Organisasi Sistem Komputer

Bagian 9
Hirarki Memori

Sekolah Teknik Elektro dan Informatika ITB
2009

Pembahasan

- Trend teknologi penyimpanan
- Referensi locality
- Cache pada hirarki memori

Random Access Memory (RAM)

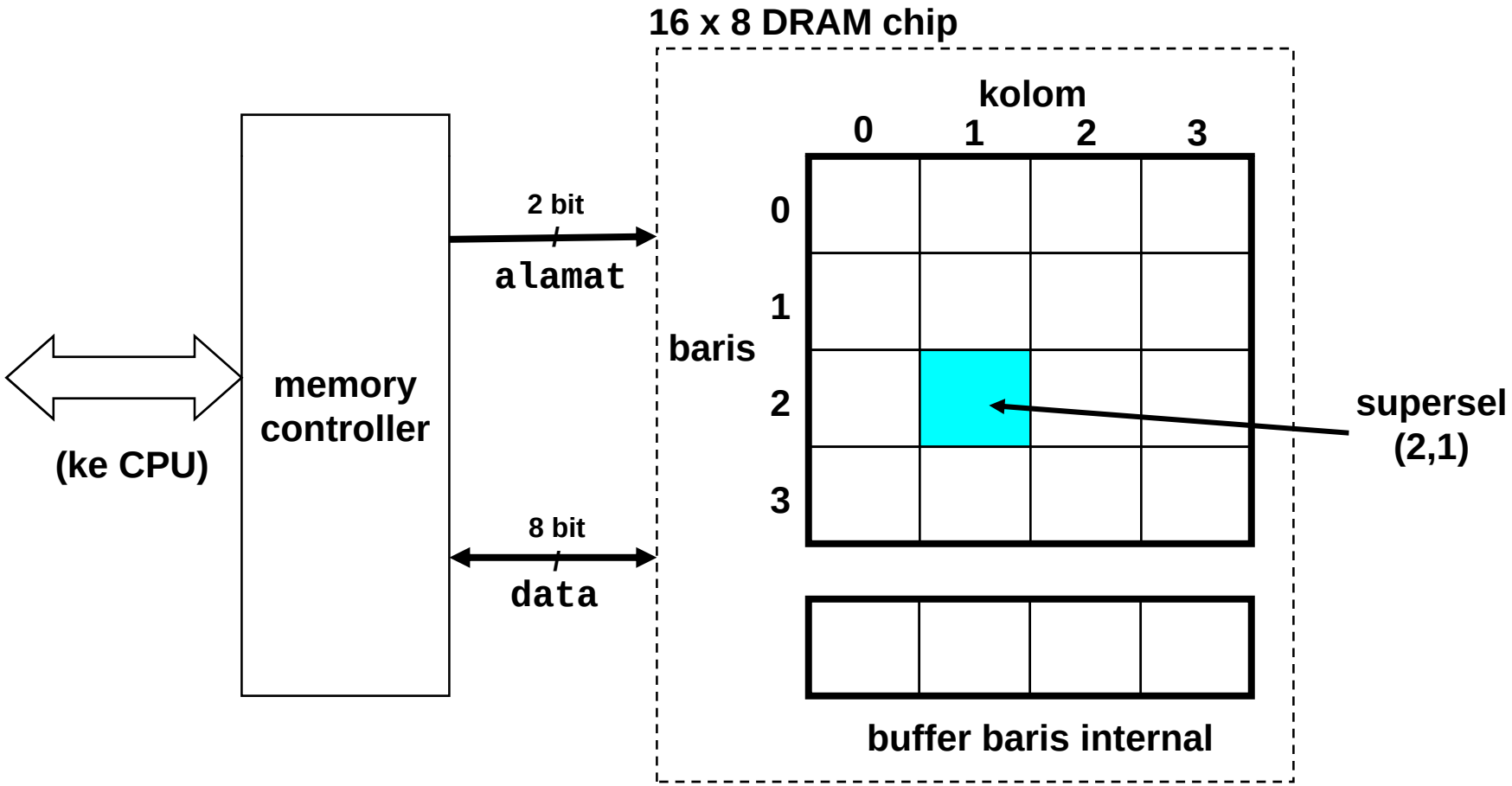
- Karakteristik
  - RAM dibungkus dalam paket berbentuk chip
  - Satuan penyimpanan dasar adalah sel (1 bit per sel)
  - Gabungan beberapa chip RAM membentuk memori
- Static RAM (SRAM)
  - Setiap sel menyimpan bit dalam rangkaian dgn enam transistor
  - Datanya akan bertahan terus, selama diberi daya
  - Relatif tahan terhadap gangguan, seperti noise
  - Lebih cepat dan mahal dari DRAM
- Dynamic RAM (DRAM)
  - Setiap sel menyimpan bit dalam kapasitor dan transistor
  - Datanya harus di-refresh setiap 10-100 ms
  - Sensitif terhadap gangguan
  - Lebih lambat dan murah dibandingkan dengan SRAM

Perbandingan SRAM vs DRAM

	Transistor per bit	Waktu akses	Data bertahan	Sensitif	Harga	Aplikasi
SRAM	6	x 1	ya	tidak	x 100	Cache memory
DRAM	1	x 10	tidak	ya	x 1	Main memory

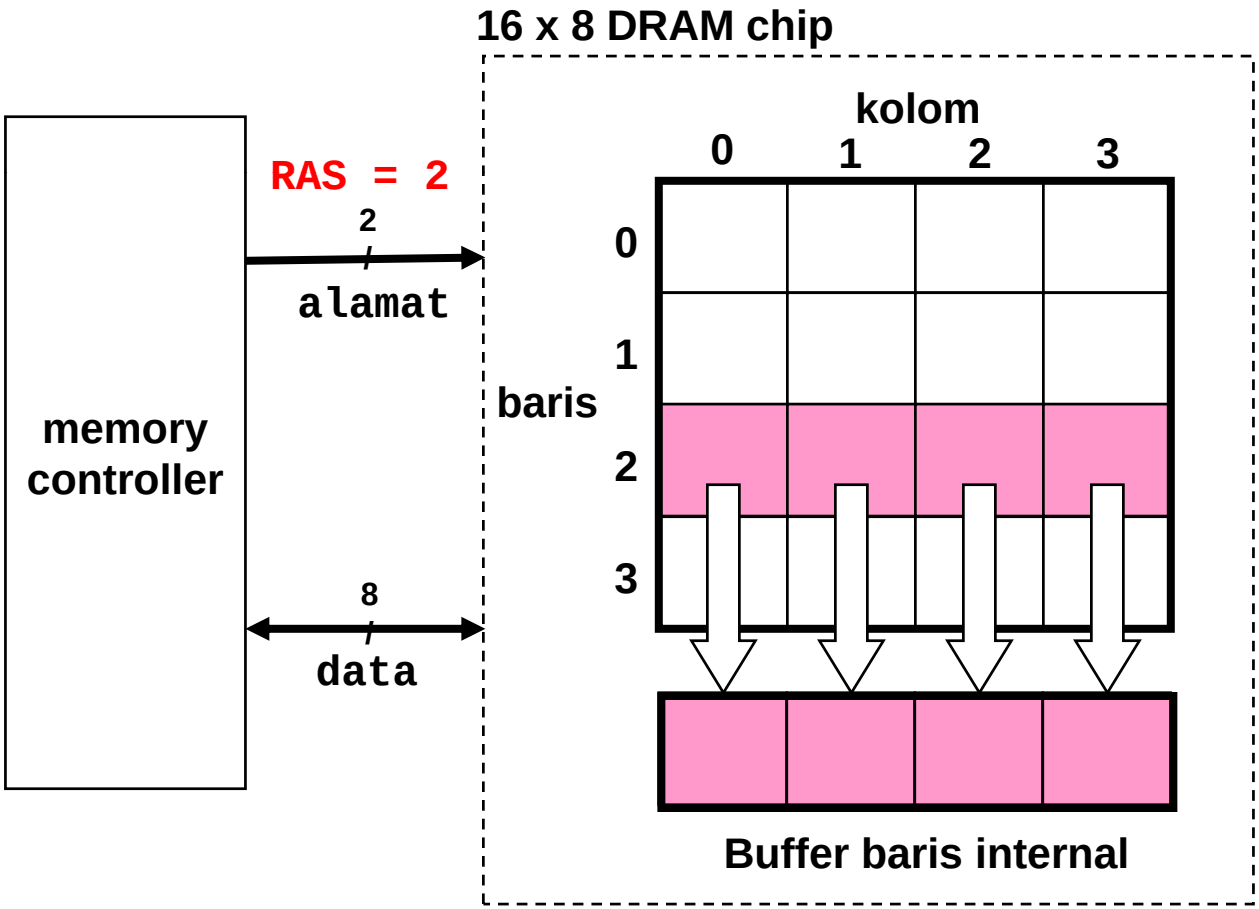
Organisasi DRAM Konvensional

- Total d x w bit, disimpan dalam d buah supersel berukuran w bit



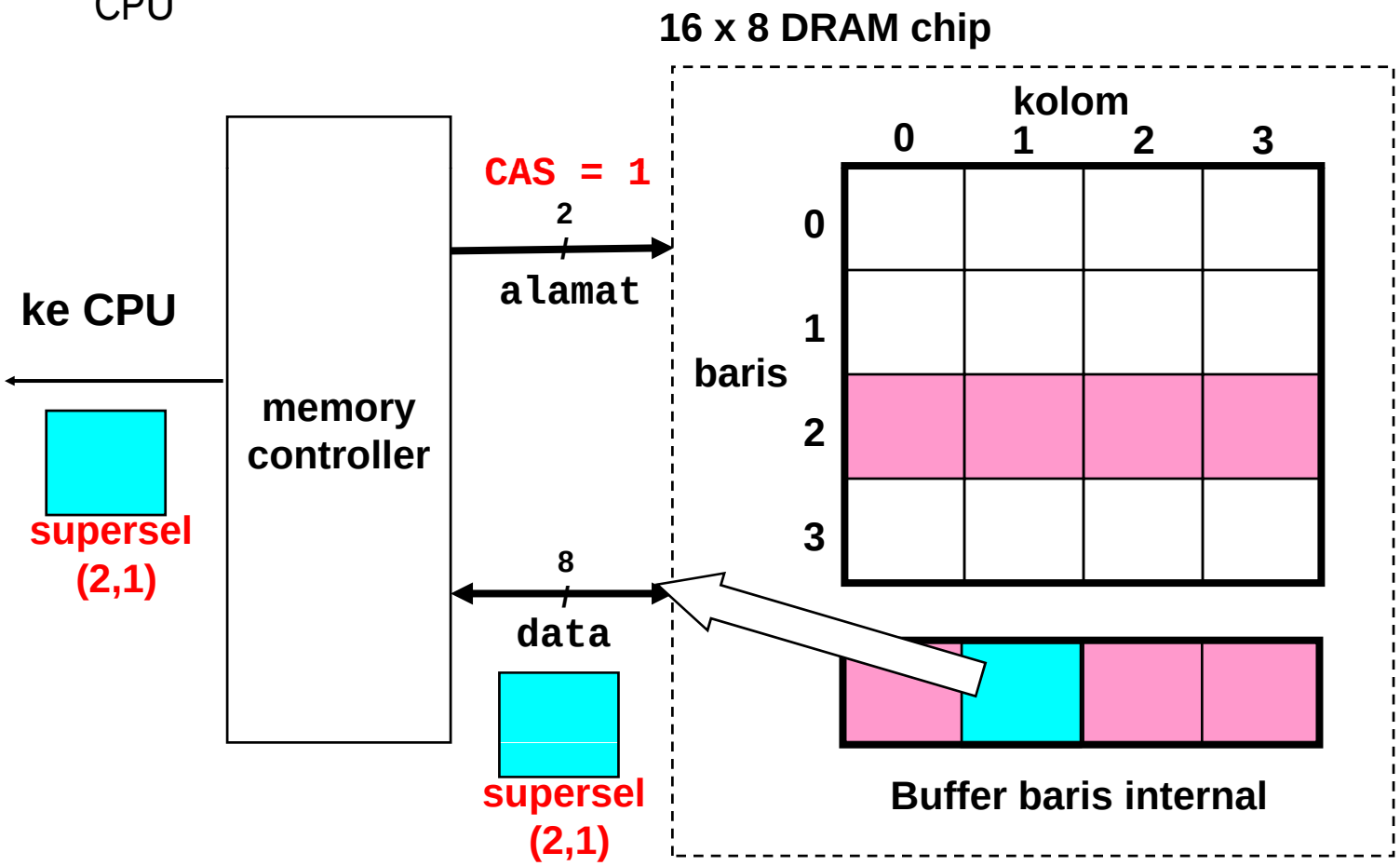
Membaca DRAM – Supersel (2,1)

