

Organisasi Sistem Komputer

Bagian 11

Virtual Memory

Sekolah Teknik Elektro dan Informatika – ITB
2009



Pembahasan

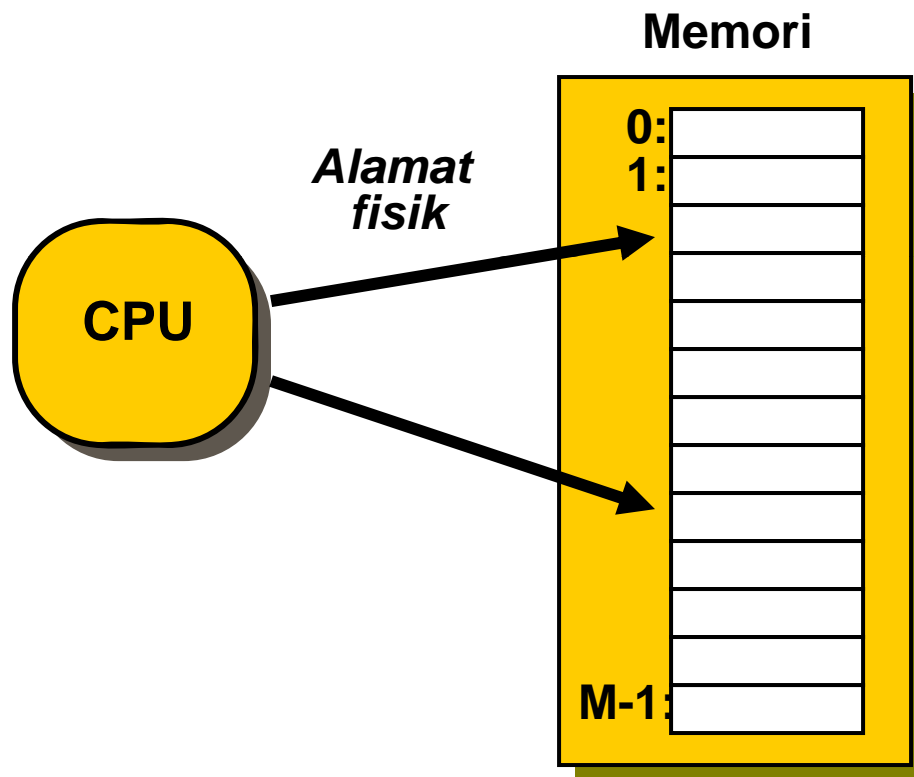
- + Fungsi virtual memory
- + Penerjemah (translasi) alamat pada virtual memory
- + Mempercepat translasi dengan TLB
- + Sistem memori sederhana

Fungsi Virtual Memory (VM)

Virtual Memory (VM) memiliki tiga fungsi penting :

1. Sebagai perangkat **cache**
 - VM memanfaatkan memori utama (DRAM) lebih efisien dengan memperlakukannya sebagai cache alamat yang tersimpan dalam disk
 - Memori utama hanya menyimpan data yang aktif
 - VM melakukan transfer data antara disk dan memori seperlunya saja
2. Sebagai perangkat **manajemen memori**
 - VM menyederhanakan manajemen memori dengan melakukan teknik pengalamatan yang seragam untuk setiap proses
 - Pengalamatan disk, memori, cache dilakukan secara uniform
3. Sebagai perangkat untuk **memproteksi memori**
 - VM melindungi alokasi alamat setiap proses terhadap gangguan
 - Suatu proses tidak dapat diganggu oleh proses lain.
 - Proses tidak dapat mengakses informasi yang bukan hak-nya
 - Setiap alokasi alamat memiliki hak yang berbeda-beda

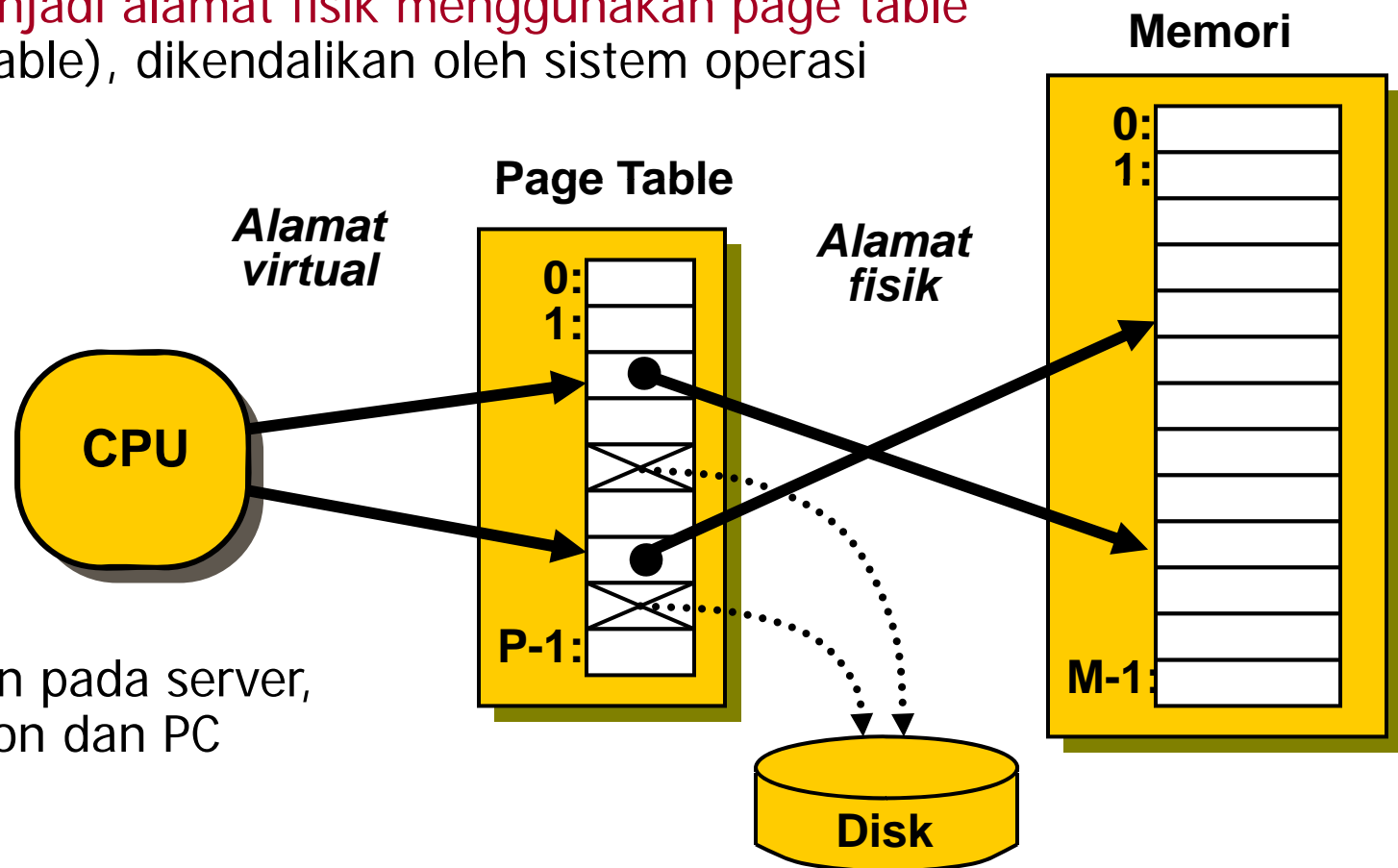
Pengalamatan secara Fisik



- + Disebut '**physical addressing**'
- + Alamat yang dihasilkan CPU menunjukkan langsung lokasi byte dalam memori secara fisik
- + Digunakan pada :
 - mesin Cray
 - PC generasi pertama
 - sistem embedded pada umumnya

Sistem dengan Virtual Memory

- ✚ Penerjemahan/translasi alamat (address translation) : berbentuk perangkat keras yang **mengkonversi alamat virtual menjadi alamat fisik menggunakan page table** (look-up table), dikendalikan oleh sistem operasi



- ✚ Digunakan pada server, workstation dan PC modern

Address Space

- ✚ Address space adalah kumpulan alamat-alamat yang letaknya berurutan

$\{0, 1, 2, \dots\}$

- ✚ Pada sistem dengan virtual memory, CPU menghasilkan alamat virtual dari address space $N = 2^n$ alamat, disebut virtual address space

