

eBook

## **MEMORI INTERNAL**

**Minggu 9**

Penyusun :

1. Imam Purwanto, S.Kom, MMSI
2. Ega Hegarini, S.Kom., MM
3. Rifki Amalia, S.Kom., MMSI
4. Arie Kusumawati, S.Kom

**Fakultas Teknologi Industri  
Universitas Gunadarma  
2013**

# Karakteristik Memori

- Lokasi
- Kapasitas
- Unit transfer
- Metode Akses
- Kinerja
- Jenis fisik
- Sifat-sifat fisik
- Organisasi

# Lokasi

- CPU (register)
- Internal (main memori)
- External (secondary memori)

# Kapasitas

- Ukuran Word
  - Satuan alami organisasi memori
- Banyaknya words
  - atau Bytes

# Satuan Transfer

- Internal
  - Jumlah bit dalam sekali akses
  - Sama dengan jumlah saluran data (= ukuran word)
- External
  - Dalam satuan block yg merupakan kelipatan word
- Addressable unit
  - Lokasi terkecil yang dpt dialamati secara unig
  - Secara internal biasanya sama dengan Word
  - Untuk disk digunakan satuan Cluster

# Metode Akses

- **Sekuensial**
  - Mulai dari awal sampai lokasi yang dituju
  - Waktu akses tergantung pada lokasi data dan lokasi sebelumnya
  - Contoh tape
- **Direct**
  - Setiap blocks memiliki address yg unique
  - Pengaksesan dengan cara lompat ke kisaran umum (general vicinity) ditambah pencarian sekuensial
  - Waktu akses tdk tergantung pada lokasi dan lokasi sebelumnya
  - contoh disk

# Metode Akses

- **Random**
  - Setiap lokasi memiliki alamat tertentu
  - Waktu akses tdk tergantung pada urutan akses sebelumnya
  - Contoh RAM
- **Associative**
  - Data dicari berdasarkan isinya bukan berdasarkan alamatnya
  - Waktu akses tdk tergantung terhadap lokasi atau pola akses sebelumnya
  - Contoh: cache

# Hierarki Memori

- Register
  - Dalam CPU
- Internal/Main memory
  - Bisa lebih dari satu level dengan adanya cache
  - “RAM”
- External memory
  - Penyimpan cadangan



# Performance

- Access time
  - Waktu untuk melakukan operasi baca-tulis
- Memory Cycle time
  - Diperlukan waktu tambahan untuk recovery sebelum akses berikutnya
  - Access time + recovery
- Transfer Rate
  - Kecepatan transfer data ke/dari unit memori

# Jenis Fisik

- Semiconductor
  - RAM
- Magnetic
  - Disk & Tape
- Optical
  - CD & DVD
- Others
  - Bubble
  - Hologram

# Karakteristik

- Decay
- Volatility
- Erasable
- Power consumption

## **Organisasi**

- Susunan fisik bit-bit untuk membentuk word

# Kendala Rancangan

- Berapa banyak?
  - Capacity
- Seberapa cepat?
  - Time is money
- Berapa mahal?

# Hierarki

- Registers
- L1 Cache
- L2 Cache
- Main memory
- Disk cache
- Disk
- Optical
- Tape

# Ingin Komputer yg Cepat?

- Komputer hanya menggunakan static RAM
- Akan sangat cepat
- Tidak diperlukan cache
  - Apa perlu cache untuk cache?
- Harga menjadi sangat mahal

# Locality of Reference

- Selama berlangsungnya eksekusi suatu program, referensi memori cenderung untuk mengelompok (cluster)
- Contoh: loops

# Memori Semiconductor

- RAM
  - Penamaan yang salah karena semua memori semiconductor adalah random access (termasuk ROM)
  - Read/Write
  - Volatile
  - Penyimpan sementara
  - Static atau dynamic



# Dynamic RAM

- Bit tersimpan berupa muatan dalam capacitor
- Muatan dapat bocor
- Perlu di-refresh
- Konstruksi sederhana
- Ukuran per bit nya kecil
- Murah
- Perlu refresh-circuits
- Lambat
- Main memory

# Static RAM

- Bit disimpan sebagai switches on/off
- Tidak ada kebocoran
- Tidak perlu refreshing
- Konstruksi lebih kompleks
- Ukuran per bit lebih besar
- Lebih mahal
- Tidak memerlukan refresh-circuits
- Lebih cepat
- Cache

# Read Only Memory (ROM)

- Menyimpan secara permanen
- Untuk
  - Microprogramming
  - Library subroutines
  - Systems programs (BIOS)
  - Function tables