- Langkah 1(a): Row access strobe (RAS) memilih baris ke 2.
- Langkah 1(b): Baris 2 disalin dari DRAM array ke buffer baris.

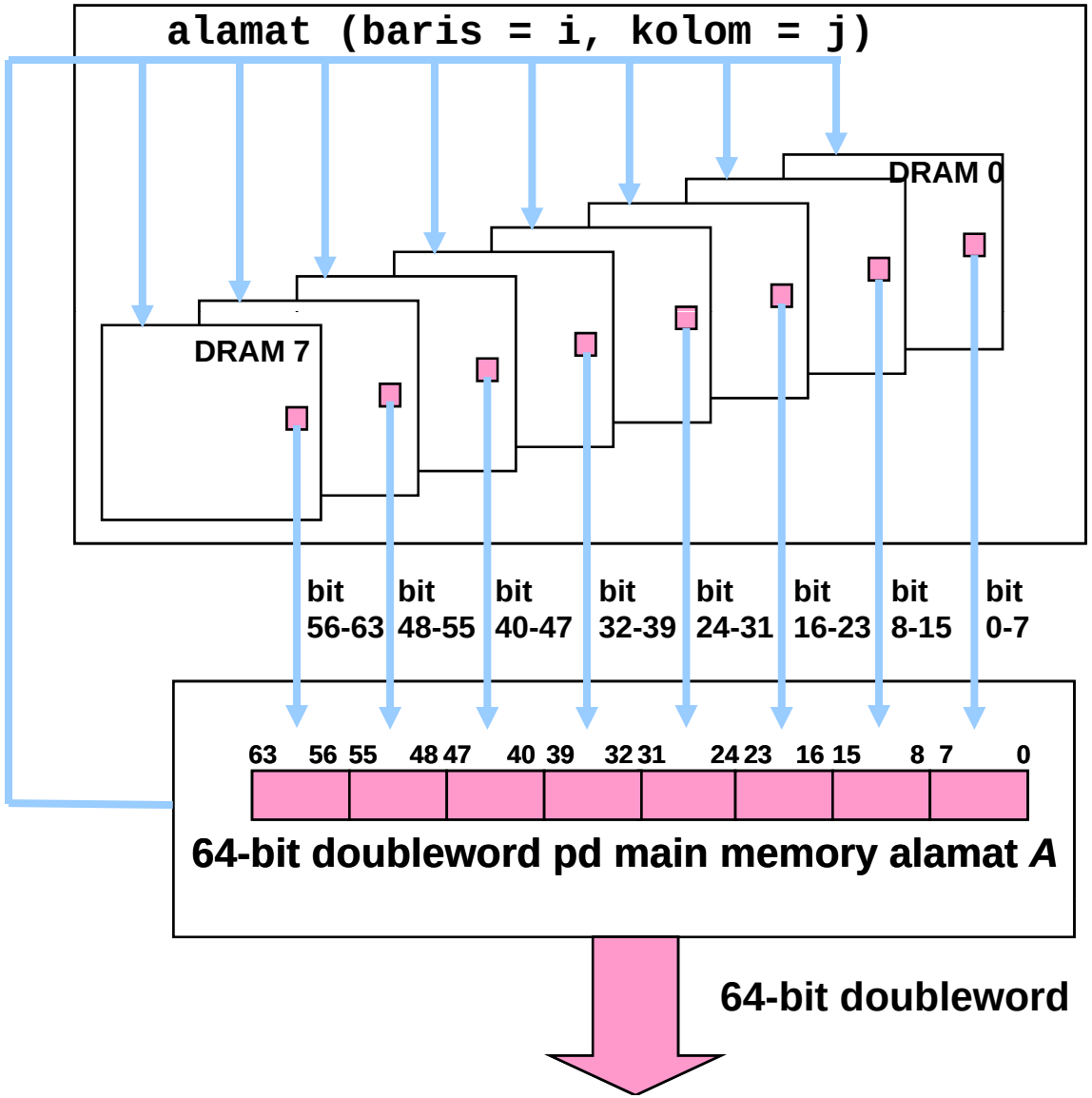


Membaca DRAM – Supersel (2,1)

- Langkah 2(a): Column access strobe (CAS) memilih kolom 1.
- Langkah 2(b): Supercell (2,1) disalin dari buffer ke saluran data & dikirim ke CPU



Modul Memori



: supersel (i,j)

Modul memori 64 MB terdiri dari :
8M x 8 DRAM (baris x kolom)

Memory controller

Enhanced DRAM

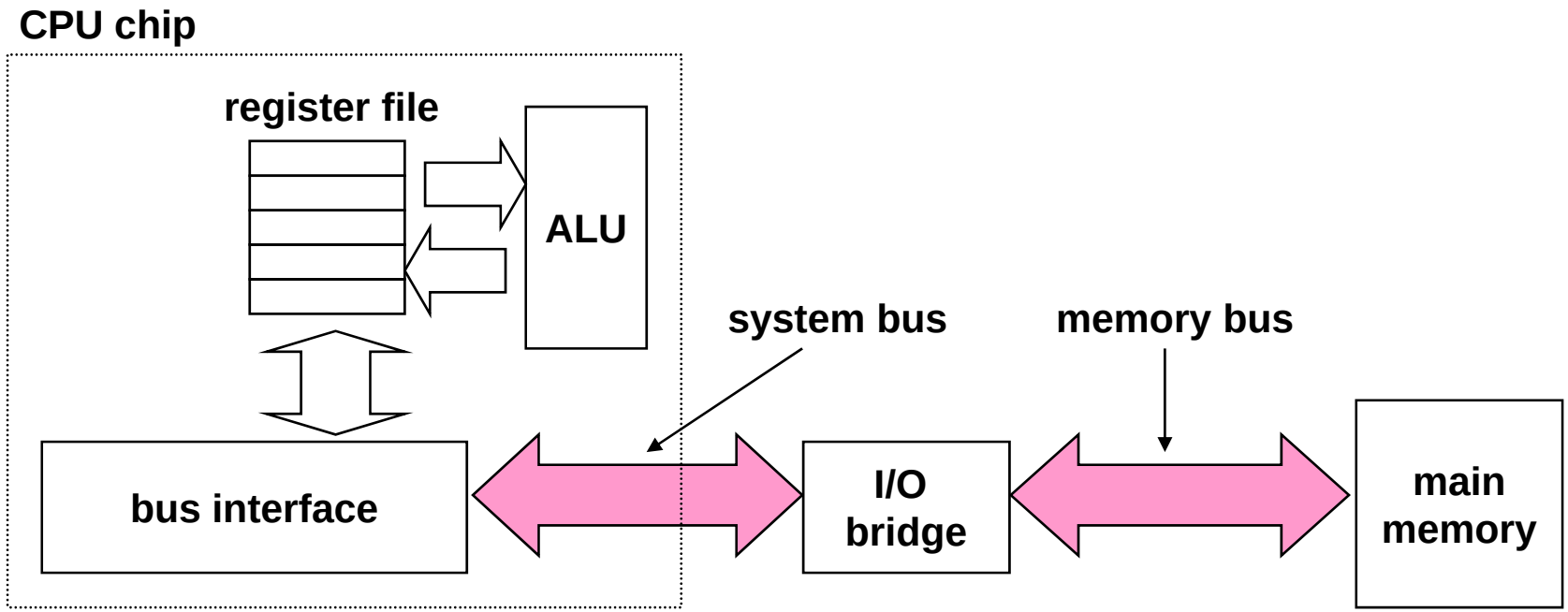
- Seluruh DRAM yang ada saat ini dibangun berdasarkan DRAM konvensional, yaitu :
  - Fast Page Mode DRAM (FPM DRAM)
    - Mengakses isi baris secara [RAS, CAS, CAS, CAS], bukan [(RAS,CAS), (RAS,CAS), (RAS,CAS), (RAS,CAS)]
  - Extended Data Output DRAM (EDO DRAM)
    - Peningkatan FPM DRAM, jarak antara sinyal CAS lebih dekat
  - Synchronous DRAM (SDRAM)
    - Dikendalikan secara sinkron pada sisi clock naik
  - Double Data-Rate Synchronous DRAM (DDR SDRAM)
    - Peningkatan SDRAM, sinyal kontrol menggunakan kedua sisi clock
  - Video RAM (VRAM)
    - Seperti FPM DRAM, output diperoleh dengan menggeser buffer baris (shift)
    - Memiliki dua port (dapat melakukan proses baca dan tulis secara bersamaan

Nonvolatile Memory

- DRAM dan SRAM termasuk kategori volatile memory
  - Informasi hilang jika daya listrik dimatikan
- Nonvolatile memory dapat mempertahankan isinya walaupun daya dimatikan
  - Nama generiknya adalah read-only memory (ROM)
  - Terjadi salah arti, karena beberapa ROM dapat dimodifikasi
- Jenis-jenis ROM
  - Programmable ROM (PROM)
  - Eraseable Programmable ROM (EPROM)
  - Electrically Eraseable PROM (EEPROM)
  - Flash memory
- Firmware
  - Program yang disimpan di ROM
    - Boot time code, BIOS (basic input/output system)
    - Graphic card, disk controller

