Pertemuan ke – 9
Memori

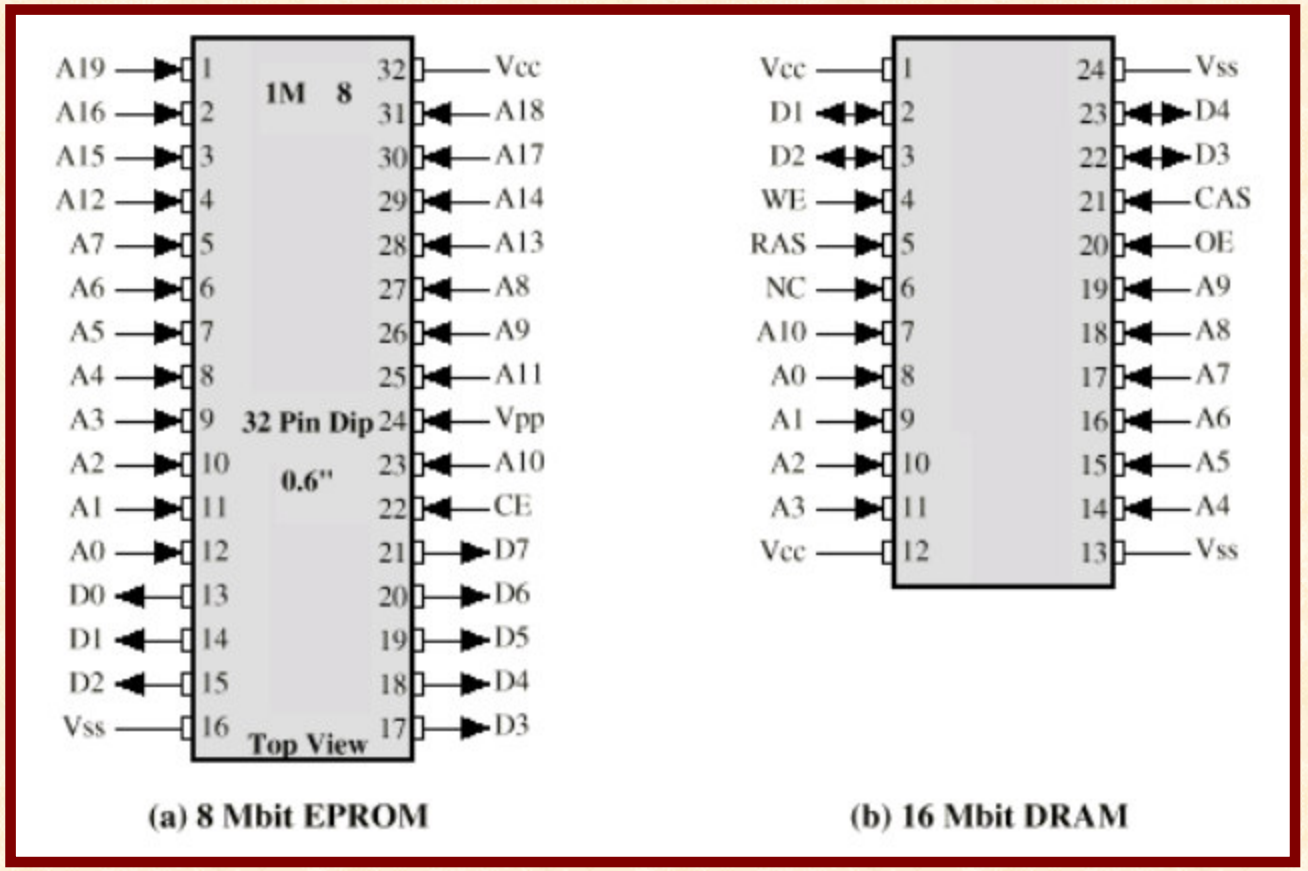
Riyanto Sigit, ST.
Nur Rosyid, S.kom
Setiawardhana, ST
Hero Yudo M, ST

Politeknik Elektronika Negeri Surabaya

Tujuan

1. Menjelaskan tentang memori utama komputer
2. Menjelaskan tipe dari memori, waktu dan pengontrolan
3. Menjelaskan pembetulan kesalahan
4. Menjelaskan cache memori termasuk didalamnya adalah fungsi pemetaan

Pengemasan (Packging)



(a) 8 Mbit EPROM

(b) 16 Mbit DRAM

Pengemasan (Packging)

Gambar (a)

- EPROM yang merupakan keping 8 Mbit yang diorganisasi sebagai 1Mx8.
- Organisasi dianggap sebagai kemasan satu word per keping.
- Kemasan terdiri dari 32 pin, yang merupakan salah satu ukuran kemasan keping standar

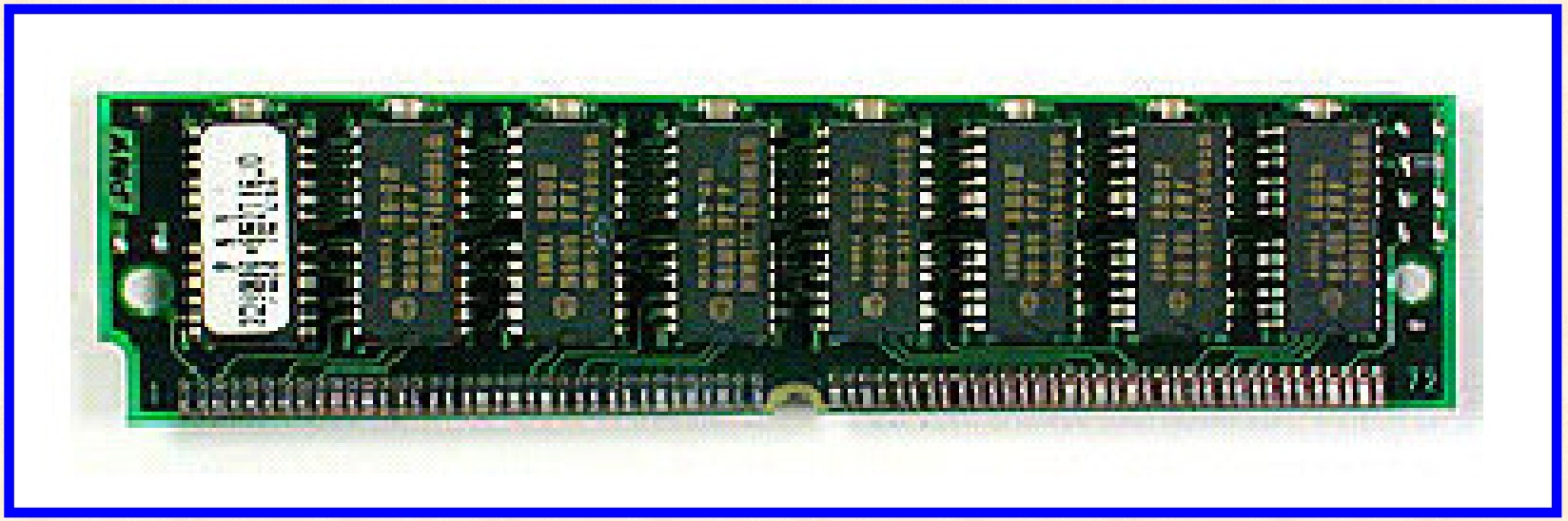
Gambar (b)

