

Pertemuan 3-4

Teknologi Cache

Direct Mapped Cache

Tujuan Pembelajaran

- Mahasiswa mampu memahami konsep teknologi cache, khususnya direct mapped cache
- Mahasiswa mampu menerapkan metode pembelajaran tutorial dan praktikum untuk memahami direct mapped cache
- Mahasiswa mampu menerapkan konsep Direct Mapped Cache dalam praktikum.

Durasi

2 x 50 menit

Materi

Bagian 1: Tutorial

1. Pengantar Teknologi Cache (xx menit)

- Definisi dan fungsi cache dalam sistem komputer.

Cache dalam Sistem Komputer

Definisi Cache

Cache adalah komponen penyimpanan berkecepatan tinggi yang dapat menyediakan data atau instruksi yang sering diakses oleh prosesor dengan lebih cepat daripada memori utama. Cache biasanya lebih kecil daripada memori utama (RAM) tetapi lebih cepat, sehingga dapat meningkatkan efisiensi dan kinerja komputer.

Fungsi Cache

1. **Meningkatkan Kecepatan Akses:** Cache menyimpan data atau instruksi yang sering diakses oleh prosesor. Ketika prosesor membutuhkan data tersebut, ia akan mencarinya di cache terlebih dahulu sebelum memeriksa memori utama. Hal ini mengurangi waktu tunggu prosesor dan meningkatkan kecepatan eksekusi.
2. **Mengurangi Beban pada Memori Utama:** Dengan menyimpan data yang sering diakses di cache, beban pada memori utama berkurang. Ini mengurangi kemungkinan terjadinya bottleneck pada sistem.
3. **Optimasi Penggunaan Bandwidth:** Cache mengurangi jumlah akses ke memori utama, sehingga mengoptimalkan penggunaan bandwidth antara prosesor dan memori.

Teknologi Cache

1. **Level Cache**: Terdapat beberapa level cache, yaitu:

- **L1 Cache**: Terletak di dalam chip prosesor dan memiliki kapasitas yang kecil tetapi kecepatan akses yang sangat tinggi.
- **L2 Cache**: Biasanya terletak di luar chip prosesor tetapi masih berada di dalam paket yang sama. Kapasitasnya lebih besar daripada L1 tetapi kecepatannya sedikit lebih lambat.
- **L3 Cache**: Lebih besar lagi dan biasanya dibagi antara beberapa inti pada prosesor multi-inti.

2. **Metode Penyimpanan**:

- **Direct Mapped Cache**: Setiap blok memori memiliki lokasi khusus di cache.
- **Full Associative Cache**: Blok memori dapat ditempatkan di mana saja di cache.
- **Set Associative Cache**: Kompromi antara direct mapped dan full associative, di mana cache dibagi menjadi beberapa set dan setiap set memiliki beberapa blok.

3. **Algoritma Penggantian**: Ketika cache penuh dan data baru perlu disimpan, algoritma tertentu digunakan untuk menentukan blok mana yang harus digantikan. Beberapa algoritma populer termasuk FIFO (First In First Out), LRU (Least Recently Used), dan Optimal.

4. **Write Policy**: Menentukan bagaimana operasi penulisan ke memori ditangani. Dua pendekatan umum adalah write-through (data ditulis ke cache dan memori utama) dan write-back (data hanya ditulis ke cache dan ditulis kembali ke memori utama saat blok tersebut digantikan).

Kesimpulan

Cache memainkan peran penting dalam meningkatkan kinerja sistem komputer dengan menyediakan akses cepat ke data yang sering digunakan. Dengan memahami berbagai aspek teknologi cache, kita dapat lebih memahami bagaimana komputer bekerja dengan efisien dan efektif.

- Manfaat penggunaan cache.

Manfaat dan Teknologi Cache dalam Sistem Komputer

Manfaat Penggunaan Cache

Cache, sebagai komponen penyimpanan berkecepatan tinggi, memberikan sejumlah manfaat penting dalam sistem komputer:

1. **Kecepatan Akses yang Lebih Tinggi**: Cache menyediakan data atau instruksi yang sering diakses oleh prosesor dengan kecepatan yang jauh lebih tinggi dibandingkan dengan

memori utama. Hal ini mengurangi waktu tunggu prosesor dan meningkatkan kecepatan eksekusi.

2. **Efisiensi Bandwidth**: Dengan mengurangi jumlah akses ke memori utama melalui penggunaan cache, bandwidth antara prosesor dan memori dapat dioptimalkan, mengurangi kemungkinan terjadinya bottleneck.
3. **Pengurangan Latensi**: Cache mengurangi latensi yang diperlukan untuk mengakses data dari memori utama, memungkinkan prosesor untuk melanjutkan eksekusi dengan lebih cepat.
4. **Hemat Energi**: Mengakses cache memerlukan lebih sedikit energi dibandingkan dengan mengakses memori utama. Dengan demikian, penggunaan cache dapat mengurangi konsumsi energi sistem.
5. **Optimasi Multitasking**: Dalam sistem yang menjalankan banyak tugas secara bersamaan, cache memastikan bahwa tugas-tugas tersebut dapat diakses dan dieksekusi dengan cepat tanpa harus menunggu data dari memori utama.