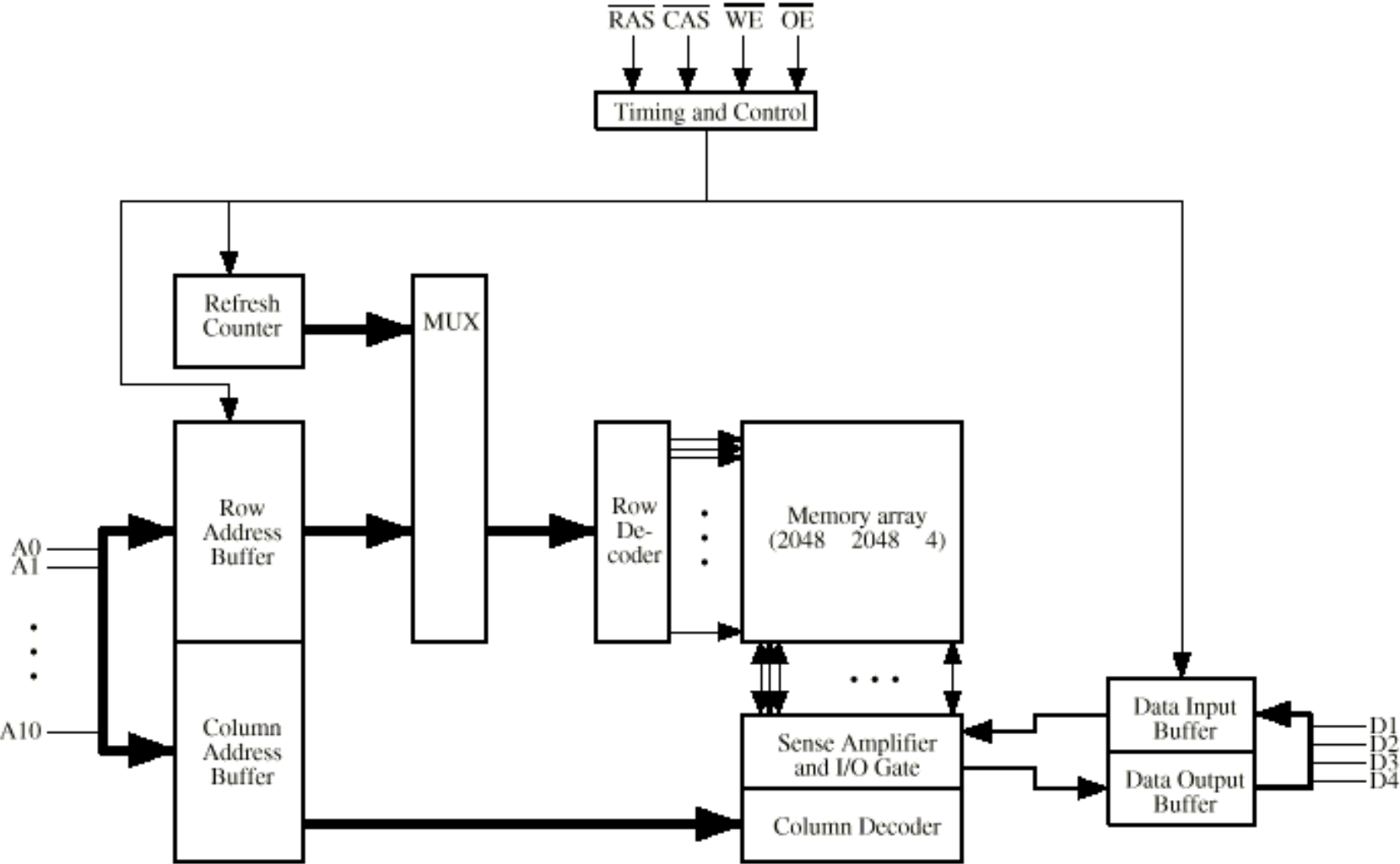
# Jenis ROM

- Ditulisi pada saat dibuat
  - Sangat mahal
- Programmable (once)
  - PROM
  - Diperlukan peralatan khusus untuk memprogram
- Read “mostly”
  - Erasable Programmable (EPROM)
    - Dihapus dg sinar UV
  - Electrically Erasable (EEPROM)
    - Perlu waktu lebih lama untuk menulisi
  - Flash memory
    - Menghapus seluruh memori secara electric

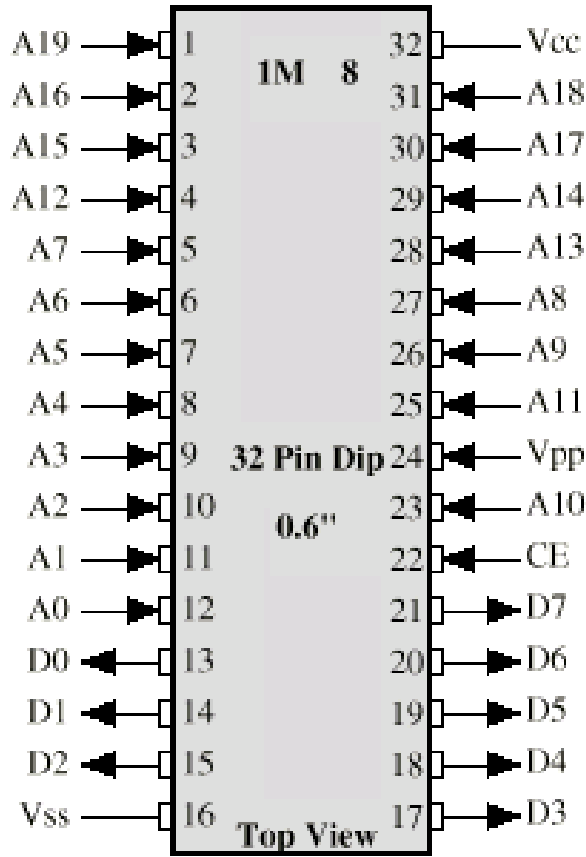
# Organisasi

- 16Mbit chip dapat disusun dari 1M x 16 bit word
- 1 bit/chip memiliki 16 lots dengan bit ke 1 dari setiap word berada pada chip 1
- 16Mbit chip dapat disusun dari array: 2048 x 2048 x 4bit
  - Mengurangi jumlah address pins
  - Multiplex row address dg column address
  - 11 pins untuk address ( $2^{11}=2048$ )
  - Menambah 1 pin kapasitas menjadi 4x

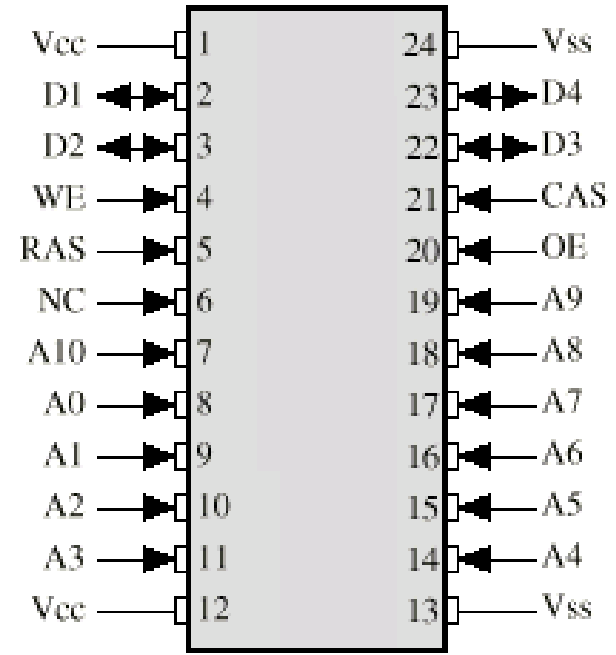
# Contoh: 16 Mb DRAM (4M x 4)