- Keping 16 Mbit yang diorganisasikan sebagai 4M x 4.
- Terdapat sejumlah perbedaan dengan keping ROM, karena ada operasi tulis maka pin – pin data merupakan input/output yang dikendalikan oleh WE (*write enable*) dan OE (*output enable*).

Pengemasan (Packging)

- Alamat word yang sedang diakses. Untuk 1M word, diperlukan sejumlah 20 buah (220 = 1M).
- Data yang akan dibaca, terdiri dari 8 saluran (D0 –D7)
- Catu daya keping adalah Vcc
- Pin grounding Vss
- Pin chip enable (CE). Karena mungkin terdapat lebih dari satu keping memori yang terhubung pada *bus* yang sama maka pin CE digunakan untuk mengindikasikan valid atau tidaknya pin ini. Pin CE diaktifkan oleh logik yang terhubung dengan bit berorde tinggi *bus* alamat (diatas A19)
- Tegangan program (Vpp).

Pengemasan (Packging)



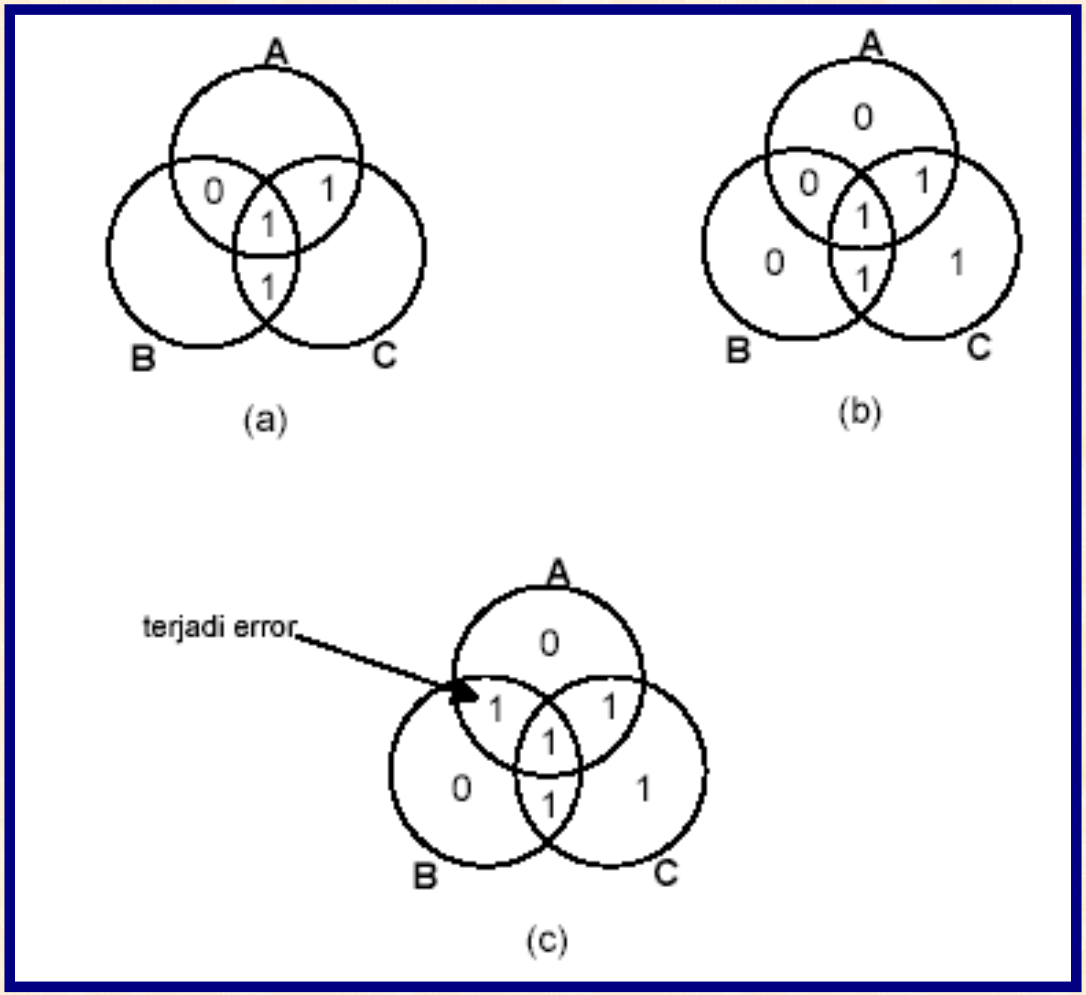
Koreksi Error

- Dalam melaksanakan fungsi penyimpanan, memori semikonduktor dimungkinkan mengalami kesalahan.
- Kesalahan berat yang biasanya merupakan kerusakan fisik memori
- Kesalahan ringan yang berhubungan data yang disimpan.
- Kesalahan ringan dapat dikoreksi kembali.
- Koreksi kesalahan data yang disimpan diperlukan dua mekanisme
  - Mekanisme pendeteksian kesalahan
  - Mekanisme perbaikan kesalahan

