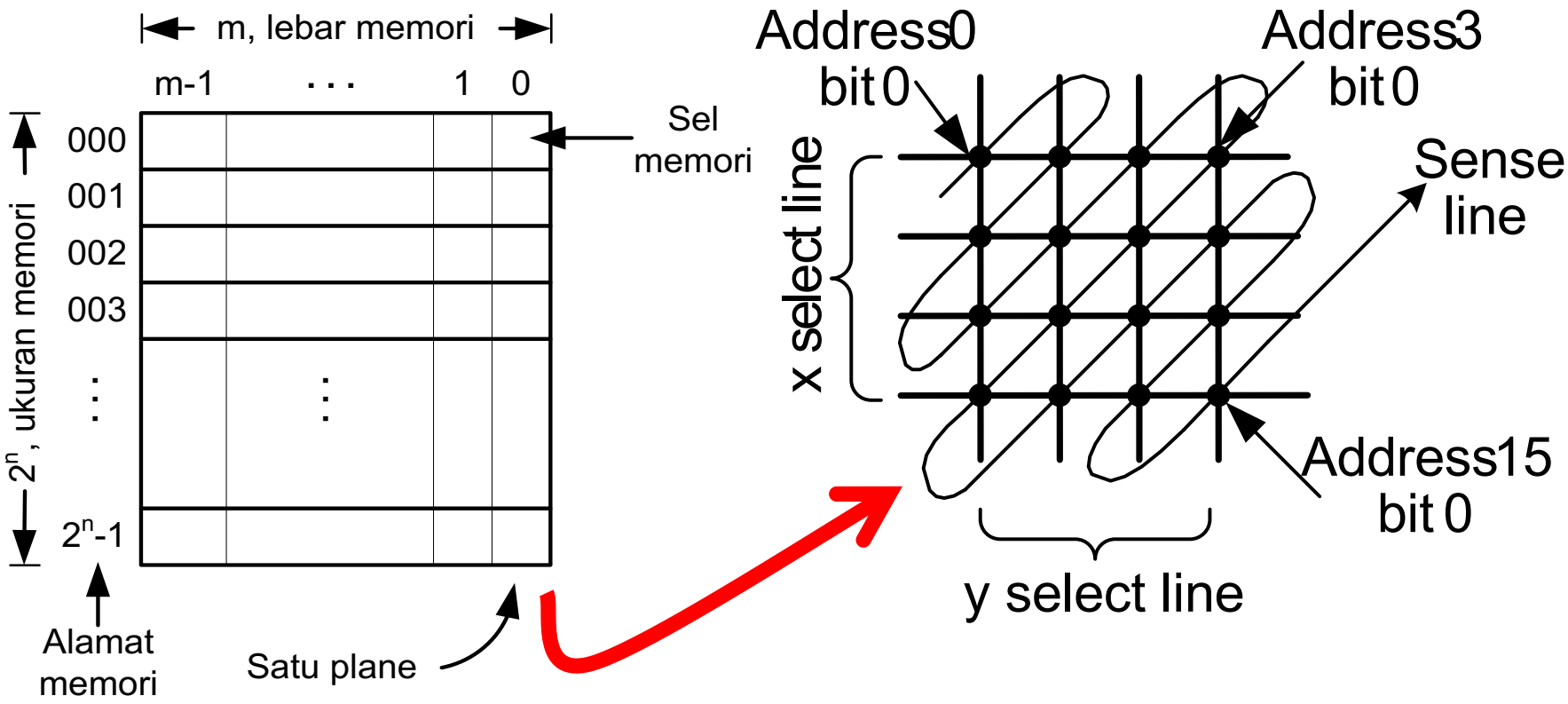
# Organisasi dan Arsitektur Komputer

## Pertemuan 3: Memori Utama

## Memori Controller

- MAR = memory address register
- MBR = memory buffer register (MDR = memory data register)
- WE = write enable; menulis dari MBR ke memori
- OE = output enable; menulis dari memori ke MBR

# Organisasi Memori (1)



# Organisasi Memori (2)

- Memori dapat digambarkan seperti sebuah matriks berukuran m kali n, dimana:
  - m = lebar memori = jumlah bit dalam satu alamat
    = *addressable unit*
  - n = ukuran memori = jumlah alamat
- Setiap bit pada setiap alamat yang mempunyai posisi yang sama disusun ke dalam sebuah *memory plane*
- *Memory plane* merupakan array 2 dimensi → dibutuhkan 2 buah *select line* (x dan y)
- Setiap satu *memory plane* terdiri dari **sebuah** *sensor line* → pada waktu diakses, dalam setiap *memory plane* **hanya satu bit** saja yang dibaca

# Organisasi Memori (3)

- Contoh soal:

  Sebuah memori terdiri dari 32 alamat dan setiap alamat terdiri dari 16 bit.