2. Jenis-jenis Cache (xx menit)

- Cache L1, L2, dan L3.

Jenis-Jenis Cache dan Teknologi dalam Sistem Komputer

Cache adalah komponen penyimpanan berkecepatan tinggi yang bertujuan untuk menyediakan data atau instruksi yang sering diakses oleh prosesor dengan kecepatan yang lebih tinggi dibandingkan dengan memori utama. Dalam arsitektur komputer modern, terdapat beberapa jenis cache yang dikenal dengan L1, L2, dan L3.

1. L1 Cache (Level 1 Cache)

- **Lokasi**: Terintegrasi langsung dalam chip prosesor.
- **Kapasitas**: Relatif kecil, biasanya berkisar antara 16KB hingga 128KB.
- **Kecepatan**: Memiliki kecepatan akses tertinggi di antara jenis cache lainnya.
- **Fungsi**: Menyimpan instruksi dan data yang paling sering digunakan oleh prosesor.
- **Karakteristik**: Karena berada sangat dekat dengan unit pemrosesan, L1 cache memiliki latensi yang sangat rendah.

2. L2 Cache (Level 2 Cache)

- **Lokasi**: Pada awalnya berada di luar chip prosesor, tetapi pada generasi prosesor modern, L2 cache terintegrasi dalam chip yang sama dengan prosesor.
- **Kapasitas**: Lebih besar daripada L1, biasanya berkisar antara 128KB hingga 1MB.
- **Kecepatan**: Lebih lambat dibandingkan L1 tetapi lebih cepat dibandingkan L3.
- **Fungsi**: Menyimpan data yang tidak muat di L1 cache atau data yang belum diakses baru-baru ini.

- **Karakteristik**: Berfungsi sebagai jembatan antara L1 cache yang super cepat dan memori utama yang relatif lambat.

3. L3 Cache (Level 3 Cache)

- **Lokasi**: Terintegrasi dalam chip prosesor, biasanya dibagi antara beberapa inti pada prosesor multi-inti.
- **Kapasitas**: Lebih besar lagi, mulai dari 1MB hingga 8MB atau lebih.
- **Kecepatan**: Lebih lambat dibandingkan L2.
- **Fungsi**: Menyimpan data yang sering diakses oleh semua inti prosesor, memungkinkan data untuk dibagi antar inti dengan efisien.
- **Karakteristik**: Membantu meningkatkan kinerja dalam aplikasi yang memanfaatkan banyak inti prosesor.

- Perbandingan performa dan kapasitas.

Jenis-Jenis Cache: Perbandingan Performa dan Kapasitas dalam Sistem Komputer

Cache adalah komponen penyimpanan berkecepatan tinggi yang bertujuan untuk menyediakan data atau instruksi yang sering diakses oleh prosesor dengan kecepatan yang lebih tinggi dibandingkan dengan memori utama. Dalam arsitektur komputer modern, terdapat beberapa jenis cache yang dikenal dengan L1, L2, dan L3.

1. L1 Cache (Level 1 Cache)

- **Kapasitas**: Relatif kecil, biasanya berkisar antara 16KB hingga 128KB.
- **Performa**: Memiliki kecepatan akses tertinggi di antara jenis cache lainnya karena terintegrasi langsung dalam chip prosesor.
- **Karakteristik**: Karena berada sangat dekat dengan unit pemrosesan, L1 cache memiliki latensi yang sangat rendah. Namun, kapasitasnya yang terbatas membuatnya tidak dapat menyimpan banyak data.

2. L2 Cache (Level 2 Cache)

- **Kapasitas**: Lebih besar daripada L1, biasanya berkisar antara 128KB hingga 1MB.
- **Performa**: Lebih lambat dibandingkan L1 tetapi lebih cepat dibandingkan L3. Meskipun demikian, L2 masih memiliki kecepatan yang cukup tinggi untuk mendukung operasi prosesor.
- **Karakteristik**: Berfungsi sebagai jembatan antara L1 cache yang super cepat dan memori utama yang relatif lambat. Kapasitas yang lebih besar memungkinkan L2 untuk menyimpan lebih banyak data dibandingkan L1.

3. L3 Cache (Level 3 Cache)

- **Kapasitas**: Lebih besar lagi, mulai dari 1MB hingga 8MB atau lebih.
- **Performa**: Lebih lambat dibandingkan L2, tetapi memiliki kapasitas yang jauh lebih besar.
- **Karakteristik**: Membantu meningkatkan kinerja dalam aplikasi yang memanfaatkan banyak inti prosesor. L3 biasanya dibagi antara beberapa inti pada prosesor multi-inti, memungkinkan data untuk dibagi antar inti dengan efisien.