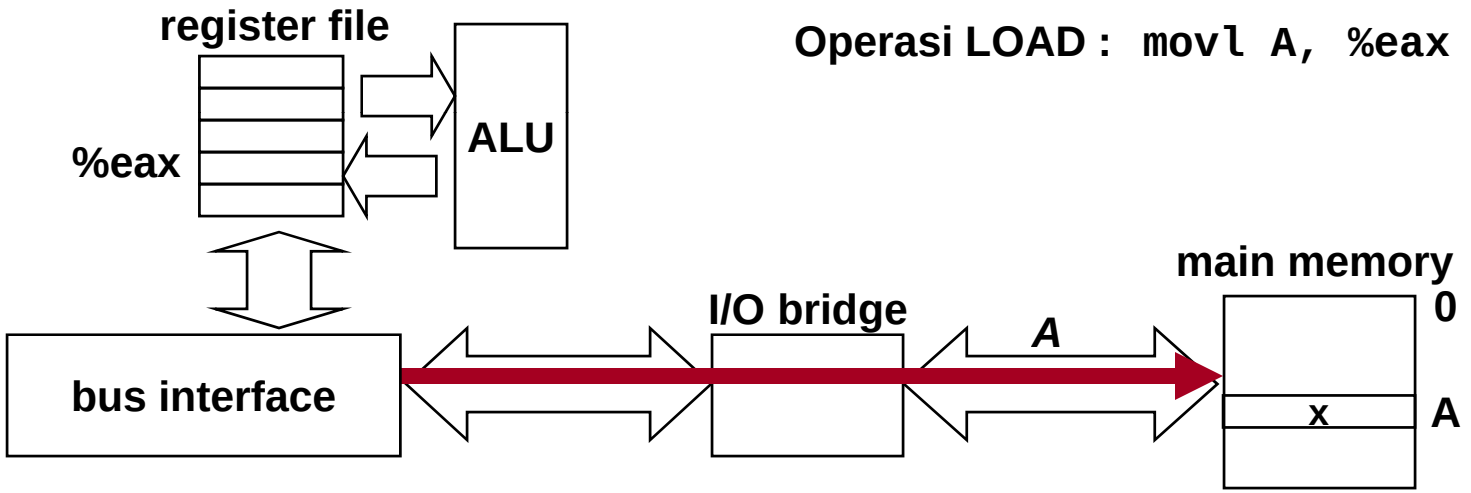
Struktur Bus CPU-Memori

- Bus adalah kumpulan saluran paralel yang mengalirkan sinyal alamat, data dan kontrol.
- Umumnya digunakan bersama oleh beberapa devais.



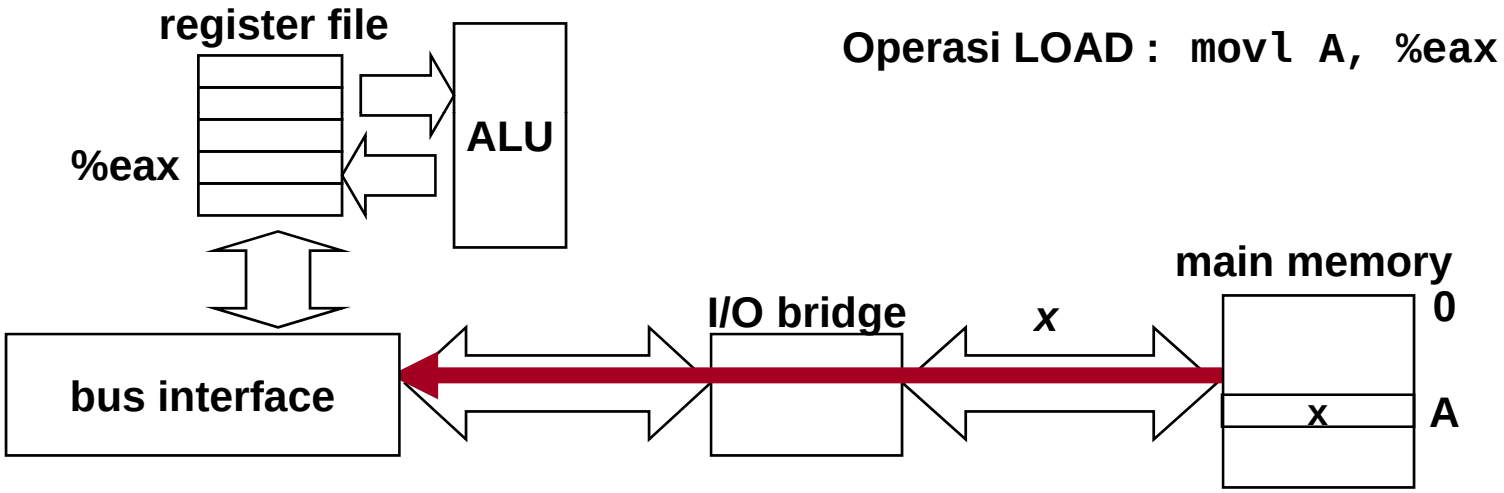
Proses Membaca Memori (1)

- CPU meletakkan alamat A pada memory bus.



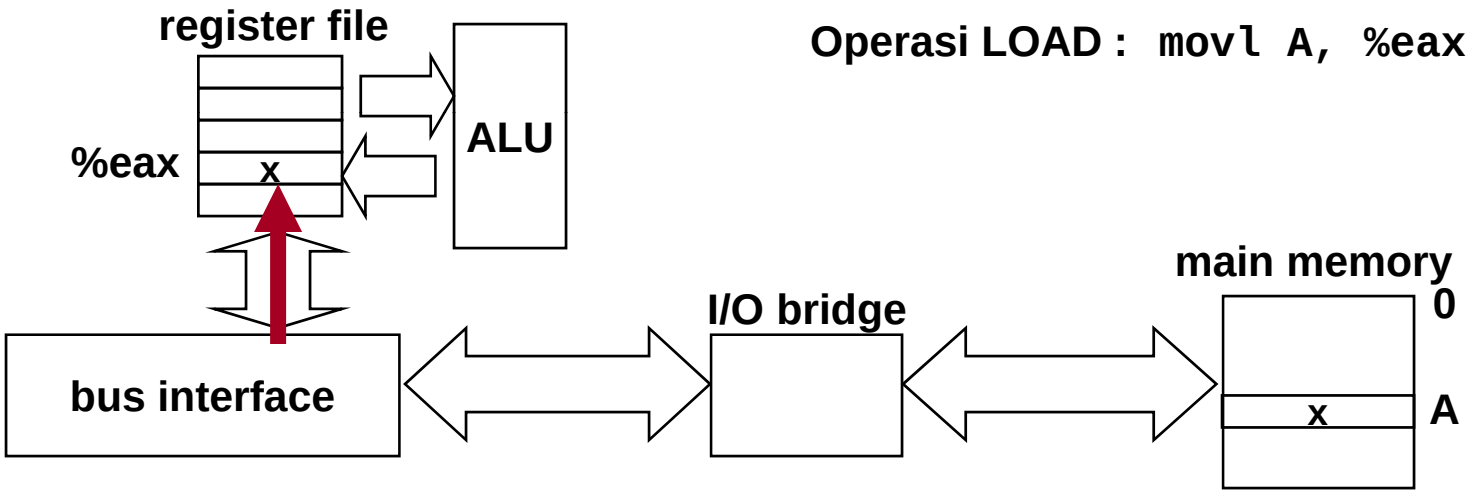
Proses Membaca Memori (2)

- Main memory membaca A dari memory bus, mengambil word x, dan meletakkannya pada bus



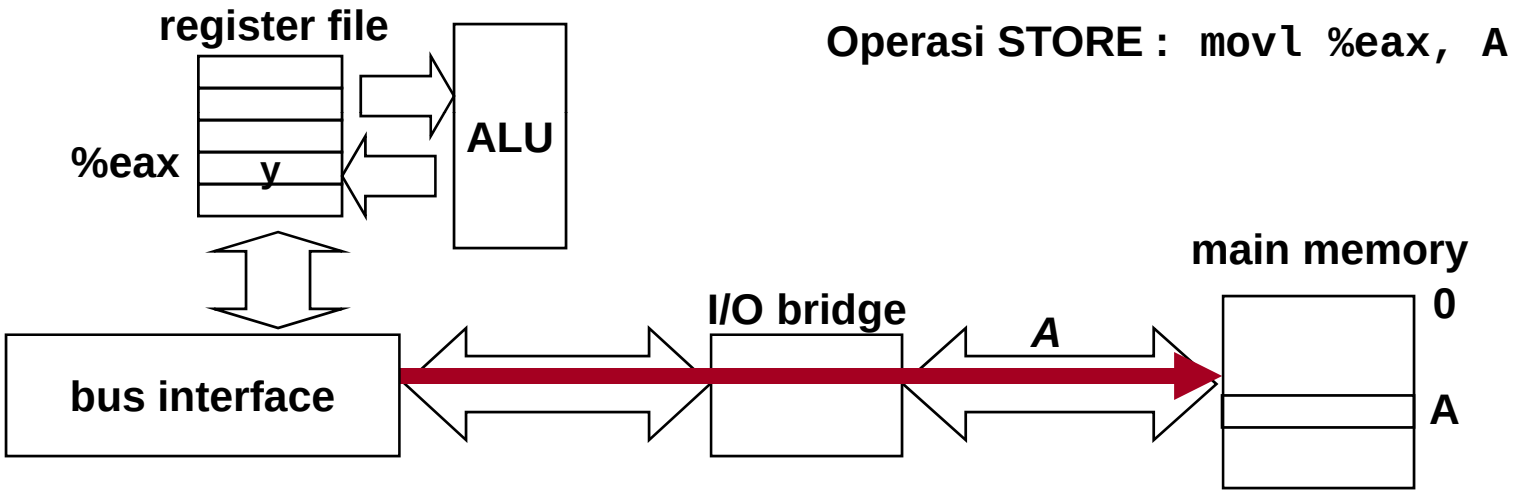
Proses Membaca Memori (3)

