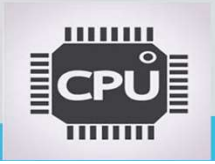
Arsitektur dan Organisasi Komputer
COM 60011

Bab #4 –Cache Memory

Bab 4
Memori Cache



Tabel 4.1
Karakteristik Sistem Memori Komputer

Lokasi	Performa
Internal (misalnya register prosesor, cache, memori utama)	Waktu akses
Eksternal (misalnya disk optik, disk magnetis, kaset)	Waktu siklus
Kapasitas	Kecepatan transfer
Jumlah kata	Tipe Fisik
Jumlah byte	Semikonduktor
Unit Transfer	Magnetik
Kata	Optik
Blok	Magneto-optical
Metode Akses	Karakter fisik
Sekuensial	Volatile / nonvolatile
Langsung	Dapat dihapus / tidak dapat dihapus
Acak	Organisasi
Asosiatif	Modul memori

Karakteristik Sistem Memori

Lokasi

- Mengacu pada apakah memori internal dan eksternal ke komputer
- Memori internal sering kali disamakan dengan memori utama
- Prosesor membutuhkan memori lokalnya sendiri, dalam bentuk register
- Cache adalah bentuk lain dari memori internal
- Memori eksternal terdiri dari perangkat penyimpanan periferal yang dapat diakses oleh prosesor melalui pengontrol I / O

Kapasitas

- Memori biasanya dinyatakan dalam byte

Unit transfer

- Untuk memori internal, unit transfer sama dengan jumlah saluran listrik yang masuk dan keluar dari modul memori

Metode akses Data

Sequential access

- Memori disusun ke dalam satuan data yang disebut rekaman
- Akses harus dilakukan dalam urutan linear tertentu
- Waktu akses bervariasi

Direct access

- Melibatkan mekanisme baca-tulis bersama
- Blok atau catatan individual memiliki alamat unik berdasarkan lokasi fisik
- Waktu akses bervariasi

Random access

- Setiap lokasi yang dapat diatasi dalam memori memiliki mekanisme mengatasi yang unik dan berkabel secara fisik
- Waktu untuk mengakses lokasi tertentu tidak terbatas pada urutan akses sebelumnya dan berurutan
- Lokasi apa pun dapat dipilih secara acak dan langsung ditangani dan diakses
- Memori utama dan beberapa sistem cache adalah akses acak

Associative

- Sebuah kata diambil berdasarkan sebagian isinya daripada alamatnya
- Setiap lokasi memiliki mekanisme mengatasi sendiri dan waktu pengambilan konstan independen dari lokasi atau pola akses sebelumnya
- Memori cache dapat menggunakan akses asosiatif

Kapasitas dan Kinerja:

Dua karakteristik memori yang paling penting

Tiga parameter kinerja digunakan:

Waktu akses (latensi)

- Untuk memori akses acak, ini adalah waktu yang diperlukan untuk melakukan operasi baca atau tulis
- Untuk memori akses non-acak, inilah waktu yang diperlukan untuk memposisikan mekanisme baca-tulis di lokasi yang diinginkan

Memory cycle time

- Waktu akses ditambah waktu tambahan yang diperlukan sebelum akses kedua dapat dimulai
- Waktu tambahan mungkin diperlukan bagi sementara untuk mati pada baris sinyal atau untuk meregenerasi data jika mereka dibaca secara destruktif
- Prihatin dengan bus sistem, bukan prosesor

Transfer rate

- Tingkat di mana data dapat ditransfer ke atau keluar dari unit memori
- Untuk memori akses acak sama dengan 1/(waktu siklus)

Penyimpanan

Bentuk yang paling umum adalah:

- Memori semikonduktor
- Memori permukaan magnetik
- Optik
- Magneto-optical

Beberapa karakteristik fisik penyimpanan data penting:

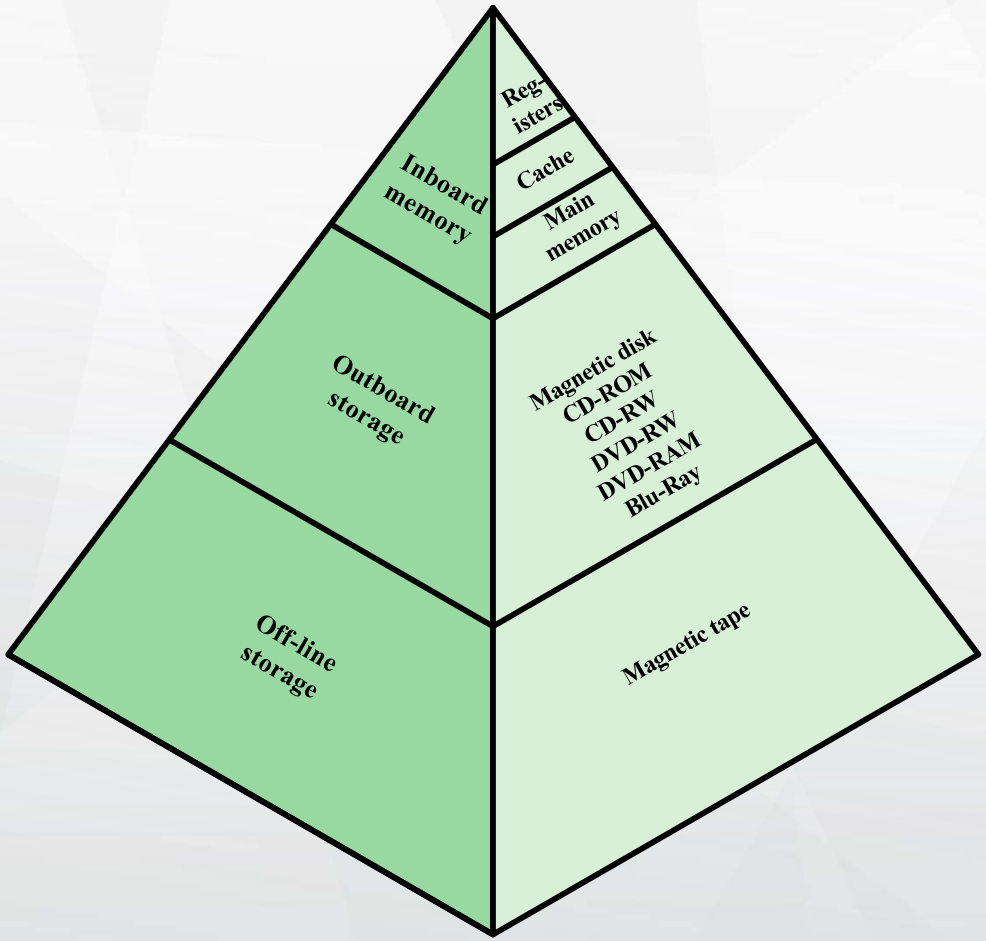
- Memori yang mudah menguap
  - Informasi membusuk secara alami atau hilang ketika daya listrik dimatikan
- Memori nonvolatile
  - Setelah direkam, informasi tetap tanpa kerusakan sampai sengaja diubah
  - Tidak ada daya listrik yang diperlukan untuk menyimpan informasi
- Kenangan permukaan magnetik
  - Tidak mudah menguap
- Memori semikonduktor
  - Mungkin volatile atau nonvolatile
- Memori nonerasable
  - Tidak dapat diubah, kecuali dengan menghancurkan unit penyimpanan
  - Memori semikonduktor jenis ini dikenal sebagai read-only memory (ROM)

Untuk memori akses-acak, organisasi merupakan masalah desain utama

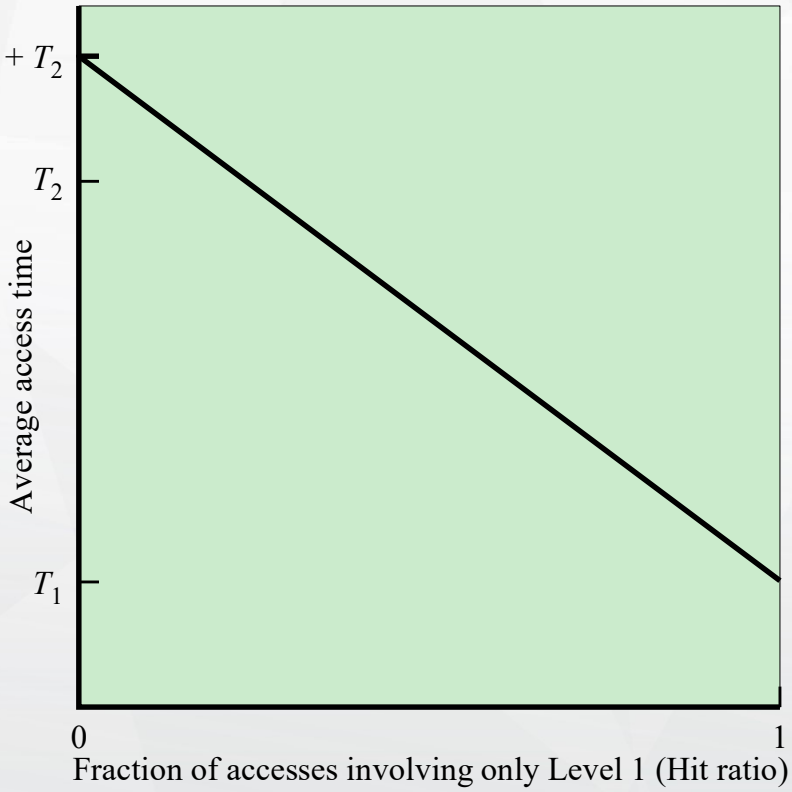
- Organisasi mengacu pada pengaturan fisik bit untuk membentuk kata-kata

