

MC722

**Organização de Computadores
Teoria e Prática**

2012

Prof. Paulo Cesar Centoducatte
ducatte@ic.unicamp.br
www.ic.unicamp.br/~ducatte

MC722
6.1

MC542

Arquitetura de Computadores

Sistemas de Memória Virtual

"DDCA" - (Capítulo 8)
 "COD" - (Capítulo 7)

MC722
6.2

Sumário

- Memória Virtual
- Páginas: Virtual Memory Blocks
- Virtual Page Number X Physical Page Number
- Page Tables (PT)
- Política de Substituição e Tamanho da Page Table
- TLB: Translation Lookaside Buffer
- TLB e cache (DEC 3100)
- TLB, Virtual memory e Cache
- Proteção com Virtual Memory
- Tratamento de Page Faults e TLB misses
- Miss rate vs set associativity

MC722
6.3

Hierarquia de Memórias

Technology	cost / GB	Access time
SRAM	~ \$10,000	~ 1 ns
DRAM	~ \$100	~ 100 ns
Hard Disk	~ \$1	~ 10,000,000 ns

- Physical Memory: DRAM
- Virtual Memory: Hard disk
 - Lento, Grande, Barato

MC722
6.4

Hard Disk

Gasta-se Milisegundos para localizar (*seek*) a posição no disco

MC722
6.5

Memória Virtual

- Ilusão de uma memória maior do que a que realmente se tem (sem os altos custos da DRAM)
- A memória física (Physical Memory - DRAM) age como uma cache para a memória virtual (Virtual memory - hard disk)
- Somente um sub-conjunto da memória virtual está na memória física
- O Processador gera endereços virtuais (virtual addresses) porém faz acesso à memória física (physical addresses)
- Endereços virtuais têm que serem traduzidos em endereços físicos

MC722
6.6

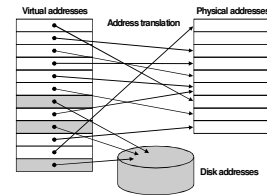
Memória Virtual

- **Page Size:** Quantidade de memória transferida do disco para a DRAM (bloco)
- **Address Translation:** determina o endereço físico a partir do endereço virtual
- **Page Table:** lookup table usada para traduzir o endereço virtual em endereço físico

MCT22
6.7

Memória Virtual

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



- **Vantagens:**
 - Ilusão de que há mais memória física (programa independente da configuração do hardware)
 - Relocação de programa
 - Proteção (do espaço de endereçamento)

MCT22
6.8

Cache/Virtual Memory

Cache	Virtual Memory
Block	Page
Block Size	Page Size
Block Offset	Page Offset
Miss	Page Fault
Tag	Virtual Page Number

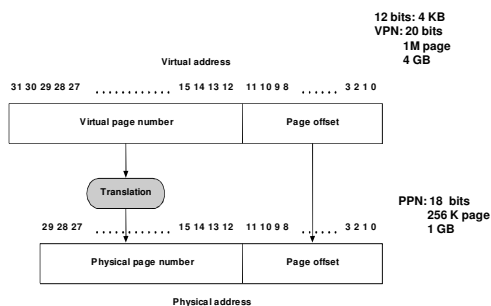
MCT22
6.9

Páginas: Virtual Memory Blocks

- **Page faults:** O dado não está na memória, deve ser buscado no disco
 - miss penalty elevado, assim as páginas devem ser relativamente grandes (i.e., 4KB)
 - Redução de faults é importante (LRU)
 - Pode-se tratar as faults por software no lugar de hardware
 - Uso de write-through é muito caro, usa-se write-back