- CPU membaca word x dari bus dan menyalinnya ke register %eax



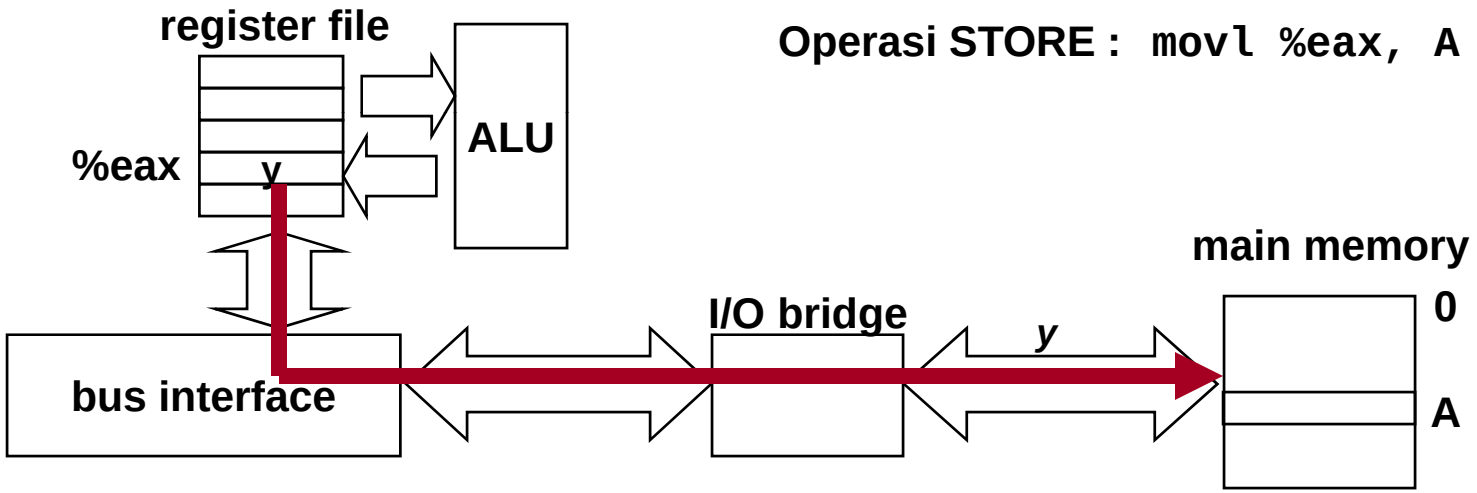
Proses Menulis ke Memori (1)

- CPU meletakkan alamat A pada bus. Main memory membacanya dan menunggu munculnya word data.



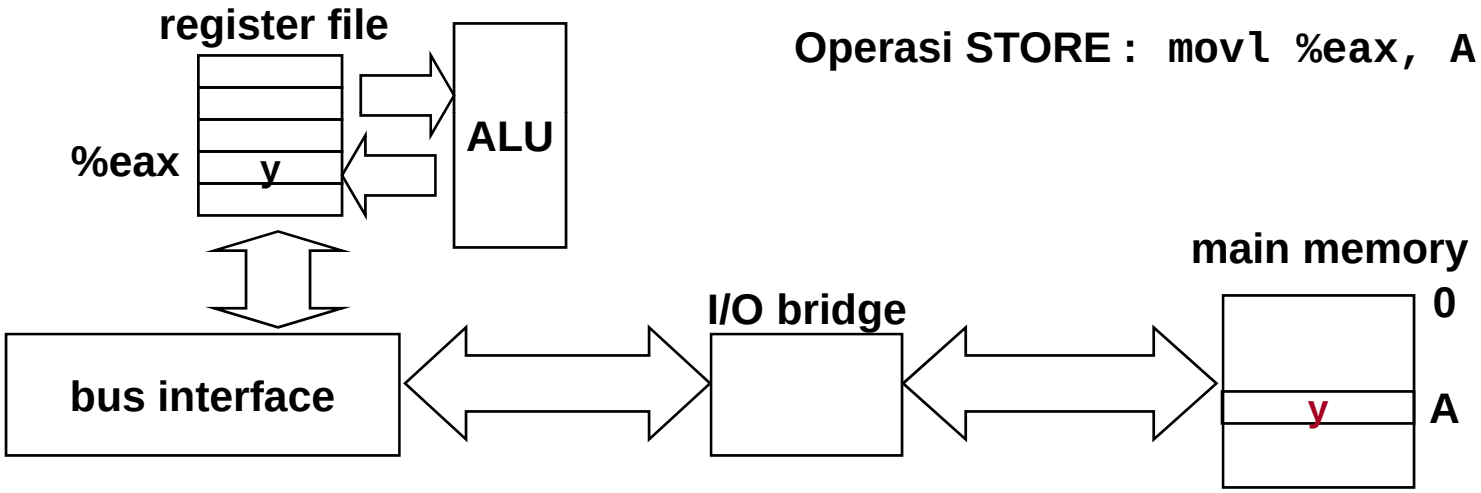
Proses Menulis ke Memori (2)

- CPU meletakkan word data y pada bus



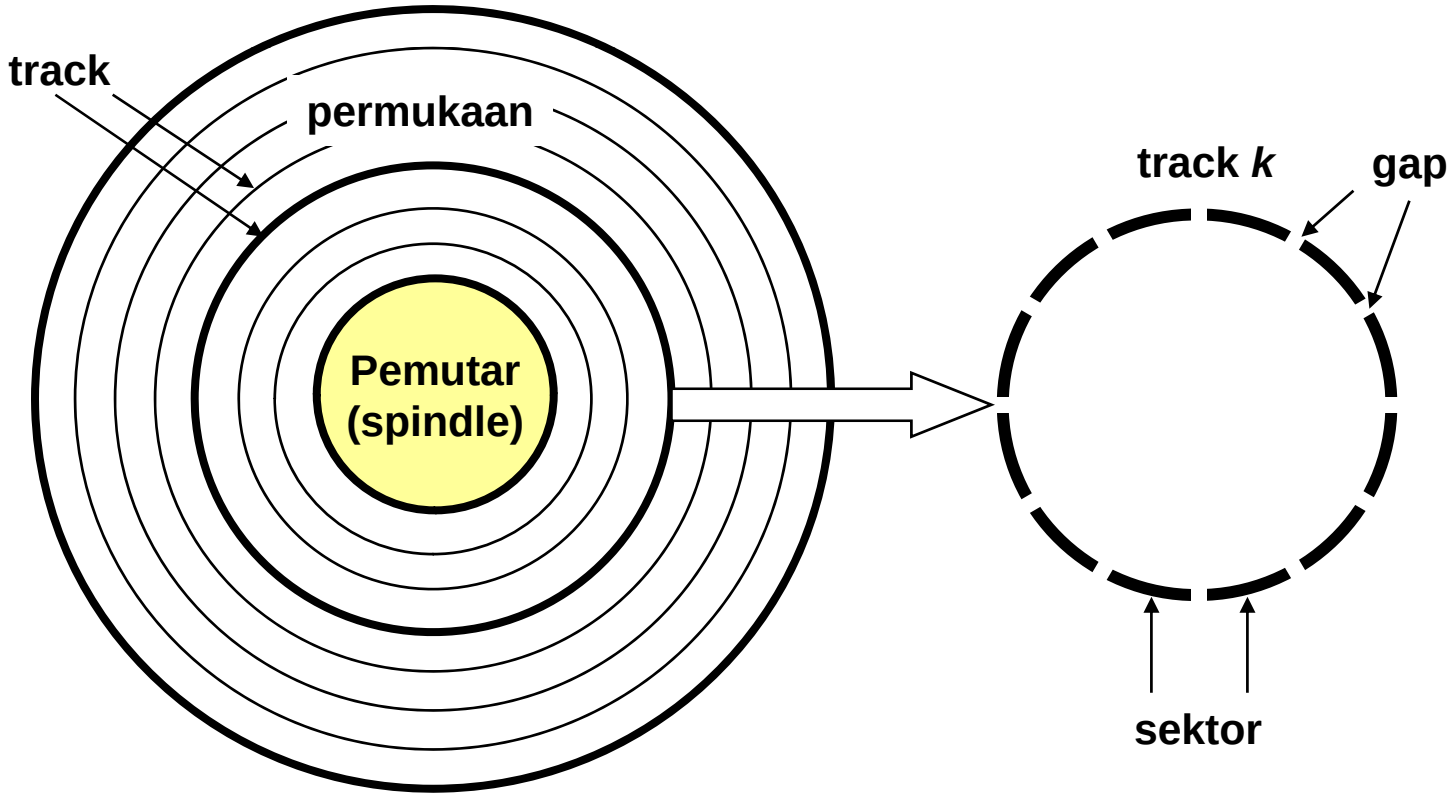
Proses Menulis ke Memori (3)

- Main memory membaca word data y dari bus dan menyimpannya di alamat A.



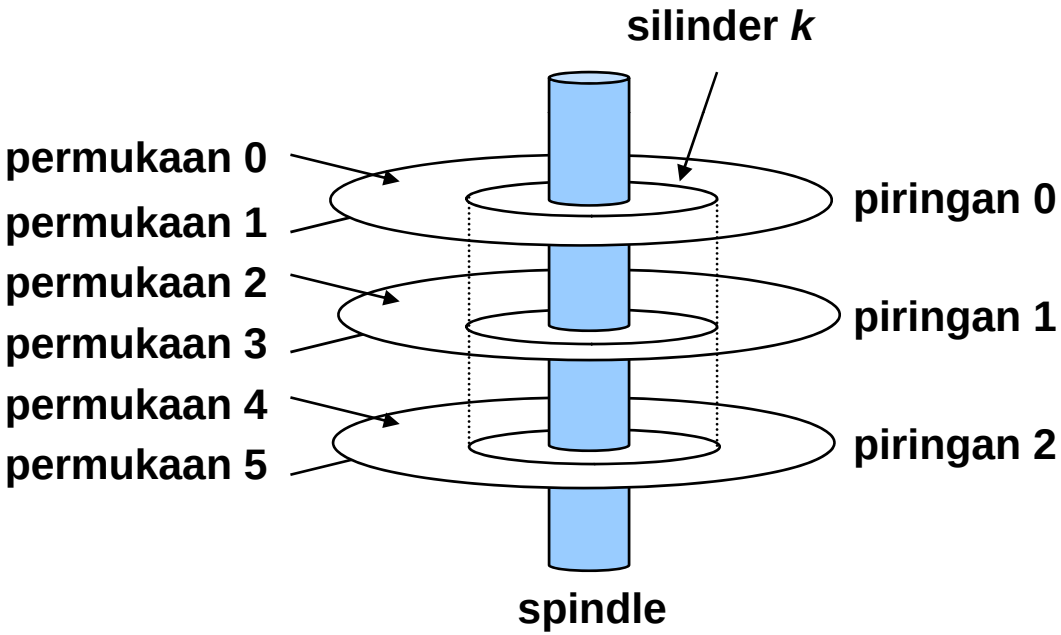
Hard Disk

- Hard disk terdiri dari beberapa piringan, masing-masing memiliki dua permukaan.
- Pada setiap permukaan terdapat lingkaran konsentrik yang disebut track.
- Setiap track terbagi atas beberapa sektor yang dipisahkan oleh jarak tertentu (gap).