# Packaging



(a) 8 Mbit EPROM



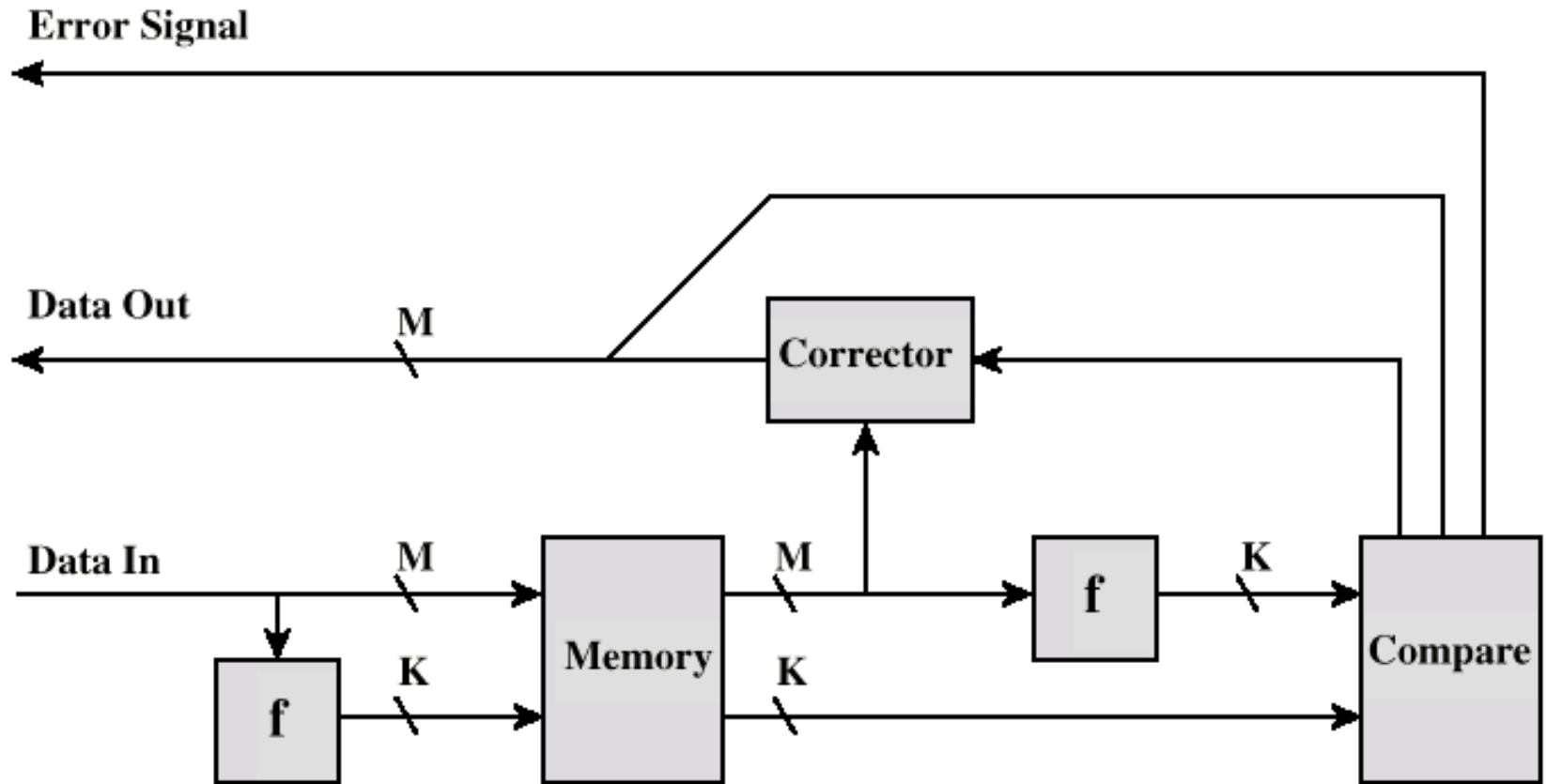
(b) 16 Mbit DRAM

# Koreksi kesalahan

- Rusak berat
  - Cacat/rusak Permanent
- Rusak ringan
  - Random, non-destructive
  - Rusak non permanent
- Dideteksi menggunakan Hamming code

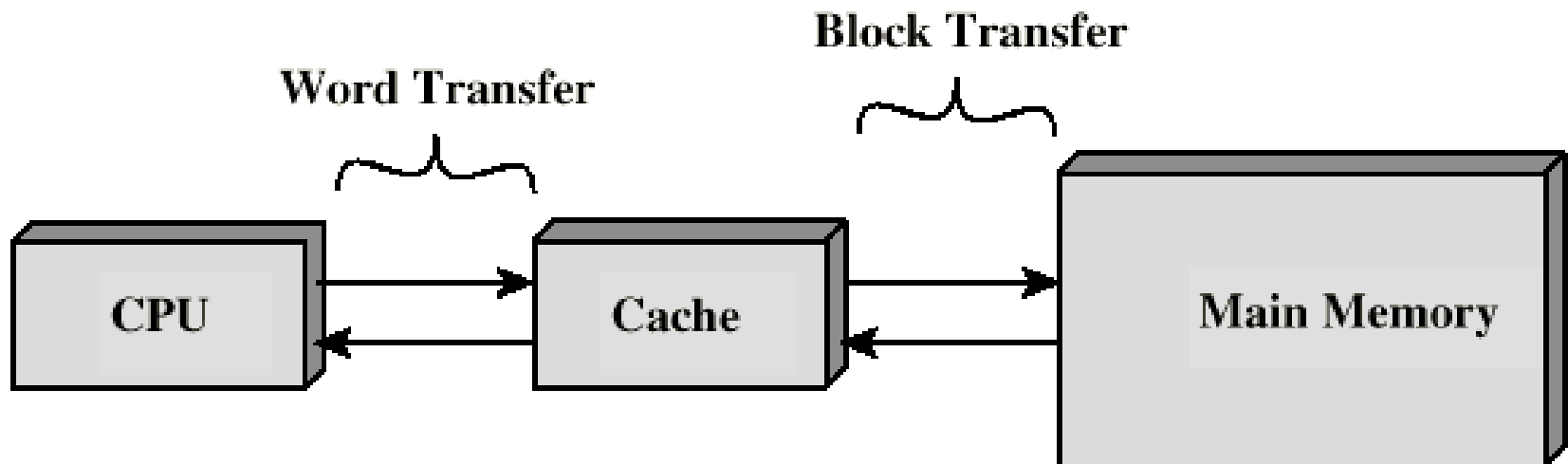


# Error Correcting Code Function



# Cache

- Memori cepat dg kapasitas yg sedikit
- Terletak antara main memory dengan CPU
- Bisa saja diletakkan dalam chip CPU atau module tersendiri