Hirarki Memori

- **Batasan desain pada memori komputer dapat diringkas dengan tiga pertanyaan:**
  - Berapa, seberapa cepat, seberapa mahal
- **Ada trade-off antara kapasitas, waktu akses, dan biaya**
  - Waktu akses lebih cepat, biaya per bit lebih besar
  - Kapasitas lebih besar, biaya per bit lebih kecil
  - Kapasitas lebih besar, waktu akses lebih lambat
- **Jalan keluar dari dilema memori tidak bergantung pada satu komponen memori atau teknologi, tetapi menggunakan hierarki memori**



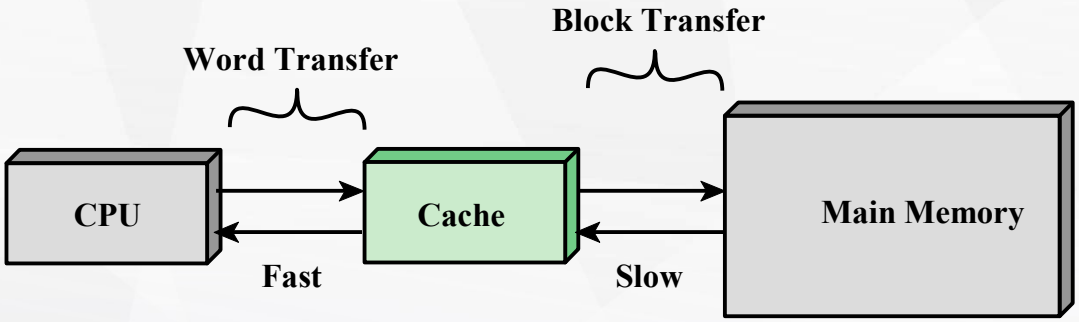
Gambar 4.1 The Memory Hierarchy



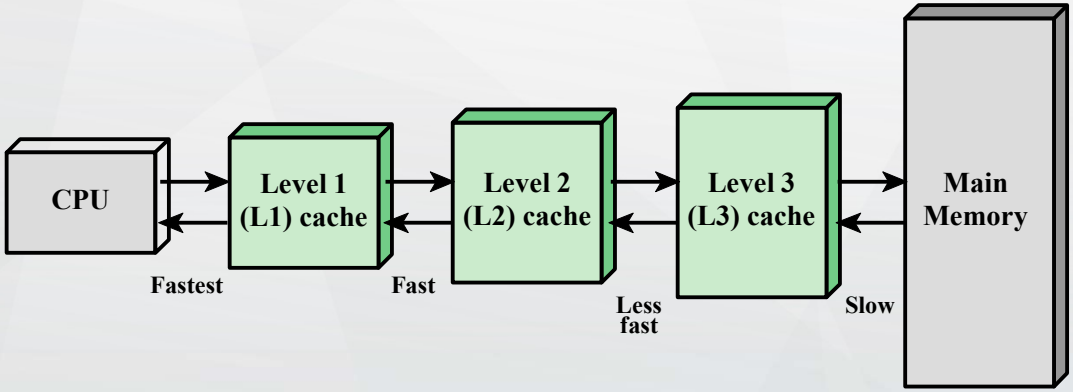
Gambar 4.2 Performa Memori Dua Tingkat Sederhana

Penyimpanan

- **Penggunaan tiga level mengeksploitasi fakta bahwa memori semikonduktor hadir dalam berbagai tipe yang berbeda dalam kecepatan dan biaya**
- **Data disimpan lebih permanen di perangkat penyimpanan massal eksternal**
- **Memori eksternal, nonvolatile juga disebut sebagai sekunder Penyimpanan atau bantu Penyimpanan**
- **Cache disk**
  - Sebagian dari memori utama dapat digunakan sebagai buffer untuk menyimpan data sementara yang akan dibacakan ke disk
  - Beberapa transfer data besar dapat digunakan sebagai pengganti banyak transfer data kecil
  - Data dapat diambil dengan cepat dari cache perangkat lunak daripada perlahan dari disk