3. Direct Mapped Cache (xx menit)

- Definisi Direct Mapped Cache.

Pengertian Direct Mapped Cache

Direct Mapped Cache adalah salah satu metode penyimpanan dalam cache memory yang mengaitkan setiap blok memori dengan tepat satu blok cache. Dengan kata lain, setiap alamat memori memiliki lokasi khusus di cache tempat data tersebut dapat disimpan. Lokasi ini ditentukan dengan mengambil beberapa bit dari alamat memori dan menggunakan bit tersebut sebagai indeks untuk cache.

Fungsi Direct Mapped Cache

1. **Penyimpanan Data**: Seperti cache pada umumnya, Direct Mapped Cache berfungsi untuk menyimpan data atau instruksi yang sering diakses oleh prosesor. Namun, metodenya unik karena setiap blok memori memiliki lokasi tertentu di cache.
2. **Pemetaan Alamat**: Direct Mapped Cache menggunakan sebagian dari alamat memori untuk menentukan lokasi penyimpanan di cache. Ini memudahkan proses pencarian data saat diperlukan.
3. **Penggantian Data**: Ketika cache penuh dan data baru perlu disimpan, blok cache yang sesuai dengan alamat memori data baru tersebut akan digantikan. Ini berarti bahwa data lama di lokasi tersebut akan digantikan oleh data baru.

Manfaat Direct Mapped Cache

1. **Kecepatan**: Salah satu keuntungan utama dari Direct Mapped Cache adalah kecepatannya. Karena setiap blok memori memiliki lokasi khusus di cache, prosesor dapat dengan cepat menentukan apakah data yang diperlukan berada di cache atau tidak.
2. **Sederhana**: Dibandingkan dengan metode penyimpanan cache lainnya, seperti Fully Associative atau Set Associative, Direct Mapped lebih sederhana dalam desain dan implementasinya. Hal ini membuatnya lebih mudah untuk diimplementasikan pada chip prosesor.
3. **Efisiensi**: Meskipun Direct Mapped Cache mungkin tidak selalu menyediakan tingkat keberhasilan cache (cache hit rate) tertinggi, metodenya yang sederhana dan cepat seringkali menghasilkan kinerja yang cukup baik untuk banyak aplikasi.

4. **Pengurangan Latensi**: Dengan menyimpan data yang sering diakses di cache, latensi yang diperlukan untuk mengakses data tersebut dari memori utama dapat dikurangi, meningkatkan kinerja keseluruhan sistem.

Kesimpulan

Direct Mapped Cache adalah metode penyimpanan cache yang mengaitkan setiap blok memori dengan satu blok cache. Meskipun memiliki kelemahan, seperti kemungkinan tingginya tingkat kegagalan cache (cache miss) jika dua blok memori yang sering diakses dipetakan ke blok cache yang sama, keuntungannya dalam kecepatan dan kesederhanaan membuatnya menjadi pilihan populer dalam desain sistem komputer. Memahami cara kerja Direct Mapped Cache dan manfaatnya dapat membantu dalam memahami kinerja dan efisiensi sistem komputer.

- Cara kerja Direct Mapped Cache.

Cara Kerja Direct Mapped Cache

Pengertian Direct Mapped Cache

Direct Mapped Cache adalah salah satu metode penyimpanan dalam cache memory yang mengaitkan setiap blok memori dengan tepat satu blok cache. Dengan kata lain, setiap alamat memori memiliki lokasi khusus di cache tempat data tersebut dapat disimpan. Lokasi ini ditentukan dengan mengambil beberapa bit dari alamat memori dan menggunakan bit tersebut sebagai indeks untuk cache.

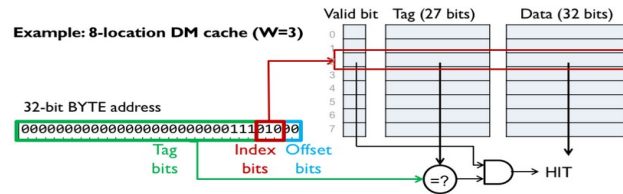
Mekanisme Kerja

1. **Pemetaan Alamat**: Saat prosesor membutuhkan data, alamat memori dari data tersebut dipecah menjadi tiga bagian: tag, indeks, dan offset. Indeks digunakan untuk menentukan lokasi di cache tempat data mungkin disimpan, sedangkan tag digunakan untuk memverifikasi apakah data yang benar berada di lokasi tersebut.
2. **Pengecekan Tag**: Setelah indeks menunjukkan lokasi di cache, tag dari alamat memori dibandingkan dengan tag yang disimpan di lokasi cache tersebut. Jika keduanya cocok, ini menunjukkan bahwa data yang benar berada di cache (cache hit). Jika tidak cocok, data harus diambil dari memori utama (cache miss).
3. **Penggantian Data**: Jika terjadi cache miss, data dari memori utama diambil dan disimpan di lokasi cache yang ditunjukkan oleh indeks. Data yang sebelumnya berada di lokasi tersebut digantikan.
4. **Pengambilan Data**: Jika terjadi cache hit, data diambil dari cache dan dikirim ke prosesor. Jika terjadi cache miss, setelah data baru disimpan di cache, data tersebut juga dikirim ke prosesor.