# Operasi pada Cache

- CPU meminta isi data dari lokasi memori tertentu
- Periksa data tersebut di cache
- Jika ada ambil dari cache (cepat)
- Jika tidak ada, baca 1 block data dari main memory ke cache
- Ambil dari cache ke CPU
- Cache bersisi tags untuk identitas block dari main memory yang berada di cache

# Operasi pada Cache

- CPU meminta isi data dari lokasi memori tertentu
- Periksa data tersebut di cache
- Jika ada ambil dari cache (cepat)
- Jika tidak ada, baca 1 block data dari main memory ke cache
- Ambil dari cache ke CPU
- Cache bersisi tags untuk identitas block dari main memory yang berada di cache

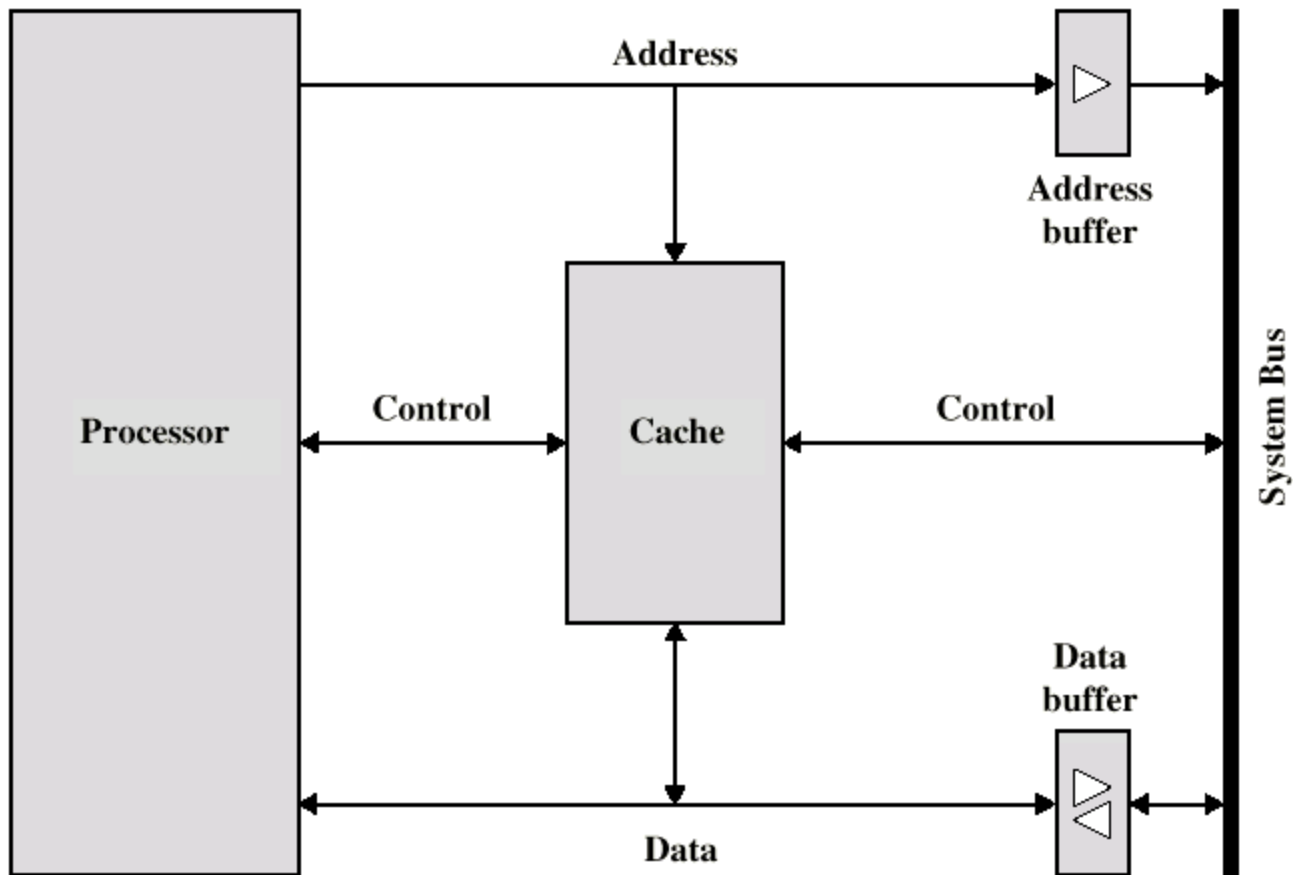
# Desain Cache

- Ukuran (size)
- Fungsi Mapping
- Algoritma penggantian (replacement algrthm)
- Cara penulisan (write policy)
- Ukuran Block
- Jumlah Cache

# Size

- Cost
  - Semakin besar semakin mahal
- Speed
  - Semakin besar semakin cepat
  - Check data di cache perlu waktu

# Organisasi Cache



# Fungsi Mapping

- Ukuran Cache 64kByte
- Ukuran block 4 bytes
  - diperlukan 16k ( $2^{14}$ ) alamat per alamat 4 bytes
  - Jumlah jalur alamat cache 14
- Main memory 16MBytes
- Jalur alamat perlu 24 bit
  - ( $2^{24}=16M$ )



# Direct Mapping

- Setiap block main memory dipetakan hanya ke satu jalur cache
  - Jika suatu block ada di cache, maka tempatnya sudah tertentu
- Address terbagi dalam 2 bagian
- LS-w-bit menunjukkan word tertentu
- MS-s-bit menentukan 1 blok memori
- MSB terbagi menjadi field jalur cache  $r$  dan tag sebesar  $s-r$  (most significant)

