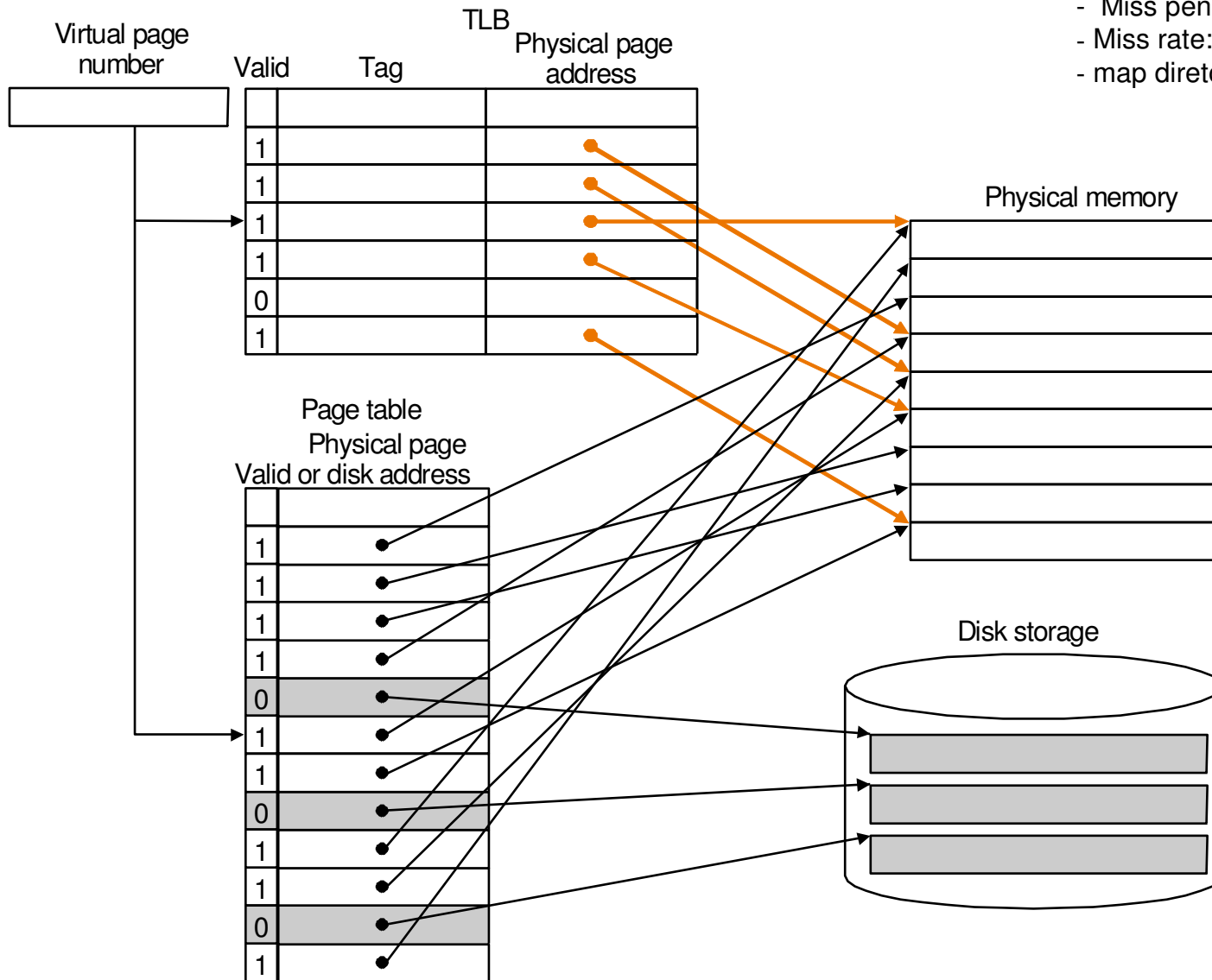
- **Tamanho da PT (p/ end 32 bits, pag de 4KB, 4B / linha da PT)**
 - número de linhas: $2^{32} / 2^{12} = 2^{20}$
 - tamanho da PT = 4 MB
 - 1 PT por programa ativo !!
 - para reduzir área dedicada para PT: registradores de limite superior e inferior