Geometri Hard Disk

- Track pada jalur yang sama membentuk silinder



Kapasitas Hard Disk

- Kapasitas : banyaknya bit maksimum yang dapat disimpan
  - Produsen menulis kapasitas dalam satuan gigabyte (GB)
  - 1 GB = 10^9
- Kapasitas ditentukan oleh faktor teknologi berikut :
  - Recording density (bit/inci): banyaknya bit yang dapat ditempatkan dalam segmen track sepanjang 1 inci.
  - Track density (track/inci): banyaknya track yang dapat ditempatkan dalam satu segmen radial sepanjang 1 inci.
  - Areal density (bit/inci^2): perkalian dari recording density dan track density.
- Hard disk saat ini membagi track menjadi beberapa bagian yang tidak saling berhubungan, disebut recording zones
  - Setiap track dalam satu zona memiliki jumlah sektor yang sama, ditentukan oleh panjang track yang paling dalam.
  - Setiap zona memiliki jumlah sektor/track yang berbeda

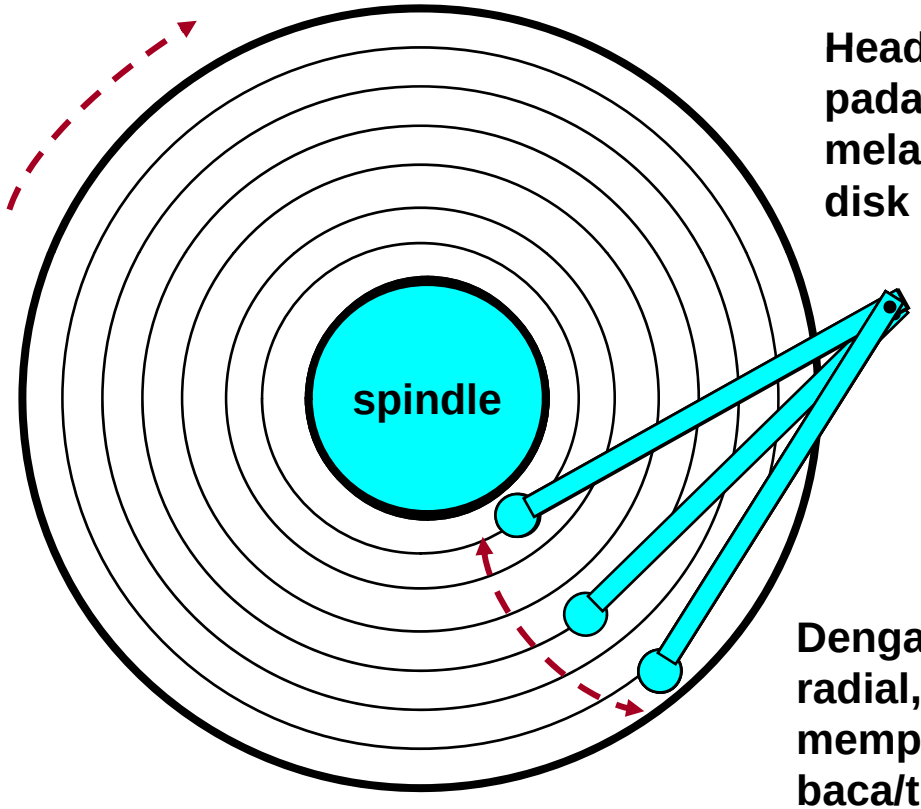
Menghitung Kapasitas HardDisk

- Kapasitas = (# byte/sektor) x (rata2 # sektor/track) x
  (# track/permukaan) x (# permukaan/piringan) x
  (# piringan/disk)
- Contoh :
  - 512 byte/sektor
  - 300 sektor/track (rata-rata)
  - 20,000 track/permukaan
  - 2 permukaan/piringan
  - 5 piringan/disk

- Kapasitas = 512 x 300 x 20000 x 2 x 5
  = 30,720,000,000
  = 30.72 GB

Operasi Hard Disk

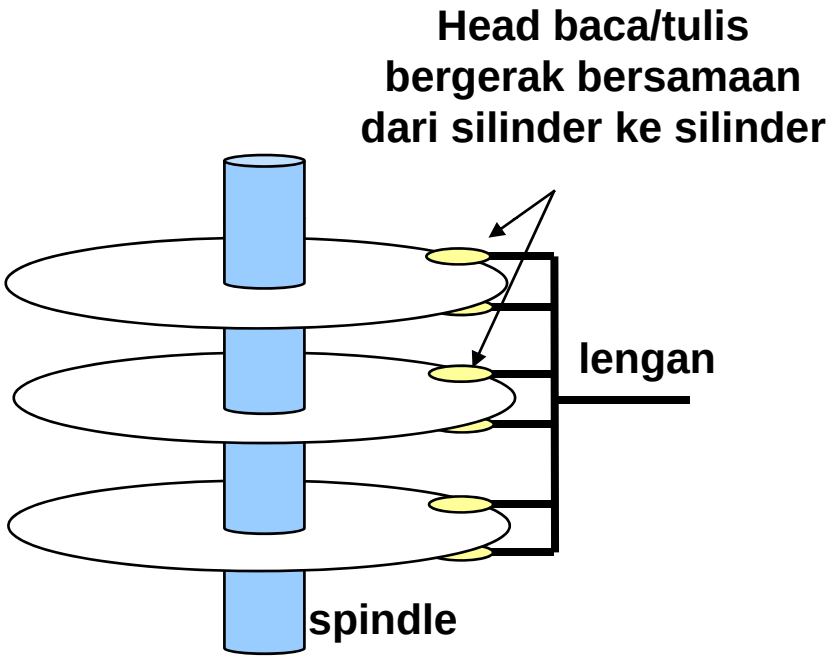
Permukaan hard disk berputar dengan kecepatan rotasi tetap



Head baca/tulis diletakkan pada ujung lengan, dan melayang di atas permukaan disk ketika disk berputar

Dengan bergerak secara radial, lengan dapat memposisikan head baca/tulis pada berbagai track

Operasi Hard Disk



Waktu Akses Hard Disk

- Waktu rata-rata untuk mengakses suatu sektor target dapat dihitung :
  - Taccess = Tavg seek + Tavg rotasi + Tavg transfer
- Waktu pencarian (Tavg seek)
  - Waktu yang diperlukan untuk meletakkan head di atas silinder yang mengandung sektor target
  - Umumnya Tavg seek = 9 mdetik
- Delay rotasi (Tavg rotasi)
  - Waktu untuk menunggu bit pertama sektor target berada di bawah head baca/tulis
  - Tavg rotasi = ½ x 1/RPM x 60 detik/1 menit
- Waktu transfer (Tavg transfer)
  - Waktu untuk membaca bit-bit pada sektor target
  - Tavg transfer = 1/RPM x 1 (rata2 # sektor/track) x 60 detik/1 menit

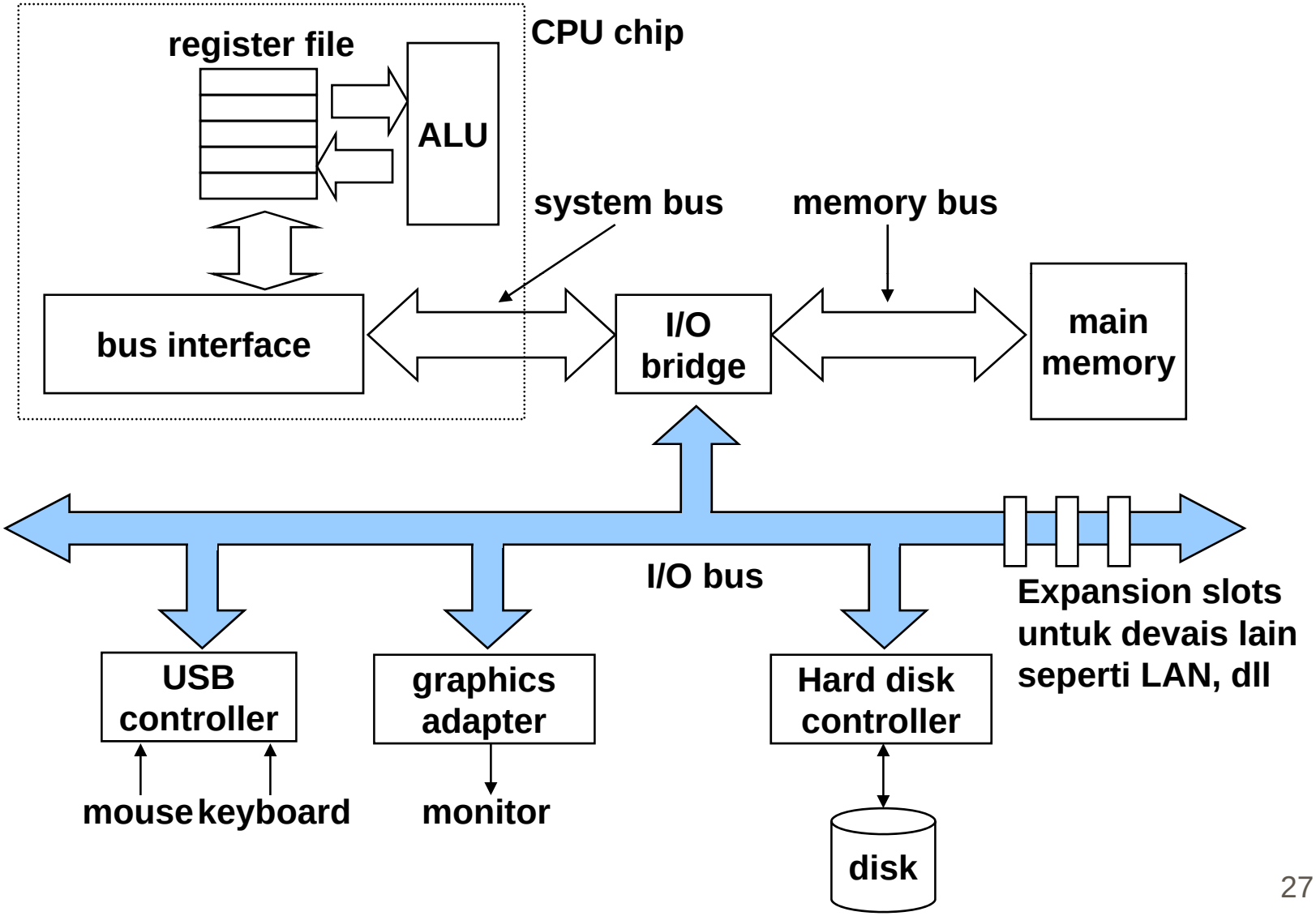
Contoh Waktu Akses HardDisk

- Diketahui :
  - Kecepatan rotasi = 7,200 RPM
  - Waktu pencarian rata-rata = 9 ms.
  - Rata2 # sektor/track = 400.
- Diturunkan :
  - Tavg rotasi = 1/2 x (60 det/7200 RPM) x 1000 mdetik/detik = 4 mdetik
  - Tavg transfer = 60/7200 RPM x 1/400 sektor/track x 1000 mdetik/detik = 0.02 mdetik
  - Taccess = 9 mdetik + 4 mdetik + 0.02 mdetik
- Hal penting :
  - Waktu akses didominasi oleh waktu pencarian dan delay rotasi
  - Bit pertama pada sektor adalah yang paling berpengaruh.
  - Waktu akses SRAM sekitar 4 ndetik/doubleword, DRAM 60 ndetik
    - Hard disk sekitar 40.000 kali lebih lambat dari SRAM
    - 2.500 lebih lambat dari DRAM