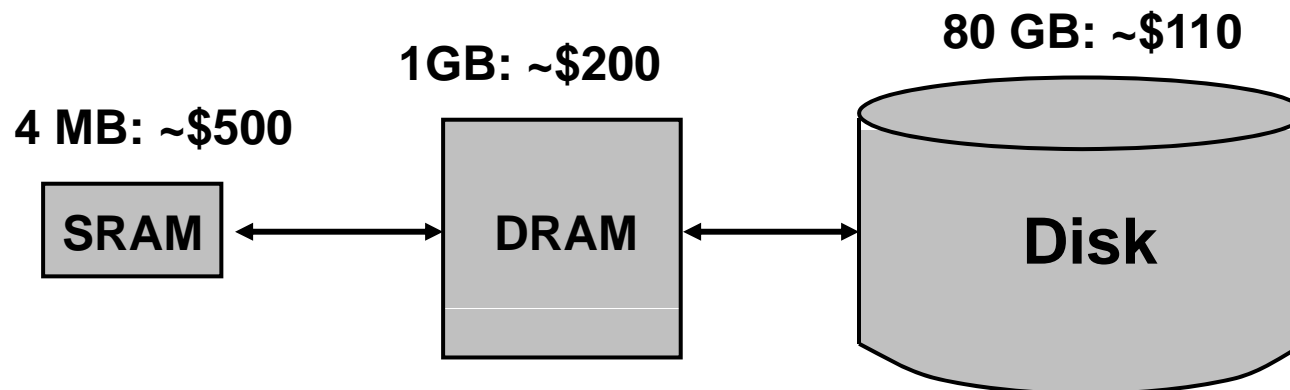
$\{0, 1, 2, \dots, N-1\}$

- ✚ Sistem juga memiliki physical address space yang merupakan memori fisik sebesar M byte

$\{0, 1, 2, \dots, M-1\}$

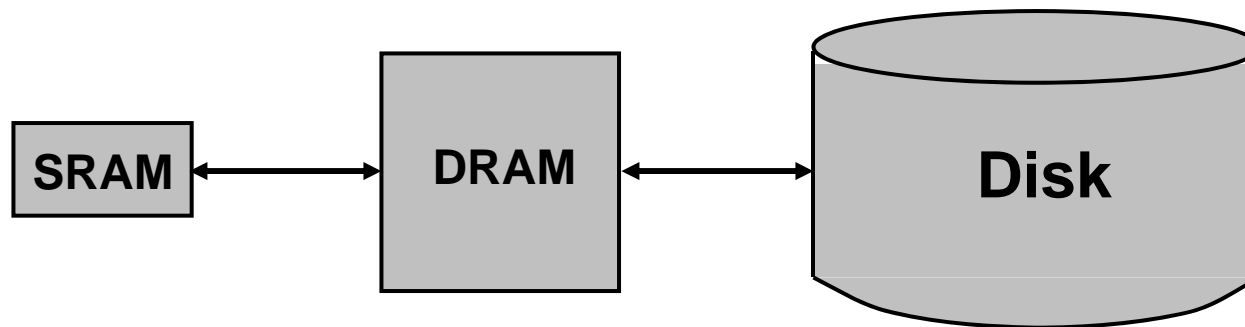
1. VM : Perangkat Caching

- ✚ Ukuran seluruh address space sangat besar :
 - Bus alamat 32-bit : dapat mengalami ~4,000,000,000 (4 milyar) byte
 - Bus alamat 64-bit : ~16,000,000,000,000,000,000 (16 quintillion) byte
- ✚ Harga Harddisk kira-kira 300 kali lebih murah dari DRAM
 - 80 GB DRAM: ~ \$33,000
 - 80 GB harddisk: ~ \$110
- ✚ Agar data berukuran besar dapat diakses secara murah, maka data tersebut harus disimpan dalam disk, sementara data yang sedang aktif saja yang disimpan dalam DRAM/SRAM



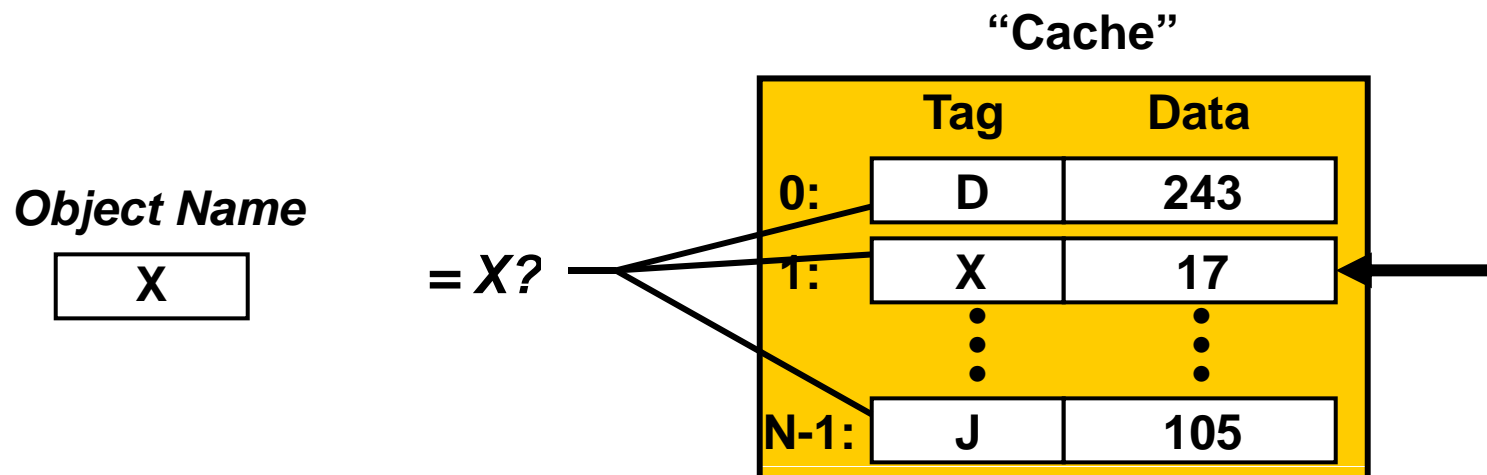
DRAM vs SRAM sebagai Cache

- ✚ DRAM vs. disk lebih ekstrim dari SRAM vs. DRAM
 - Waktu akses :
 - DRAM ~10X lebih lambat SRAM
 - Disk ~**100,000X** lebih lambat DRAM
 - Pentingnya spatial locality:
 - Byte pertama ~**100,000X** lebih lambat dari byte yang berurutan pada disk
 - Pertimbangan desain cache menggunakan DRAM harus memperhatikan 'cost' yang timbul akibat miss



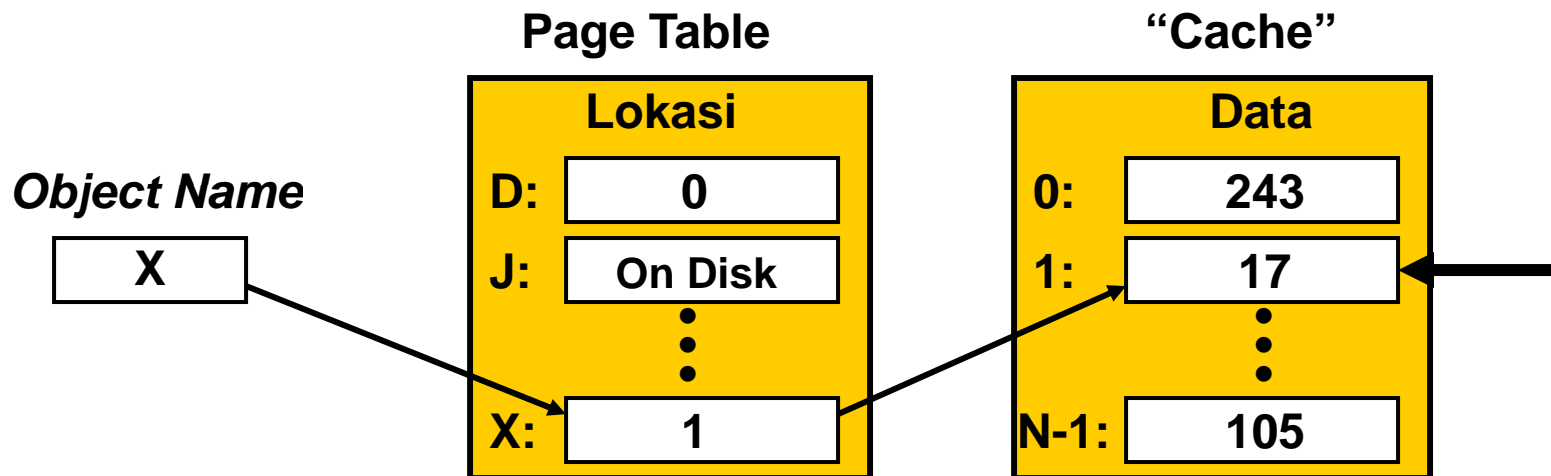
SRAM Cache

- Tag disimpan bersama baris cache
- Memetakan dari blok cache ke blok memori
 - Dari bentuk cached ke uncached
- Tag tidak ada bila blok tidak ada dalam cache