Kode Hamming

- Diciptakan Richard Hamming di Bell Lab 1950
- Mekanisme pendeteksian kesalahan dengan menambahkan data word (D) dengan suatu kode, biasanya bit cek paritas (C).
- Data yang disimpan memiliki panjang D + C.
- Kesalahan diketahui dengan menganalisa data dan bit paritas tersebut

Kode Hamming



Kode Hamming

# Data Bits	# Bit Paritas SEC	# Bit Paritas DEC
8	4	5
16	5	6
32	6	7
64	7	8
128	8	9
512	9	10

Penambahan bit cek paritas untuk koreksi kode Hamming

Kode Hamming

Posisi Bit	Posisi Angka				Cek Bit	Data Bit
12	1	1	0	0		D8
11	1	0	1	1		D7
10	1	0	1	0		D6
9	1	0	0	1		D5
8	1	0	0	0	C4	
7	0	1	1	1		D4
6	0	1	1	0		D3
5	0	1	0	1		D2
4	0	1	0	0	C3	
3	0	0	1	1		D1
2	0	0	1	0	C2	
1	0	0	0	1	C1	

Kode Hamming

- Bit cek paritas ditempatkan dengan perumusan 2N dimana N = 0,1,2, ……, sedangkan bit data adalah sisanya.
  Kemudian dengan *exclusive-OR* dijumlahkan:

$$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$$
$$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$$
$$C4 = D2 \oplus D3 \oplus D4 \oplus D8$$
$$C8 = D5 \oplus D6 \oplus D7 \oplus D8$$

- Setiap cek bit (C) beroperasi pada setiap posisi bit data yang nomor posisinya berisi bilangan 1 pada kolomnya

Kode Hamming

- masukkan data : 00111001 kemudian ganti bit data ke 3 dari 0 menjadi 1 sebagai error-nya.

- Bagaimanakah cara mendapatkan bit data ke 3 sebagai bit yang terdapat error?

Kode Hamming

Jawab :

Masukkan data pada perumusan cek bit paritas :

C1 = 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 = 1

C2 = 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 = 1

C4 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1

C8 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

Sekarang bit 3 mengalami kesalahan data menjadi: 00111**1**01

C1 = 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 = 1

C2 = 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 = 0

C4 = 0 ⊕ 1 ⊕ 1 ⊕ 0 = 0

C8 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

Kode Hamming

Apabila bit – bit cek dibandingkan antara yang lama dan baru maka terbentuk *syndrom word* :

C8	C4	C2	C1	
0	1	1	1	
0	0	0	1	
0	1	1	0	= 6

Sekarang kita lihat posisi bit ke-6 adalah data ke-3.

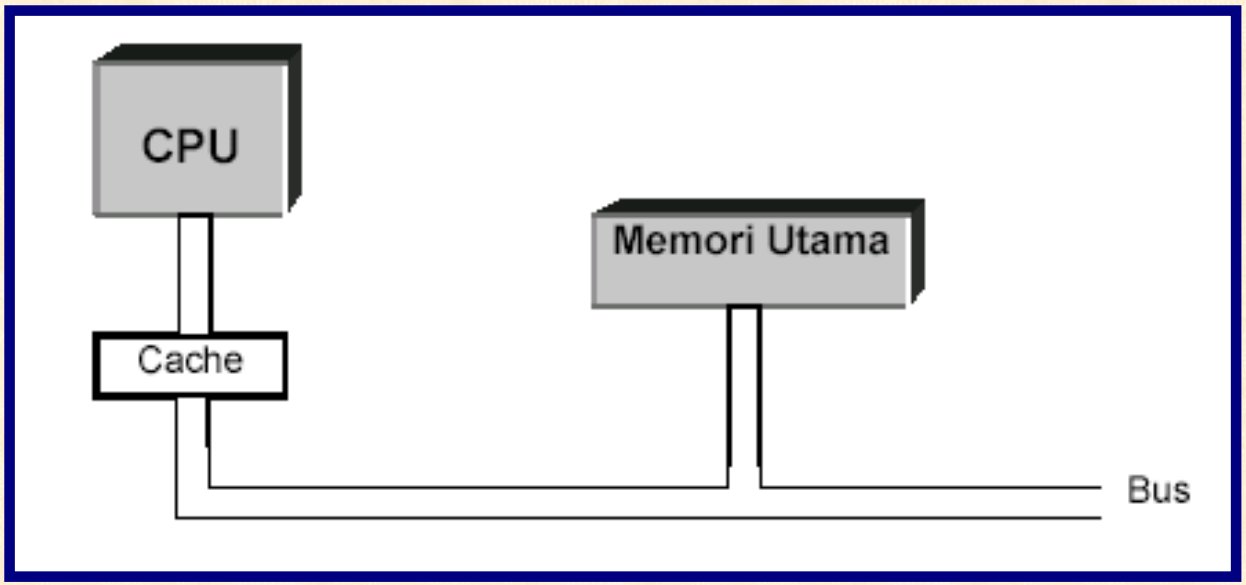
Kode Hamming

- Mekanisme koreksi kesalahan akan meningkatkan realibitas bagi memori
- Menambah kompleksitas pengolahan data.
- Menambah kapasitas memori karena adanya penambahan bit – bit cek paritas.
- Memori akan lebih besar beberapa persen atau dengan kata lain kapasitas penyimpanan akan berkurang karena beberapa lokasi digunakan untuk mekanisme koreksi kesalahan