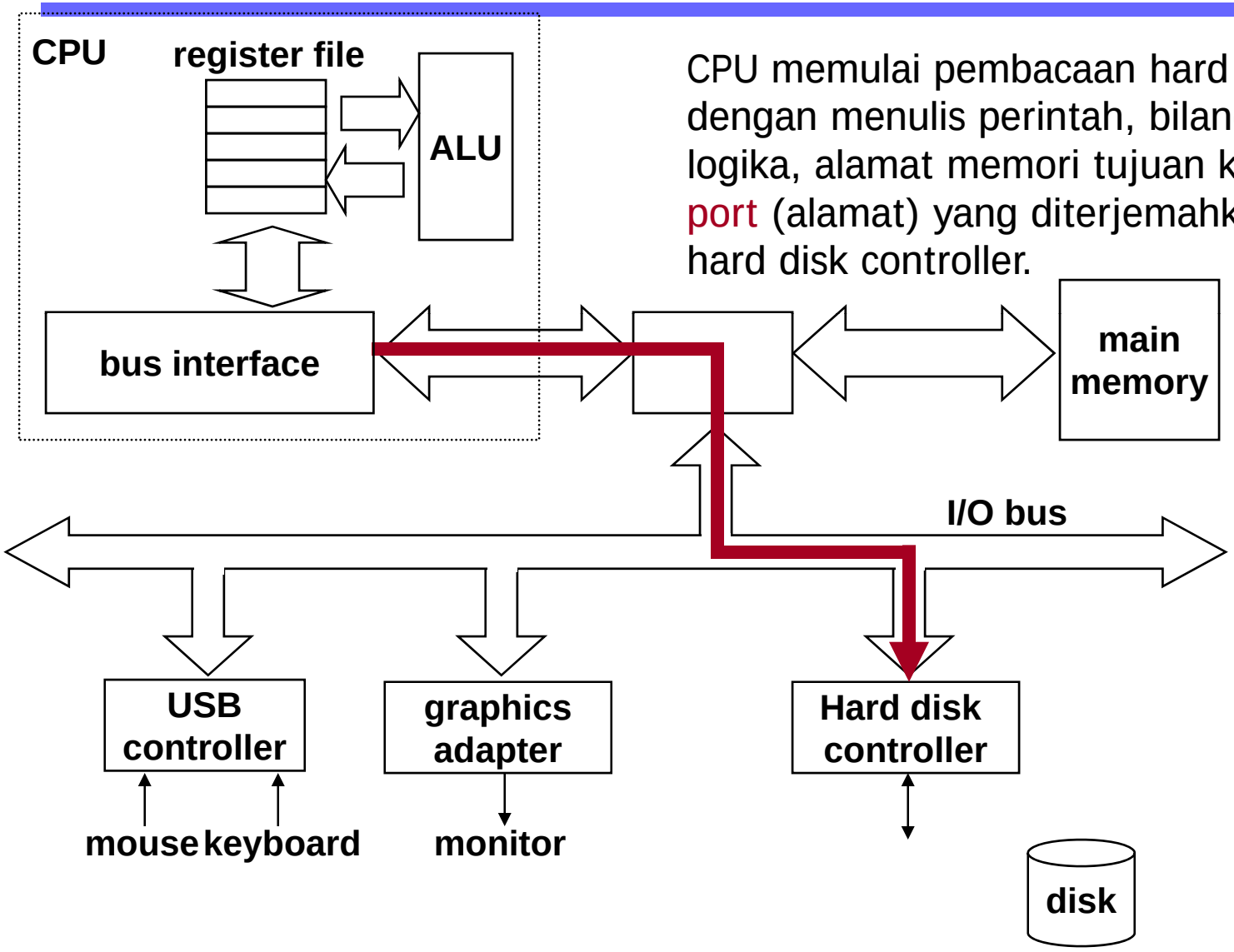
Blok Logika Hard Disk

- Pada hard disk modern, sektor geometri yang rumit dapat direpresentasikan dengan sudut pandang yang lebih sederhana
  - Set dari sektor yang tersedia dimodelkan dalam urutan blok logika berukuran b (0, 1, 2, ...)
- Memetakan blok logika dan sektor fisik sesungguhnya
  - Dikelola oleh suatu devais perangkat keras yang disebut harddisk controller.
  - Menerjemahkan permintaan akan blok logika menjadi urutan permukaan-track-sektor
- Harddisk controller mengatur penggunaan silinder untuk setiap zona
  - Menghitung perbedaan antara “kapasitas setelah diformat” dan “kapasitas maksimum”

Bus I/O

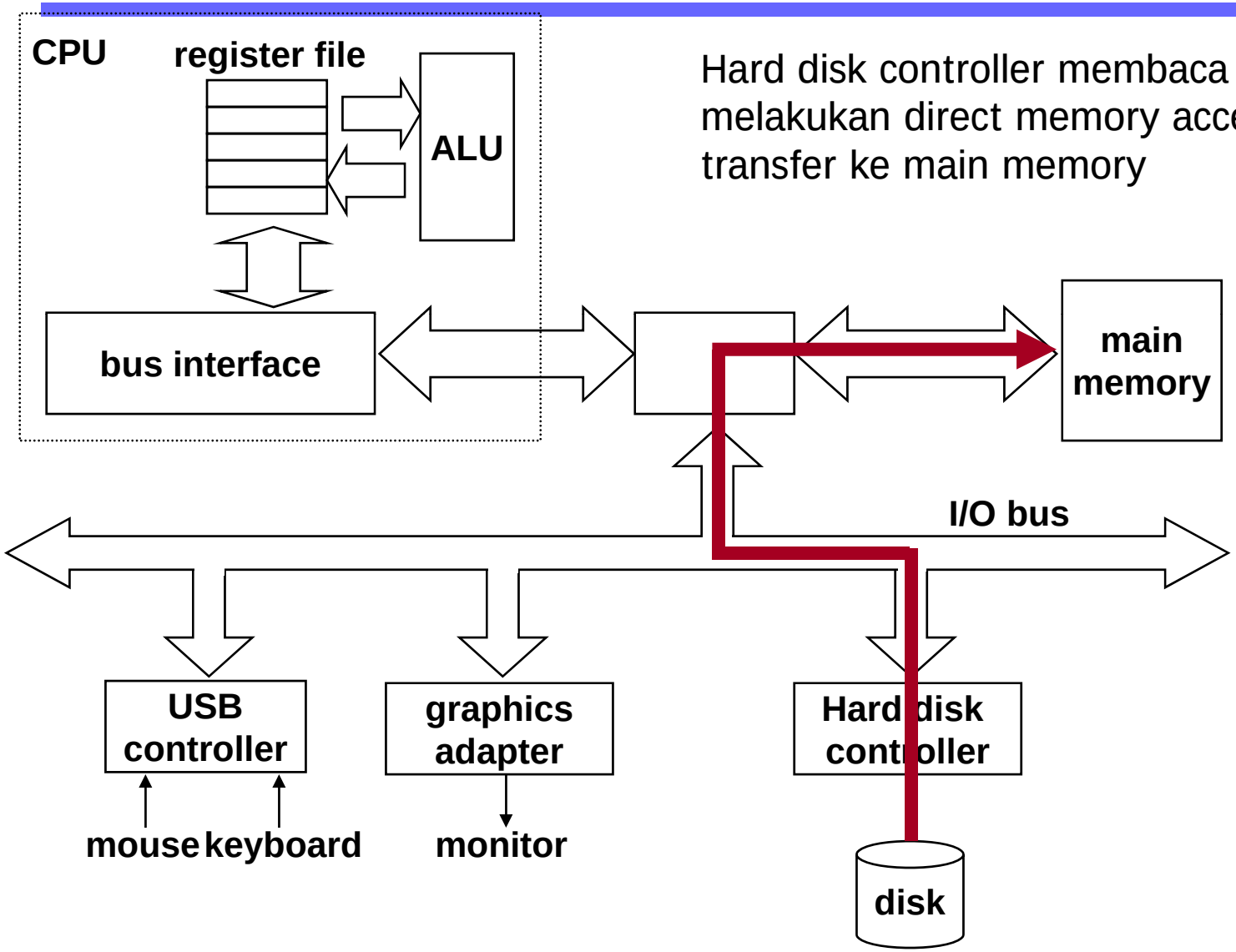

CPU chip
register file
ALU
system bus
memory bus
bus interface
I/O bridge
main memory
I/O bus
Expansion slots untuk devais lain seperti LAN, dll
USB controller
graphics adapter
Hard disk controller
mouse
keyboard
monitor
disk

Membaca Sektor Hard Disk (1)



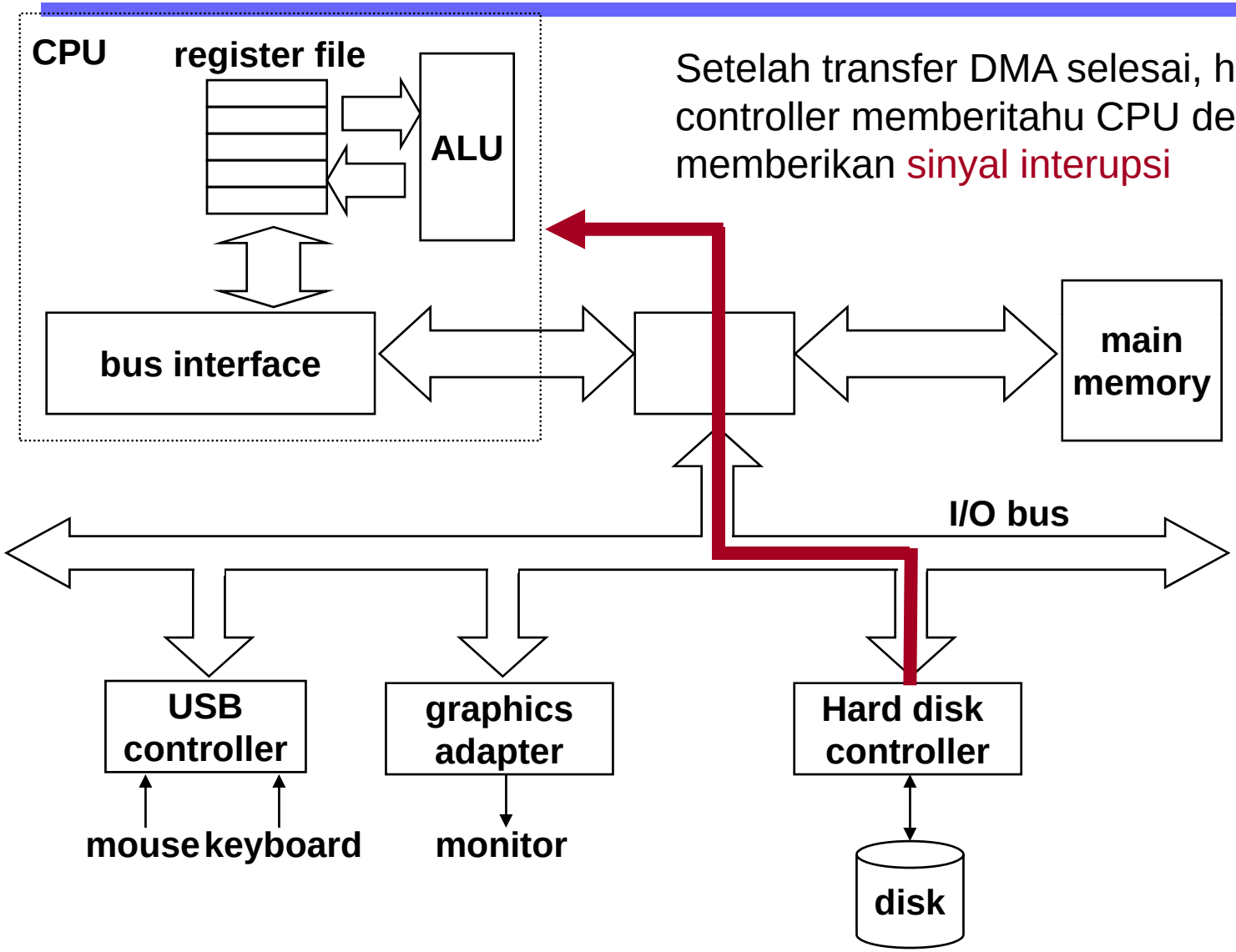
CPU memulai pembacaan hard disk dengan menulis perintah, bilangan blok logika, alamat memori tujuan ke suatu port (alamat) yang diterjemahkan oleh hard disk controller.