- **PT também são paginados**

TLB: Translation Lookaside Buffer

Typical values

- TLB size: 32 - 4,096 entries
- Block size: 1 - 2 page table entries
- Hit time: 0.5 - 1 clock cycle
- Miss penalty: 10 - 30 clock cycle
- Miss rate: 0.01% - 1%
- map direto ou fully associativo



TLBs

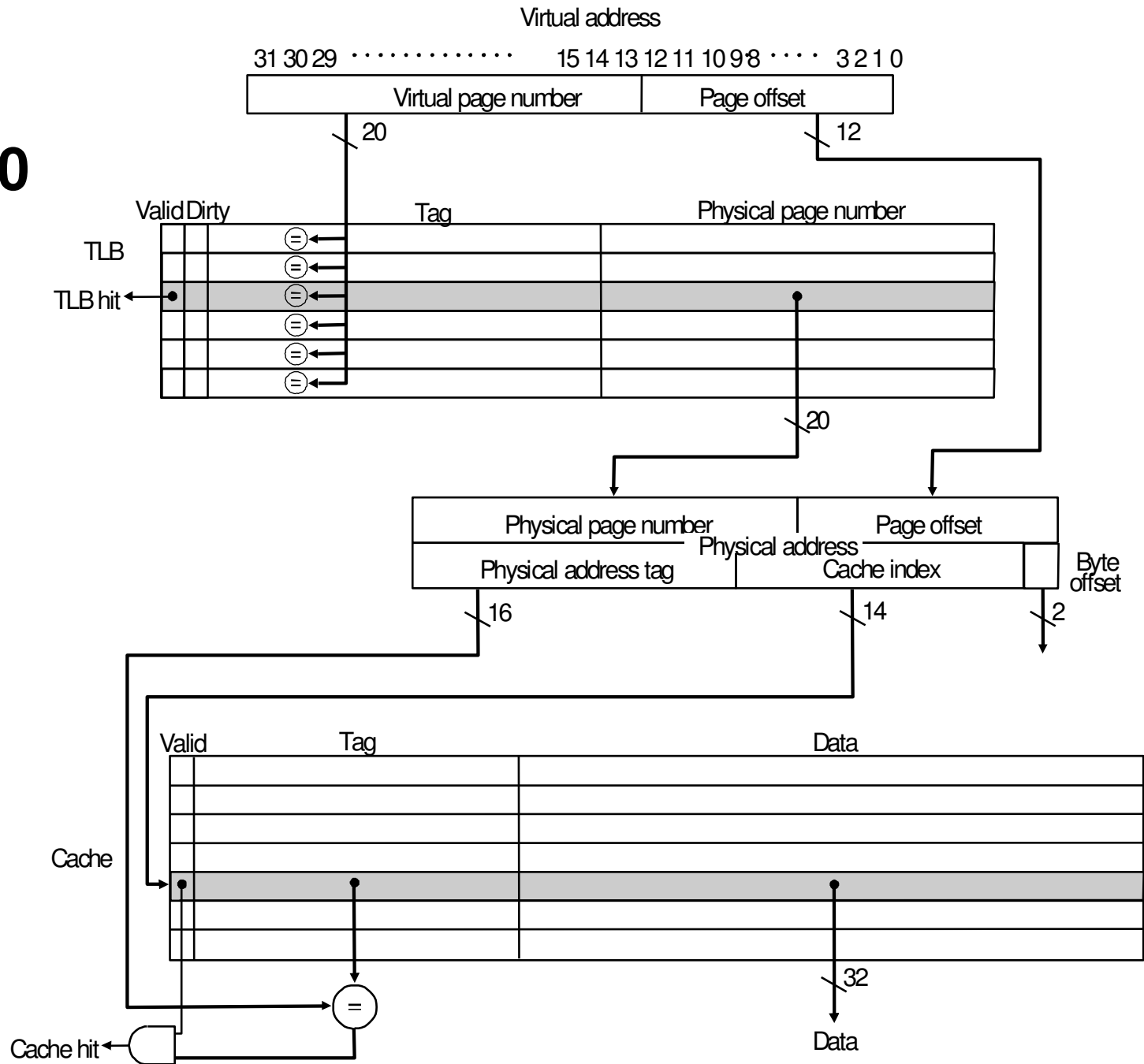
Cache

DEC 3100

- mapeamento fully associative

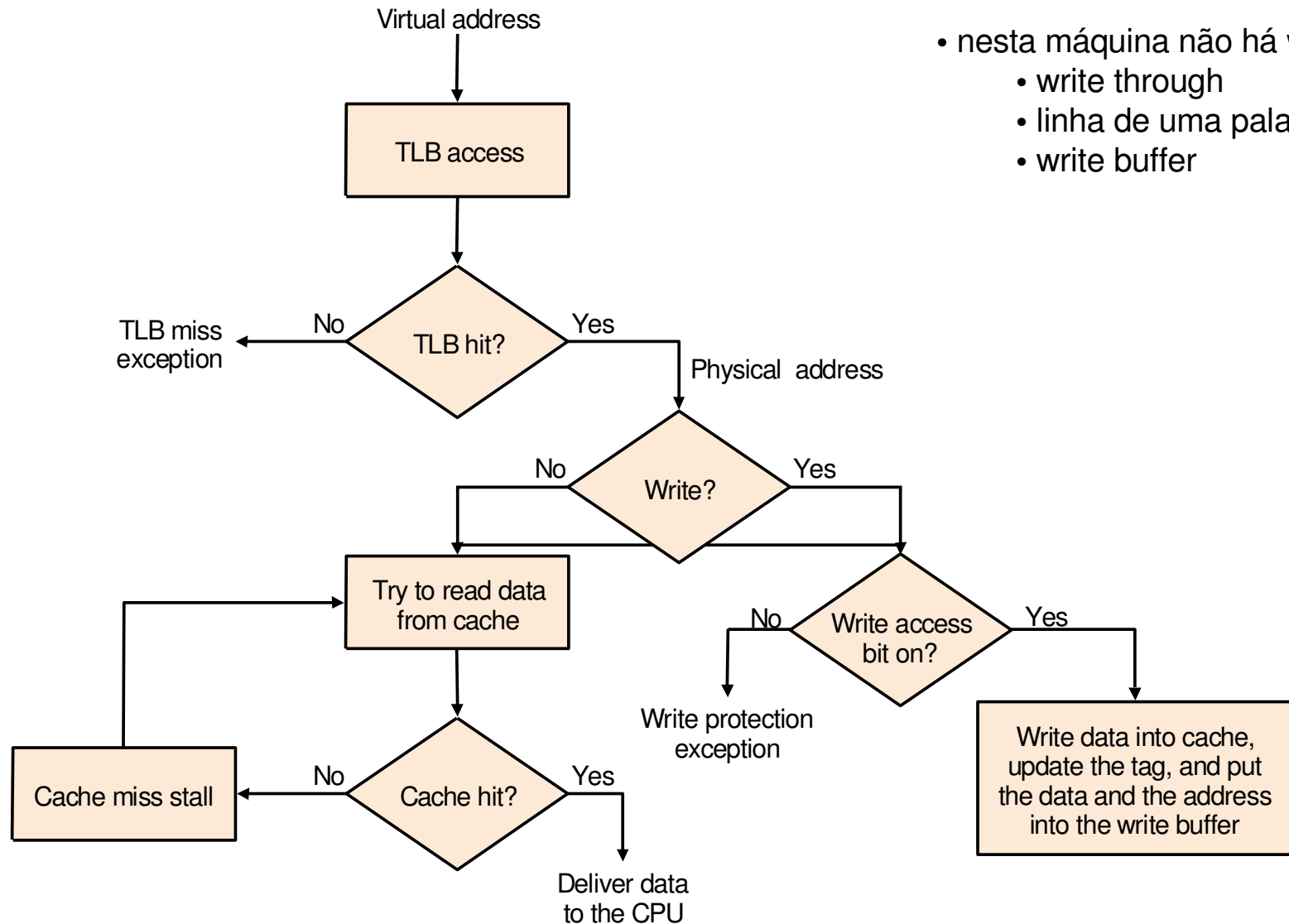
- mapeamento direto

- pior caso: 3 misses TLB, PT, cache



TLBs and caches (DEC 3100)

- nesta máquina não há write hit
 - write through
 - linha de uma palavra
 - write buffer