MCT22
6.10

Virtual Page Number X Physical Page Number

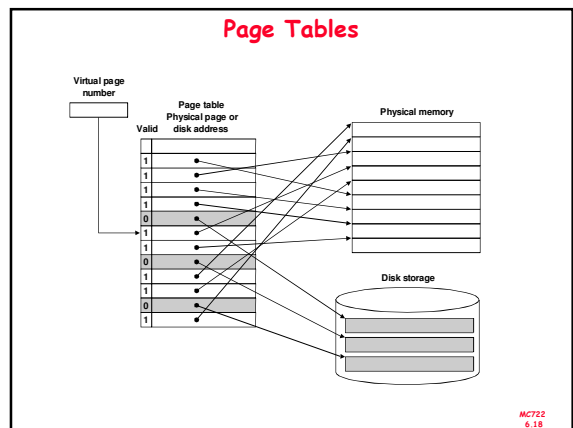
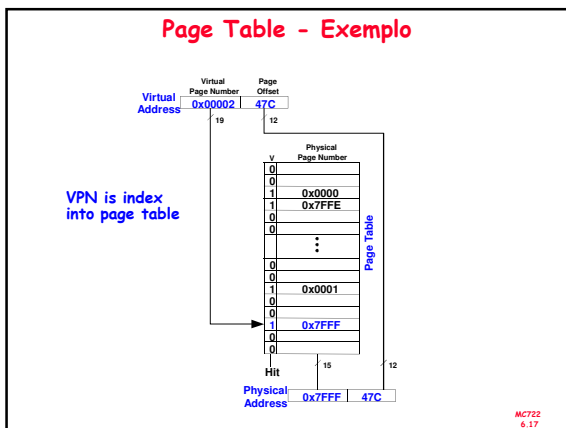
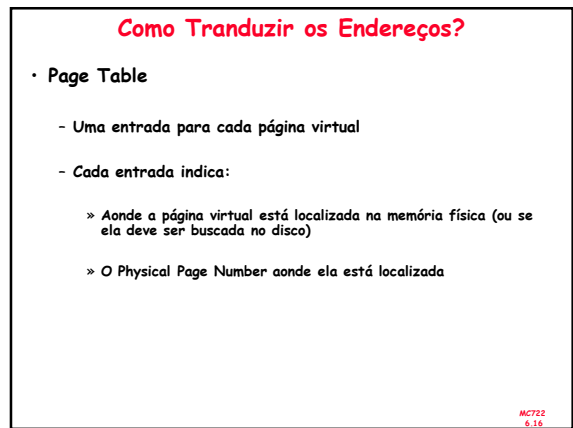
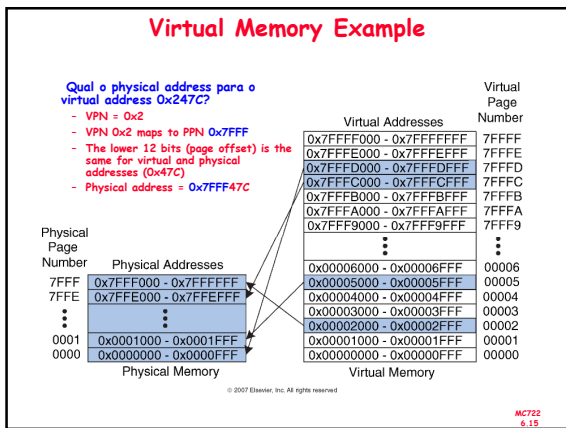
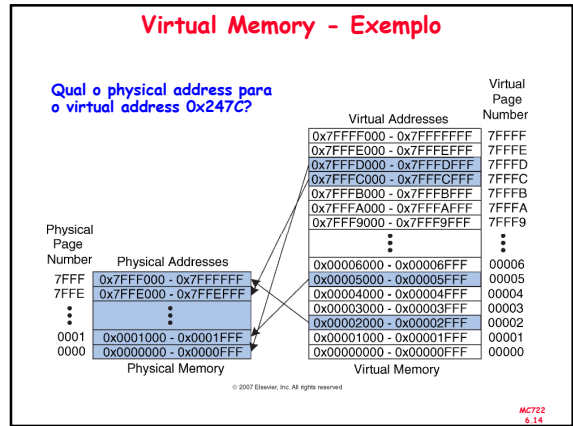
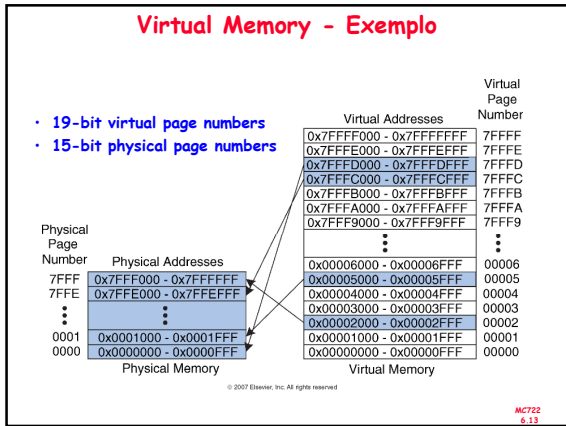