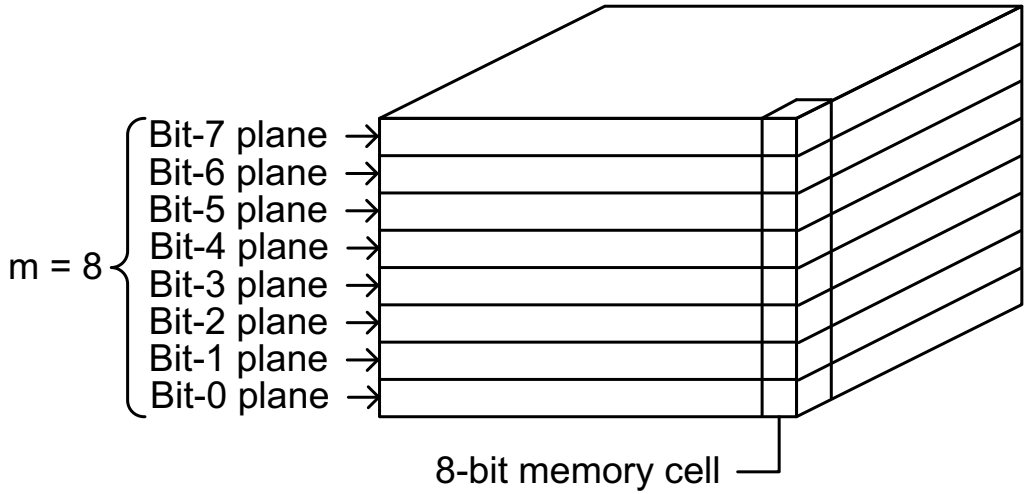
  - Berapakah jumlah jalur untuk setiap *select line pada memori plane ke-0 ?*
  - Berapakah jumlah *memory plane* yang diperlukan ?
  - Berapakah kapasitas memori ?

  Jawab:

  - Jumlah jalur setiap *select line* = 1 dan 32, 2 dan 16, 4 dan 8, atau 8 dan 4, atau 16 dan 2
  - Jumlah *memory plane* = jumlah bit setiap alamat = 16 *memory plane*
  - Kapasitas memori = 32 x 16 bit = 512 bit = 64 byte

# Organisasi Memori (4)

- *Memory bank*
  - Merupakan memori yang tersusun dari sejumlah *memory plane*
  - Sebuah *memory bank* tersusun dari beberapa chip
  - Pada gambar tersebut, berapakah:
    - Jumlah bit setiap alamat ?
    - Jumlah alamat total jika setiap select line = 8 ?



# Organisasi Memori (5)

- Contoh soal:

  Sebuah RAM berkapasitas 128 MB dipasang pada sebuah komputer dengan prosesor Intel dan menggunakan format instruksi yang terdiri dari 32 bit. Setiap satu *memory plane* dikemas ke dalam sebuah chip.