TLB, Memória Virtual e Caches

Cache	TLB	Virtual memory	Possible? If so, under what circumstance?
Miss	Hit	Hit	Possible, although the page table is never really checked if TLB hits
Hit	Miss	Hit	TLB misses, but entry found in page table; after retry data is found in cache
Miss	Miss	Hit	TLB misses, but entry found in page table; after retry data misses in cache
Miss	Miss	Miss	TLB misses and is followed by a page fault; after retry, data must miss in cache
Miss	Hit	Miss	Impossible: cannot have a translation in TLB if page is not present in memory
Hit	Hit	Miss	Impossible: cannot have a translation in TLB if page is not present in memory
Hit	Miss	Miss	Impossible: data cannot be allowed in cache if the page is not in memory

Proteção com Memória Virtual

- **Suporta, pelo menos, dois modos**
 - **Processo do usuário**
 - **Processo do sistema operacional** (*kernel, supervisor, executive*)
- **Estado da CPU que o processo do usuário pode ler mas não escrever**
 - Instruções especiais que são disponíveis apenas no modo supervisor
- Mecanismos onde a CPU pode ir do modo usuário para o supervisor e vice-versa
 - Usuário para supervisor: *chamada de exceção*
 - Supervisor para usuário: retorno de exceção
- OBS: tabelas de páginas (espaço de endereços do SO).

Manipulando Falhas de Páginas e Misses na TLB

- TLB miss (*software ou hardware*).
 - Página não está na memória, necessita criar apenas um miss na entrada na TLB
 - Página não está na memória, necessita transferir controle para o SO para que “ele” manipule falhas de páginas
- Falha de página (*mecanismo de exceção*).
 - SO sabe o estado do processo ativo
 - EPC = endereço virtual da página que gerou o miss
 - SO deve finalizar após três passos:
 - Consultar a entrada na TP using o endereço virtual e encontrar a localização da página referenciada no disco
 - Escolher uma página física a ser substituída; se a página escolhida está **dirty**, deve ser escrita no disco antes de trazer a nova página virtual para essa localização física
 - Iniciar uma leitura para trazer a página referenciada do disco para a página física no disco