Membaca Sektor Hard Disk (2)



Hard disk controller membaca sektor dan melakukan direct memory access (DMA) transfer ke main memory

Membaca Sektor Hard Disk (3)



Setelah transfer DMA selesai, hard disk controller memberitahu CPU dengan memberikan sinyal interupsi

Perkembangan Memori

SRAM

metrik	1980	1985	1990	1995	2000	*2000:1980*
$/MB	19,200	2,900	320	256	100	*190*
akses(ndetik)	300	150	35	15	2	*100*

DRAM

metrik	1980	1985	1990	1995	2000	*2000:1980*
$/MB	8,000	880	100	30	1	*8,000*
akses (ndetik)	375	200	100	70	60	*6*
ukuran (MB)	0.064	0.256	4	16	64	*1,000*

Disk

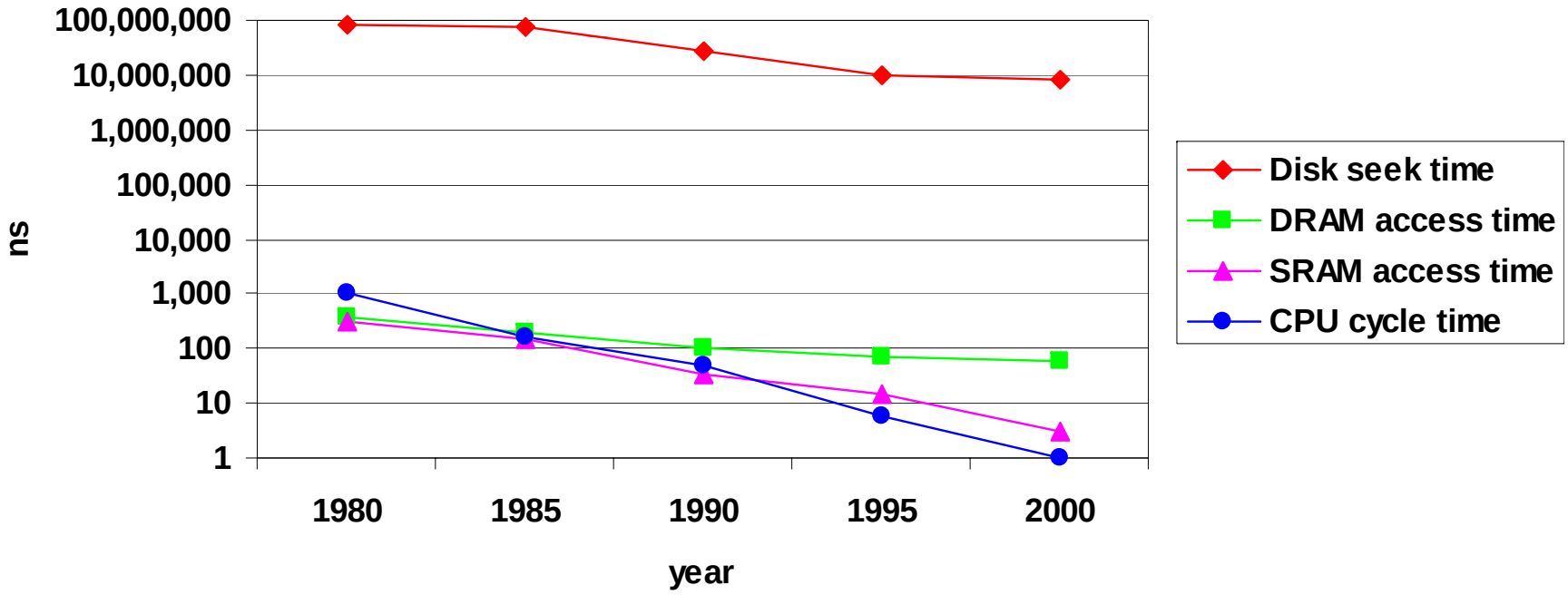
metrik	1980	1985	1990	1995	2000	*2000:1980*
$/MB	500	100	8	0.30	0.05	*10,000*
akses (mdetik)	87	75	28	10	8	*11*
ukuran (MB)	1	10	160	1,000	9,000	*9,000*

Kecepatan Clock CPU

	1980	1985	1990	1995	2000	*2000:1980*
Prosesor	**8080**	**286**	**386**	**Pent**	**P-III**	
clock rate(MHz)	**1**	**6**	**20**	**150**	**750**	**750**
cycle time(ns)	**1,000**	**166**	**50**	**6**	**1.6**	**750**

Gap Kecepatan CPU - Memori

- Perbedaan kecepatan antara DRAM, Hard disk dan CPU terus meningkat



Locality

- Prinsip locality :
  - Program cenderung untuk memakai ulang data dan instruksi yang letaknya berdekatan dengan yang sebelumnya digunakan, atau yang pernah mereferensikannya.
  - Temporal locality : sesuatu yang pernah direferensikan cenderung akan direferensikan kembali pada waktu yang tidak lama.
  - Spatial locality : sesuatu yang letak alamatnya berdekatan cenderung untuk direferensikan secara bersama pada satu waktu

Locality

- Contoh locality :

```
sum = 0;
for (i = 0; i < n; i++)
    sum += a[i];
return sum;
```

  - Data
    - Mereferensikan elemen array secara berurutan (pola stride-1 reference) : Spatial locality
    - Mereferensikan `sum` pada setiap iterasi : Temporal locality
  - Instruksi
    - Mereferensikan instruksi secara berurutan : Spatial locality
    - Berputar dalam loop secara berulang-ulang : Temporal locality

Contoh Locality

- Kemampuan untuk melihat kode dan mengetahui locality secara kualitatif merupakan kunci yang harus dimiliki oleh seorang programmer profesional.

- Apakah fungsi di bawah ini memiliki locality yang baik ?

```
int sumarrayrows(int a[M][N])
{
    int i, j, sum = 0;

    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum += a[i][j];
    return sum
}
```

Contoh Locality

- Apakah fungsi berikut memiliki locality yang baik ?

```
int sumarraycols(int a[M][N])
{
    int i, j, sum = 0;

    for (j = 0; j < N; j++)
        for (i = 0; i < M; i++)
            sum += a[i][j];
    return sum
}
```

Contoh Locality

- Lakukan permutasi pada loop sehingga fungsi dapat melacak array 3D **`a[]`** dengan pola stride-1 reference (dan menghasilkan locality spasial yang baik).

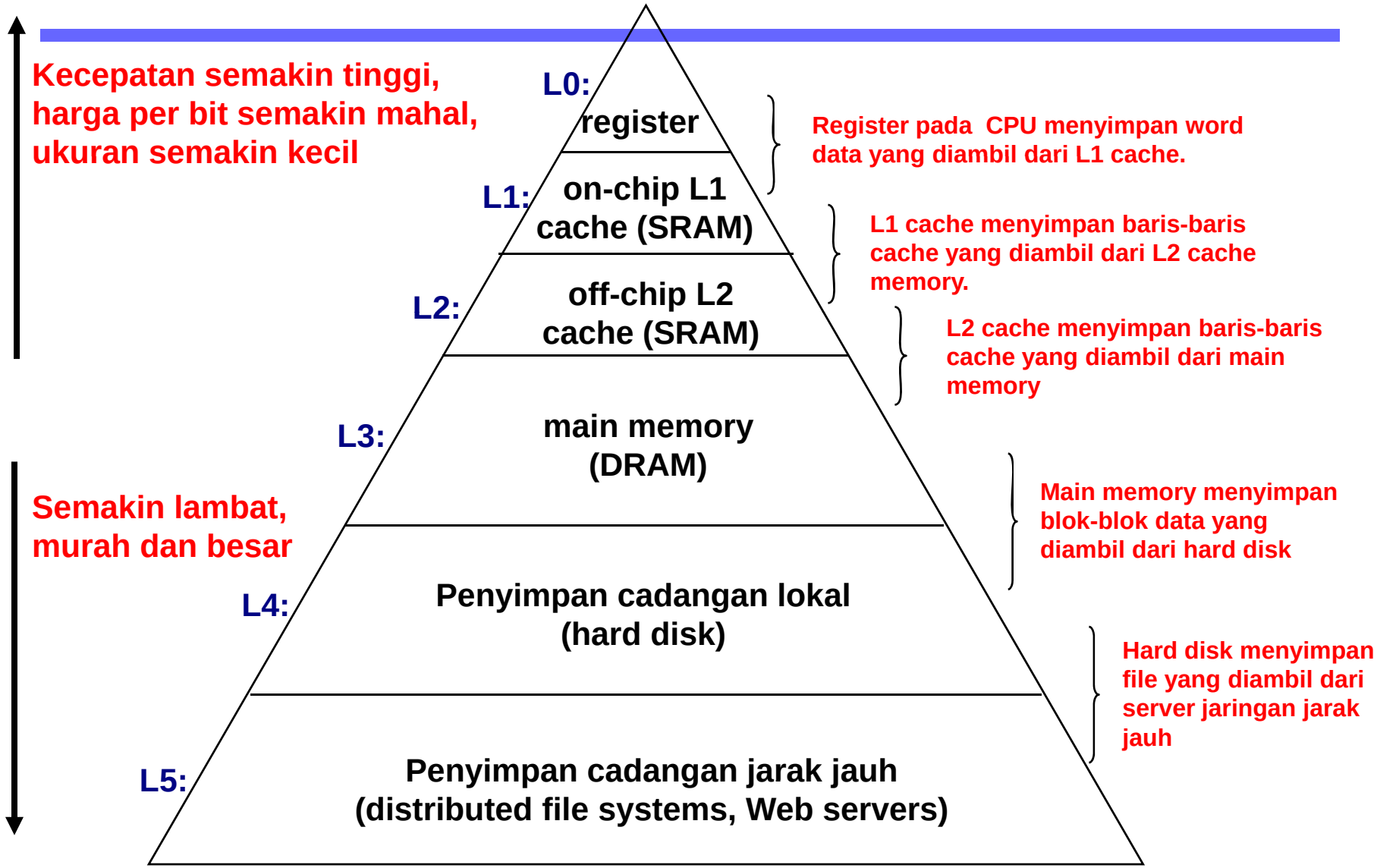
```
int sumarray3d(int a[M][N][N])
{
    int i, j, k, sum = 0;

    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            for (k = 0; k < N; k++)
                sum += a[k][i][j];
    return sum
}
```