DRAM Cache

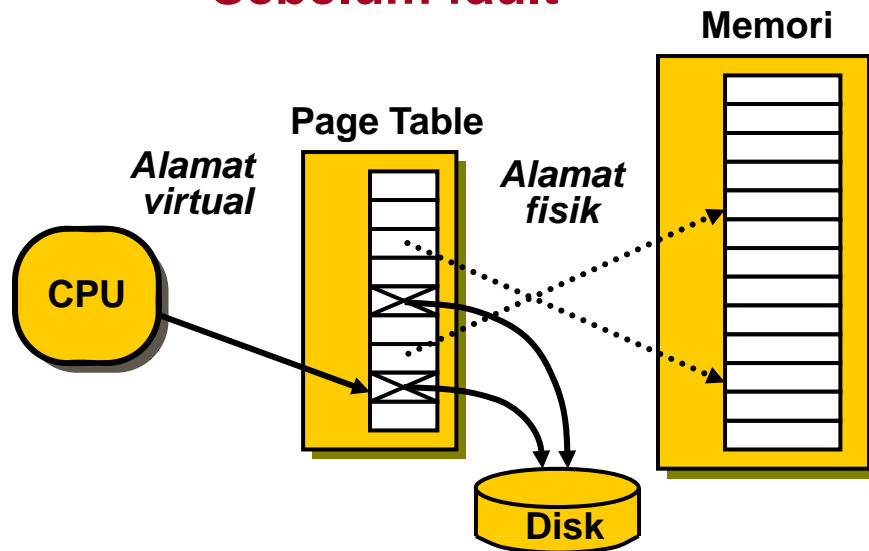
- Setiap page yang dialokasikan oleh virtual memory memiliki entry pada page table
- Memetakan virtual page ke physical page
 - Dari bentuk uncached menjadi bentuk cached
- Page table entry tetap ada walaupun page belum ada di memori
 - Menunjukkan alamat pada disk



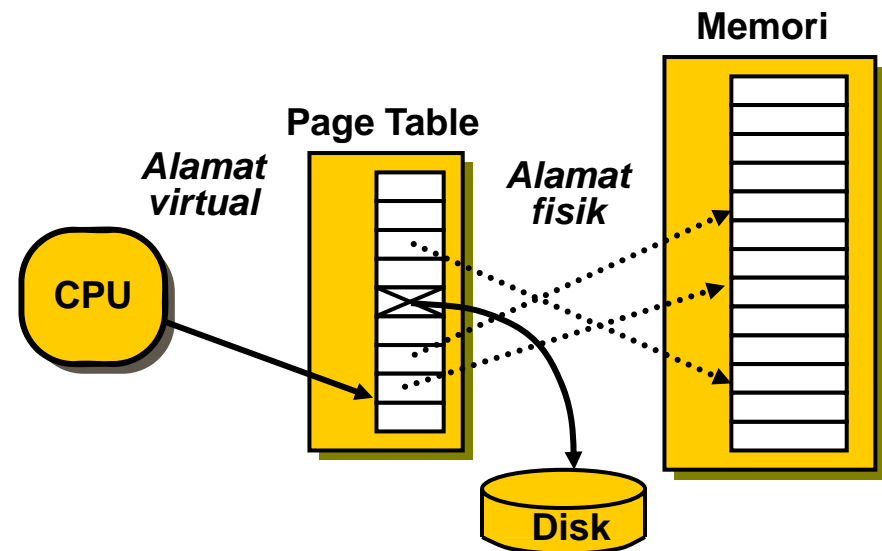
DRAM Cache

- ✚ Jika suatu obyek berada dalam disk, bukan di memori, maka :
 - Page table entry memperlihatkan alamat virtual tidak ada di memori
 - OS exception handler melakukan pemindahan data dari disk ke memori

Sebelum fault

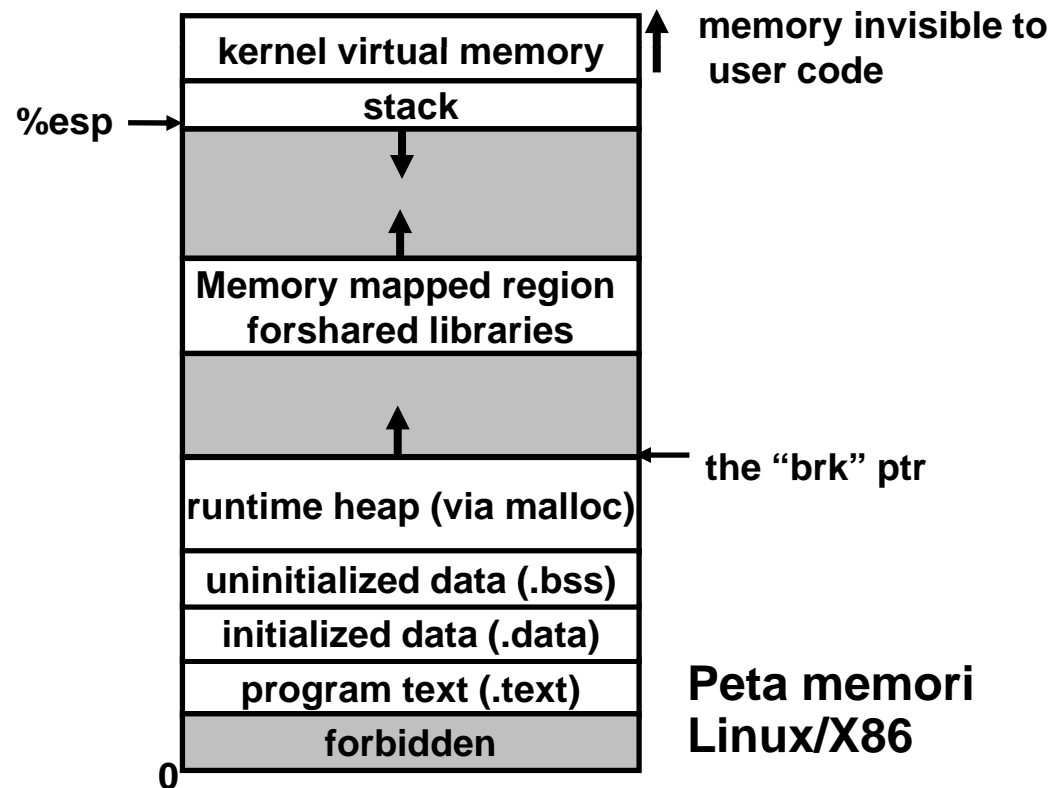


Setelah fault



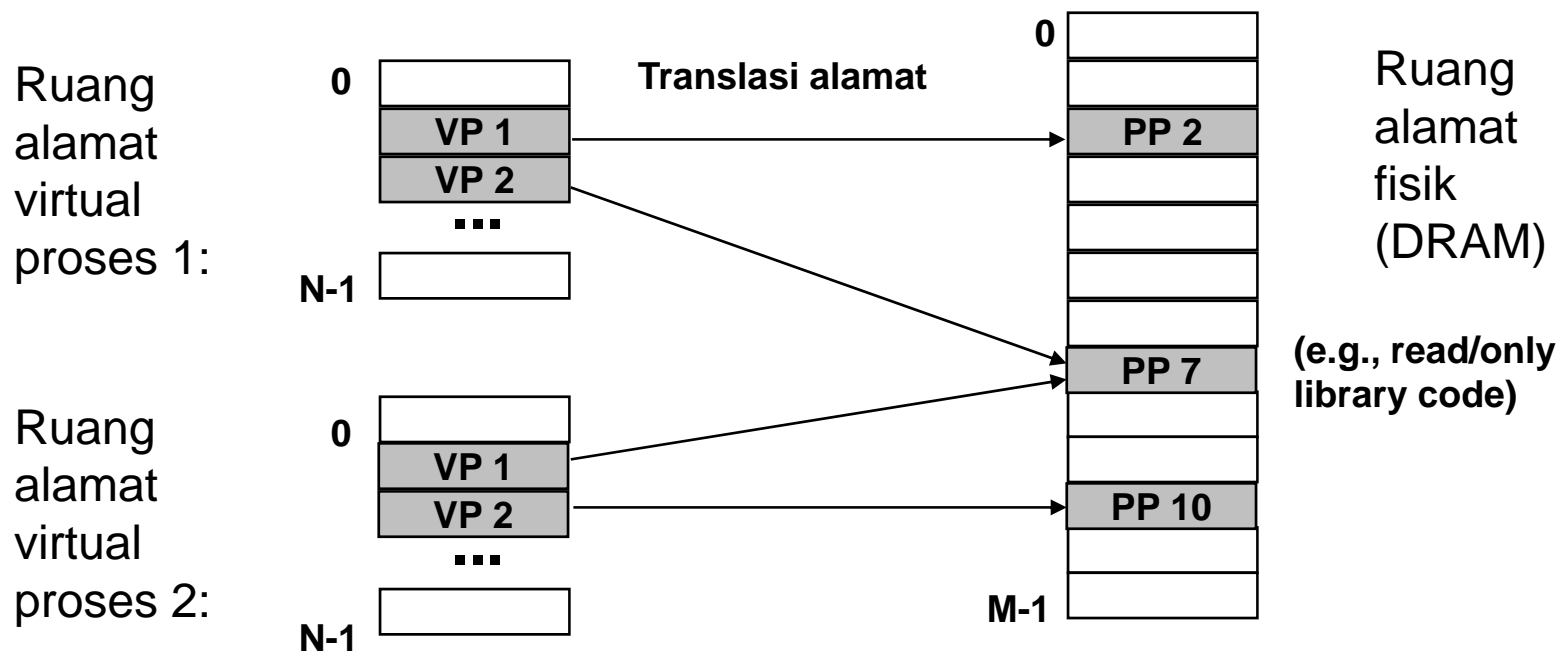
2. VM : Manajemen Memori

- ✚ Beberapa proses dapat tersimpan secara bersamaan dalam memori
- ✚ Masalah akan timbul bila dua proses mengakses alamat yang sama
 - Konflik alamat tersebut harus diatasi