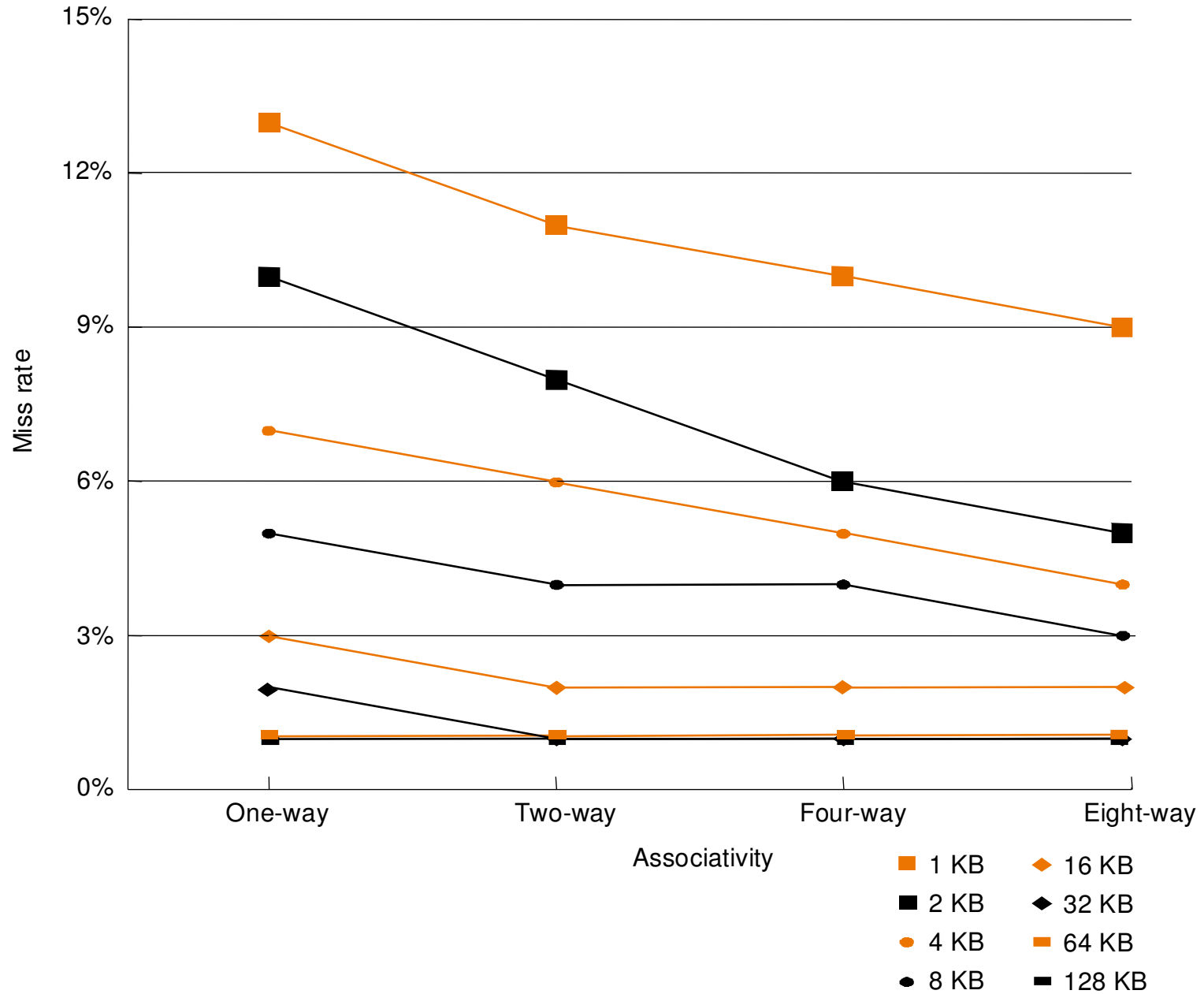
Hierarquias de Memória

- Onde um bloco pode ser colocado?

Scheme name	Number of sets	Block per set
Direct mapped	Number of blocks in cache	1
Set associative	$\frac{\text{Number of blocks in cache}}{\text{Associativity}}$	Associativity (typically 2 – 8)
Fully associative	1	Number of block in the cache

Feature	Typical values for cache	Typical values for page memory	Typical values for a TLB
Total size in blocks	1000 – 100,000	2000 – 250,000	32 – 4,000
Total size in kilobytes	8 – 8,000	8000 – 8,000,000	0.254 – 32
Block size in bytes	16 – 256	4000 – 64,000	4 – 32
Miss penalty in clocks	10 – 100	1,000,000 – 10,000,000	10 – 100
Miss rate	0.1% -- 10%	0.00001% -- 0.0001%	0.01% -- 2%

Miss rate vs conjuntos asociativos



Hierarquias de Memória

- Como um bloco é encontrado?

Associativity	Location method	Comparisons required
Direct mapped	Index	1
Set associative	Index the set, search among elements	Degree of associativity
Full	Search all cache entries	Size of the cache
	Separate lookup table	0

- OBS.: Em sistemas de memória virtual
 - Associatividade completa é benéfico, desde que misses são muito custosos
 - Associatividade completa possibilita que o software utilize esquemas de substituição projetados para reduzir o tempo de miss
 - Associatividade completa pode ser facilmente indexado sem hardware extra e sem pesquisa requerida
 - Uma página “grande” significa que a página tem pouca sobrecarga

Hierarquias de Memória

- Por que o bloco deveria ser substituído em um cache miss?
 - Random : blocos candidatos são randomicamente selecionados
 - Least Recently Used (LRU): O bloco substituído é o bloco com o maior tempo de inatividade

Hierarquias de Memória

- O que acontece em uma escrita?
 - Write-through
 - Misses são simples e baratos porque eles nunca requerem um bloco ser escrito em um nível mais baixo
 - É mais fácil de implementar que write-back, embora, para ser útil, necessita de um write-buffer
 - Write-back (copy-back)
 - Palavras individuais podem ser escritas pelo processador na taxa em que a cache aceita-as.
 - Múltiplos writes dentro de um bloco requerem apenas 1 escrita no nível mais baixo de memória.
 - Quando blocos são reescritos, o sistema pode fazer uso efetivo de uma transferência com alta taxa de largura de banda, pelo fato que o bloco interior é escrito

Algumas questões

- **Velocidade do processador continua aumentando muito rápido**
 - muito mais rápido que evolução das DRAMs ou mesmo tempo de acesso aos disco
- **Desafio de projeto: lidar com essa disparidade**
- **Tendências:**
 - **SRAMs síncronas**
 - **Reprojetar chips de DRAM para fornecer alta largura de banda e processamento**
 - **Reestruturar código para aumentar localidade**
 - **Usar pré-busca (torna cache visível ao ISA)**

Evolução desempenho CPU vs Mem