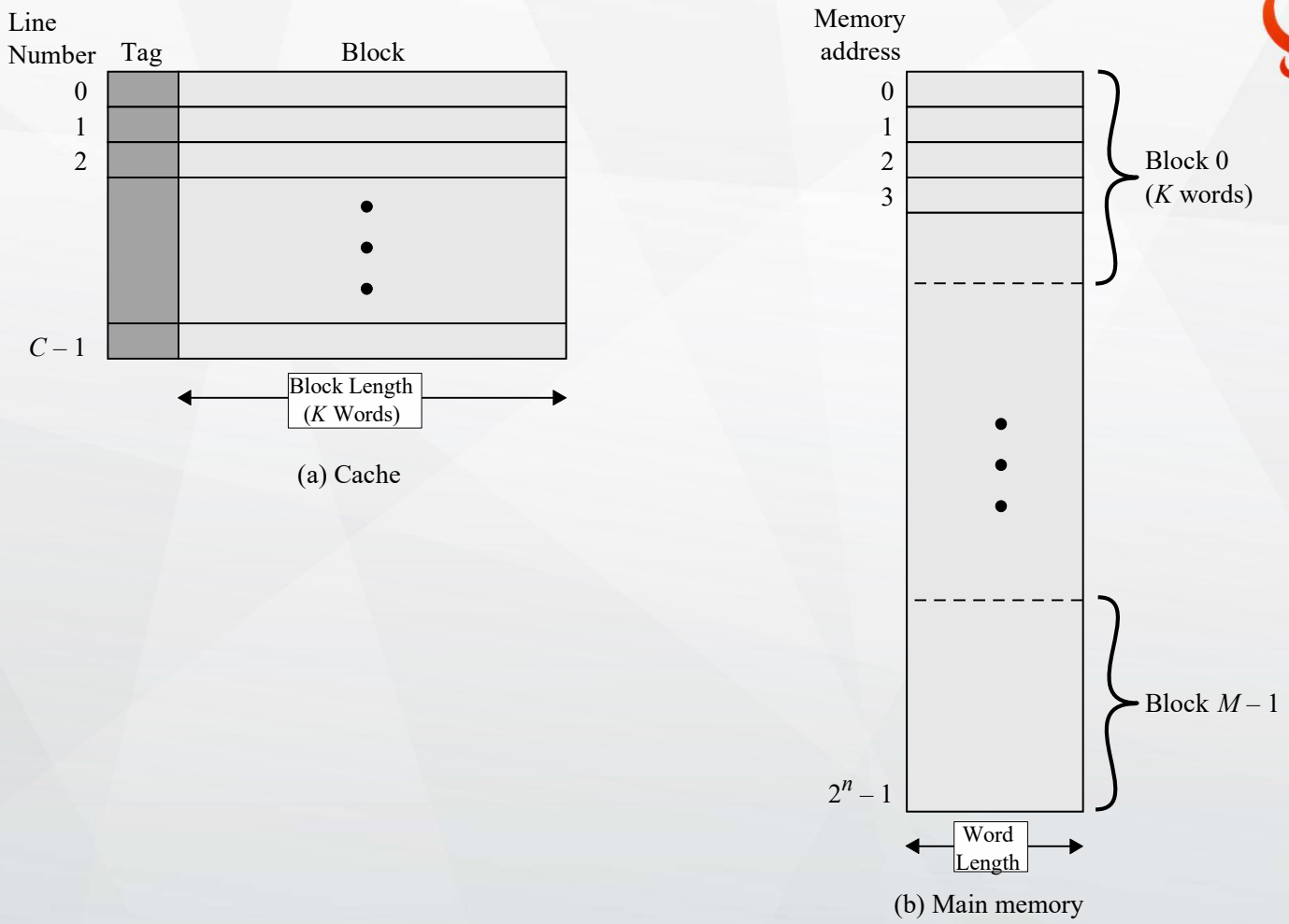


(a) Single cache

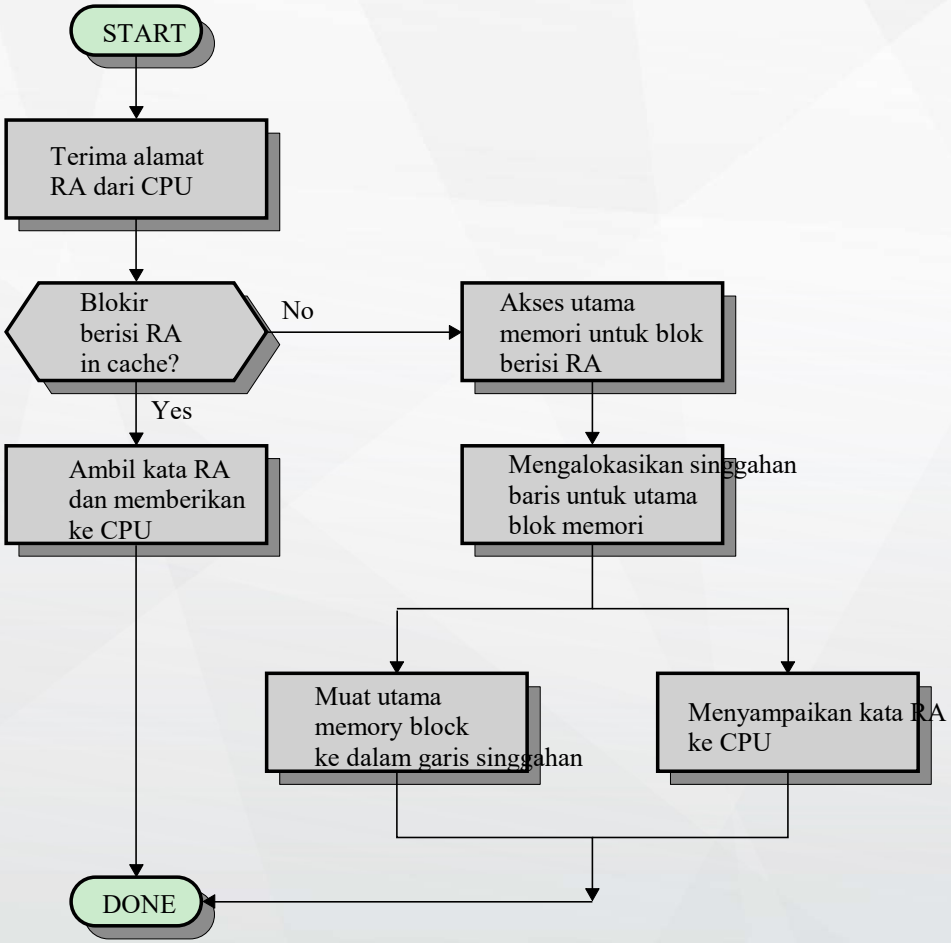


(b) Tiga level cache organization

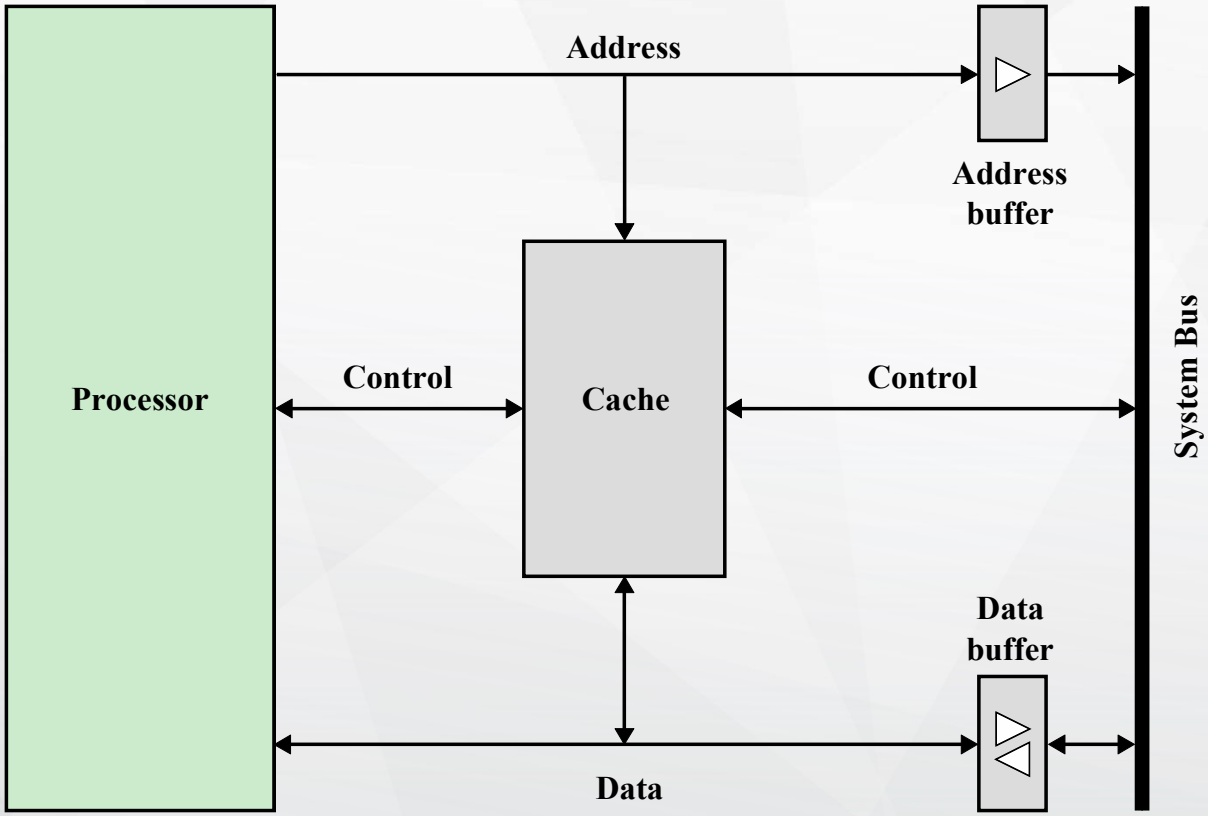
Gambar 4.3 Cache and Main Memory



Gambar 4.4 Cache/Main-Memory Structure



Gambar 4.5 Cache Read Operation



Gambar 4.6 Diagram Organisasi Cache

Cache Addresses	**Write Policy**
Logical	Write through
Physical	Write back
Cache Size	**Line Size**
Mapping Function	**Number of caches**
Direct	Single or two level
Associative	Unified or split
Set Associative	
Replacement Algorithm	
Least recently used (LRU)	
First in first out (FIFO)	
Least frequently used (LFU)	
Random	

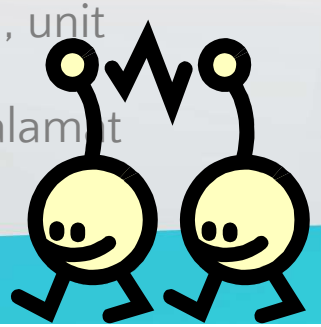
Tabel 4.2
Elemen Desain Cache

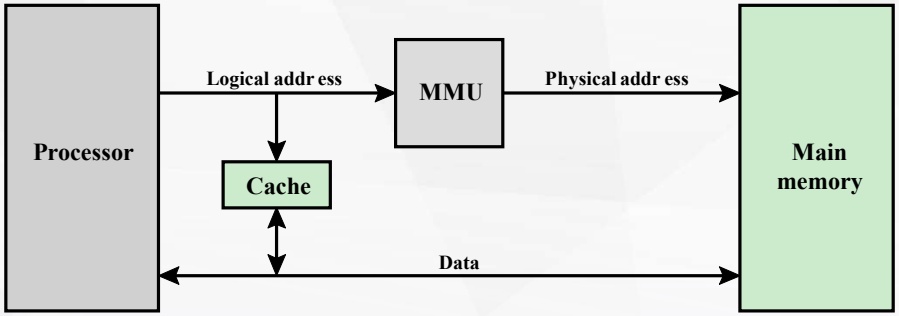
Alamat Cache

Memori Virtual

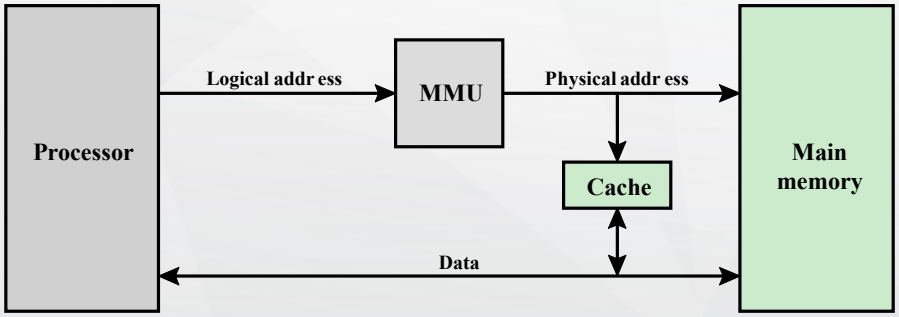
Memori virtual

- Fasilitas yang memungkinkan program untuk mengalamatkan memori dari sudut pandang logis, tanpa memperhatikan jumlah memori utama yang tersedia secara fisik
- Saat digunakan, bidang alamat instruksi mesin berisi alamat virtual
- Untuk membaca dan menulis dari memori utama, unit manajemen memori perangkat keras (MMU) menerjemahkan setiap alamat virtual menjadi alamat fisik di memori utama





(a) Logical Cache



(b) Physical Cache

Gambar 4.7 Logical and Physical Caches

Processor	Type	Tahun Pembuatan	L1 Cachea	L2 cache	L3 Cache
IBM 360/85	Mainframe	1968	16 to 32 kB	—	—
PDP-11/70	Minicomputer	1975	1 kB	—	—
VAX 11/780	Minicomputer	1978	16 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128 to 256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256 to 512 KB	—
PowerPC 601	PC	1993	32 kB	—	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
PowerPC G4	PC/server	1999	32 kB/32 kB	256 KB to 1 MB	2 MB
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/server	2000	8 kB/8 kB	256 KB	—
IBM SP	High-end server/ supercomputer	2000	64 kB/32 kB	8 MB	—
CRAY MTAb	Supercomputer	2000	8 kB	2 MB	—
Itanium	PC/server	2001	16 kB/16 kB	96 KB	4 MB
Itanium 2	PC/server	2002	32 kB	256 KB	6 MB
IBM POWER5	High-end server	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 kB/64 kB	1MB	—
IBM POWER6	PC/server	2007	64 kB/64 kB	4 MB	32 MB
IBM z10	Mainframe	2008	64 kB/128 kB	3 MB	24-48 MB
Intel Core i7 EE 990	Workstaton/ server	2011	6 ´ 32 kB/32 kB	1.5 MB	12 MB
IBM zEnterprise 196	Mainframe/ Server	2011	24 ´ 64 kB/ 128 kB	24 ´ 1.5 MB	24 MB L3 192 MB L4

Tabel 4.3

Ukuran Cache dari Beberapa Prosesor

[Sebuah] Dua nilai yang dipisahkan oleh garis miring mengacu pada instruksi dan cache data.