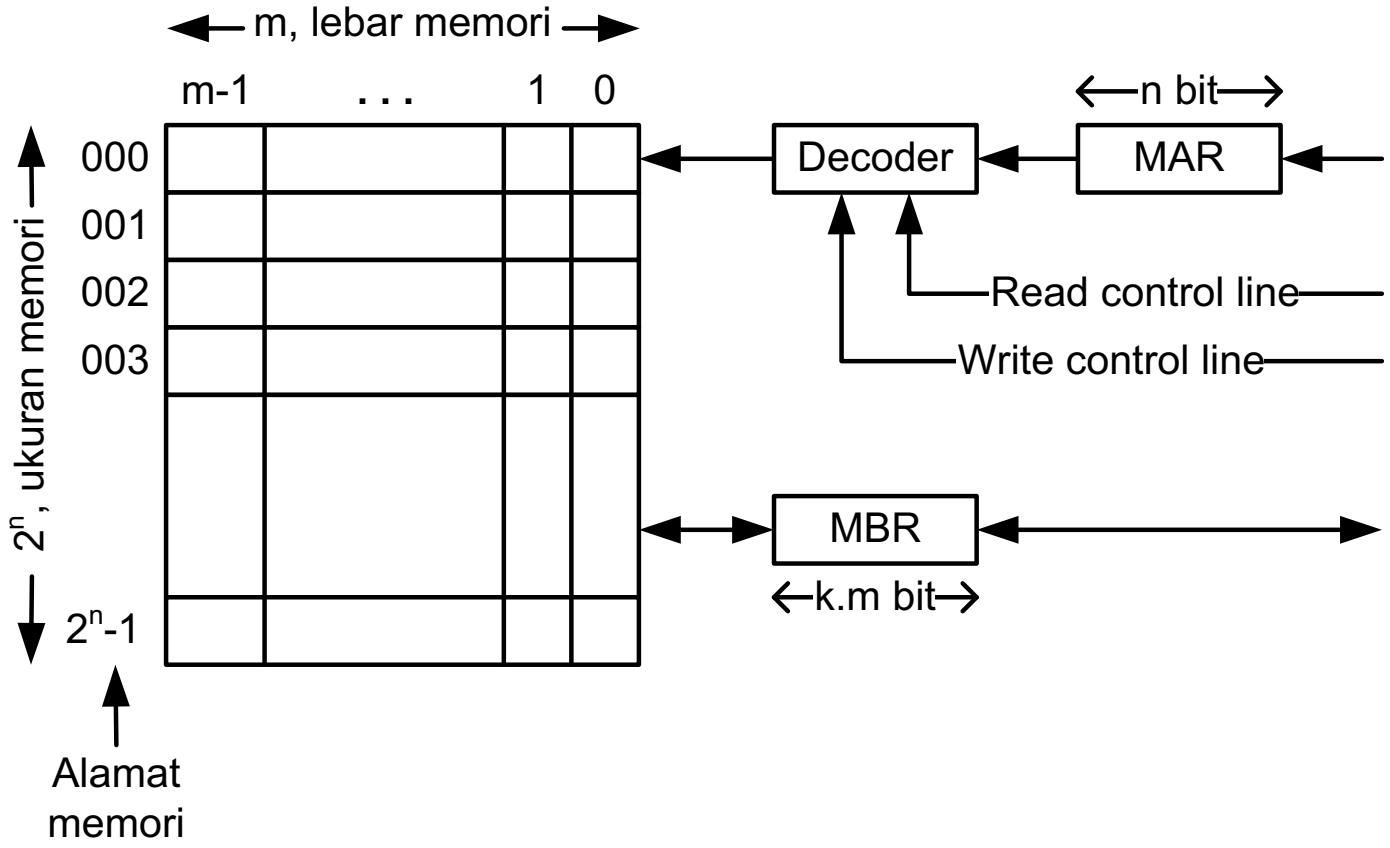
  - Berapakah jumlah chip dalam RAM tersebut *?*
  - Berapakah jumlah alamat yang tersedia ?
  - Berapakah jumlah jalur untuk *select line* pada memori plane ke-0 ?

  Jawab:

  - Jumlah bit dalam setiap alamat untuk komputer dengan prosesor Intel adalah 8 bit → diperlukan 8 buah *memory plane* → terdapat 8 chip
  - Jumlah alamat yang tersedia = kapasitas setiap *memory plane*/chip yaitu (128 x 8/8) M = 128 M alamat
  - Kapasitas setiap *memory plane*/chip = 128 MB / 8 chip = 16 MB = 128 Mb = $2^7$ x $2^{20}$ bit → jumlah jalur *select line* = $2^{27}$ , 2 dan $2^{26}$ , 4 dan $2^{25}$ jalur dst.

# Pengaksesan Memori (1)

- Bagaimana memori diakses ?



# Pengaksesan Memori (2)

- MAR (*Memory Address Register*):
  - Memuat alamat dari lokasi memori yang akan diakses (baca/tulis)
  - Jumlah bit MAR menentukan jumlah maksimum dari memori fisik yang dapat dipasang dalam suatu komputer
  - Jika MAR terdiri dari n bit → alamat memori yang valid adalah 0 hingga $2^n$ - 1
- MBR (*Memory Buffer Register*):
  - Memuat isi informasi yang akan dituliskan ke memori atau baru saja dibaca dari memori pada alamat yang ditunjukkan oleh isi MAR
  - MBR dapat berukuran m bit, 2m bit, 4m bit, dst dimana m = jumlah bit minimal dalam satu alamat (*minimum addressable unit* )
- *Memory Decoder*:
  - Untuk menerjemahkan alamat yang disimpan dalam MAR menjadi pasangan x dan y

# Pengaksesan Memori (3)

- Urutan **baca** dari memori:
  - Taruh alamat memori yang akan dibaca (dalam unsigned binary) ke MAR (range 0 hingga $2^n - 1$)
  - Kirim READ signal melalui READ control line
  - Decode isi MAR sehingga diperoleh nilai x dan y (nilai MAR tidak berubah)
  - Taruh isi alamat yang ditunjuk ke dalam MBR
- Urutan **tulis** ke memori:
  - Taruh alamat memori yang akan ditulisi (dalam unsigned binary) ke MAR (range 0 hingga $2^n - 1$)
  - Taruh data yang akan ditulis ke MBR
  - Kirim signal WRITE melalui WRITE control line
  - Decode isi MAR sehingga diperoleh nilai x dan y (nilai MAR tidak berubah)
  - Copy-kan isi MBR ke memori (isi MBR tidak berubah)