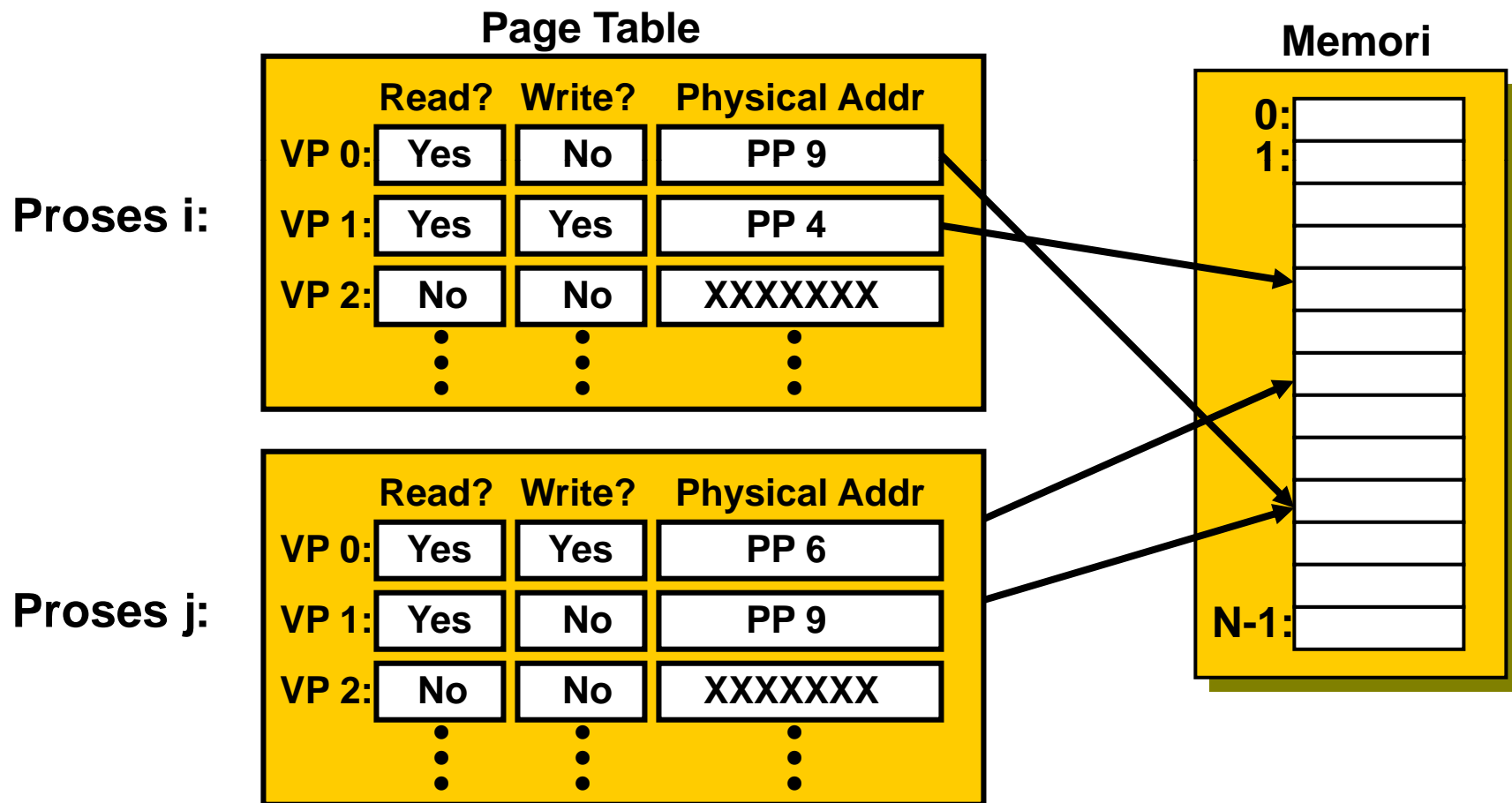
Solusi : Pisahkan Alamat Virtual

- Ruang alamat virtual dan fisik dibagi menjadi beberapa blok berukuran sama
 - Block disebut “pages” (virtual dan fisik)
- Setiap proses memiliki ruang alamat virtual sendiri
 - Sistem operasi mengendalikan konversi virtual pages ke memori fisik



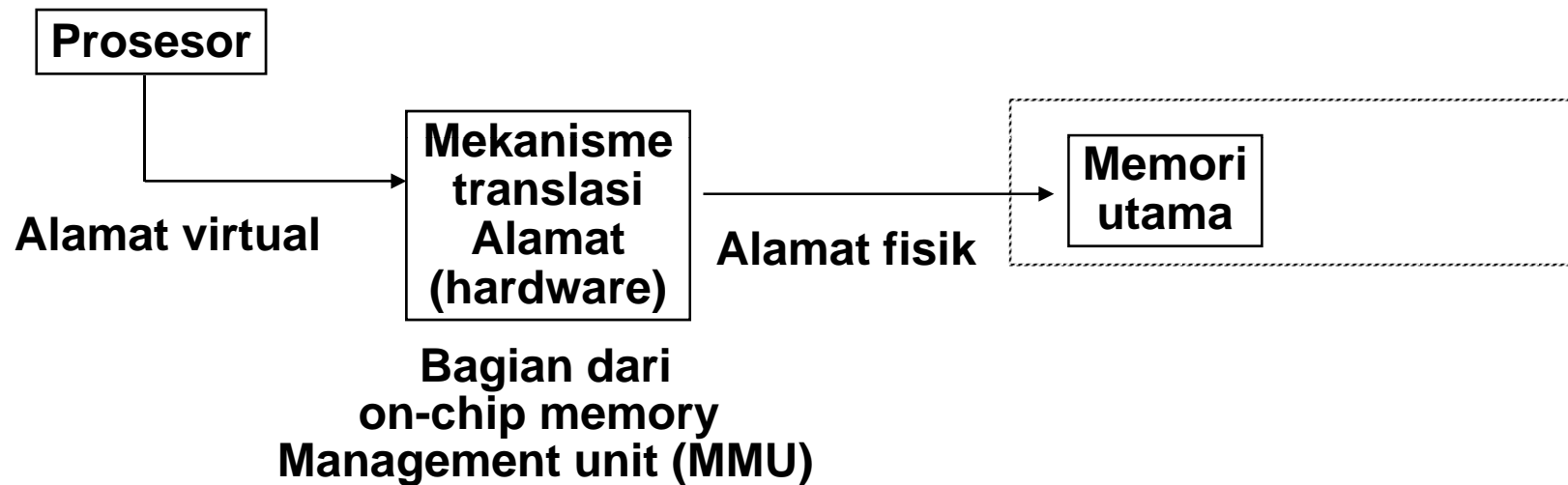
3. VM : Proteksi Memori

- ✚ Page table entry berisi informasi tentang hak akses
 - Proteksi dilakukan secara hardware