[b]Kedua cache hanya untuk instruksi; tidak ada cache data.

(Tabel dapat ditemukan di halaman 134 di buku teks.)

Fungsi Pemetaan

Karena ada lebih sedikit baris cache daripada blok memori utama, algoritme diperlukan untuk memetakan blok memori utama ke dalam baris cache

Tiga teknik dapat digunakan:

Direct

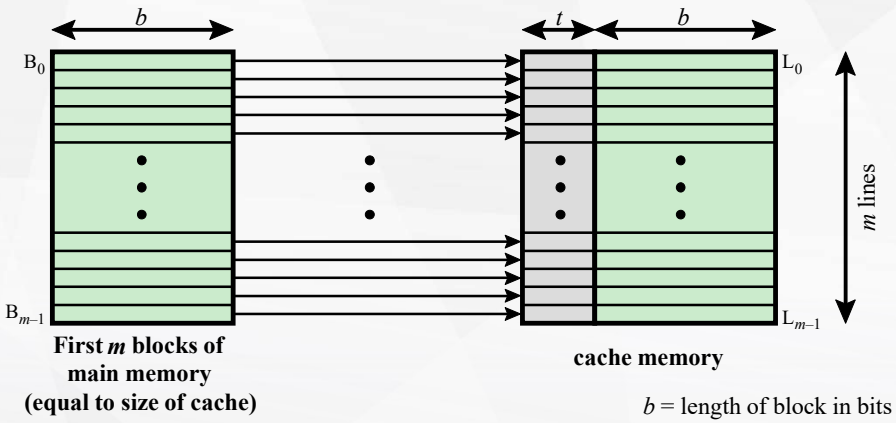
- Teknik paling sederhana
- Memetakan setiap blok memori utama menjadi hanya satu baris cache yang mungkin

Associative

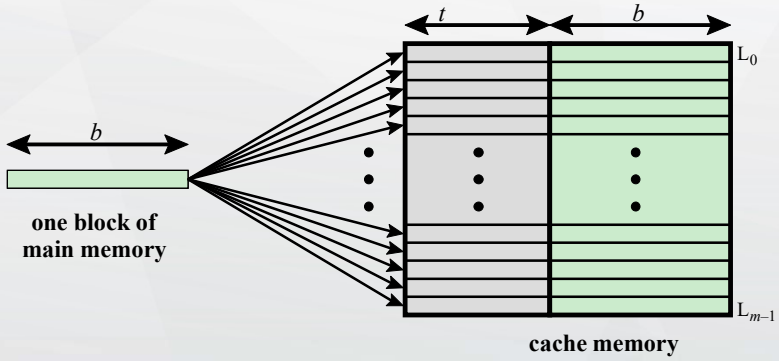
- Mengizinkan setiap blok memori utama dimuat ke dalam baris cache apa pun
- Logika kontrol singgahan menginterpretasikan alamat memori hanya sebagai tag dan bidang Word
- Untuk menentukan apakah blok berada dalam cache, logika kontrol cache harus secara bersamaan memeriksa Tag setiap baris untuk kecocokan

Set Associative

- Kompromi yang menunjukkan kekuatan pendekatan langsung dan asosiatif sambil mengurangi kerugian mereka