# Pen-decode-an Memori (1)

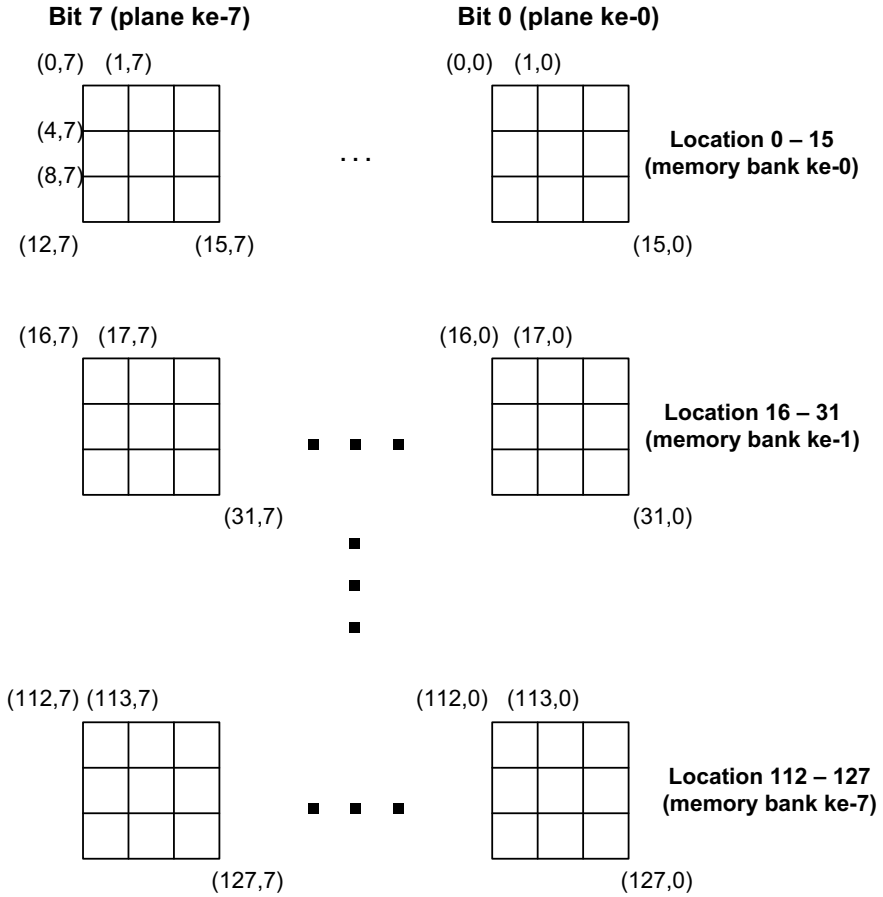
**Bagaimana cara men-*decode* alamat memori ?**

- Contoh:
  - Sebuah memori mempunyai 128 alamat (maksimum) yang setiap alamat terdiri dari 8 bit. Memori tersebut tersusun dari 8 *memory bank* , maka:
    - Daya tampung setiap *memory bank* = 128/8 = 16 alamat
    - 16 alamat = 4x4 alamat → diperlukan 4 jalur untuk setiap *select line*
    - 128 alamat = $2^7$ alamat → jumlah bit MAR = 7 bit
    - Memory bank / Alamat / Dalam biner

      | Memory bank | Alamat | Dalam biner |
      |---|---|---|
      | ke-0 | 0 – 15 | 0000000 – 0001111 |
      | ke-1 | 16 – 31 | 0010000 – 0011111 |
      | ... | ... | ... |
      | ke-7 | 112 – 127 | 1110000 – 1111111 |

      - Bit ke-4, 5, 6 nilainya **tetap** untuk setiap *memory bank* dan **berbeda** untuk *memory bank* yang berbeda → Bit ke-4, 5, 6 (warna merah) menunjukkan **nomor bank (*bank selector*)**

# Pen-decode-an Memori (2)




- **Distribusi lokasi memori sebanyak 128 alamat**

* (a,b) = alamat ke-a, posisi bit ke-b

# Pen-decode-an **Memori (3)**

**Bagaimana cara menentukan nilai x dan y ?**

- Alamat ke-0, 1, 2, dan 3 terletak pada **baris pertama** pada **plane** ke-0 dan pada **bank** ke-0 adalah sbb:

| Alamat | Dalam biner |
|---|---|
| 0 | 000 00 00 |
| 1 | 000 00 01 |
| 2 | 000 00 10 |
| 3 | 000 00 11 |

- Alamat ke-4, 5, 6, dan 7 terletak pada **baris kedua** pada **plane** ke-0 dan pada **bank** ke-0 adalah sbb:

| Alamat | Dalam biner |
|---|---|
| 4 | 000 01 00 |
| 5 | 000 01 01 |
| 6 | 000 01 10 |
| 7 | 000 01 11 |