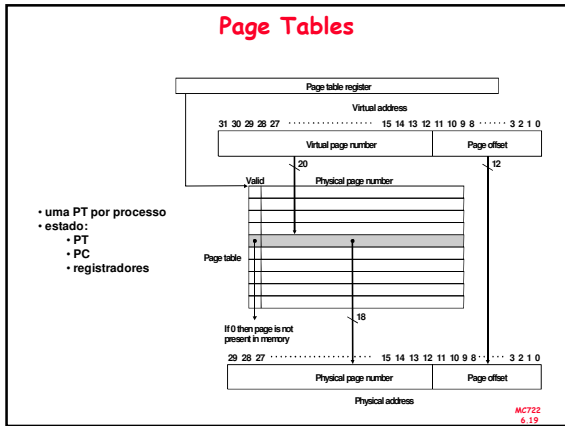
MCT22
6.11

Virtual Memory - Exemplo

- Virtual address size: 31 bits
- Physical address size: 27 bits
- Page size: 4 KB = 2^{12} bytes
- Assim,
 - Virtual memory size = 2^{31} bytes = 2 GB
 - Physical memory size = 2^{27} bytes = 128 MB
 - Page offset = 12 bits
 - $2^{31}/2^{12} = 2^{19}$ virtual pages (VPN = 19 bits)
 - $2^{27}/2^{12} = 2^{15}$ physical pages (PPN = 15 bits)