Cache Memori

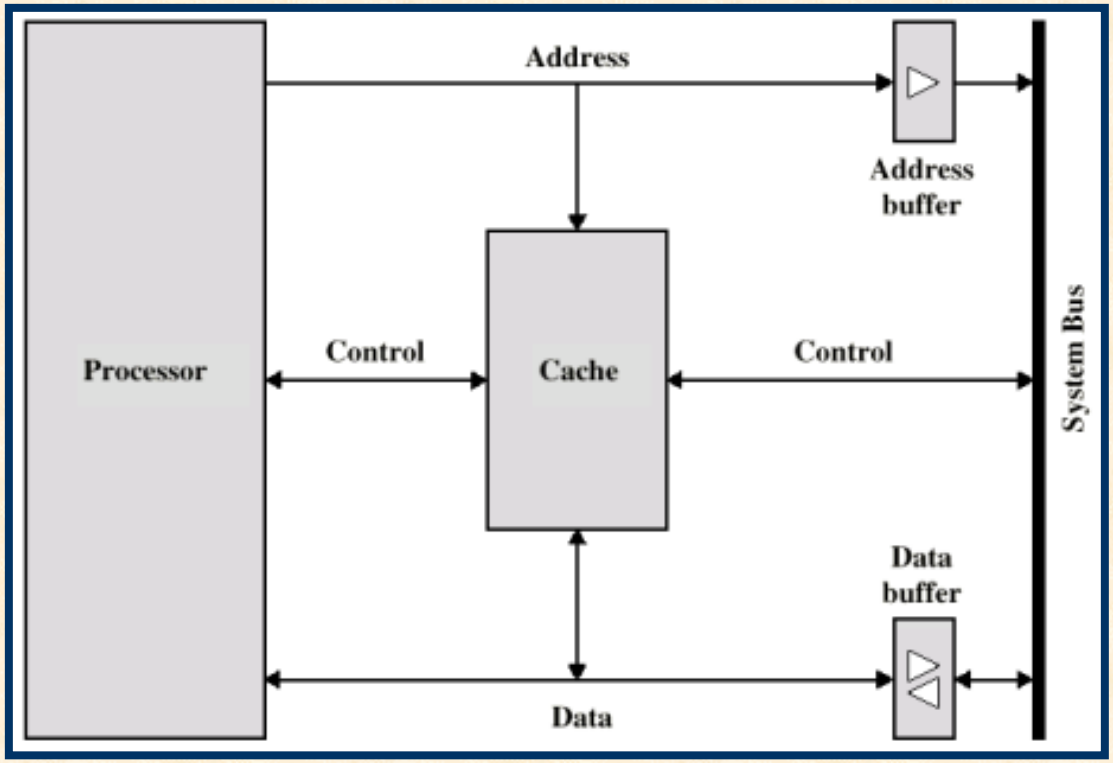
- Mempercepat kerja memori sehingga mendekati kecepatan prosesor.
- Memori utama lebih besar kapasitasnya namun lambat operasinya, sedangkan cache memori berukuran kecil namun lebih cepat.
- Cache memori berisi salinan memori utama

Cache Memori



Ukuran cache memori adalah kecil, semakin besar kapasitasnya maka akan memperlambat proses operasi cache memori itu sendiri, disamping harga cache memori yang sangat mahal

Organisasi Cache Memori



Elemen Cache Memori

Unsur	Macam
Kapasitas	-
Ukuran blok	-
Mapping	1. Direct Mapping 2. Assosiative Mapping 3. Set Assosiative Mapping
Algoritma pengganti	1. Least recently used (LRU) 2. First in first out (FIFO) 3. Least frequently used (LFU) 4. Random
Write Policy	1. Write Througth 2. Write Back 3. Write Once
Jumlah Cache	1. Singe atau dua level 2. Unified atau split

Kapasitas Cache

- AMD mengeluarkan prosesor K5 dan K6 dengan cache yang besar (1MB), kinerjanya tidak bagus
- Intel mengeluarkan prosesor tanpa cache untuk alasan harga yang murah, yaitu seri Intel Celeron pada tahun 1998-an, kinerjanya sangat buruk terutama untuk operasi data besar, floating point, 3D
- Sejumlah penelitian telah menganjurkan bahwa ukuran cache antara 1KB dan 512KB akan lebih optimum [STA96]

Ukuran Blok Cache

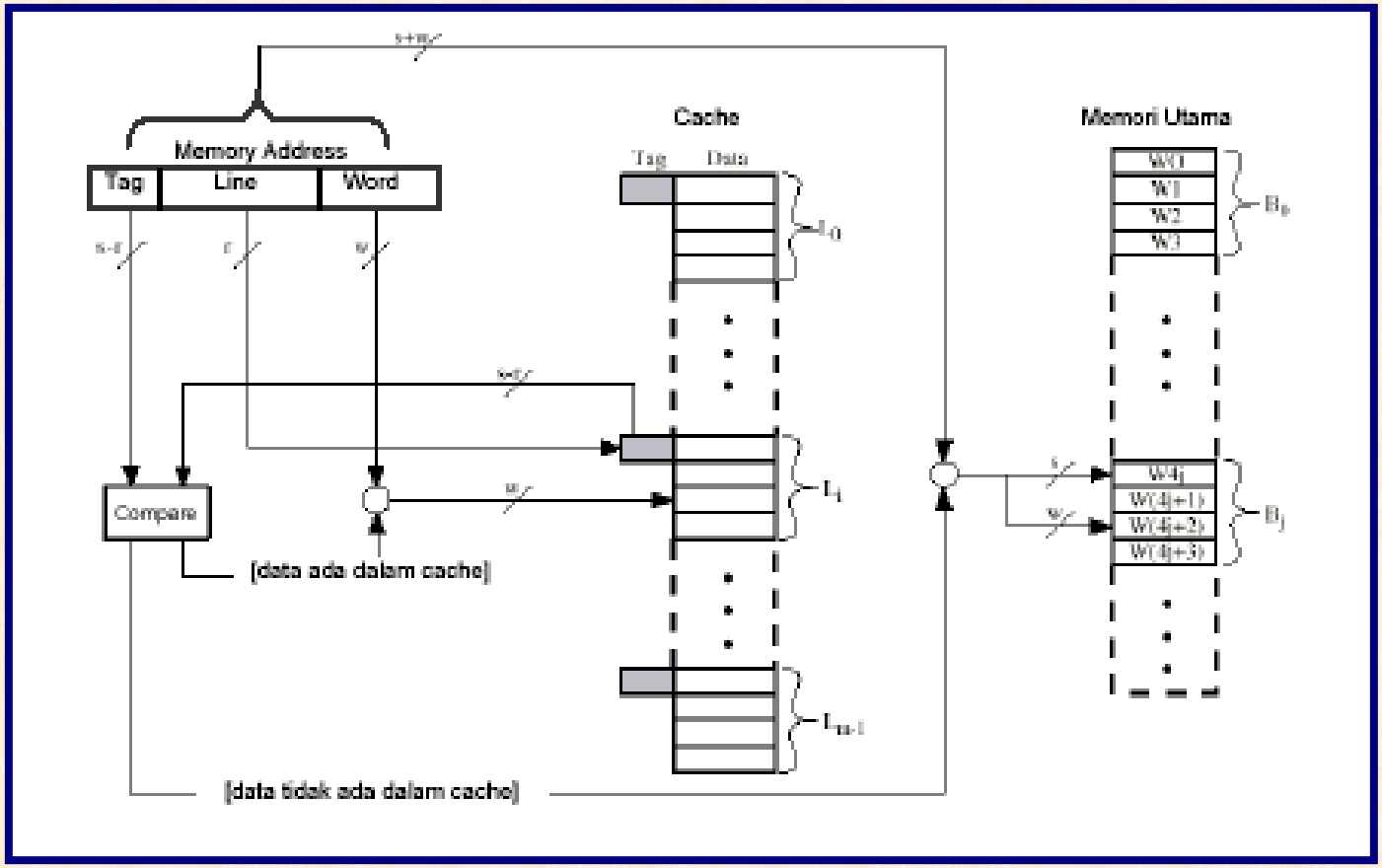
- Hubungan antara ukuran blok dan hit ratio sangat rumit untuk dirumuskan, tergantung pada karakteristik lokalitas programnya dan tidak terdapat nilai optimum yang pasti telah ditemukan.
- Ukuran antara 4 hingga 8 satuan yang dapat dialamati (word atau byte) cukup beralasan untuk mendekati nilai optimum [STA96]