# Struktur Alamat Direct Mapping

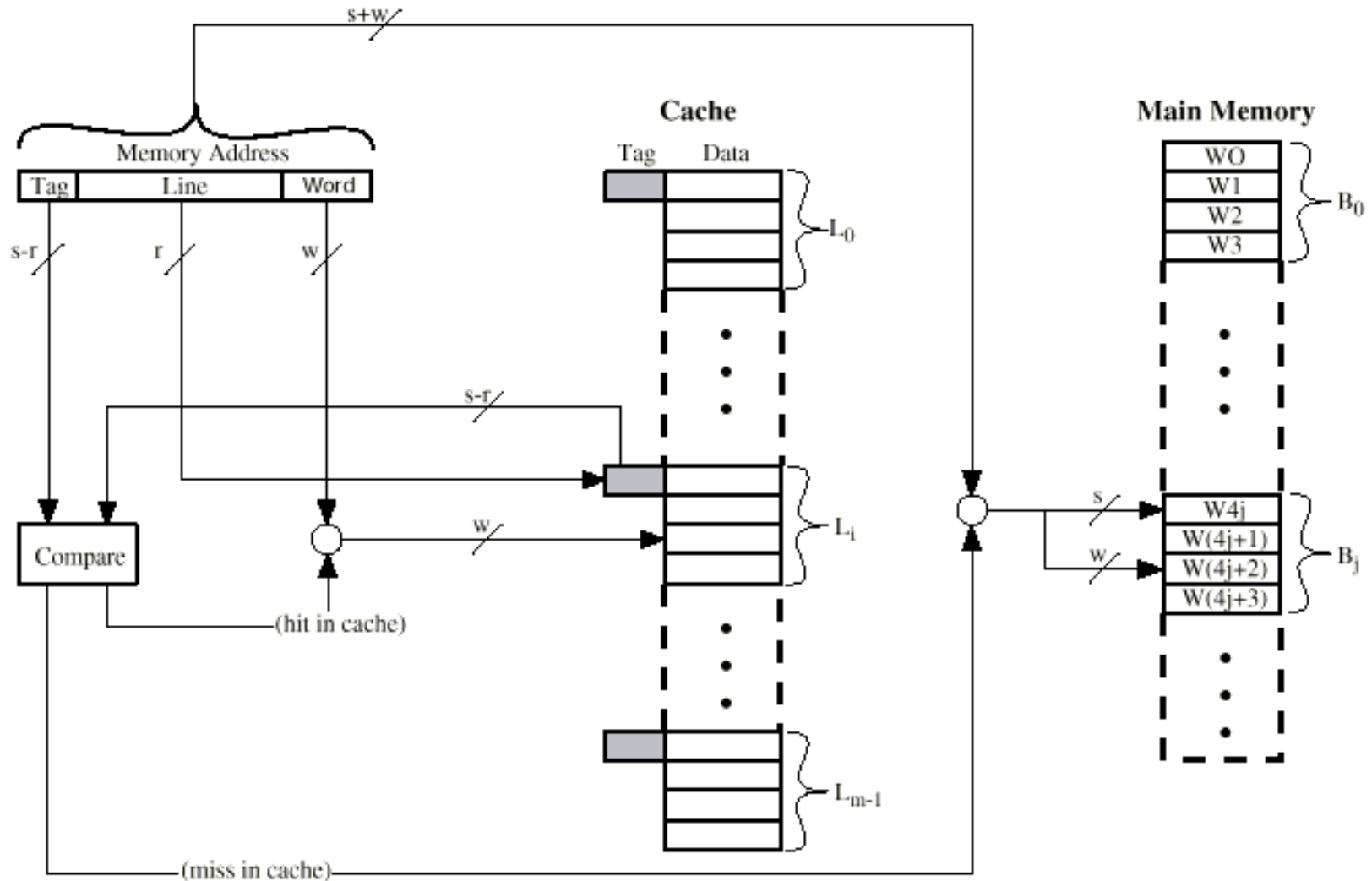
Tag s-r	Line or Slot r	Word w
8	14	2

- 24 bit address
- 2 bit : word identifier (4 byte block)
- 22 bit: block identifier
  - 8 bit tag (=22-14)
  - 14 bit slot atau line
- 2 blocks pada line yg sama tidak boleh memiliki tag yg sama
- Cek isi cache dengan mencari line dan Tag