MCT22
6.12





Política de Substituição e Tamanho da PT

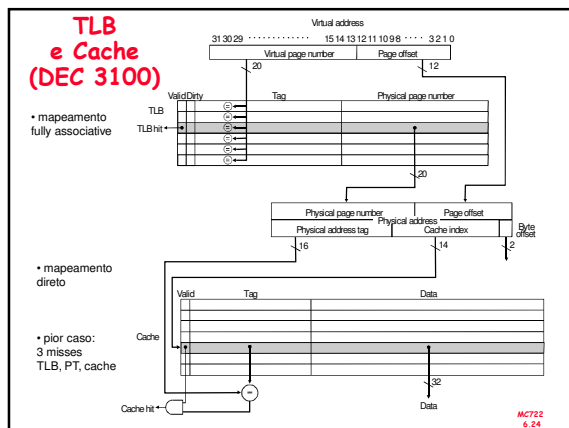
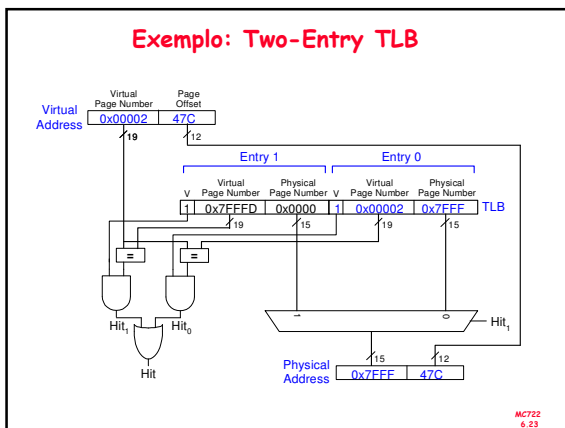
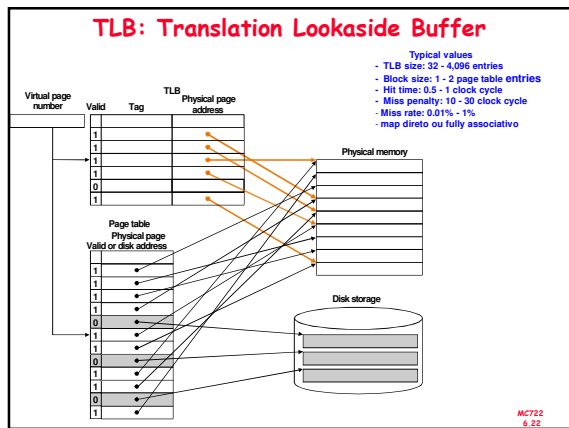
- Se **page fault** (bit válido= 0)
 - sistema operacional executa a carga da página
 - Para minimizar **page faults**, política de substituição mais usada: **LRU**
- Tamanho da PT (p/ end 32 bits, pág. de 4KB, 4B / linha da PT)
 - número de linhas: $2^{32} / 2^{12} = 2^{20}$
 - tamanho da PT = 4 MB
 - 1 PT por programa ativo !!
 - para reduzir área dedicada para PT: registradores de limite superior e inferior
- PT também são paginados

MCT22 6.20

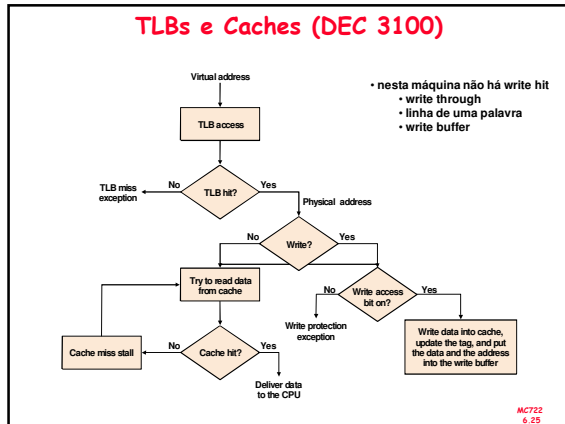
Page Table- Desafios

- A Page Table é grande, assim usualmente está localizada na memória física
- Cada processo de load/store requer dois acessos à memória:
 - 1 para a translação do endereço (read page table)
 - 1 para o acesso do dado
- **Translation Lookaside Buffer (TLB)**
 - Pequena Cache dos endereços mais recentes transladados
 - Reduz o número de acessos à memória requeridos por cada load/store de dois para um

MCT22 6.21



TLBs e Caches (DEC 3100)



TLB, Virtual memory e Cache (pag 595)

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.

MCT22 6.26

Proteção com Virtual Memory

- Suporte a pelo menos dois modos de operação
 - user process
 - operating system process (kernel, supervisor, executive)
 - CPU garante que um user process pode ler mas não escreve na **page table e na TLB**
 - Instruções especiais que somente são disponíveis no modo supervisor
 - Mecanismo que permite a CPU mudar de modo *user* para modo *supervisor*, e vice-versa
 - user para supervisor : *system call exception*
 - supervisor para user : *return from exception (RFE)*
 - OBS: page tables (no espaço de endereçamento do sistema operacional)
- MCT22 6.27