Hirarki Memori

- Beberapa perilaku dasar pada hardware dan software :
  - Teknologi penyimpanan yang cepat memiliki biaya per byte lebih tinggi dan kapasitas lebih rendah
  - Perbedaan kecepatan antara CPU dan memori bertambah lebar
  - Program yang dibuat dengan baik cenderung untuk memiliki locality yang baik
- Setiap perilaku dasar tersebut saling mengisi satu sama lain dengan baik
- Diusulkan untuk mengelola memori dan sistem penyimpanan yang dikenal dengan hirarki memori.

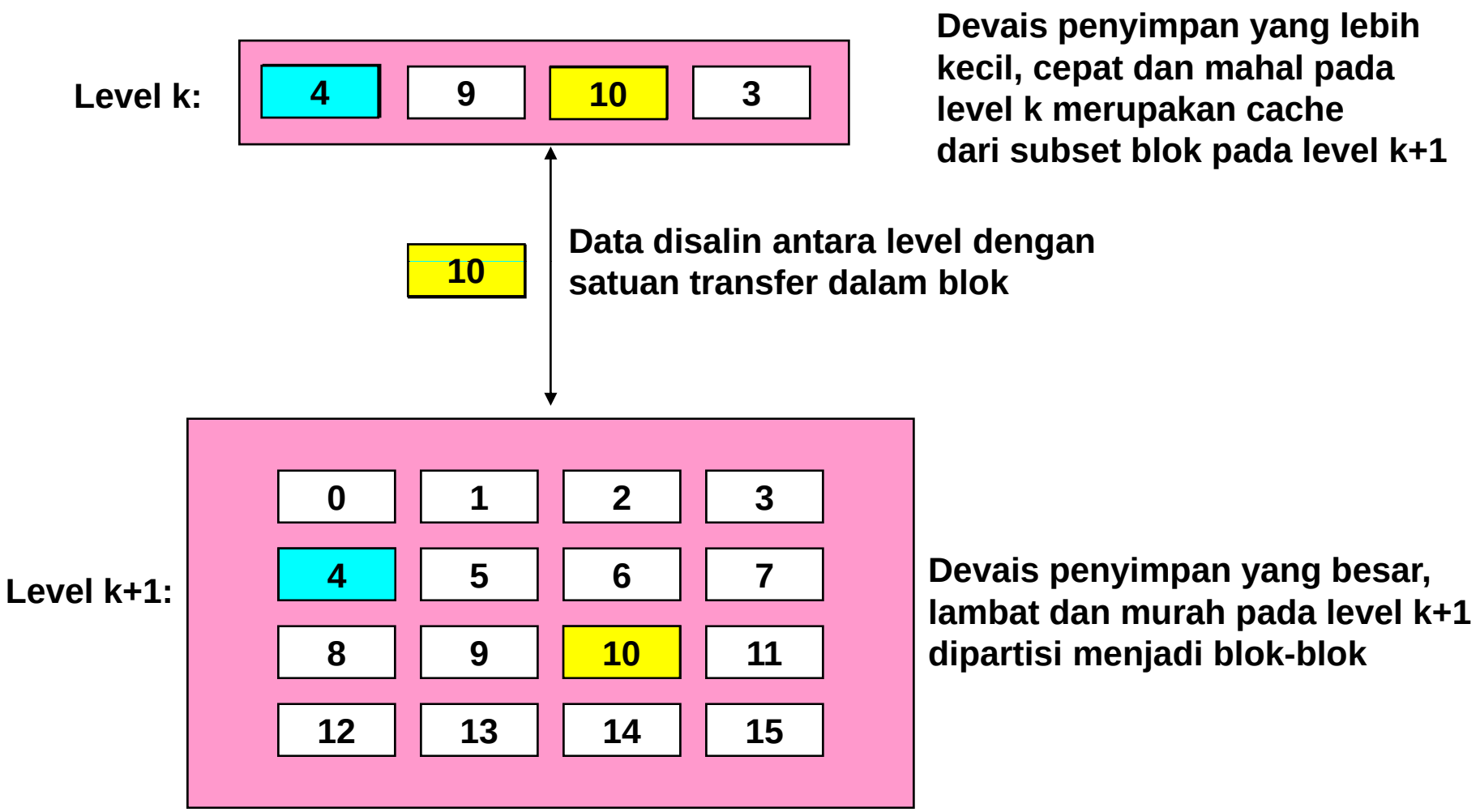
Hirarki Memori



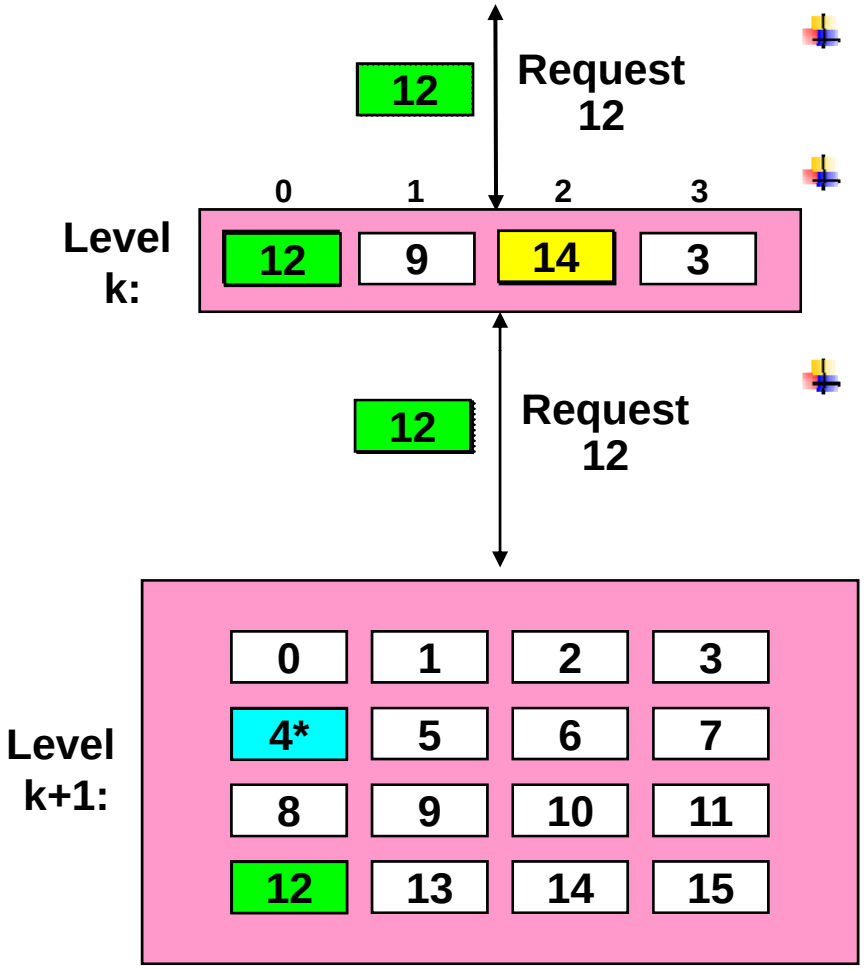
Cache

- Perangkat penyimpan cepat dan kecil, berfungsi sebagai area antara dengan data yang berada pada penyimpan yang lambat dan besar.
- Ide dasar dari hirarki memori :
  - Untuk setiap k, devais pada level k yang lebih cepat dan kecil merupakan cache dari devais yang lebih lambat dan besar pada level k+1
- Mengapa hirarki memori digunakan ?
  - Program cenderung untuk mengakses data pada level k lebih sering dari data pada level k+1
  - Penyimpan pada level k+1 dapat lebih lambat, besar dan harga per bit lebih rendah

Cache Pada Hirarki Memori



Konsep Umum Cache



- Program memerlukan obyek d yang disimpan dalam suatu blok b
- Cache hit
  - Program menemukan b dalam cache level k. Misalnya pada blok 14.
- Cache miss
  - B tidak terdapat pada level k, sehingga cache level k harus mengambilknya dari level k+1. Misalnya blok 12.
  - Jika cache level k penuh, maka suatu blok harus diganti isinya. Blok mana yang menjadi “korban” ?
    - Placement policy : dimana blok baru diletakkan. Misalnya b mod 4
    - Replacement policy : blok mana yang harus terusir ? Misalnya LRU

Konsep Umum Cache

- Jenis cache miss :
  - Cold (compulsary) miss
    - Cold miss terjadi jika cache kosong.
  - Conflict miss
    - Cache pada level k+1 membagi blok menjadi subset berukuran kecil yang dapat diletakkan pada level k.
    - Misalnya, blok i pada level k+1 harus diletakkan pada blok (i mod 4) pada level k+1.
    - Conflict miss terjadi jika cache level k berukuran cukup besar, tetapi suatu kumpulan obyek data dipetakan pada blok yang sama di level k.
    - Misalnya jika mereferensikan blok 0, 8, 0, 8, 0, 8, ... akan terjadi miss secara terus menerus.
  - Capacity miss
    - Terjadi ketika blok data yang aktif (working set) lebih besar dari kapasitas cache

Cache Dalam Hirarki Memori

Jenis Cache	Obyek Cache	Lokasi Cache	Delay	Pengelola
Registers	4-byte word	CPU registers	0	Compiler
TLB	Address translations	On-Chip TLB	0	Hardware
L1 cache	32-byte block	On-Chip L1	1	Hardware
L2 cache	32-byte block	Off-Chip L2	10	Hardware
Virtual Memory	4-KB page	Main memory	100	Hardware+ OS
Buffer cache	Parts of files	Main memory	100	OS
Network buffer cache	Parts of files	Local disk	10,000,000	AFS/NFS client
Browser cache	Web pages	Local disk	10,000,000	Web browser
Web cache	Web pages	Remote server disks	1,000,000,000	Web proxy server