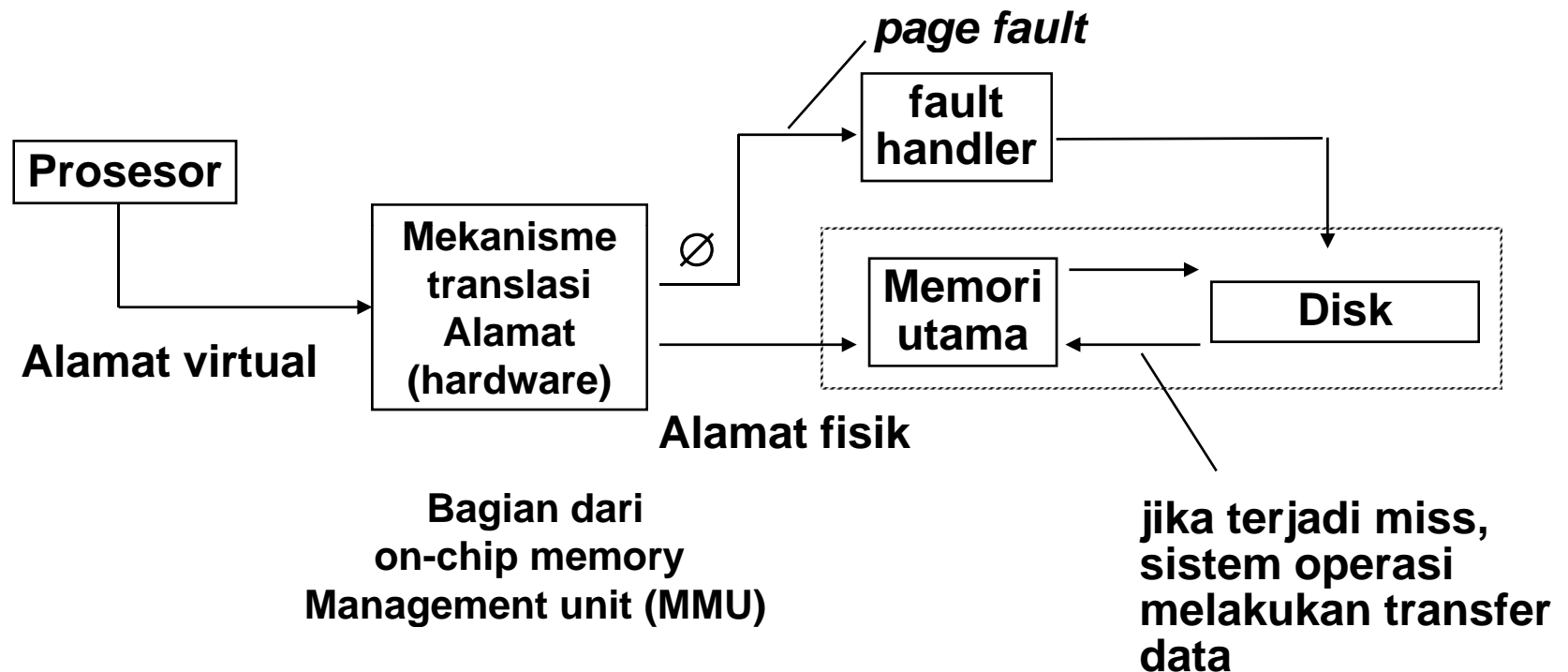


Translasi Alamat (Address Translation) Virtual Memory

Translasi Alamat : HIT



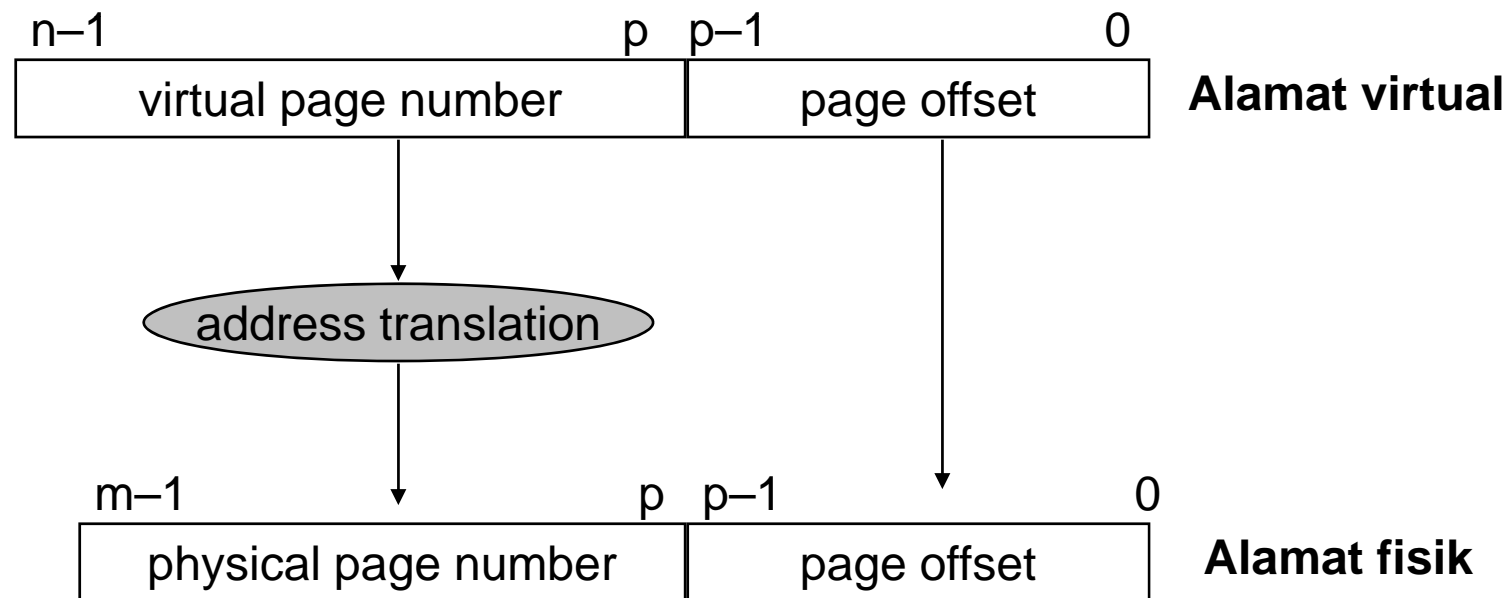
Translasi Alamat : MISS



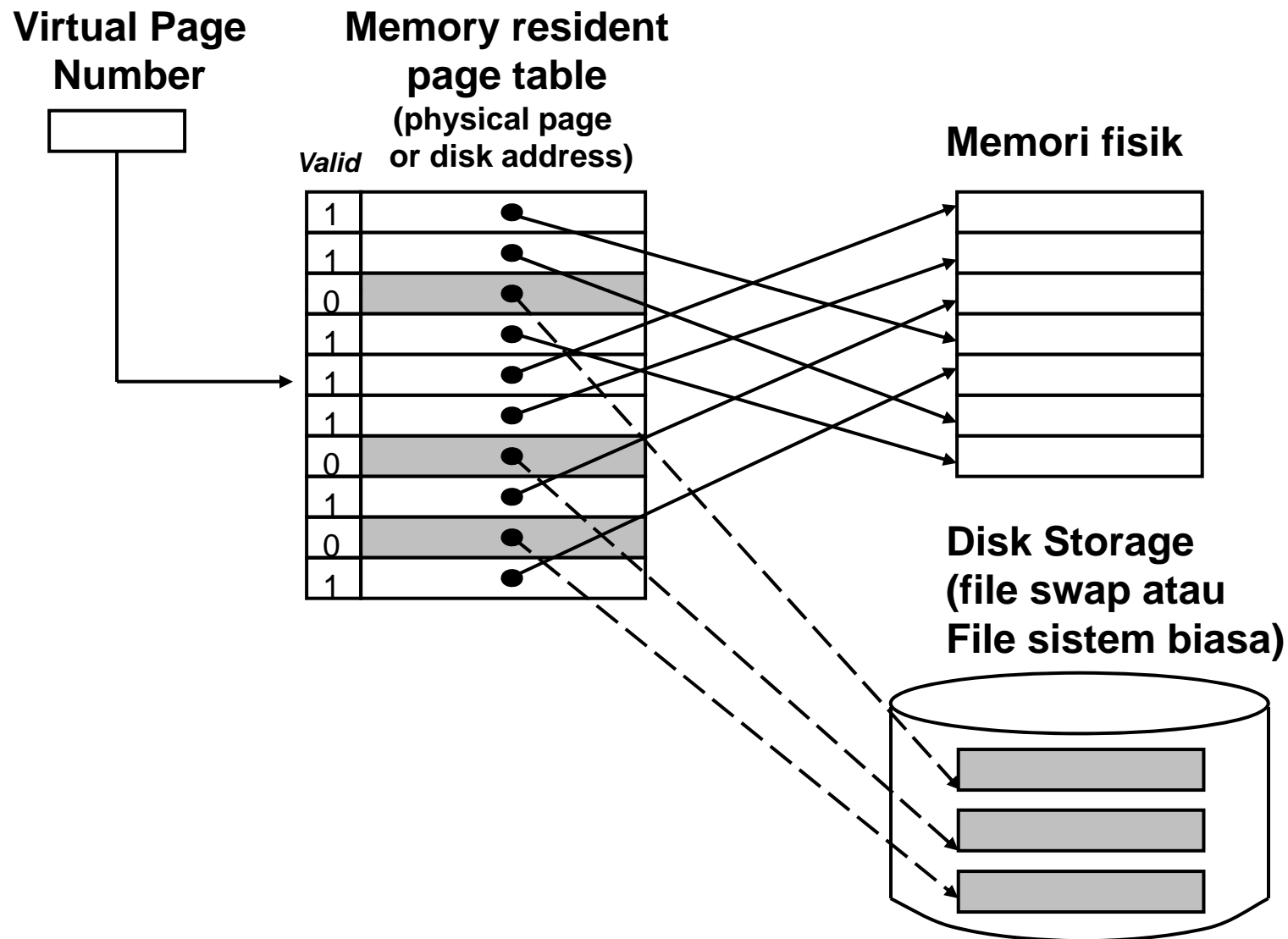
Translasi Alamat VM

Parameters

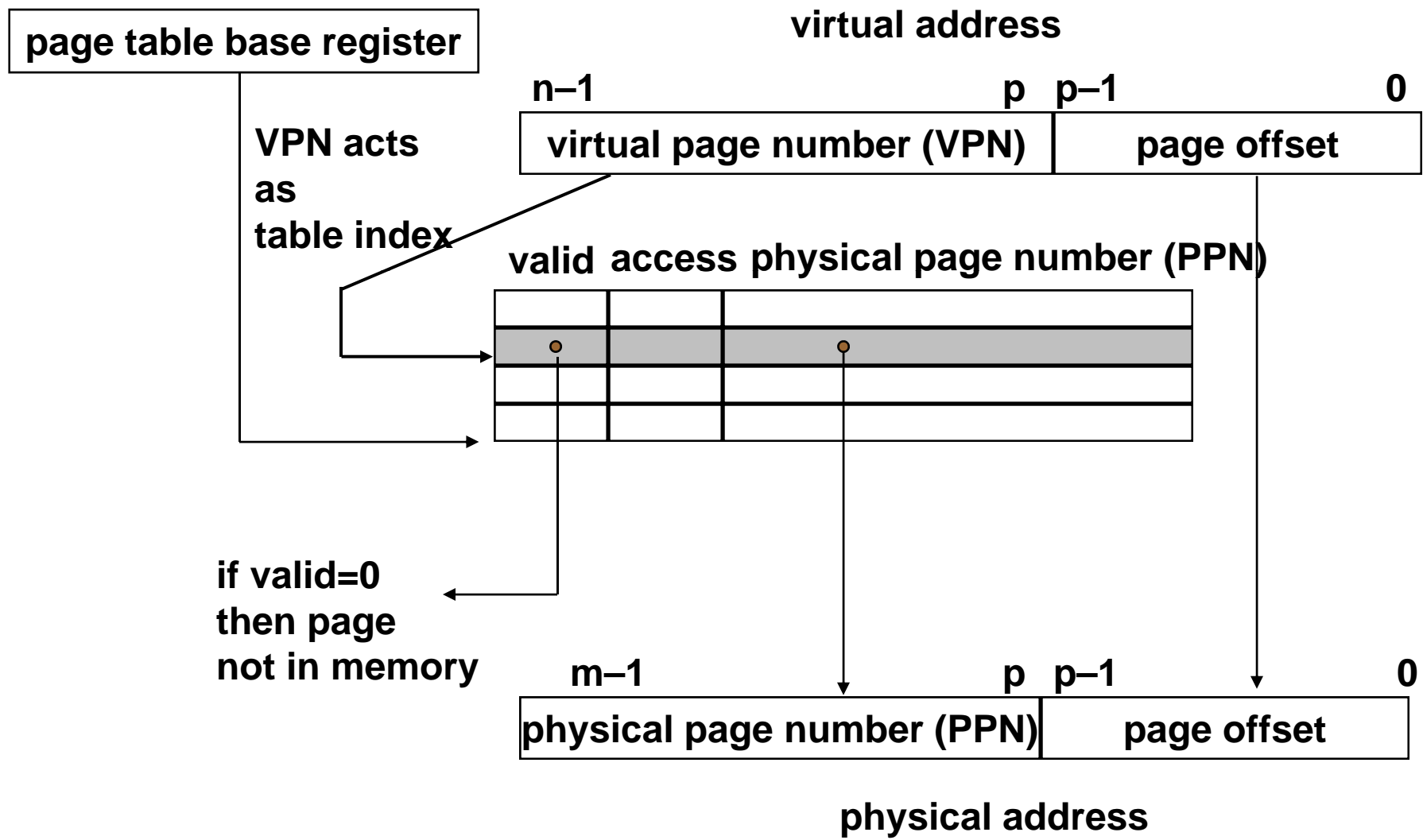
- $P = 2^p =$ page size (byte)
- $N = 2^n =$ batas alamat virtual
- $M = 2^m =$ batas alamat fisik



Page Table



Operasi Page Table



Operasi Page Table

Translasi

- Pisahkan page tabel per proses