Tratamento de Page Faults e TLB misses

- TLB miss (*software ou hardware*).
 - Se a página está presente na memória, é necessário criar uma entrada na TLB.
 - Se a página não está presente na memória, é necessário transferir o controle para o SO para tratar a **page fault**.
 - Page fault (*exception mechanism*).
 - SO salva o estado do processo ativo.
 - EPC = virtual address da faulting page.
 - SO deve executar três tarefas :
 - » Consultar a **page table** usando o **virtual address** e encontrar a localização da referida página no disco.
 - » Escolher uma **physical page** para troca-la; se a página escolhida está **dirty**, ela deve ser escrita no disco antes de colocar a nova página na página física que ela ocupa.
 - » Iniciar a leitura, no disco, da referida página e coloca-la na página física escolhida.
- MCT22 6.28

Hierarquia de Memórias

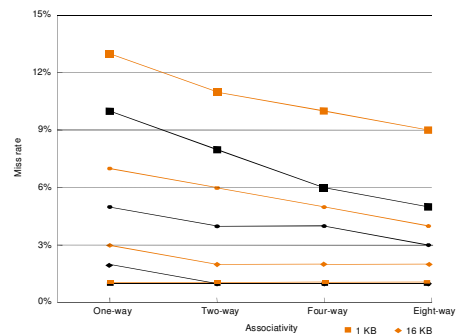
- Onde um Bloco deve ser colocado?

Scheme name	Number of sets	Block per set
Direct mapped	Number of blocks in cache	1
Set associative	Number of blocks in cache 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%

MCT22 6.29

Miss Rate vs Set Associativity



Hierarquia de Memórias

• Como Encontrar um bloco?

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 Separate lookup table	Size of the cache 0

• OBS.: Em virtual memory systems

- Full associativity é bom, já que um miss tem alto custo
- Full associativity permite que, por software, se use esquemas de reposição sofisticados projetados para reduzirem o miss rate.

MCT22
6.31

Hierarquia de Memórias

• Qual Bloco Deve Ser Trocado em um Cache Miss?

- Random : o bloco candidato é selecionado randomicamente, possivelmente usando algum hardware.
- Least Recently Used (LRU): O bloco trocado é aquele que foi não foi usado a mais tempo

MCT22
6.32

Hierarquia de Memórias

• O que Ocorre em um Write?

- Write-through

- » Misses são simples e baratos já que o bloco não precisa ser escrito no lower level.
- » Mais fácil de implementar do que write-back, porém em sistemas com alta velocidade, para ser usável, uma write-through cache necessita usar um write buffer

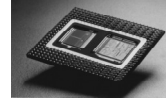
- Write-back (copy-back)

- » Words individuais podem ser escritas pelo processador na velocidade da cache e não da memória.
- » Múltiplos writes em um mesmo bloco requer somente uma escrita no lower level na hierarquia.
- » Quando um bloco é escrito na memória, o sistema pode fazer uso efetivo de high bandwidth transfer.

MCT22
6.33

Sistemas Modernos

Characteristic	Intel Pentium Pro	PowerPC 604
Virtual address	32 bits	32 bits
Physical address	32 bits	32 bits
Page size	4 KB, 4 MB	4 KB, selectable, and 256 MB
TLB organization	A TLB for instructions and a TLB for data Both four-way set associative Pseudo-LRU replacement Instruction TLB: 32 entries Data TLB: 64 entries TLB misses handled in hardware	A TLB for instructions and a TLB for data Both two-way set associative LRU replacement Instruction TLB: 128 entries Data TLB: 128 entries TLB misses handled in hardware



Characteristic	Intel Pentium Pro	PowerPC 604
Cache organization	Split instruction and data caches	Split instruction and data caches
Cache size	8 KB each for instructions/data	16 KB each for instructions/data
Cache associativity	Four-way set associative	Four-way set associative
Replacement	Approximated LRU replacement	LRU replacement
Block size	32 bytes	32 bytes
Write policy	Write-back	Write-back or write-through

MCT22
6.34

MCT22
6.35

ERROR: undefined
OFFENDING COMMAND: pp_by2

STACK:

-180
1248
/pp_sy
331
/pp_by2