Pemetaan (Cache)

- Cache mempunyai kapasitas yang kecil dibandingkan memori utama.
- Aturan blok – blok mana yang diletakkan dalam cache.
- Terdapat tiga metode, yaitu pemetaan langsung, pemetaan asosiatif, dan pemetaan asosiatif set

Pemetaan Langsung

- Teknik paling sederhana, yaitu teknik ini memetakan blok memori utama hanya ke sebuah saluran cache saja



Pemetaan Langsung

i = j modulus m dan **m = 2r**

dimana :

i = nomer saluran cache

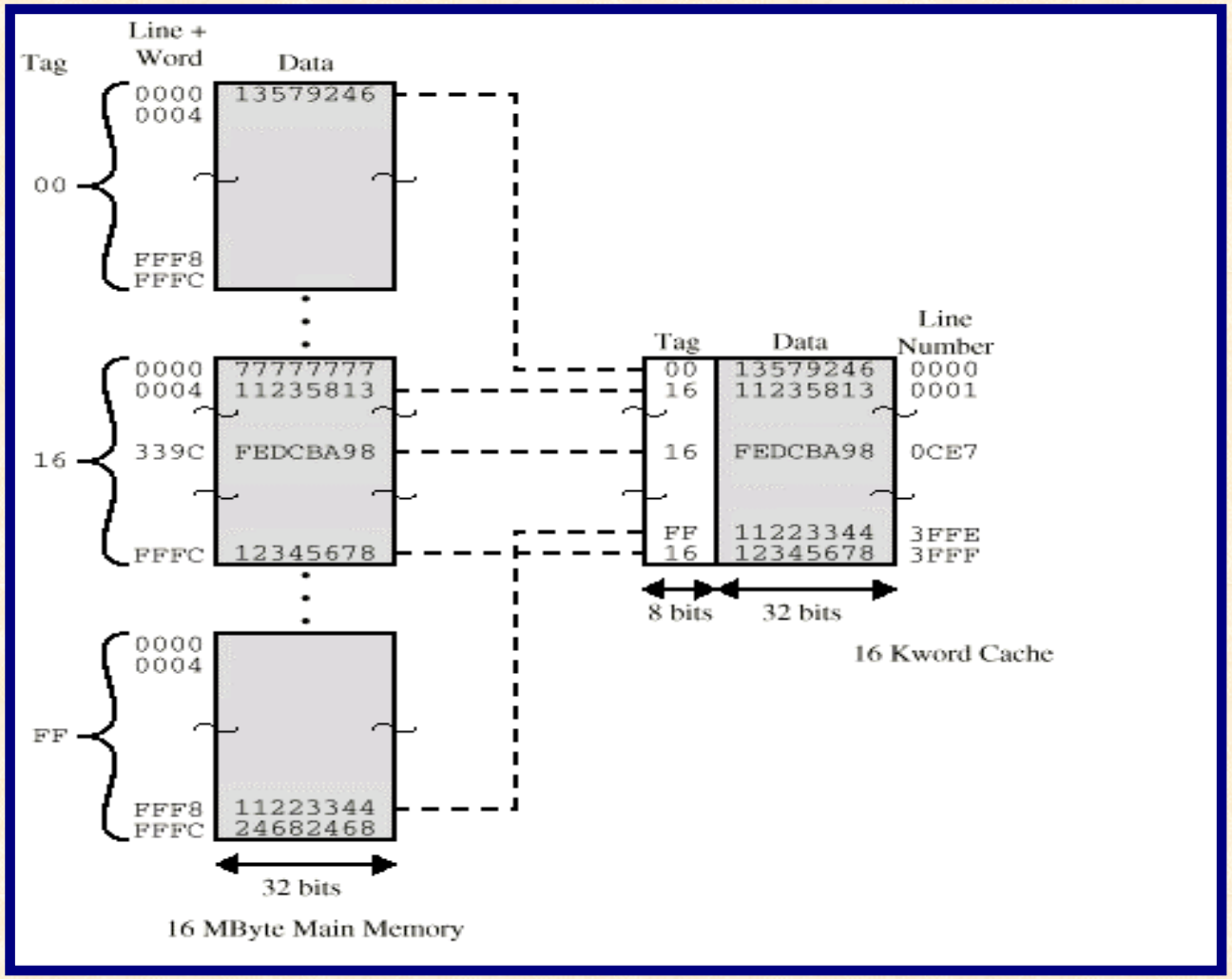
j = nomer blok memori utama

m = jumlah saluran yang terdapat dalam cache

Pemetaan Langsung

Saluran cache	Blok – blok memori utama
0	0,m, ...,2S - m
1	1, (m+1), ... ,2S – (m+1)
m-1	(m-1), (2m-1), ... ,2S - 1