- VPN membentuk indeks pada page table (menunjukkan page table entry)

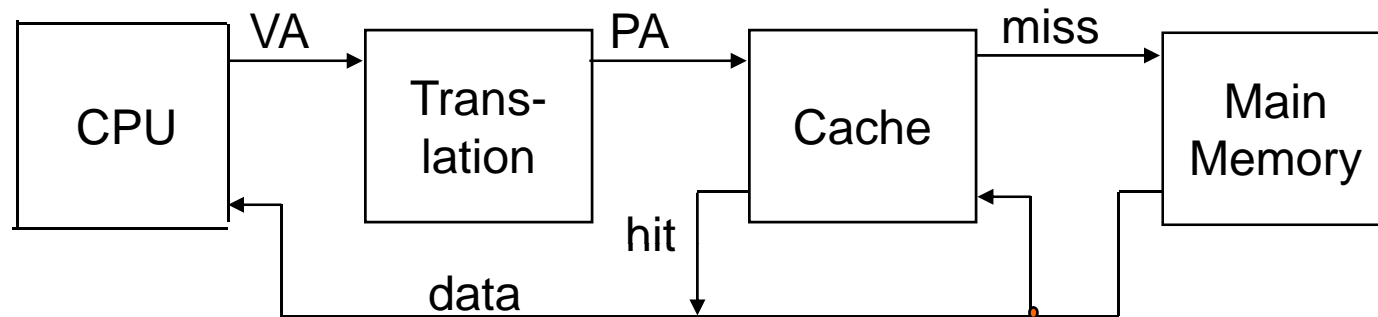
Menghitung alamat fisik

- Page Table Entry (PTE) memberikan informasi tentang page
 - Jika bit valid = 1, maka page berada dalam memori, gunakan physical page number (PPN) untuk mengkonstruksi alamat
 - Jika bit valid = 0, maka page berada dalam disk (page fault)

Cek proteksi

- Cek 'right access' untuk menentukan akses yang diijinkan
 - Read-only, read-write, executy only