Ilustrasi

Direct-Mapped Caches

- Each word in memory maps into a single cache line
- Access (for cache with 2^W lines):
 - Index into cache with W address bits (the **index bits**)
 - Read out valid bit, tag, and data
 - If valid bit == 1 and tag matches upper address bits, HIT



Gambar di atas menunjukkan bagaimana alamat memori dipecah menjadi tag, indeks, dan offset. Indeks menunjukkan lokasi di cache, sedangkan tag digunakan untuk verifikasi.

Example on Cache Placement & Misses

- Consider a small direct-mapped cache with 32 blocks
 - Cache is initially empty, Block size = 16 bytes
 - The following memory addresses (in decimal) are referenced: 0x3E8, 0x3EC, 0x3F0, 0x9F4, 0x9F8, 0x9FC.
 - Map addresses to cache blocks and indicate whether hit or miss

• Solution:

	23	5	4
	Tag	Index	offset
– 0x3E8	cache index = 0x1E	Miss (first access)	
– 0x3EC	cache index = 0x1E	Hit	
– 0x3F0	cache index = 0x1F	Miss (first access)	
– 0x9F4	cache index = 0x1F	Miss (different tag)	
– 0x9F8	cache index = 0x1F	Hit	
– 0x9FC	cache index = 0x1F	Hit	

Source [Direct Mapped Cache Example With Hit Miss \(city-mapss.blogspot.com\)](http://city-mapss.blogspot.com)

PPT - Caching II PowerPoint Presentation, free download - ID:2400431 (slideserve.com)

Kelebihan dan Kekurangan

Kelebihan dari Direct Mapped Cache adalah desainnya yang sederhana dan cepat. Namun, metode ini memiliki kekurangan dalam hal kemungkinan terjadinya konflik. Karena setiap blok memori hanya memiliki satu lokasi potensial di cache, dua alamat yang berbeda tetapi memiliki indeks yang sama akan saling menggantikan satu sama lain di cache, menyebabkan peningkatan cache miss.

Kesimpulan

Direct Mapped Cache adalah metode penyimpanan cache yang sederhana namun efektif. Meskipun memiliki potensi konflik, desainnya yang cepat dan efisien membuatnya menjadi pilihan yang populer dalam banyak sistem komputer. Memahami cara kerja Direct Mapped Cache adalah kunci untuk memahami kinerja dan efisiensi cache dalam sistem komputer.

4. Diskusi dan Tanya Jawab (xx menit)

Bagian 2: Praktikum

1. Pengenalan Alat Praktikum (10 menit)

- Perangkat keras dan perangkat lunak yang digunakan.
- Tujuan praktikum.

2. Simulasi Direct Mapped Cache (30 menit)

- Mahasiswa diberikan skenario penggunaan cache.
- Menggunakan perangkat lunak simulasi untuk melihat bagaimana Direct Mapped Cache bekerja.
- Analisis hasil simulasi.

3. Evaluasi: Presentasi (10 menit)

- Setiap kelompok mahasiswa mempresentasikan hasil praktikum mereka.
- Kriteria indikator mampu:
 - Menjelaskan konsep Direct Mapped Cache dengan benar.
 - Menunjukkan hasil simulasi dengan tepat.
 - Menganalisis kelebihan dan kekurangan Direct Mapped Cache berdasarkan hasil praktikum.

Full Associative Cache

1. Full Associative Cache (xx menit)

- Definisi Full Associative Cache.
- Cara kerja Full Associative Cache.
- Kelebihan dan kekurangan Full Associative Cache.

2. Diskusi dan Tanya Jawab (10 menit)

Bagian 2: Praktikum (50 menit)

1. Pengenalan Alat Praktikum (10 menit)

- Perangkat keras dan perangkat lunak yang digunakan.
- Tujuan praktikum.

2. Simulasi Full Associative Cache (30 menit)

- Mahasiswa diberikan skenario penggunaan cache.
- Menggunakan perangkat lunak simulasi untuk melihat bagaimana Full Associative Cache bekerja.

- Analisis hasil simulasi.

3. Evaluasi: Presentasi (10 menit)

- Setiap kelompok mahasiswa mempresentasikan hasil praktikum mereka.
- Kriteria indikator mampu:
 - Menjelaskan konsep Full Associative Cache dengan benar.
 - Menunjukkan hasil simulasi dengan tepat.
 - Menganalisis kelebihan dan kekurangan Full Associative Cache berdasarkan hasil praktikum.