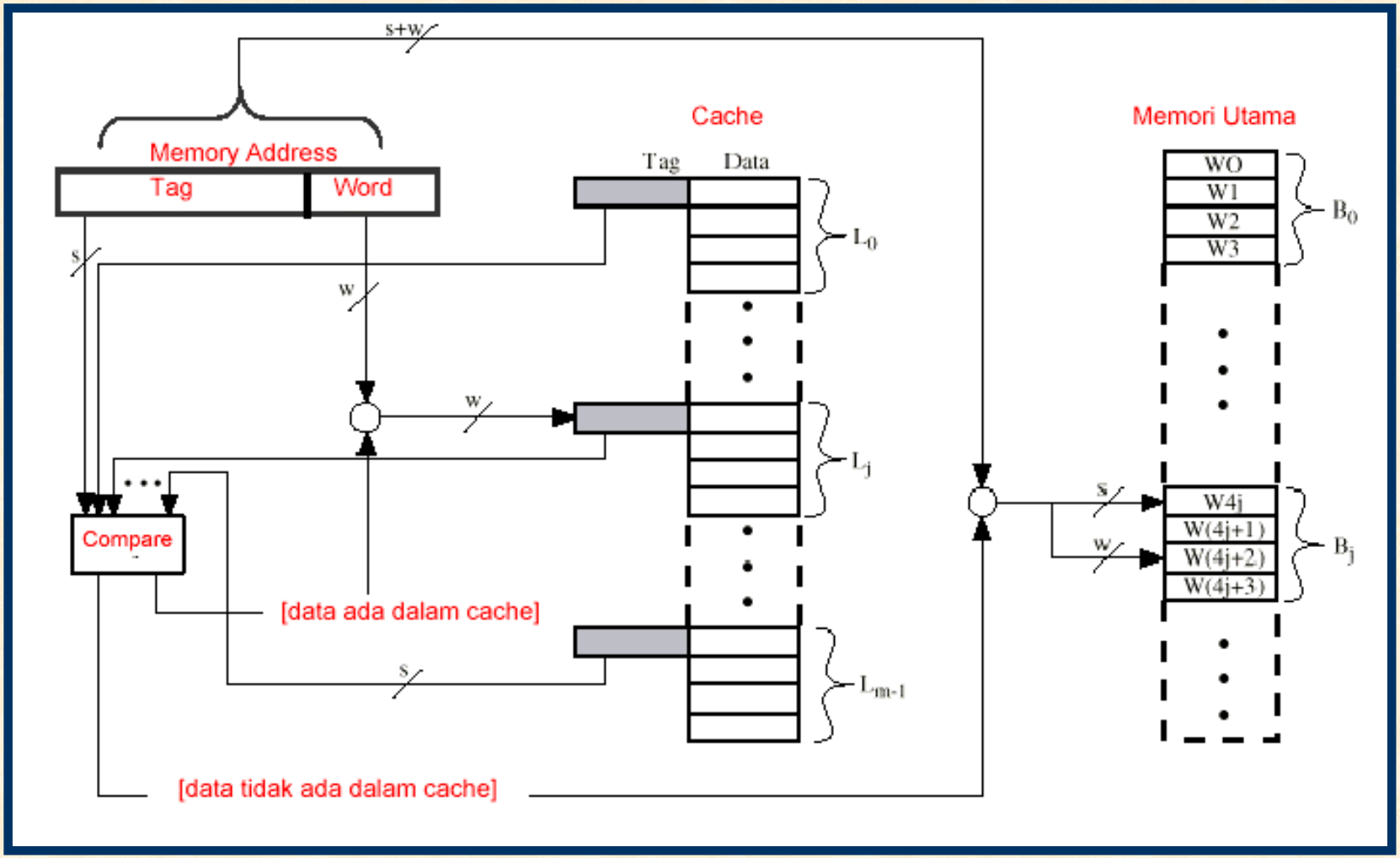
Pemetaan Langsung



Pemetaan Assosiatif

- Mengatasi kekurangan pemetaan langsung
- Tiap blok memori utama dapat dimuat ke sembarang saluran cache.
- Alamat memori utama diinterpretasikan dalam field tag dan field word oleh kontrol logika cache.
- Tag secara unik mengidentifikasi sebuah blok memori utama
- Mekanisme untuk mengetahui suatu blok dalam cache dengan memeriksa setiap tag saluran cache oleh kontrol logika cache.
- Fleksibilitas dalam penggantian blok baru yang ditempatkan dalam cache
- Kelebihan : Algoritma penggantian dirancang untuk memaksimalkan hit ratio, yang pada pemetaan langsung terdapat kelemahan
- Kekurangan : kompleksitas rangkaian sehingga mahal secara ekonomi

Pemetaan Assosiatif



Pemetaan Assosiatif Set

- Menggabungkan kelebihan yang ada pada pemetaan langsung dan pemetaan asosiatif.
- Memori cache dibagi dalam bentuk set–set.
- Alamat memori utama diinterpretasikan dalam tiga field, yaitu: field tag, field set, field word.
- Setiap blok memori utama dapat dimuat dalam sembarang saluran cache.
- Cache dibagi dalam *v* buah set, yang masing –masing terdiri dari k saluran

$$m = v \times k$$

i = j modulus v dan v = 2d dimana :