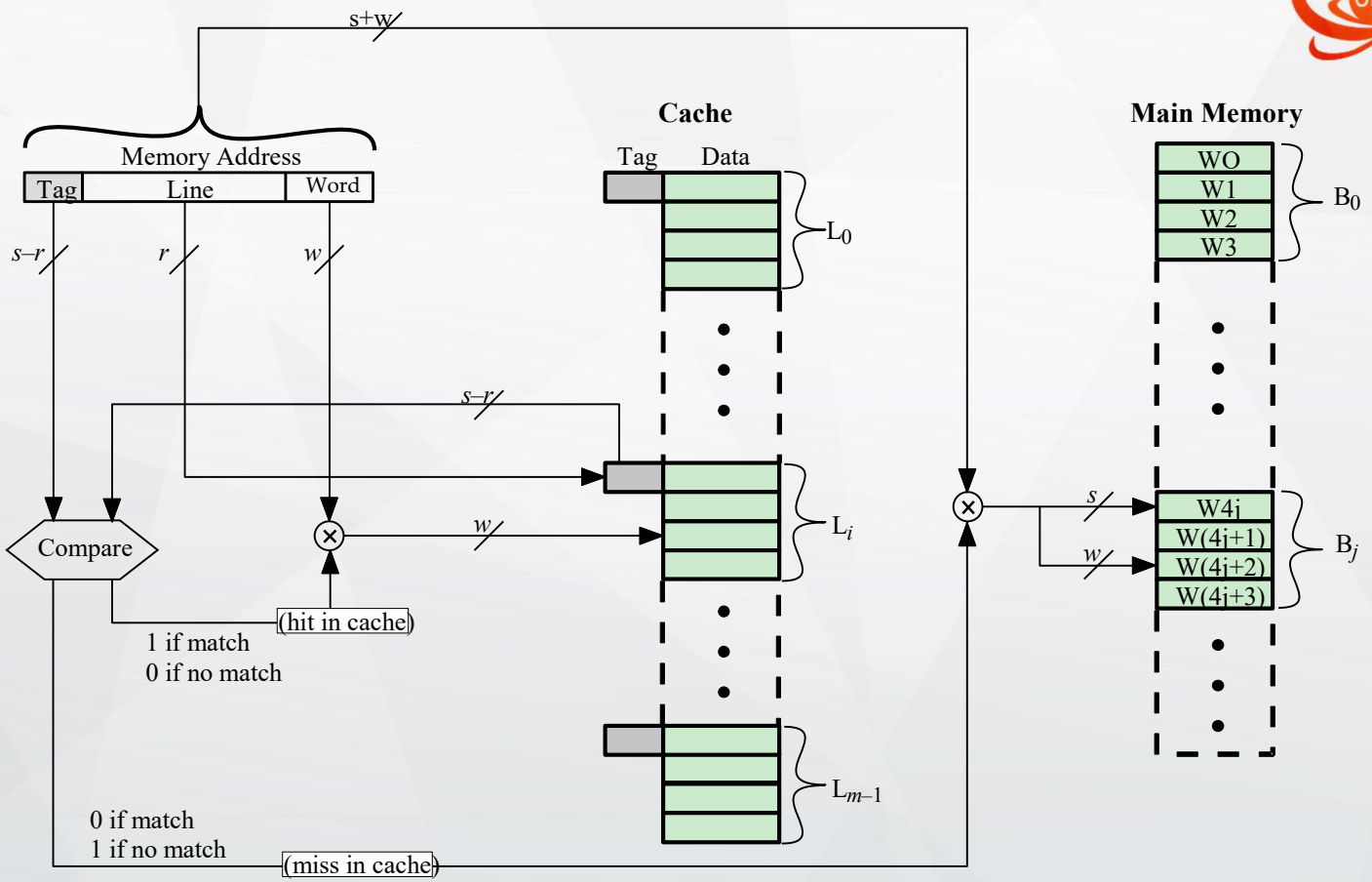


(a) Direct mapping

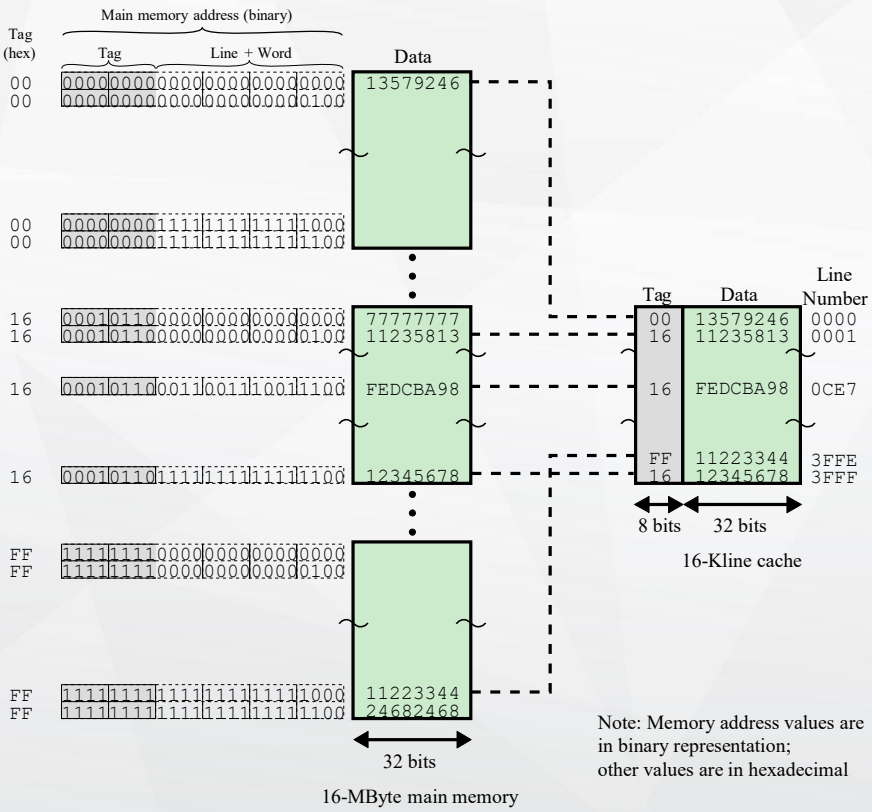


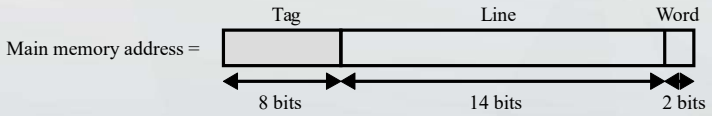
(b) Associativemapping

Gambar 4.8 Pemetaan Dari Memori Utama ke Cache: Langsung dan Asosiatif



Gambar 4.9 Direct-Mapping Organisasi Cache





Gambar 4.10 Direct Mapping Example

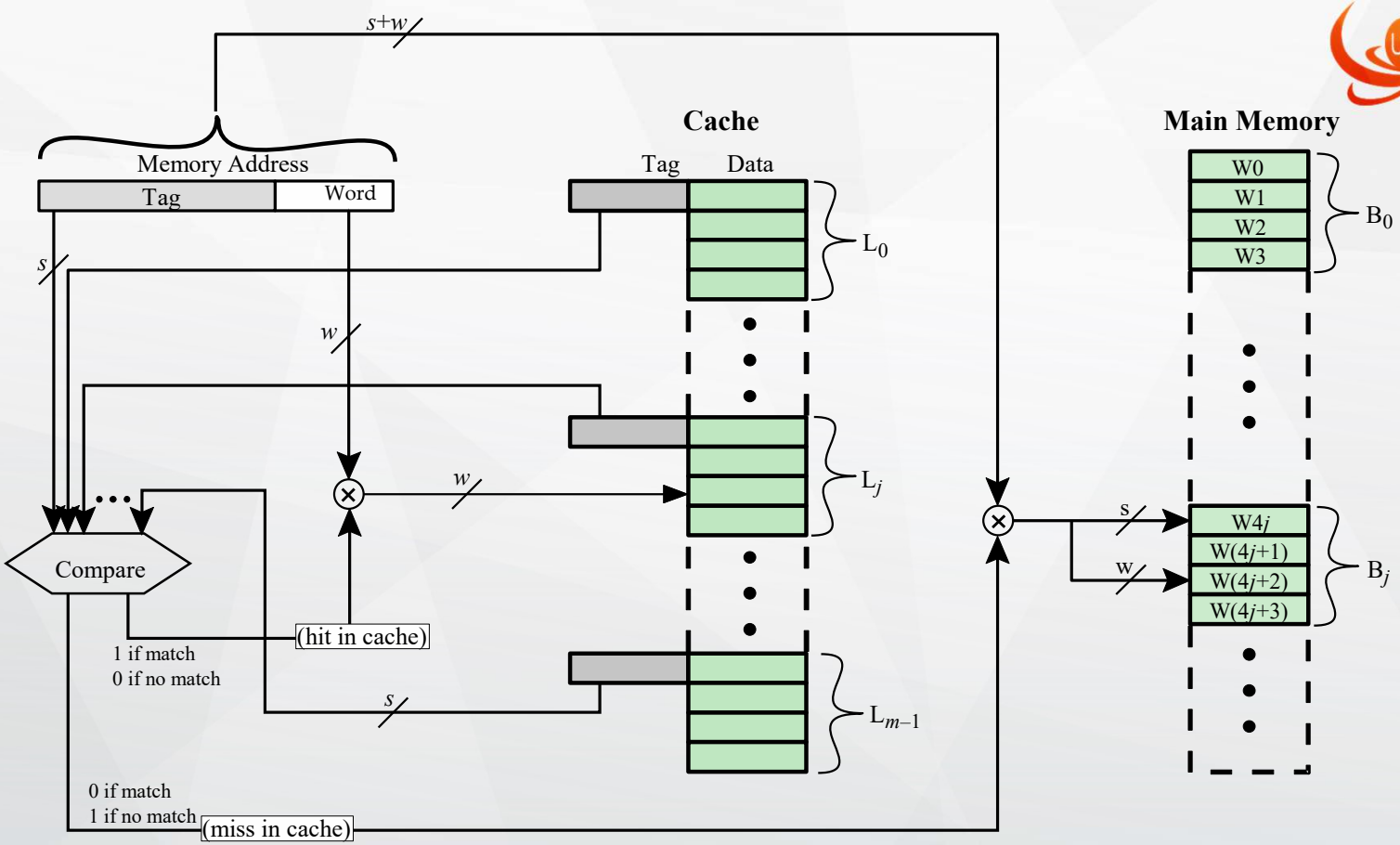
Ringkasan Pemetaan Langsung

- Panjang alamat = (s + w) bits
- Jumlah unit yang dapat dialamatkan = 2^{s+w} kata atau byte
- Blok size = ukuran garis = 2^{w} kata atau byte
- Jumlah blok di memori utama = $2^{s+w} / 2^{w} = 2^{s}$
- Jumlah baris di cache = m = 2^{r}
- Ukuran tag = (s – r) bit

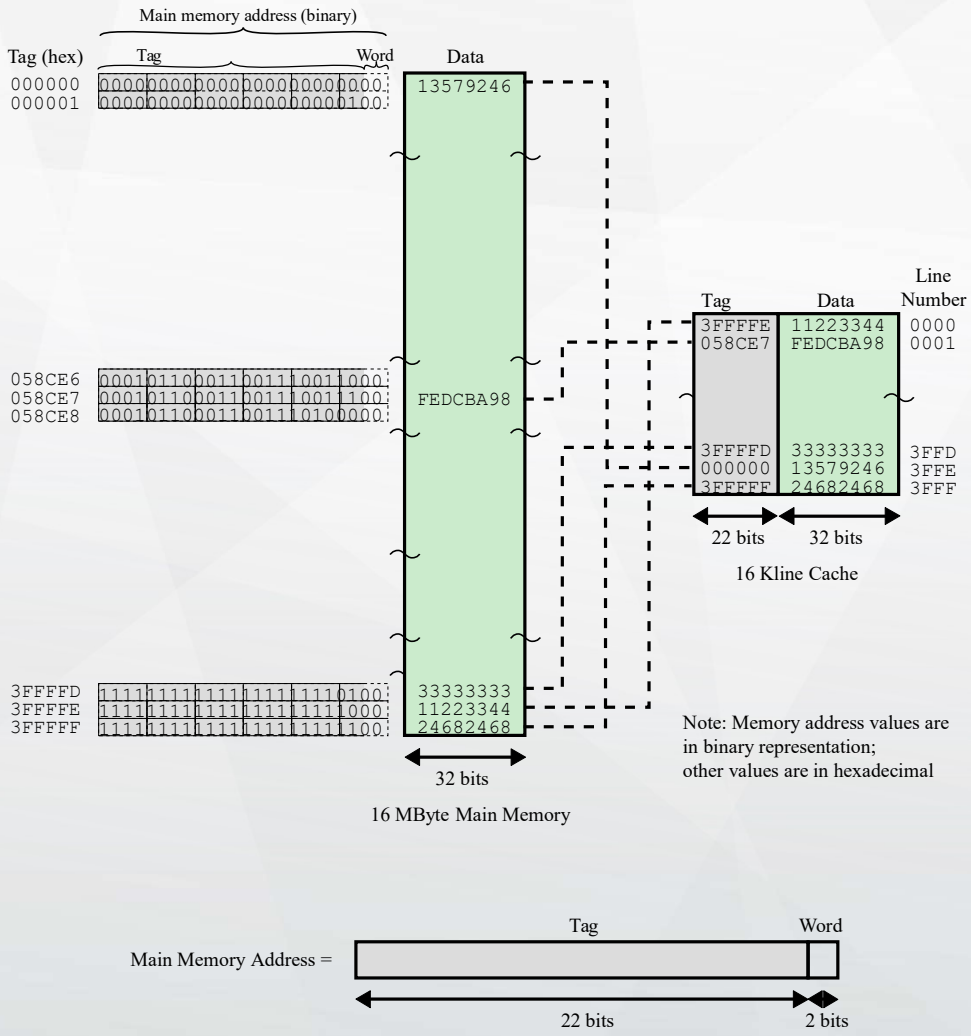
Cache Victim

Awalnya diusulkan sebagai pendekatan untuk mengurangi konflik yang hilang dari cache yang dipetakan langsung tanpa mempengaruhi waktu aksesnya yang cepat

- **Cache asosiatif penuh**
- **Ukuran tipikal adalah 4 hingga 16 baris cache**
- **Berada diantara cache L1 yang dipetakan langsung dan tingkat memori berikutnya**



Gambar 4.11 Organisasi Cache Fully Associative



Gambare 4.12 Associative Mapping Example

Ringkasan Pemetaan Asosiatif

Panjang alamat = (s + w) bits

Jumlah unit yang dapat dialamatkan = 2^{s+w} kata atau byte

Ukuran blok = ukuran garis = 2^{w} kata atau byte

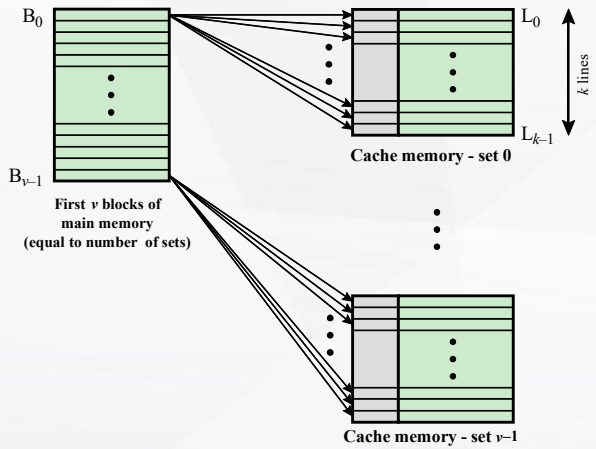
Jumlah blok di memori utama = $2^{s+w}/2^{w} = 2^{s}$