Set Associative Cache

Tujuan Pembelajaran: Mahasiswa mampu memahami teknologi cache, khususnya Set Associative Cache.

Durasi: 2x50 menit

A. Pendahuluan (xx menit)

1. Review singkat tentang cache memory dan pentingnya cache dalam meningkatkan kinerja komputer.
2. Pengenalan singkat tentang Set Associative Cache dan perbedaannya dengan Direct Mapped Cache dan Fully Associative Cache.

Pengenalan Set Associative Cache dan Perbedaannya dengan Direct Mapped Cache dan Fully Associative Cache

Set Associative Cache

Set Associative Cache adalah kompromi antara Direct Mapped Cache dan Fully Associative Cache. Dalam Set Associative Cache, setiap blok memori dapat ditempatkan dalam satu dari beberapa tempat di cache, yang dikenal sebagai "set". Setiap set terdiri dari beberapa blok cache, dan jumlah blok dalam set disebut "tingkat asosiativitas". Misalnya, cache 2-way set associative memiliki dua blok dalam setiap set.

Cara Kerja:

1. Saat data diperlukan, indeks dari alamat memori digunakan untuk menemukan set yang sesuai di cache.
2. Semua blok dalam set tersebut diperiksa secara simultan untuk mencari data yang sesuai.
3. Jika data ditemukan di salah satu blok dalam set, ini merupakan cache hit. Jika tidak, ini merupakan cache miss.

Direct Mapped Cache

Dalam Direct Mapped Cache, setiap blok memori hanya dapat ditempatkan di satu lokasi spesifik di cache. Lokasi ini ditentukan oleh indeks dari alamat memori.

Cara Kerja:

1. Saat data diperlukan, indeks dari alamat memori digunakan untuk menemukan blok cache yang sesuai.
2. Jika data di blok cache cocok dengan tag dari alamat memori, ini merupakan cache hit. Jika tidak, ini merupakan cache miss.

Fully Associative Cache

Dalam Fully Associative Cache, setiap blok memori dapat ditempatkan di mana saja di cache. Tidak ada pembatasan berdasarkan indeks atau set.

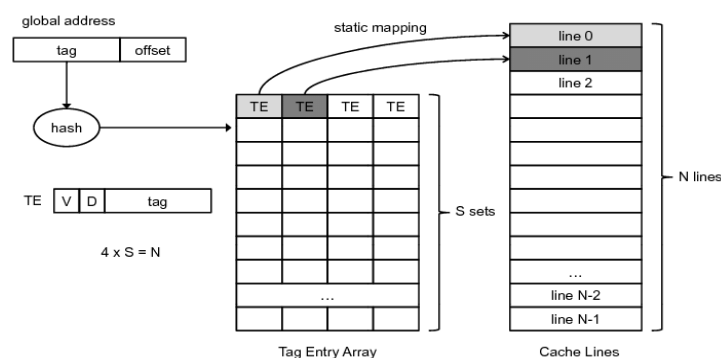
Cara Kerja:

1. Saat data diperlukan, semua blok di cache diperiksa secara simultan untuk mencari data yang sesuai.
2. Jika data ditemukan di salah satu blok, ini merupakan cache hit. Jika tidak, ini merupakan cache miss.

Perbandingan:

- **Fleksibilitas:** Fully Associative Cache paling fleksibel karena data dapat ditempatkan di mana saja di cache. Direct Mapped paling terbatas, sementara Set Associative berada di tengah-tengah.
- **Kecepatan:** Direct Mapped biasanya paling cepat karena hanya memerlukan satu lokasi untuk diperiksa. Fully Associative memerlukan waktu lebih lama karena semua blok harus diperiksa. Set Associative memerlukan waktu antara keduanya.
- **Kompromi:** Set Associative Cache menawarkan kompromi antara kecepatan dan fleksibilitas, mencoba menggabungkan kelebihan dari kedua jenis cache lainnya.

Ilustrasi:



sumber [The 4-way set-associative cache. | Download Scientific Diagram \(researchgate.net\)](#)

Gambar di atas menunjukkan bagaimana blok memori dipetakan ke cache untuk tiga jenis cache: Direct Mapped, Fully Associative, dan Set Associative.