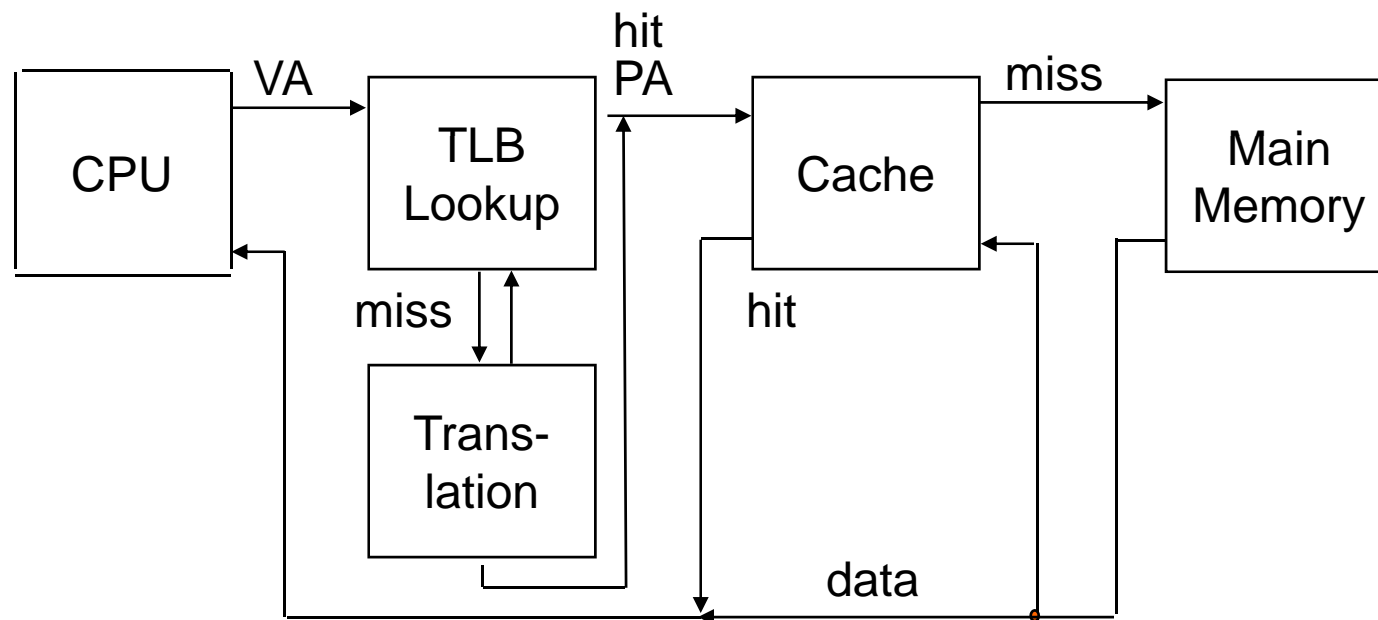
Integrasi VM dan Cache



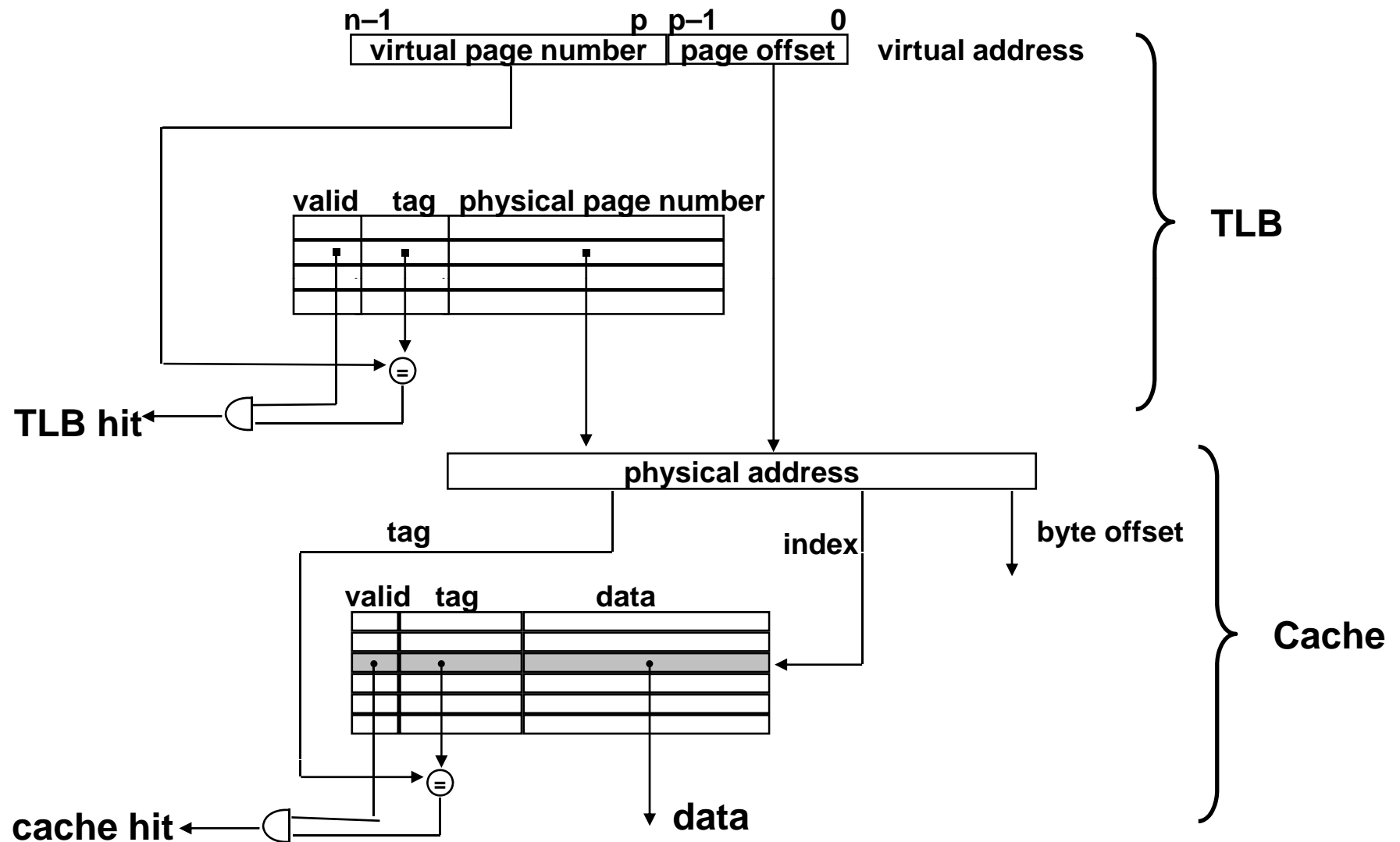
- + Cache diakses menggunakan alamat fisik
- + Translasi alamat dilakukan sebelum cache lookup

Percepat Translasi dengan TLB

- ✚ TLB = Translation Lookaside Buffer
 - Berbentuk perangkat keras cache yang terletak dalam MMU
 - Memetakan virtual page number (VPN) menjadi physical page number (PPN)
 - Dapat menggantikan page table dalam ukuran yang lebih kecil



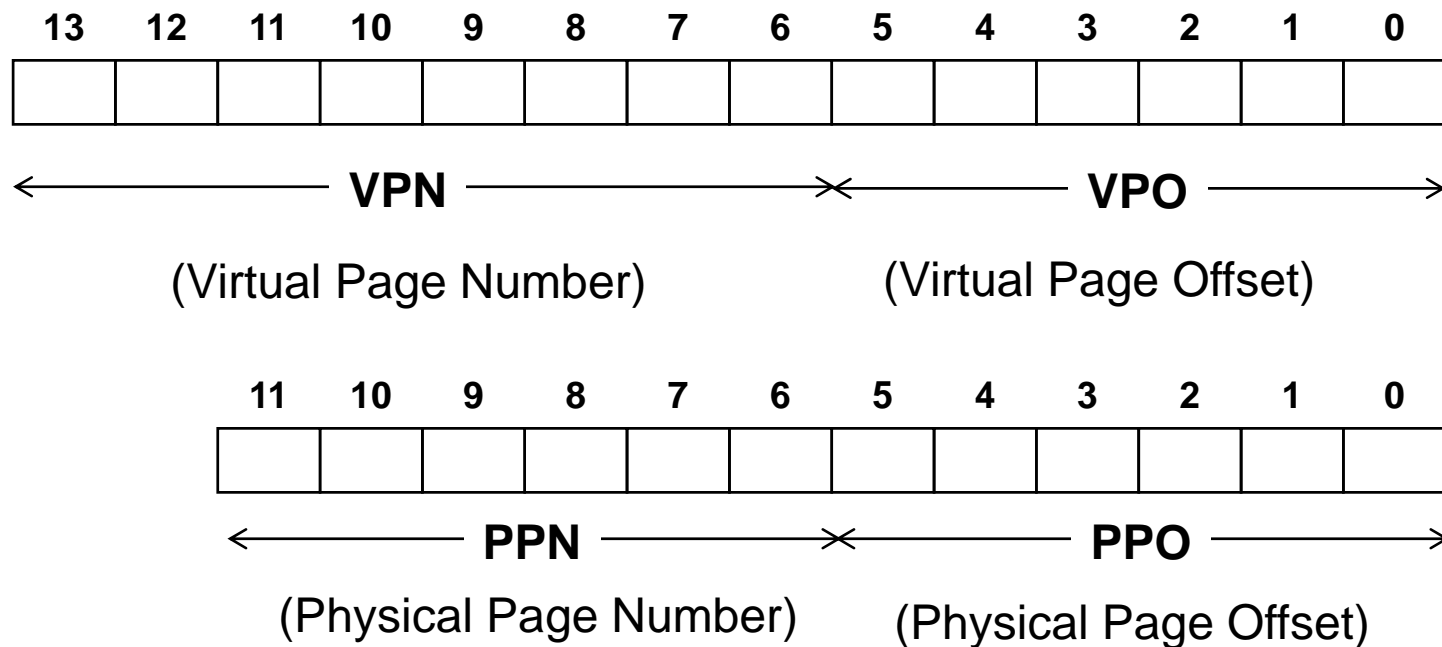
Translasi Alamat dengan TLB



Contoh Sistem Memori Sederhana

Pengalamatan

- 14-bit alamat virtual (virtual address)
- 12-bit alamat fisik (physical address)
- Page size = 64 byte = 2^6



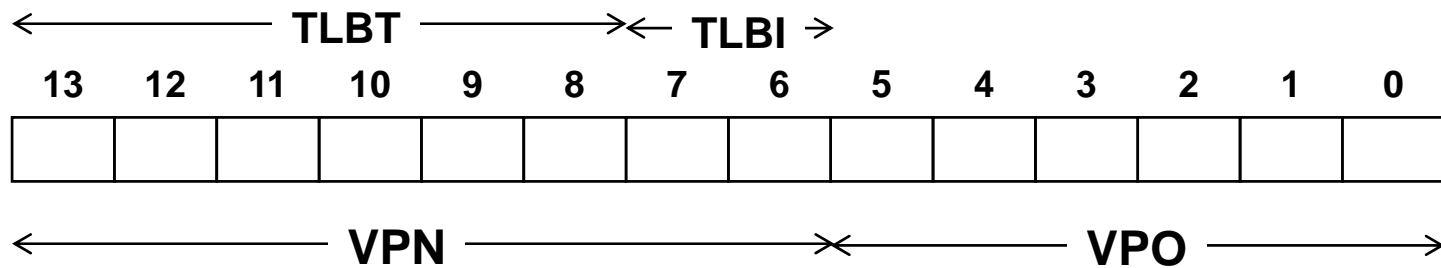
Page Table

VPN	PPN	Valid	VPN	PPN	Valid
00	28	1	08	13	1
01	–	0	09	17	1
02	33	1	0A	09	1
03	02	1	0B	–	0
04	–	0	0C	–	0
05	16	1	0D	2D	1
06	–	0	0E	11	1
07	–	0	0F	0D	1

- ✚ Tabel di atas hanya menampilkan 16 entry pertama (total 256 entry)