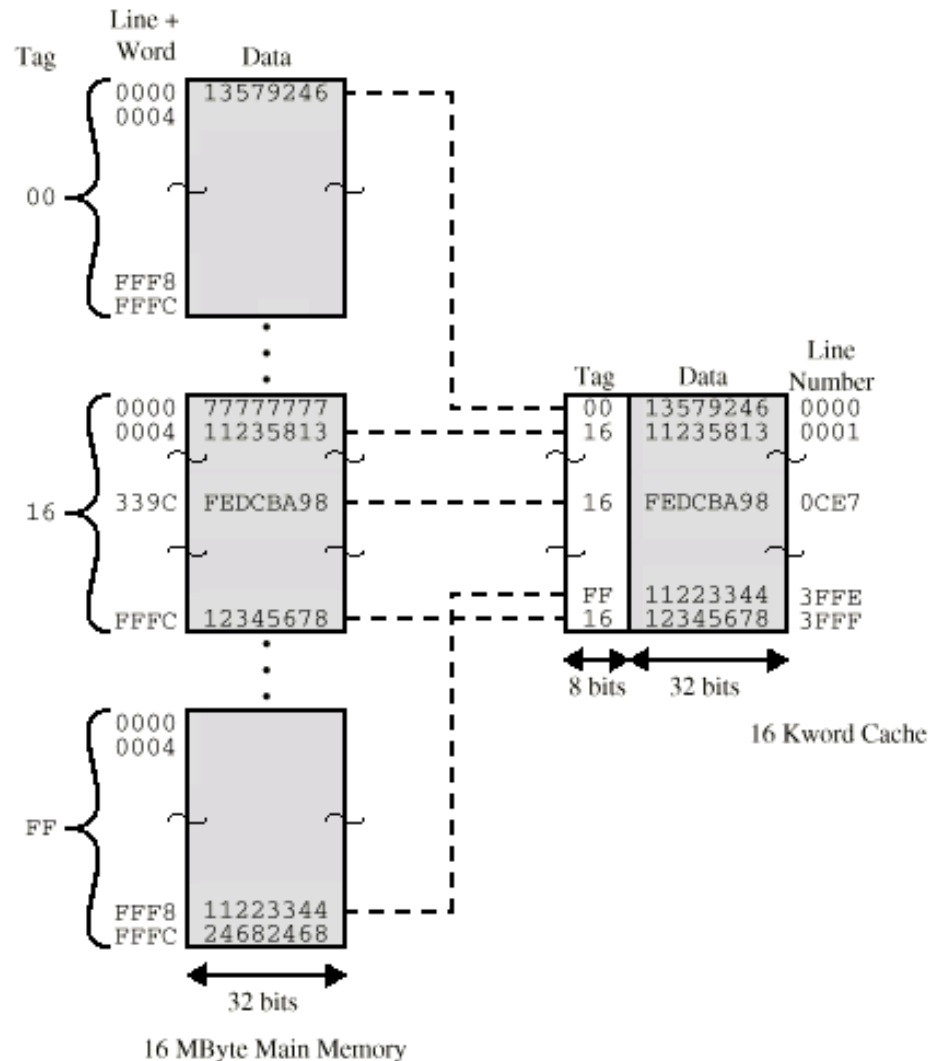
# Table Cache Line pada Direct Mapping

- Cache line                      blocks main memori
- 0                                      0, m, 2m, 3m... $2^s - m$
- 1                                      1, m+1, 2m+1... $2^s - m + 1$
  
- m-1                                  m-1, 2m-1, 3m-1... $2^s - 1$

# Organisai Cache Direct Mapping



# Contoh Direct Mapping



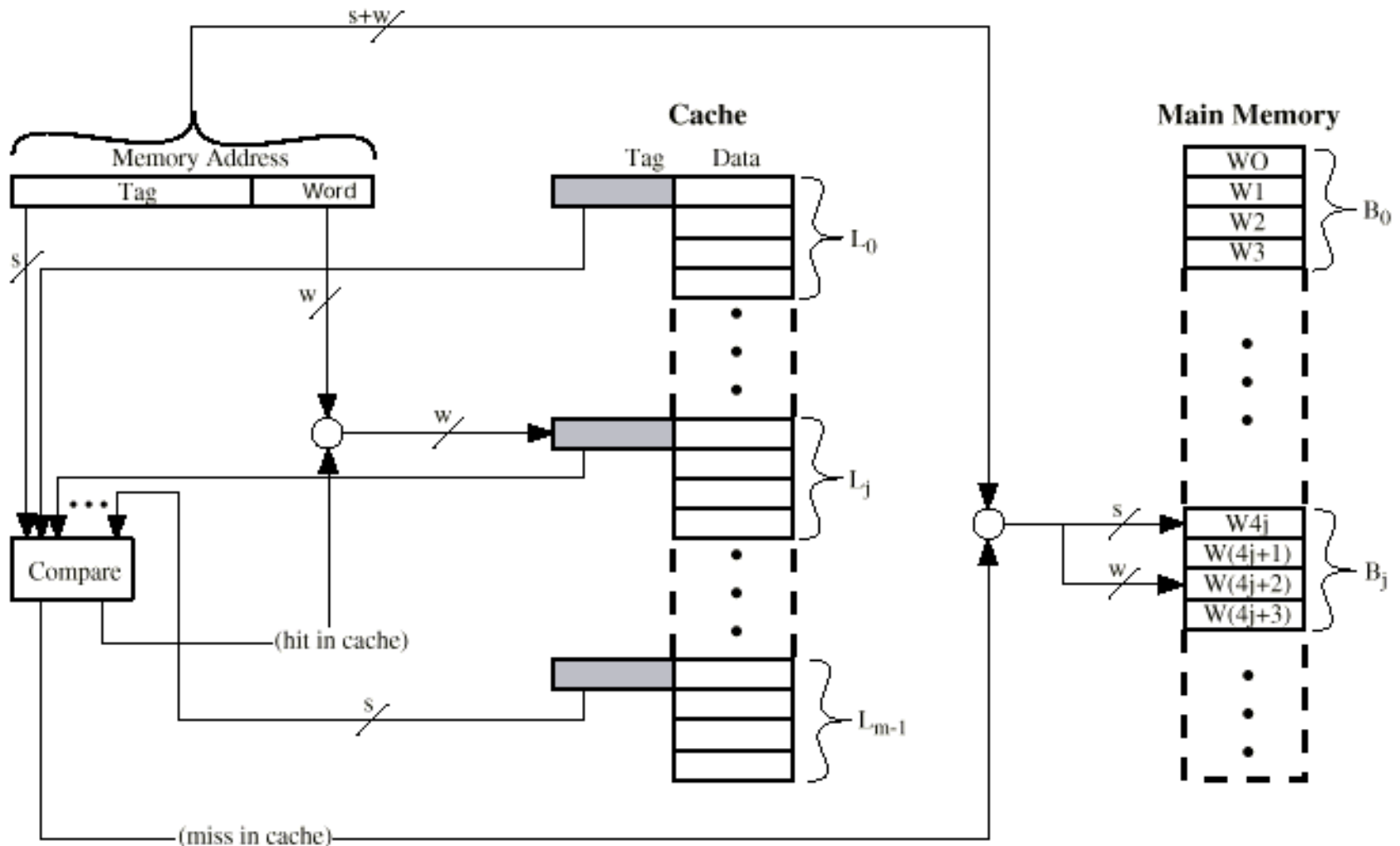
# Keuntungan & Kerugian Direct Mapping

- Sederhana
- Murah
- Suatu blok memiliki lokasi yang tetap
  - Jika program mengakses 2 block yang di map ke line yang sama secara berulang-ulang, maka cache-miss sanagat tinggi