Jumlah baris dalam cache = tidak ditentukan

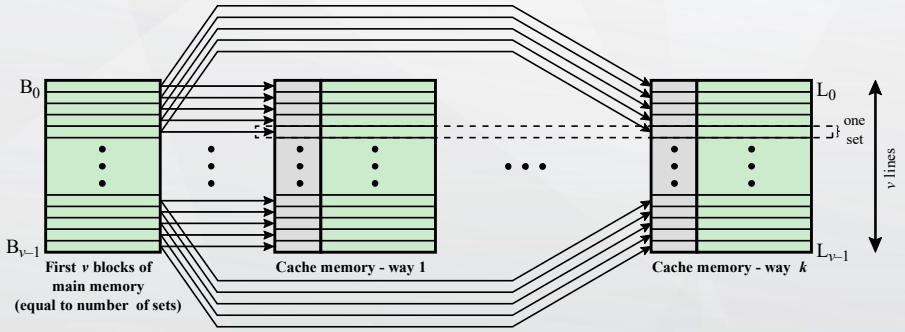
Ukuran tag = s bit

Tetapkan Pemetaan Asosiatif

- **Kompromi yang menunjukkan kekuatan pendekatan langsung dan asosiatif sekaligus mengurangi kerugiannya**
- **Cache terdiri dari a jumlah set**
- **Setiap set berisi sejumlah baris**
- **Sebuah peta blok yang diberikan untuk setiap baris di suatu set**
- **misalnya 2 baris per set**
  - 2 cara pemetaan asosiatif
  - Blok tertentu dapat berada di salah satu dari 2 baris hanya dalam satu set