B. Tutorial (xx menit)

1. Pengertian Set Associative Cache (xx menit)

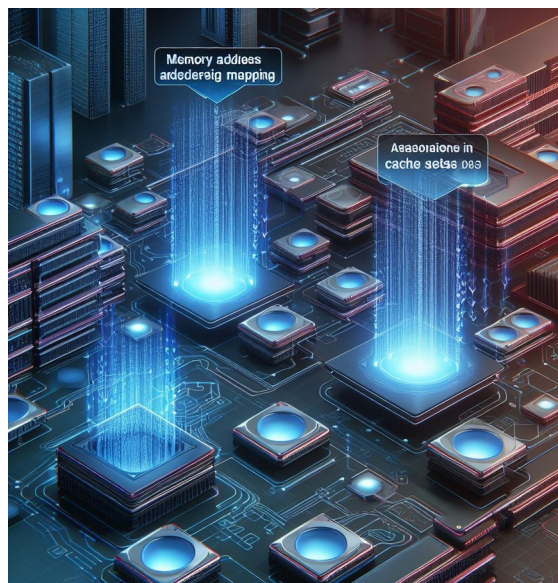
- Definisi dan konsep dasar Set Associative Cache.
- Bagaimana Set Associative Cache menggabungkan kelebihan dari Direct Mapped dan Fully Associative Cache.

2. Cara Kerja Set Associative Cache (xx menit)

- Pemetaan alamat memori ke dalam Set Associative Cache.

Pemetaan alamat memori ke dalam Set Associative Cache adalah teknik yang digunakan untuk memetakan alamat memori ke dalam cache. Pada teknik ini, cache dibagi menjadi beberapa set, dan setiap set terdiri dari beberapa blok. Alamat memori dibagi menjadi tiga bagian: tag, index, dan offset. Tag digunakan untuk menunjukkan blok mana yang ada di cache, index digunakan untuk menunjukkan set mana yang berisi blok tersebut, dan offset digunakan untuk menunjukkan lokasi byte dalam blok tersebut.

Set Associative Cache adalah teknik pemetaan yang menggabungkan antara Direct Mapping dan Fully Associative Mapping. Pada teknik ini, cache dibagi menjadi beberapa set, dan setiap set terdiri dari beberapa blok. Setiap blok dapat dipetakan ke dalam sembarang saluran cache pada set tertentu. Pemilihan saluran cache dilakukan dengan menggunakan algoritma penggabungan Direct Mapping dan Fully Associative Mapping.



Pemetaan alamat memori ke dalam Set Associative Cache

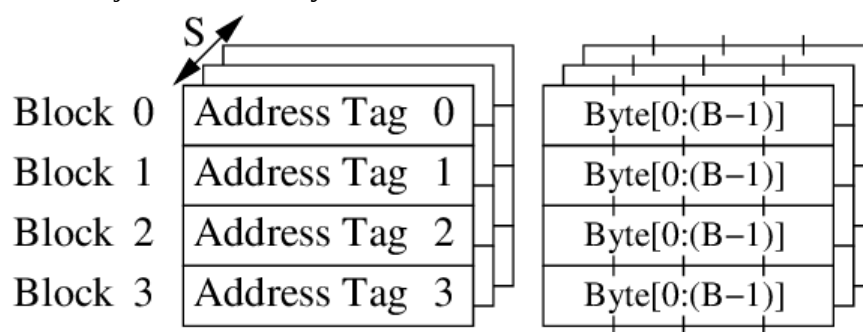
Sumber <https://www.bing.com/images/create/pemetaan-alamat-memori-ke-dalam-set-associative-ca/653bc1b1a3354da49cc66c8ddd1a6ff5?id=Bpacc46E02yN40JXZeR84w%3d%3d&view=detailv2&idpp=genimg&idpclose=1&FORM=SYDBIC>

- Konsep "set" dalam cache dan bagaimana data disimpan dan diambil.

Konsep "set" dalam cache adalah bagian dari teknik pemetaan cache yang membagi cache menjadi beberapa set, dan setiap set terdiri dari beberapa blok. Setiap blok dapat dipetakan ke dalam sembarang saluran cache pada set tertentu. Pemilihan saluran cache dilakukan dengan menggunakan algoritma penggabungan Direct Mapping dan Fully Associative Mapping.

Data disimpan dalam cache dalam bentuk blok. Setiap blok memiliki tag, data, dan bit validitas. Tag digunakan untuk menunjukkan blok mana yang ada di cache, data digunakan untuk menyimpan informasi aktual, dan bit validitas digunakan untuk menunjukkan apakah blok tersebut berisi data aktual atau tidak.

Data diambil dari cache dengan cara mencari alamat memori pada cache. Alamat memori dibagi menjadi tiga bagian: tag, index, dan offset. Tag digunakan untuk menunjukkan blok mana yang ada di cache, index digunakan untuk menunjukkan set mana yang berisi blok tersebut, dan offset digunakan untuk menunjukkan lokasi byte dalam blok tersebut.