# Associative Mapping

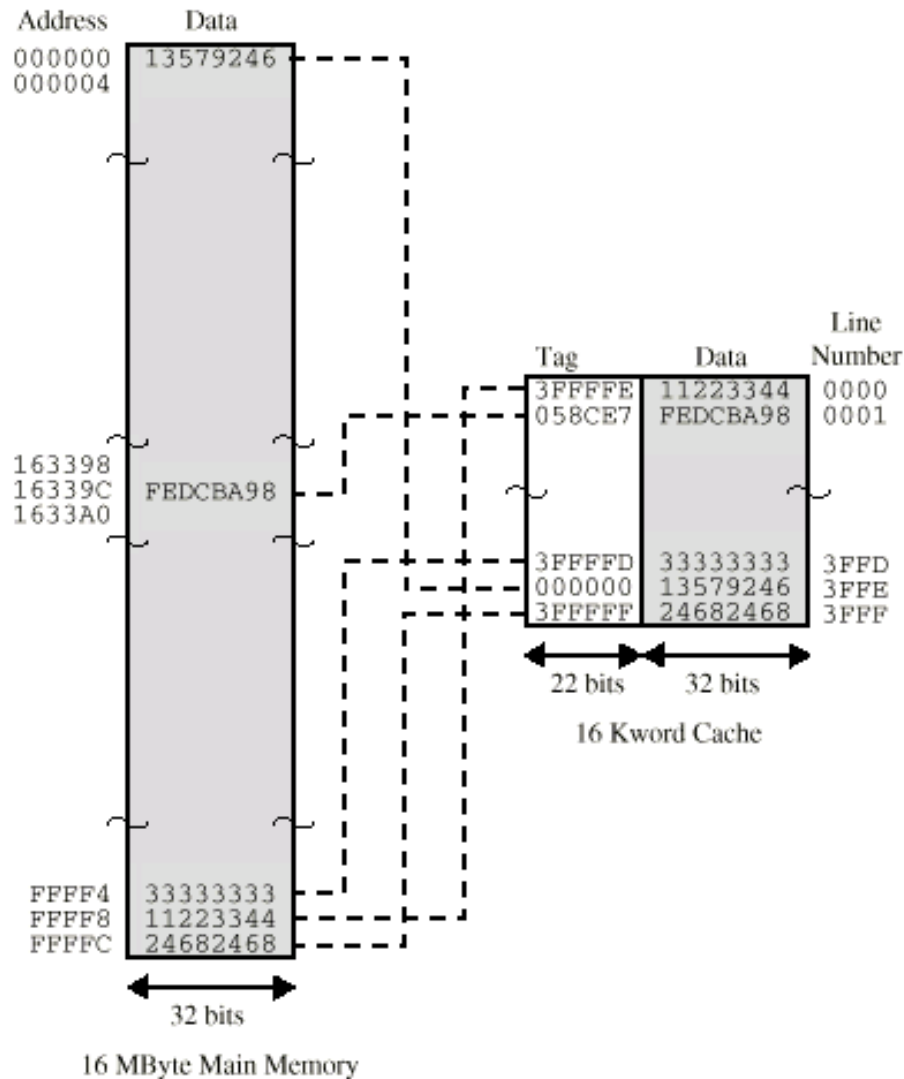
- Blok main memori dpt di simpan ke cache line mana saja
- Alamat Memori di interpretasi sbg tag dan word
- Tag menunjukkan identitas block memori
- Setiap baris tag dicari kecocokannya
- Pencarian data di Cache menjadi lama

# Organisasi Cache Fully Associative





# Contoh Associative Mapping



# Struktur Address Associative Mapping

Tag 22 bit	Word 2 bit
------------	---------------

- 22 bit tag disimpan untuk blok data 32 bit
- tag field dibandingkan dg tag entry dalam cache untuk pengecekan data
- LS 2 bits dari address menunjukkan 16 bit word yang diperlukan dari 32 bit data block
- contoh