- Dari 2 contoh di atas → bit ke-2 dan 3 (warna biru) menunjukkan nomor baris dalam sebuah plane = nilai dari x *select line*

# Pen-decode-an **Memori (4)**

- Alamat yang terletak pada **kolom pertama** pada **plane** ke-0 dan pada **bank** ke-0 adalah sbb:

| Alamat | Dalam biner |
|---|---|
| 0 | 000 00 00 |
| 4 | 000 01 00 |
| 8 | 000 10 00 |
| 12 | 000 11 00 |

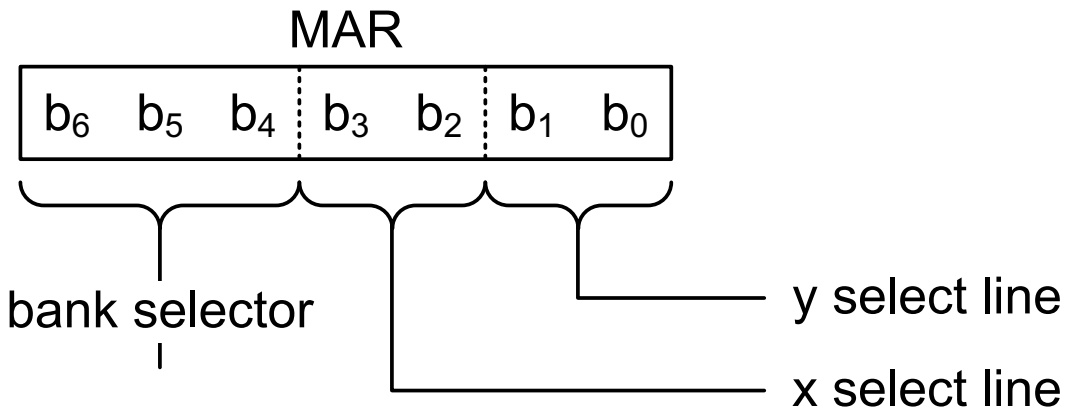
- Alamat yang terletak pada **kolom kedua** pada **plane** ke-0 dan pada **bank** ke-0 adalah sbb:

| Alamat | Dalam biner |
|---|---|
| 1 | 000 00 01 |
| 5 | 000 01 01 |
| 9 | 000 10 01 |
| 13 | 000 11 01 |

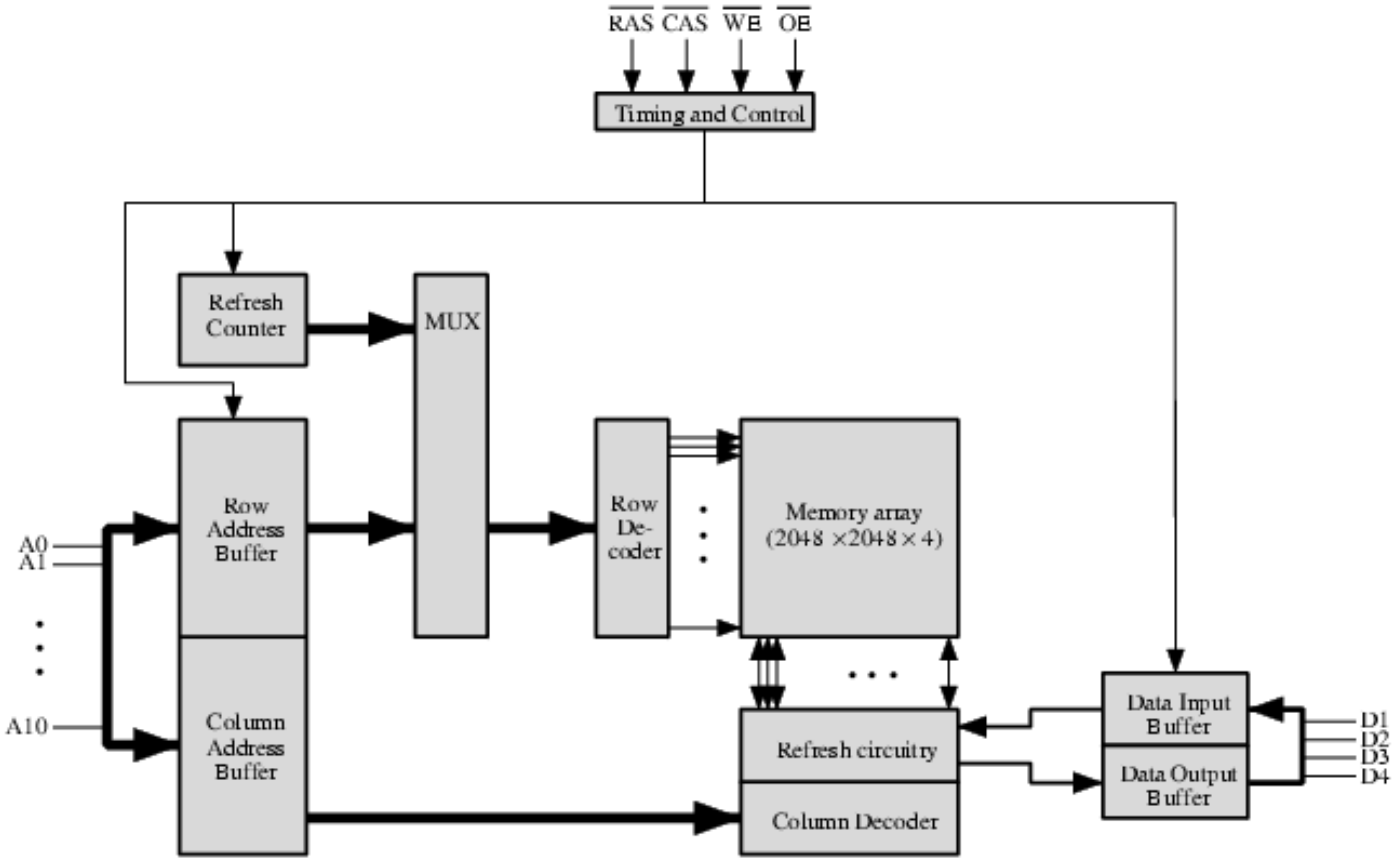
- Dari 2 contoh di atas → bit ke-0 dan 1 (warna biru) menunjukkan nomor kolom dalam sebuah plane = nilai dari y *select line*