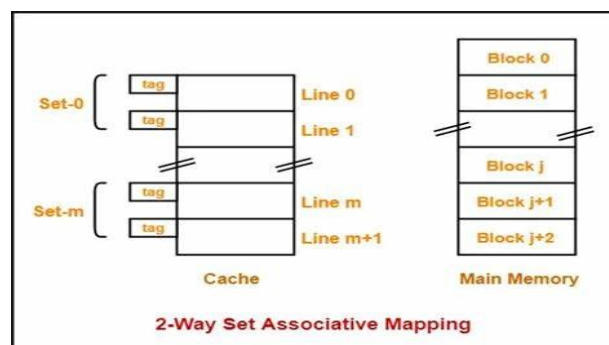
Gambar 4-Way Set Associative Cache Memory [8]

Sumber https://www.researchgate.net/figure/4-Way-Set-Associative-Cache-Memory-8_fig2_235339284

- Proses pengecekan tag dan pemilihan set.

Proses pengecekan tag dan pemilihan set adalah bagian dari teknik pemetaan cache yang membagi cache menjadi beberapa set, dan setiap set terdiri dari beberapa blok. Setiap blok dapat dipetakan ke dalam sembarang saluran cache pada set tertentu. Pemilihan saluran cache dilakukan dengan menggunakan algoritma penggabungan Direct Mapping dan Fully Associative Mapping.

Pada saat data diambil dari cache, alamat memori dibagi menjadi tiga bagian: tag, index, dan offset. Tag digunakan untuk menunjukkan blok mana yang ada di cache, index digunakan untuk menunjukkan set mana yang berisi blok tersebut, dan offset digunakan untuk menunjukkan lokasi byte dalam blok tersebut.



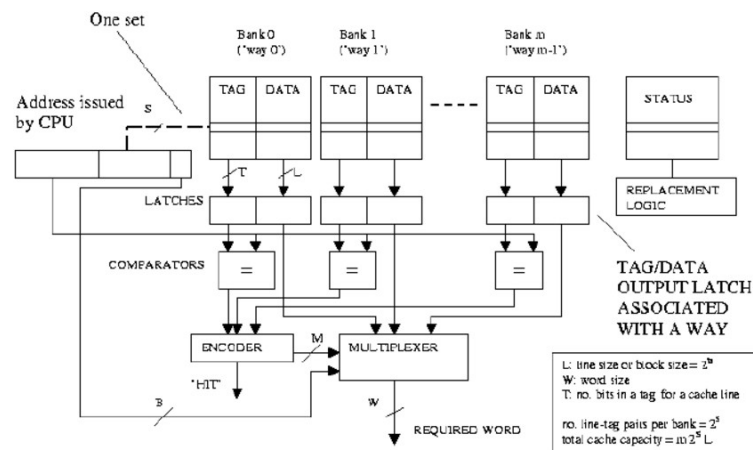
Gambar Cache Mapping | Cache Mapping Techniques

Sumber [Cache Mapping | Cache Mapping Techniques | Gate Vidyalay](#)

3. Kelebihan dan Kekurangan Set Associative Cache (xx menit)

- Mengapa Set Associative Cache sering digunakan dalam arsitektur komputer modern.

Set Associative Cache sering digunakan dalam arsitektur komputer modern karena memiliki beberapa keuntungan dibandingkan dengan jenis cache memory lainnya. Pertama, Set Associative Cache memungkinkan CPU untuk mengakses data dengan lebih cepat daripada Direct Mapped Cache. Kedua, Set Associative Cache memungkinkan CPU untuk mengakses data dengan lebih cepat daripada Fully Associative Cache. Ketiga, Set Associative Cache memungkinkan CPU untuk mengakses data dengan lebih cepat daripada Random Access Memory (RAM).



Gambar : A m-way set-associative cache organization

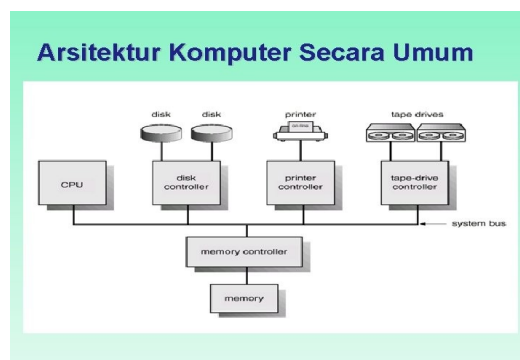
sumber [A m-way set-associative cache organization | Download Scientific Diagram \(researchgate.net\)](#)

- Pertimbangan dalam memilih jumlah set dan tingkat asosiativitas.

Pertimbangan dalam memilih jumlah set dan tingkat asosiativitas pada Set Associative Cache adalah sebagai berikut:

Jumlah set: Semakin banyak jumlah set, semakin besar kapasitas cache memory, dan semakin sedikit kemungkinan terjadinya konflik cache. Namun, semakin banyak jumlah set, semakin tinggi biaya implementasi cache memory.

Tingkat asosiativitas: Semakin tinggi tingkat asosiativitas, semakin sedikit kemungkinan terjadinya konflik cache. Namun, semakin tinggi tingkat asosiativitas, semakin besar biaya implementasi cache memory.