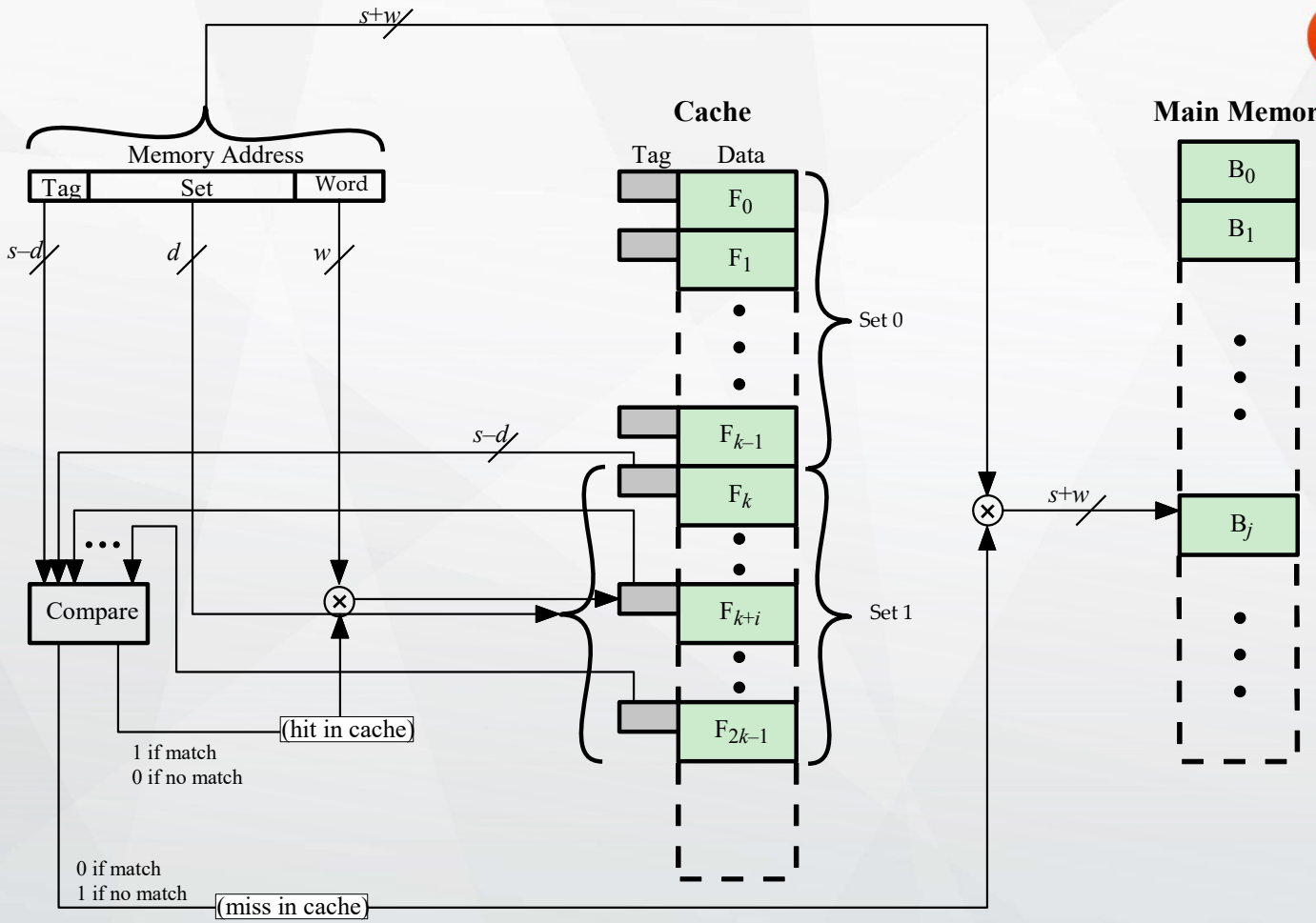


(a) v associative-mapped caches



(b) k direct-mapped caches

Gambare 4.13 Mapping From Main Memory to Cache: k-way Set Associative



Gambar 4.14 *k*-Way Set Associative Cache Organization

Tetapkan Ringkasan Pemetaan Asosiatif

Panjang alamat = (s + w) bits

Jumlah unit yang dapat dialamatkan = 2^{s+w} kata atau byte

Ukuran blok = ukuran garis = 2^w kata atau byte

Jumlah blok di memori utama = $2^{s+w}/2^w = 2^s$

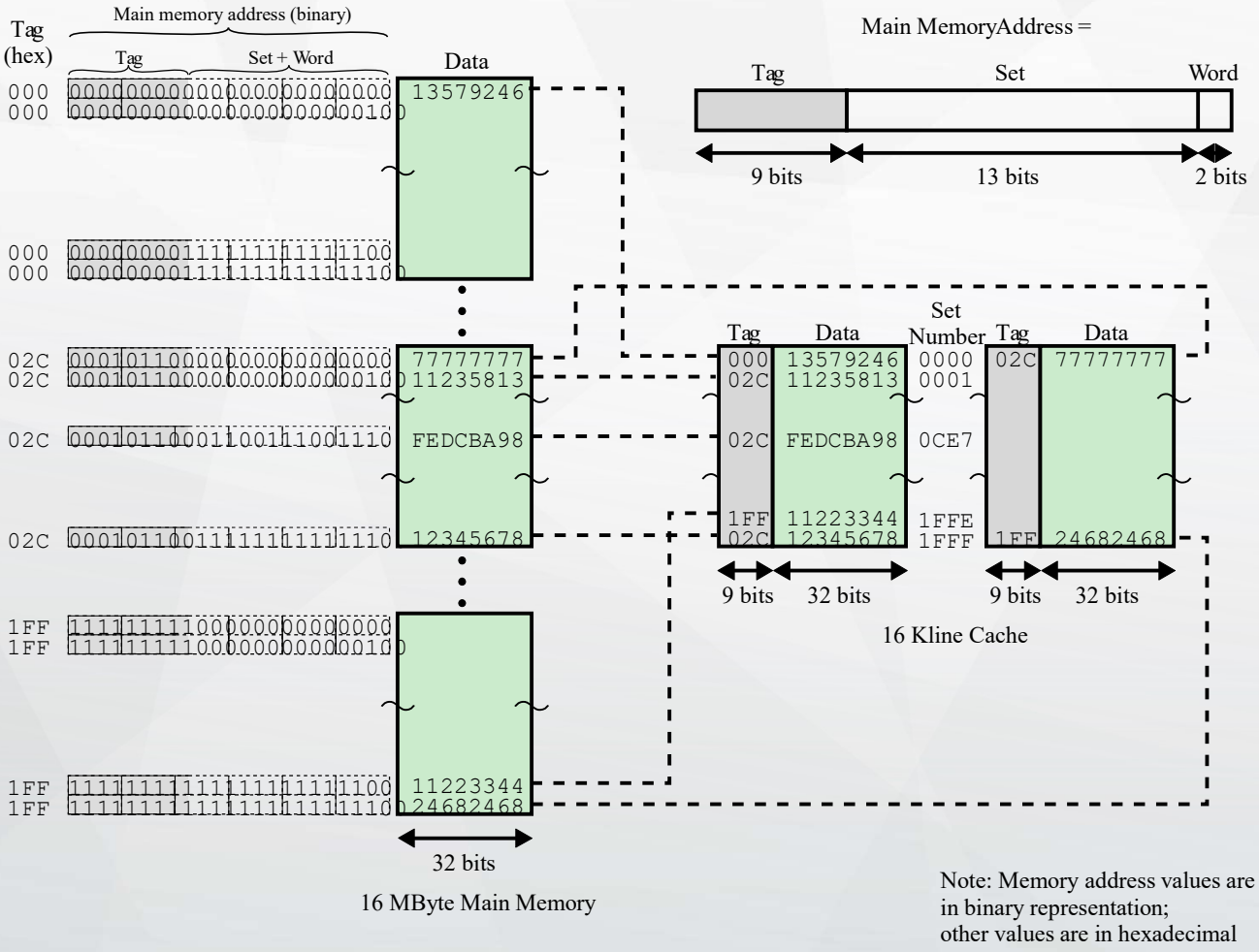
Jumlah baris dalam himpunan = k

Jumlah set = v = 2^d

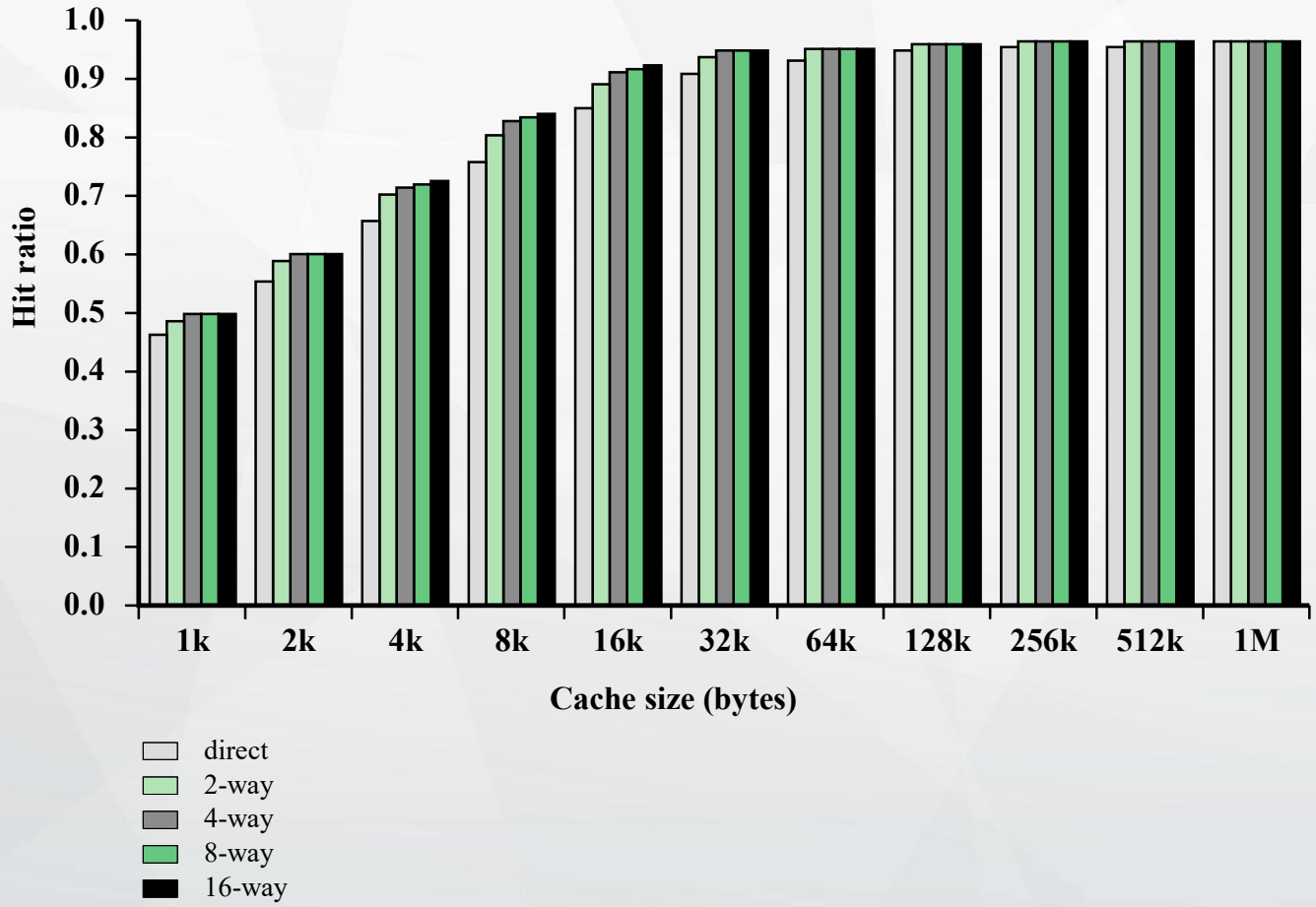
Jumlah baris dalam cache = m = kv = k * 2^d

Ukuran cache = $k * 2^{d+w}$ kata atau byte

Ukuran tag = (s – d) bit

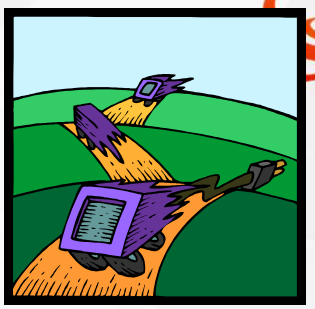


Gambare 4.15 Two-Way Set Associative Mapping Example



Gambar 4.16 Varying Associativity over Cache Size

Penggantian Algoritma



- Setelah cache diisi, ketika blok baru dibawa ke cache, salah satu blok yang ada harus diganti
- Untuk pemetaan langsung, hanya ada satu jalur yang memungkinkan untuk blok tertentu dan tidak ada pilihan
- Untuk teknik asosiatif dan set-asosiatif diperlukan algoritma pengganti
- Untuk mencapai kecepatan tinggi, algoritma harus diimplementasikan dalam perangkat keras

Algoritme pengganti yang paling umum adalah:

Paling terakhir digunakan (LRU)

- Paling efektif
- Ganti blok itu di set yang telah berada di cache paling lama tanpa referensi ke sana
- Karena kesederhanaan implementasinya, LRU adalah algoritma pengganti yang paling populer

First-in-first-out (FIFO)

- Ganti blok itu di set yang paling lama berada di cache
- Mudah diterapkan sebagai teknik penyangga round-robin atau melingkar

Paling jarang digunakan (LFU)

- Ganti blok itu di himpunan yang mengalami referensi paling sedikit
- Bisa diterapkan dengan mengaitkan penghitung dengan setiap baris

Kebijakan Penyimpanan

Ketika blok yang tinggal di cache akan diganti ada dua kasus yang perlu dipertimbangkan:

- Jika blok lama dalam cache belum diubah maka blok lama mungkin ditimpa dengan blok baru tanpa terlebih dahulu menuliskan blok lama
- Jika setidaknya satu operasi tulis telah dilakukan pada kata di baris cache itu maka memori utama harus diperbarui dengan menulis garis cache ke blok memori sebelum membawa blok baru

Ada dua masalah yang harus dihadapi:

- Lebih dari satu perangkat mungkin memiliki akses ke memori utama
- Masalah yang lebih kompleks terjadi ketika beberapa prosesor dilampirkan ke bus yang sama dan setiap prosesor memiliki cache lokal sendiri - jika kata diubah dalam satu cache itu bisa dibayangkan membatalkan kata di cache lain

Menulis Melalui dan Tulis Kembali

Menulis melalui

- Teknik paling sederhana
- Semua operasi tulis dilakukan ke memori utama serta ke cache
- Kerugian utama dari teknik ini adalah menghasilkan lalu lintas memori yang substansial dan dapat membuat kemacetan

Menulis kembali

- Meminimalkan penulisan memori
- Pembaruan hanya dilakukan di cache
- Bagian dari memori utama tidak valid dan karenanya akses oleh modul I / O hanya dapat diizinkan melalui cache
- Hal ini membuat sirkuit menjadi kompleks dan berpotensi menjadi hambatan

Garis Ukuran

Ketika blok data diambil dan ditempatkan di cache tidak hanya kata yang diinginkan tetapi juga beberapa kata yang berdekatan diambil

Karena ukuran blok meningkatkan rasio hit pada awalnya akan meningkat karena prinsip lokalitas