– Address	Tag	Data	Cache line
– FFFFFC	FFFFFC	24682468	3FFF

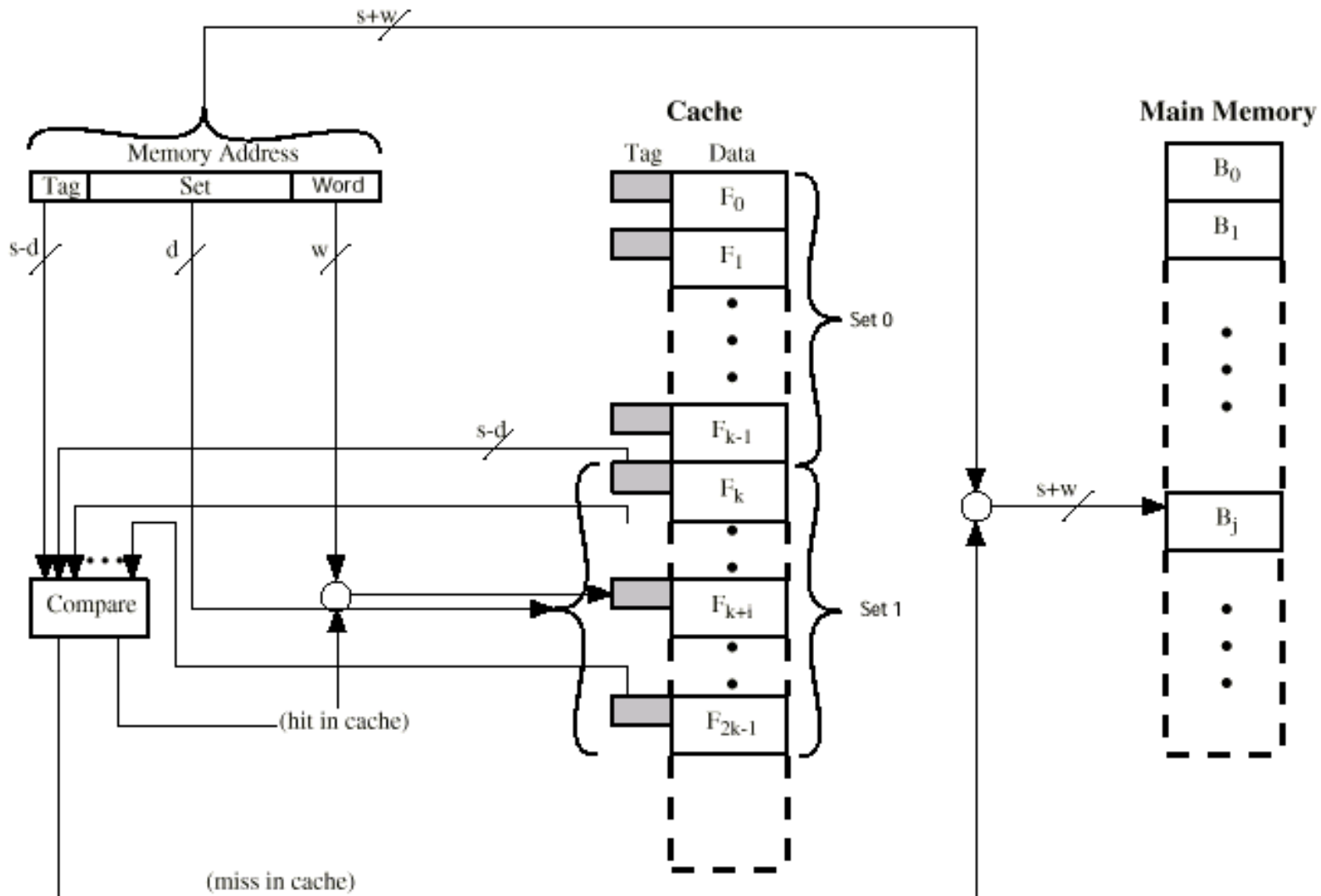
# Set Associative Mapping

- Cache dibagi dalam sejumlah sets
- Setiap set berisi sejumlah line
- Suatu blok di maps ke line mana saja dalam set
  - misalkan Block B dapat berada pada line mana saja dari set  $i$
- Contoh: per set ada 2 line
  - 2 way associative mapping
  - Suatu block dpt berada pada satu dari 2 lines dan hanya dalam 1 set

# Contoh Set Associative Mapping

- Nomor set 13 bit
- Nomor Block dlm main memori adl modulo  $2^{13}$
- 000000, 00A000, 00B000, 00C000 ... map ke set yang sama

# Organisasi Cache: Two Way Set Associative



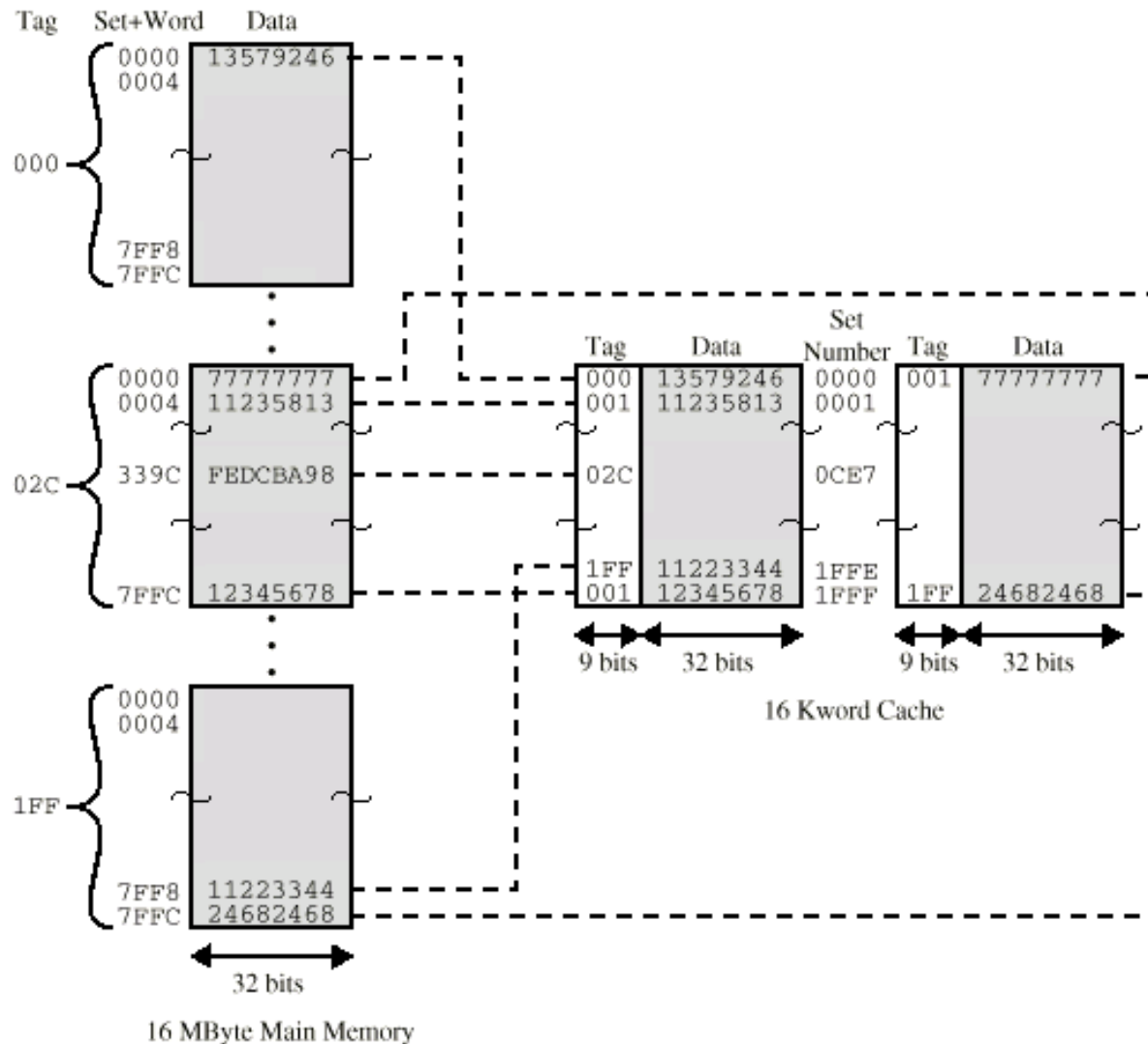
# Struktur Address: Set Associative Mapping

Tag 9 bit	Set 13 bit	Word 2 bit
-----------	------------	------------

- set field untuk menentukan set cache set yg dicari
- Bandingkan tag field untuk mencari datanya
- Contoh:

Address	Tag	Data	Set number
– 1FF 7FFC	1FF	12345678	1FFF
– 001 7FFC	001	11223344	1FFF

# Contoh Two Way Set Associative Mapping



# Replacement Algorithms (1)

## Direct mapping

- Tidak ada pilihan
- Setiap block hanya di map ke 1 line
- Ganti line tersebut



# Latihan Soal :

1. Jelaskan kapasitas untuk ukuran word dan banyaknya word?
2. Jelaskan maksud dari pengaksesan dengan cara lompat ke kisaran umum (general vicinity) ditambah pencarian sekuensial?
3. Jelaskan dan gambarkan macam-macam jenis fisik?
4. Apakah kendala rancangan dari capacity dan seberapa cepat?
5. Bagaimana cara agar komputer lebih cepat?