C. Praktikum (xx menit)

1. Simulasi Set Associative Cache (20 menit)

- Menggunakan perangkat lunak simulasi untuk memahami bagaimana Set Associative Cache bekerja.
- Melakukan beberapa operasi memori dan mengamati bagaimana data disimpan dan diambil dari cache.

2. Analisis Hasil Simulasi (10 menit)

- Diskusi tentang hasil simulasi dan bagaimana Set Associative Cache meningkatkan kinerja dibandingkan dengan jenis cache lainnya.
- Mengidentifikasi situasi di mana Set Associative Cache memberikan kinerja terbaik.

3. Penerapan dalam Sistem Nyata (10 menit)

- Demonstrasi tentang bagaimana Set Associative Cache diimplementasikan dalam prosesor komputer nyata.
- Diskusi tentang pertimbangan desain dalam memilih Set Associative Cache untuk aplikasi tertentu.

D. Evaluasi (10 menit)

1. Presentasi oleh Mahasiswa (5 menit)

- Beberapa mahasiswa dipilih untuk mempresentasikan pemahaman mereka tentang Set Associative Cache berdasarkan materi yang telah diajarkan.

2. Tanya Jawab (5 menit)

- Sesi tanya jawab untuk mengklarifikasi konsep yang mungkin belum dimengerti.

E. Penutup (10 menit)

1. Ringkasan materi yang telah diajarkan.
2. Penekanan pada pentingnya memahami Set Associative Cache dalam konteks teknologi cache dan arsitektur komputer.
3. Pemberian tugas atau bahan bacaan tambahan untuk pertemuan berikutnya.

Kriteria Indikator Kemampuan :

1. Mahasiswa dapat mendefinisikan Set Associative Cache dengan tepat.
2. Mahasiswa dapat menjelaskan cara kerja Set Associative Cache dengan benar.
3. Mahasiswa mampu membedakan Set Associative Cache dengan jenis cache lainnya.
4. Mahasiswa dapat menganalisis kelebihan dan kekurangan Set Associative Cache.
5. Mahasiswa mampu melakukan simulasi dan menganalisis hasil dari Set Associative Cache.

Page Replacement

Tujuan Pembelajaran :

- Mahasiswa mampu memahami konsep dasar teknologi cache.
- Mahasiswa mampu menjelaskan mekanisme Page Replacement dalam cache.
- Mahasiswa mampu menerapkan konsep Page Replacement dalam praktikum.

Durasi : 2x50 menit

Bagian 1: Tutorial (50 menit)

1. Pengantar Teknologi Cache (10 menit)

- Definisi dan fungsi cache dalam sistem komputer.
- Manfaat penggunaan cache.

2. Jenis-jenis Cache (10 menit)

- Cache L1, L2, dan L3.
- Perbandingan performa dan kapasitas.

3. Page Replacement dalam Cache (20 menit)

- Definisi Page Replacement.
- Algoritma Page Replacement populer (FIFO, LRU, Optimal).
- Kelebihan dan kekurangan masing-masing algoritma.

4. Diskusi dan Tanya Jawab (10 menit)

Bagian 2: Praktikum (50 menit)

1. Pengenalan Alat Praktikum (10 menit)

- Perangkat keras dan perangkat lunak yang digunakan.
- Tujuan praktikum.

2. Simulasi Page Replacement (30 menit)

- Mahasiswa diberikan skenario penggunaan cache.
- Menggunakan perangkat lunak simulasi untuk melihat bagaimana algoritma Page Replacement bekerja.
- Analisis hasil simulasi berdasarkan algoritma yang digunakan.

3. Evaluasi: Presentasi (10 menit)

- Setiap kelompok mahasiswa mempresentasikan hasil praktikum mereka.
- Kriteria indikator mampu:
 - Menjelaskan konsep Page Replacement dengan benar.
 - Menunjukkan hasil simulasi dengan tepat.
 - Menganalisis kelebihan dan kekurangan algoritma Page Replacement berdasarkan hasil praktikum.

Referensi

- [1](#) Lecture 6/9:Caches-Direct Mapped - YouTube
- [2](#) Cache introduction - Washington
- [3](#) Cache Memory in Computer Organization - GeeksforGeeks
- [4](#) Understanding Direct Mapped Cache - Stack Overflow
- [5](#) Direct mapping cache with LRU - Computer Science Stack Exchange
- [6](#) Ep 075: Direct Mapped Caches - YouTube

Dalam pertemuan ini, mahasiswa akan mempelajari teknologi cache, khususnya direct mapped cache. Materi akan disampaikan melalui metode pembelajaran tutorial dan praktikum untuk memudahkan pemahaman mahasiswa. Evaluasi dilakukan melalui presentasi dan kriteria indikator mampu. Referensi yang dapat digunakan adalah dari beberapa sumber seperti video YouTube dan artikel dari situs web.