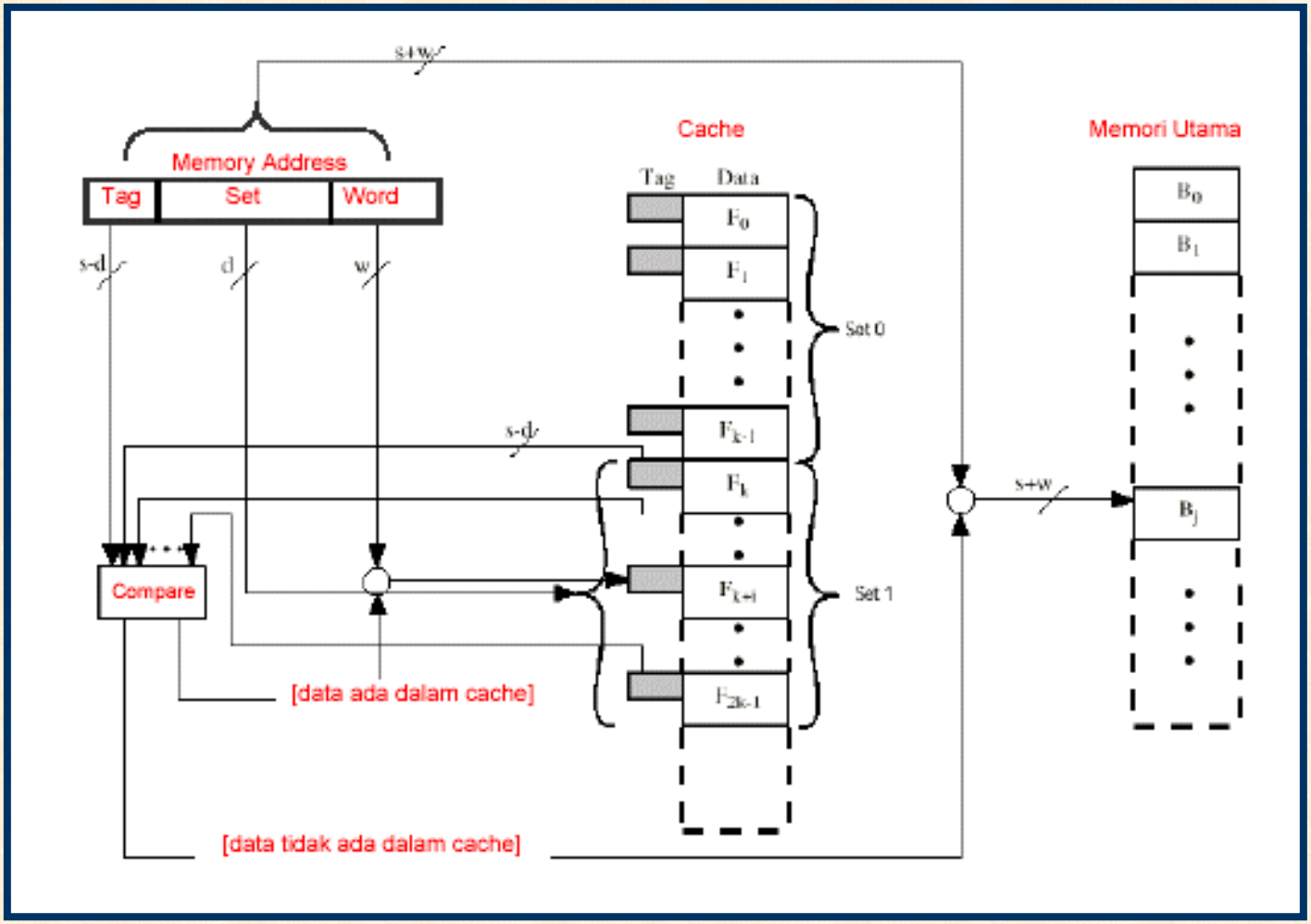
i = nomer set cache

j = nomer blok memori utama

m = jumlah saluran pada cache

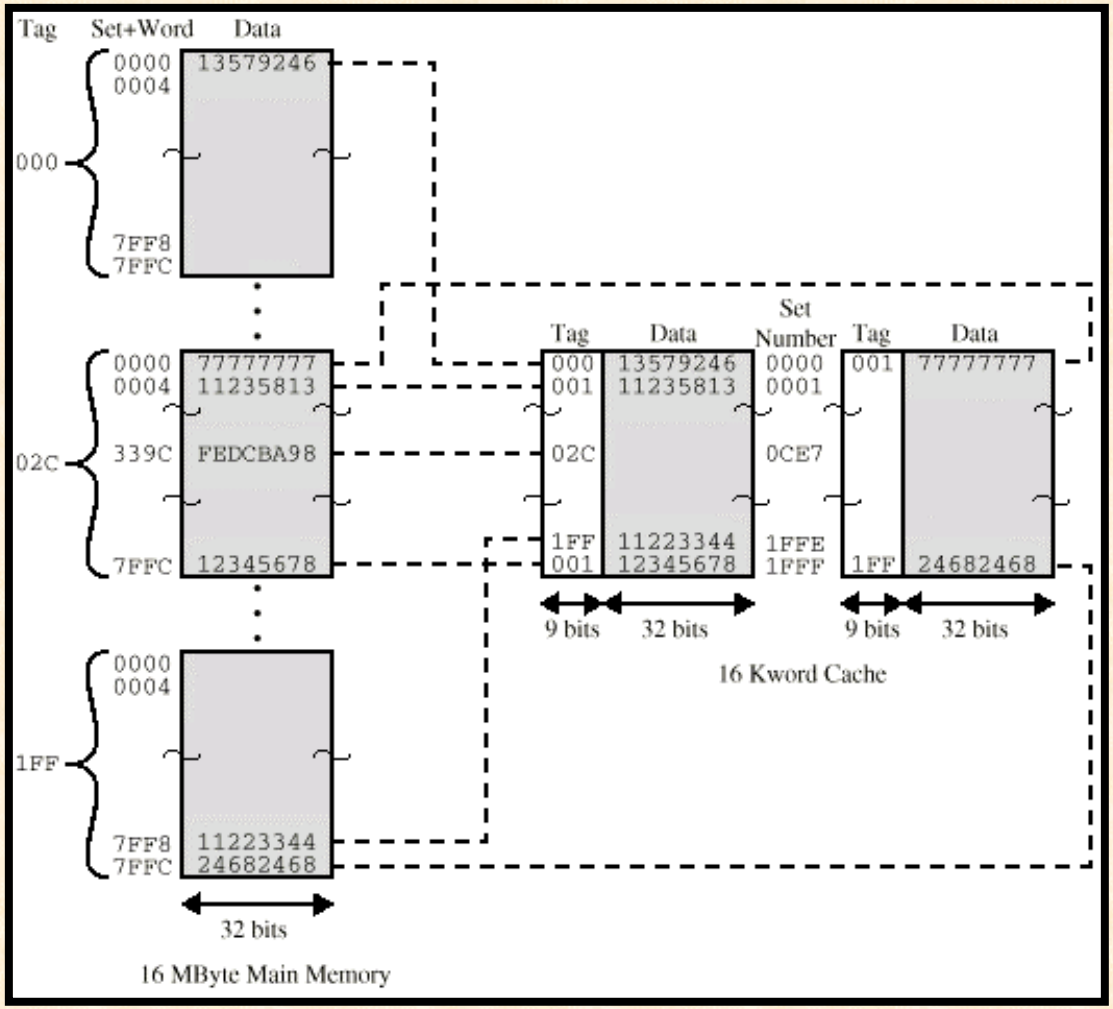
Pemetaan Assosiatif Set

(Organisasi cache dengan pemetaan asosiatif set)