Saat ukuran blok meningkatkan data yang lebih berguna dibawa ke dalam cache

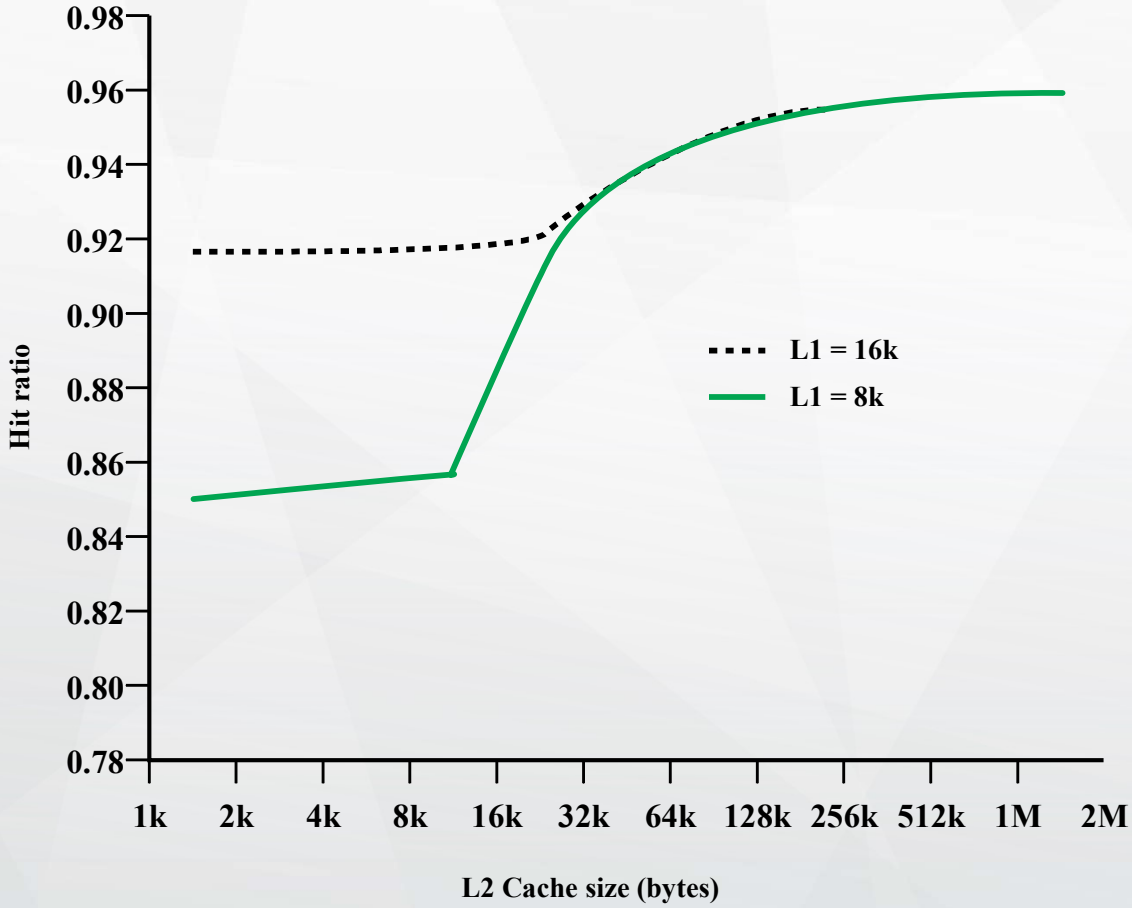
Rasio pukulan akan mulai berkurang ketika blok menjadi lebih besar dan kemungkinan menggunakan informasi yang baru diambil menjadi kurang dari kemungkinan penggunaan kembali informasi yang harus diganti

Dua efek spesifik mulai dimainkan:

- Blok yang lebih besar mengurangi jumlah blok yang masuk ke dalam singgahan
- Ketika blok menjadi lebih besar setiap kata tambahan lebih jauh dari kata yang diminta

Cache Multilevel

- **Karena kerapatan logika telah meningkat, maka menjadi mungkin untuk memiliki cache pada chip yang sama dengan prosesor**
- **Cache on-chip mengurangi aktivitas bus eksternal prosesor dan mempercepat waktu eksekusi serta meningkatkan kinerja sistem secara keseluruhan**
  - Ketika instruksi atau data yang diminta ditemukan di cache on-chip, akses bus dihilangkan
  - Akses cache pada chip akan selesai jauh lebih cepat daripada siklus bus status tanpa menunggu
  - Selama periode ini bus gratis untuk mendukung transfer lainnya
- **Cache dua tingkat:**
  - Cache internal ditetapkan sebagai level 1 (L1)
  - Cache eksternal ditetapkan sebagai level 2 (L2)
- **Potensi penghematan karena penggunaan cache L2 bergantung pada hit rate di cache L1 dan L2**
- **Penggunaan cache multilevel memperumit semua masalah desain yang terkait dengan cache, termasuk ukuran, algoritme penggantian, dan kebijakan penulisan**



Gambar 4.17 Total Hit Ratio (L1 dan L2) for 8 Kbyte and 16 Kbyte L1

Bersatu Melawan Pisahkan Cache

Telah menjadi umum untuk membagi cache:

- Satu didedikasikan untuk instruksi
- Satu didedikasikan untuk data
- Keduanya ada di level yang sama, biasanya sebagai dua cache L1

Keuntungan bersatu cache:

- Rasio hit lebih tinggi
  - Menyeimbangkan beban instruksi dan data mengambil secara otomatis
  - Hanya satu cache perlu dirancang dan diterapkan

Tren mengarah ke cache terpisah di L1 dan cache terpadu untuk level yang lebih tinggi

Keuntungan perpecahan cache:

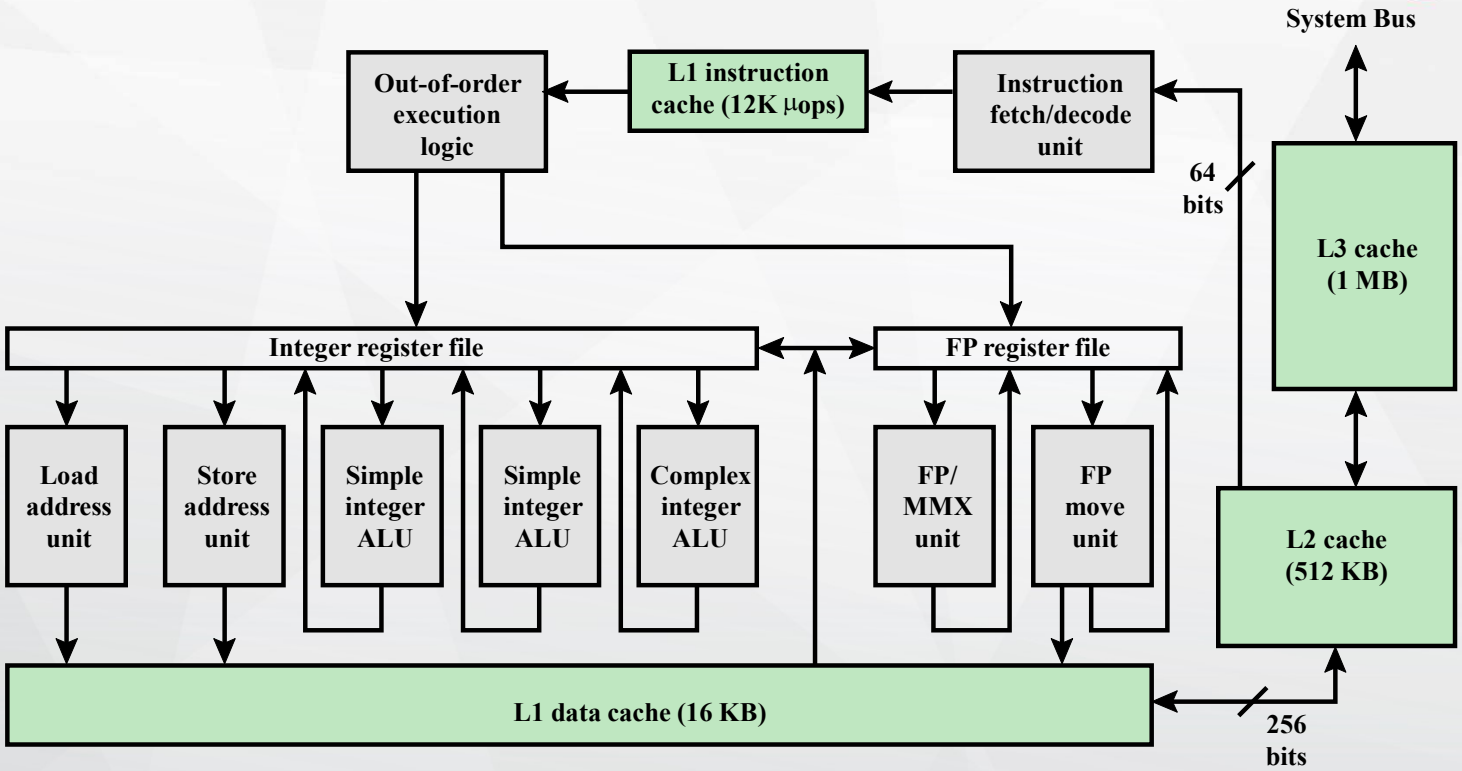
- Menghilangkan pertengkaran cache antara unit pengambilan / dekode instruksi dan unit eksekusi
  - Penting dalam pipelining

Problem	Solution	Processor on which Feature First Appears
External memory slower than the system bus.	Add external cache using faster memory technology.	386
Increased processor speed results in external bus becoming a bottleneck for cache access.	Move external cache on-chip, operating at the same speed as the processor.	486
Internal cache is rather small, due to limited space on chip	Add external L2 cache using faster technology than main memory	486
Contention occurs when both the Instruction Prefetcher and the Execution Unit simultaneously require access to the cache. In that case, the Prefetcher is stalled while the Execution Unit's data access takes place.	Create separate data and instruction caches.	Pentium
Increased processor speed results in external bus becoming a bottleneck for L2 cache access.	Create separate back-side bus that runs at higher speed than the main (front-side) external bus. The BSB is dedicated to the L2 cache.	Pentium Pro
	Move L2 cache on to the processor chip.	Pentium II
Some applications deal with massive databases and must have rapid access to large amounts of data. The on-chip caches are too small.	Add external L3 cache.	Pentium III
	Move L3 cache on-chip.	Pentium 4

Tabel 4.4

Intel Cache Evolusi

(Tabel ada di halaman 150 di buku teks.)



Gambar 4.18 Pentium 4 Block Diagram

Table 4.5 Pentium 4 Cache Operating Modes

Control Bits		Operating Mode		
CD	**NW**	**Cache Fills**	**Write Throughs**	**Invalidates**
0	0	Enabled	Enabled	Enabled
1	0	Disabled	Enabled	Enabled
1	1	Disabled	Disabled	Disabled

Note: CD = 0; NW = 1 is an invalid combination.

Ringkasan

Bab 4

Cache Penyimpanan

Gambaran umum sistem memori komputer

- Karakteristik Sistem Memori
- Hirarki Memori

Prinsip memori cache

Pentium 4 cache organisasi

Elemen desain cache

- Alamat cache
- Ukuran cache
- Fungsi pemetaan
- Algoritme penggantian
- Tulis kebijakan
- Ukuran garis
- Jumlah cache