# Pen-decode-an **Memori (5)**

Arti/fungsi setiap bit pada MAR adalah sbb:



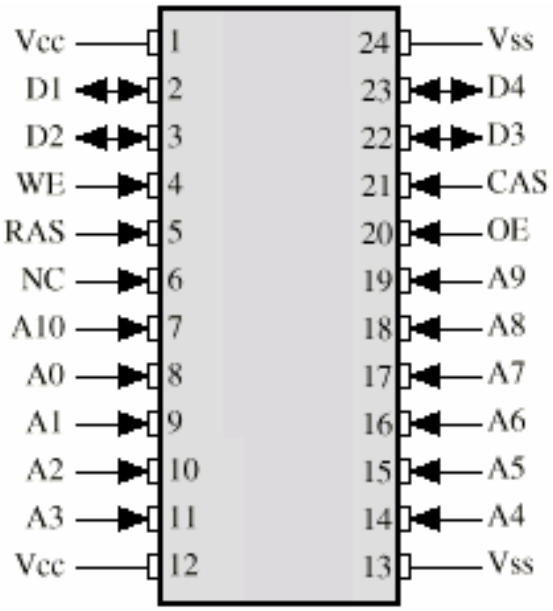
# Chip Logic DRAM 16 Mbit (4Mx4) (1)



# Chip Logic DRAM 16 Mbit (4Mx4) (2)

- Dalam satu saat dapat dibaca/ditulis data 4 bit (1 bit tiap memori plane)
- Terdiri dari 4 buah *array* berukuran 2048x2048
- Alamat tiap sel ditentukan oleh alamat baris dan kolom
- Pin alamat baris paralel dengan alamat kolom → hemat jumlah pin → digunakan RAS dan CAS
- Terdapat 4 pin untuk data *in (write)* dan *out (read)* secara paralel → digunakan WE dan OE
- *Refresh* dilakukan dengan membaca data dan menuliskannya kembali

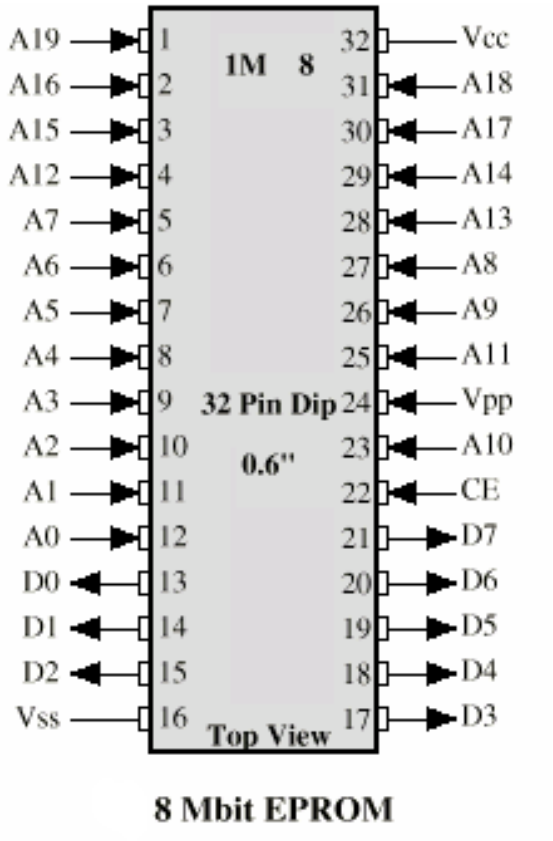
# Contoh *Chip Packaging* (2)