Pemetaan Assosiatif Set

(Contoh pemetaan asosiatif set)



Algorithma Penggantian

- Suatu mekanisme pergantian blok–blok dalam memori cache yang lama dengan data baru
- Pemetaan langsung tidak memerlukan algoritma ini
- Pemetaan asosiatif dan asosiatif set, berperanan penting meningkatkan kinerja cache memori

Algorithma Penggantian

- Algoritma *Least Recently Used* (LRU), yaitu mengganti blok data yang terlama berada dalam cache dan tidak memiliki referensi. **(EFEKTIF)**
- Algoritma *First In First Out* (FIFO), yaitu mengganti blok data yang awal masuk
- Algorithma *Least Frequently Used* (LFU) adalah mengganti blok data yang mempunyai referensi paling sedikit.
- Algoritma *Random*, yaitu penggantian tidak berdasakan pemakaian datanya, melainkan berdasar slot dari beberapa slot kandidat secara acak

Write Policy – Mengapa ?

- Apabila suatu data telah diletakkan pada cache maka sebelum ada penggantian harus dicek apakah data tersebut telah mengalami perubahan.
- Apabila telah berubah maka data pada memori utama harus di-update.
- Masalah penulisan ini sangat kompleks, apalagi memori utama dapat diakses langsung oleh modul I/O, yang memungkinkan data pada memori utama berubah, lalu bagaimana dengan data yang telah dikirim pada cache?
- Tentunya perbedaan ini menjadikan data **tidak valid**

Write Policy –"*write through*"

- Operasi penulisan melibatkan data pada memori utama dan sekaligus pada cache memori sehingga data **selalu valid**.
- Kekurangan teknik ini adalah
  - Lalu lintas data ke memori utama dan cache sangat tinggi
  - Mengurangi kinerja sistem, bisa terjadi hang

Write Policy –"*write back* "

- Teknik meminimasi penulisan dengan cara penulisan pada cache saja.
- Pada saat akan terjadi penggantian blok data cache maka baru diadakan penulisan pada memori utama.
- Masalah : manakala data di memori utama belum di-update telah diakses modul I/O sehingga data di memori utama tidak valid

Write Policy-Multi cache

- Multi cache untuk multi prosesor
  - Masalah yang lebih kompleks.
- Masalah validasi data tidak hanya antara cache dan memori utama
  - Antar cache harus diperhatikan

Heuristik :

Bus Watching with Write Through

Hardware Transparency

Non Cacheable Memory

Cache

- Cache Internal : dalam chip
  - Tidak memerlukan *bus* eksternal
  - Waktu aksesnya akan cepat sekali
- Cache Eksternal : diluar chip
  - *Cache* tingkat 2 (L2)

Cache

- Cache data
- Cache instruksi yang disebut *unified cache*

  - Keuntungan *unified cache* :
    - Hit rate yang tinggi karena telah dibedakan antara informasi data dan informasi instruksi
    - Hanya sebuah cache saja yang perlu dirancang dan diimplementasikan

Cache

- *split cache*
  - Mesin–mesin superscalar seperti Pentium dan PowerPC
  - Menekankan pada paralel proses dan perkiraan – perkiraan eksekusi yang akan terjadi.
- Kelebihan utama *split cache*
  - Mengurangi persaingan antara prosesor instruksi dan unit eksekusi untuk mendapatkan cache, hal ini sangat utama bagi perancangan prosesor–prosesor pipelining

Kesimpulan

- Memori adalah bagian dari komputer tempat program program dan data – data disimpan
- Elemen dasar memori adalah sel memori. Sel memori dipresentasikan dengan bilangan biner 1 atau 0. Sel memori mempunyai kemampuan untuk ditulisi dan dibaca.
- Untuk mempelajari sistem memori secara keseluruhan, harus mengetahui karakteristik – karakteristik kuncinya yaitu: Lokasi, Kapasitas, Satuan Transfer, Metode Akses, Kinerja, Tipe Fisik dan Karakteristik Fisik.
- Untuk memperoleh keandalan sistem ada tiga pertanyaan yang diajukan: Berapa banyak ? Berapa cepat? Berapa mahal?

Kesimpulan

- Dalam melaksanakan fungsi penyimpanan, memori semikonduktor dimungkinkan mengalami kesalahan. Untuk mengadakan koreksi kesalahan data yang disimpan diperlukan dua mekanisme, yaitu mekanisme pendeteksian kesalahan dan mekanisme perbaikan kesalahan
- Cache memori difungsikan mempercepat kerja memori sehingga mendekati kecepatan prosesor.

Soal-Soal

- Buatlah konfigurasi yang menggambarkan prosesor empat buah ROM 1K x 8-bit dan *bus* yang berisi 12 saluran alamat dan 8 saluran data. Tambahkan blok logic chip select yang memilih salah satu dari keempat modul ROM untuk masing2 alamat 4K.
- Jelaskan fungsi utama dari memori dan karakteristiknya.
- Jelaskan tipe dari memori, waktu dan pengontrolanna.
- Buatlah seuah kode SEC untuk word data 16-bt. Turunkan kode untuk word data 0101000000111001. Buktikan bahwa kode akan mengidentifikasi dengan benar sebuah error pada data bit 4.
- Cache assoiatif set terdiri dari 64 saluran, atau slot-slot yang terbagi menjadi set-set 4 slot. Memori utama berisi 4K blok masing-masing terdiri 128 word. Jelaskan format alamat-alamat memori utama