TLB

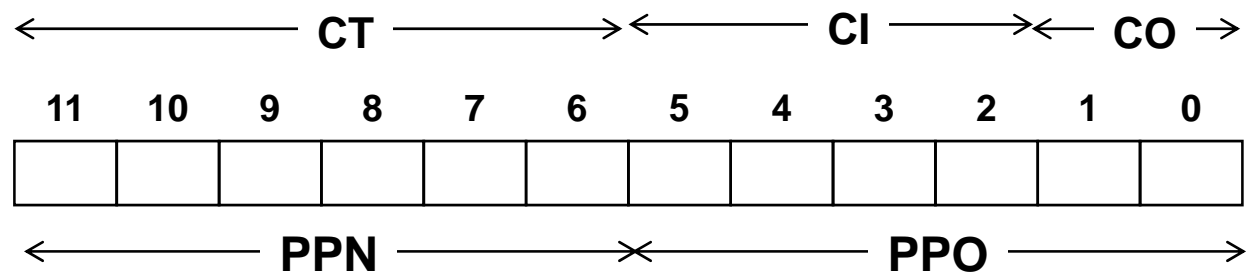
- 16 entries
- 4-way associative



Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	03	–	0	09	0D	1	00	–	0	07	02	1
1	03	2D	1	02	–	0	04	–	0	0A	–	0
2	02	–	0	08	–	0	06	–	0	03	–	0
3	07	–	0	03	0D	1	0A	34	1	02	–	0

Cache

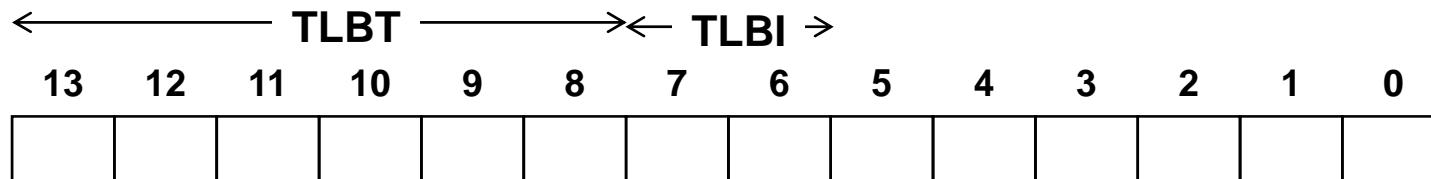
- 16 lines
- 4-byte line size
- Direct mapped



Idx	Tag	Valid	B0	B1	B2	B3	Idx	Tag	Valid	B0	B1	B2	B3
0	19	1	99	11	23	11	8	24	1	3A	00	51	89
1	15	0	–	–	–	–	9	2D	0	–	–	–	–
2	1B	1	00	02	04	08	A	2D	1	93	15	DA	3B
3	36	0	–	–	–	–	B	0B	0	–	–	–	–
4	32	1	43	6D	8F	09	C	12	0	–	–	–	–
5	0D	1	36	72	F0	1D	D	16	1	04	96	34	15
6	31	0	–	–	–	–	E	13	1	83	77	1B	D3
7	16	1	11	C2	DF	03	F	14	0	–	–	–	–

Translasi Alamat (1)

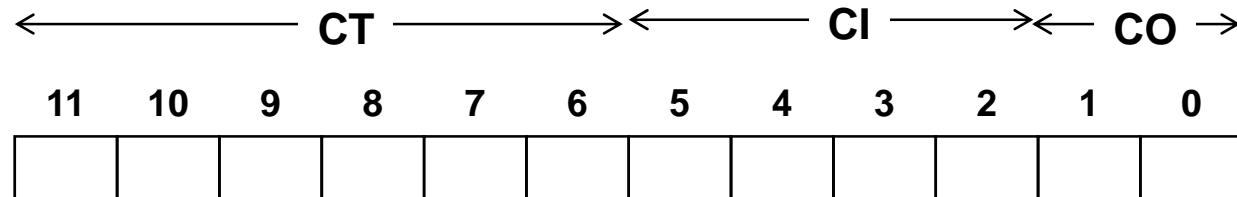
Alamat virtual 0x03D4



← **VPN** × **VPO** →

VPN ____ TLBI ____ TLBT ____ TLB Hit? __ Page Fault? __ PPN: ____

Alamat fisik

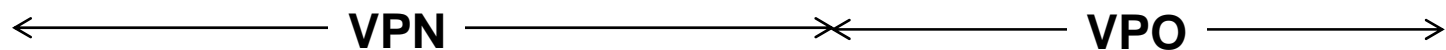
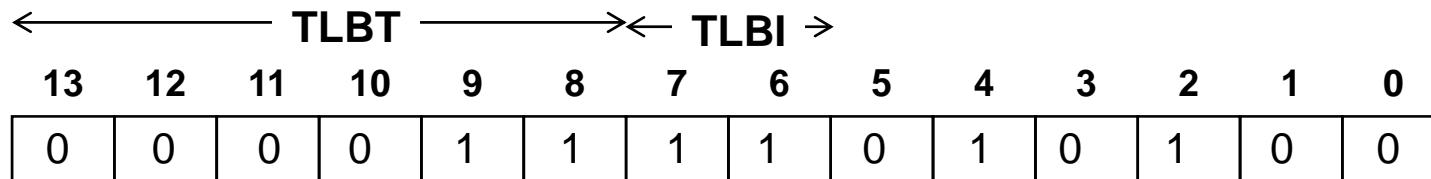


← **PPN** × **PPO** →

Offset ____ CI ____ CT ____ Hit? __ Byte: ____

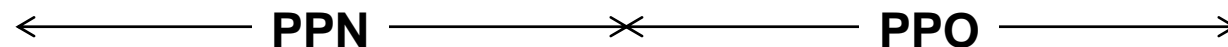
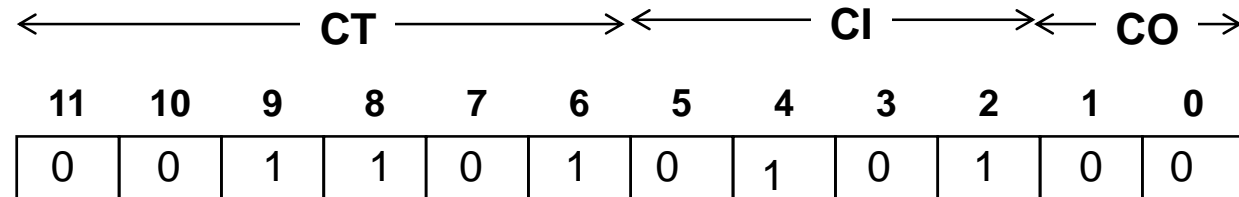
Translasi Alamat (1)

✚ Alamat virtual 0x03D4



VPN:0x0F TLBI:0x03 TLBT:0x03 TLB Hit?:Y Page Fault? :N PPN:0x0D

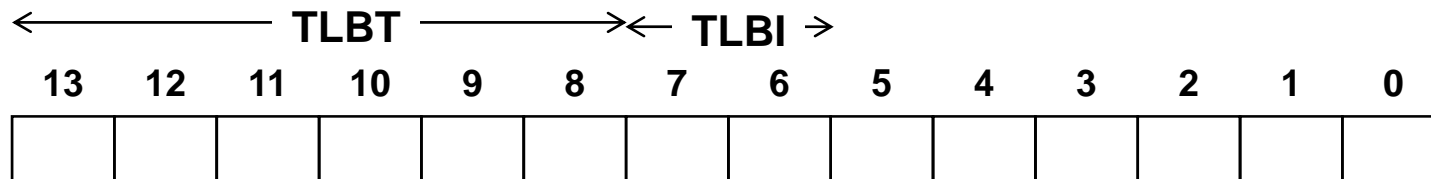
✚ Alamat fisik



Offset:0x00 CI:0x05 CT:0x0D Hit?:Y Byte:0x36

Translasi Alamat (2)

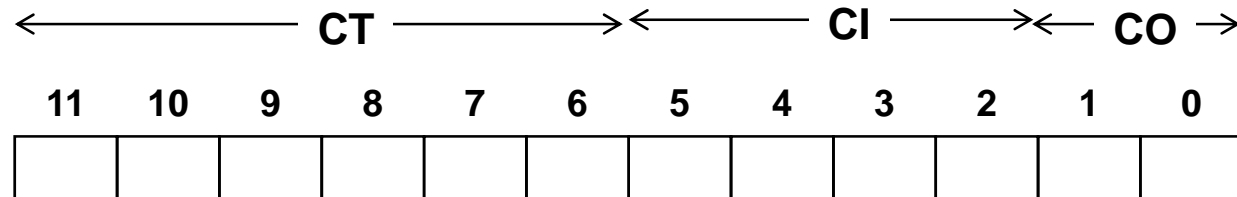
Alamat virtual 0x0B8F



← VPN → × ← VPO →

VPN ____ TLBI ____ TLBT ____ TLB Hit? __ Page Fault? __ PPN: ____

Alamat fisik



← PPN → × ← PPO →

Offset ____ CI ____ CT ____ Hit? __ Byte: ____