16 Mbit DRAM

- DRAM 16 Mbit:
  - Alamat = 11 pin = A0-A10
    Digunakan oleh baris dan kolom secara bergantian → setara dengan 2x11 pin = 22 pin = $2^{22}$ alamat = 4 M alamat
  - 1 alamat = 4 bit → kapasitas = 4 M x 4 bit = 16 bit
  - CAS untuk mengaktifkan kolom
  - RAS untuk mengaktifkan kolom
  - D0-D3 = data keluar (4 jalur)
  - OE = Output Enable (memungkinkan data dibaca)
  - WE = Write Enable = memungkinkan data disimpan ke memori

# Contoh *Chip Packaging* (1)

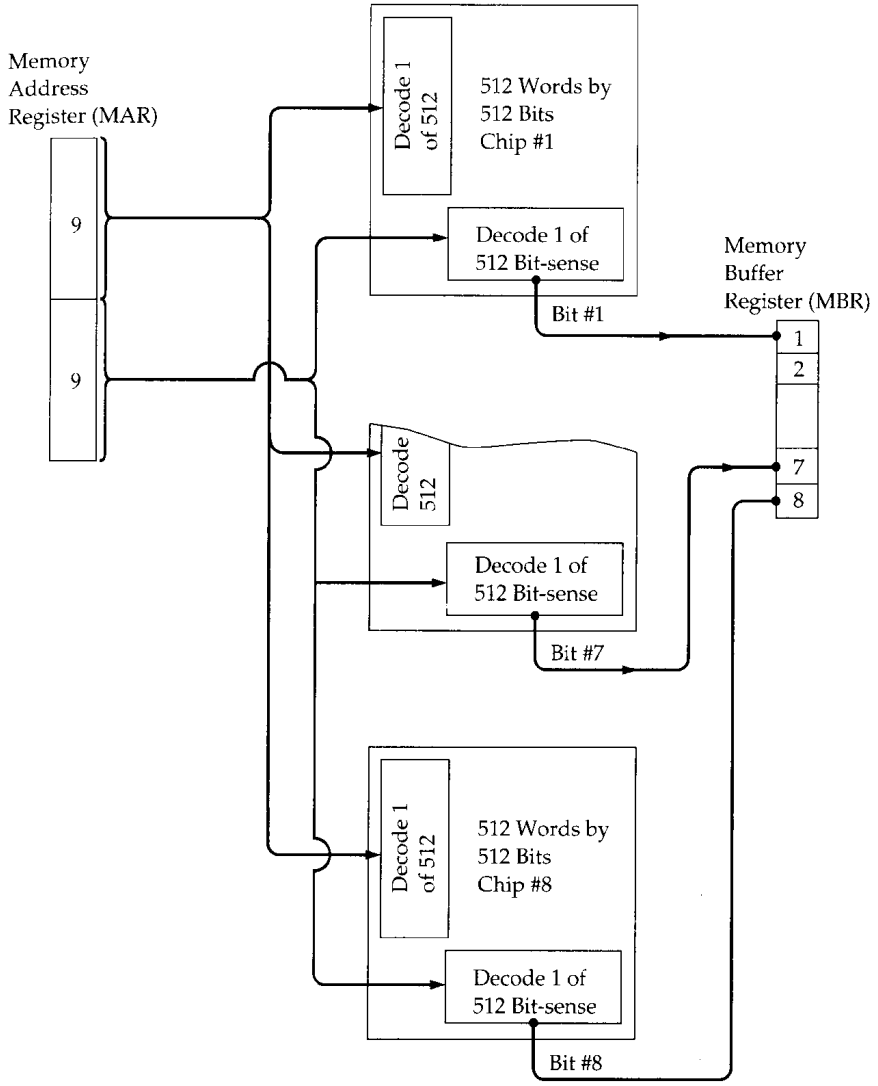



8 Mbit EPROM

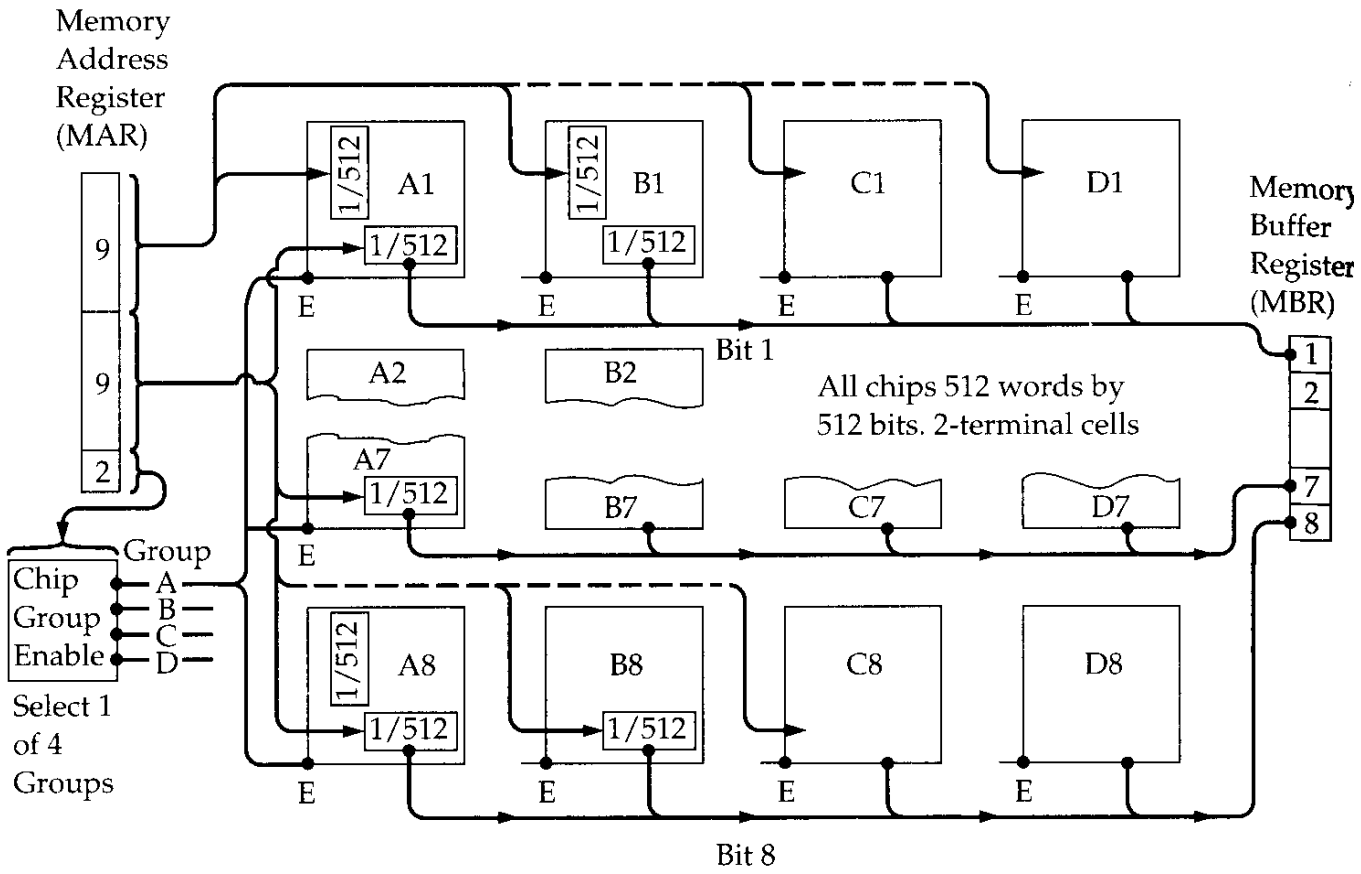
- EPROM 8 Mbit:
  - Alamat = 20 bit (pin) = A0-A19 = 1 M alamat
  - 1 alamat = 8 bit → kapasitas = 8 x 1 Mbit = 8 Mbit
  - *Chip Enable (CE)* untuk memilih chip yang aktif bila digunakan lebih dari satu chip
  - D0-D7 = data keluar (8 jalur)
  - Vss = ground
  - Vcc = tegangan positif

# Organisasi Modul (1)

- Organisasi memori 256 Kbyte
  - Terdiri dari 8 memori *plane* berukuran 512x512 bit
  - Alamat terdiri dari 18 bit (9 baris, 9 kolom)
  - Data terdiri dari 8 bit (1 byte)



# Organisasi Modul (2)



# Organisasi Modul (3)

- Organisasi memori 1 Mbyte
  - Terdiri dari 4 buah bank memori masing-masing berukuran 256 kbyte (4 kolom A, B, C, D)
  - Alamat terdiri dari 20 bit (2 bit MSB digunakan untuk memilih group chip A, B, C, atau D)

# Referensi


- Schneider, Michael G. 1985. *"The Principle of Computer Organization"*. 1st edition. John Wiley & Sons. Canada.
- Stalling, William. 2016. "*Computer Organization and Architecture: Designing for Performance*". 10th edition