Organisasi dan Arsitektur Komputer CII2A3

Organisasi Memori Utama

Tim Dosen COA

Fakultas Informatika
Universitas Telkom

Rencana Studi

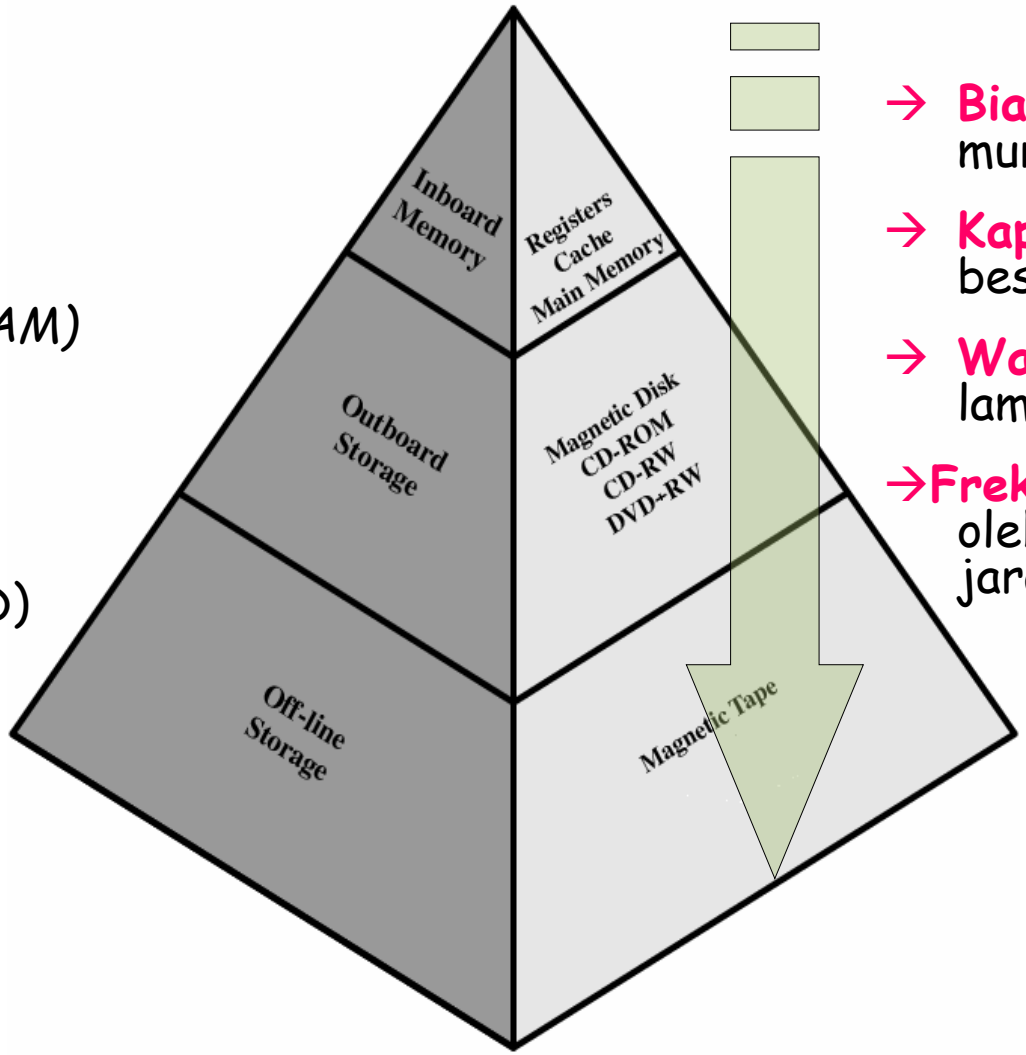
Fakultas Informatika
School of Computing
Telkom University

Rencana Studi		Rincian Nilai Kuis																Rincian Nilai Tugas									Rincian Nilai Hasil Proyek										Bobot Tiap CLO (%)			
Pertemuan Ke-	Materi	CLO 1			CLO 2					CLO 3				CLO 4				CLO 1			CLO 2					CLO	CLO 3						CLO 4									
		Kuis (Kognitif) (20 %)																Tugas Partisipatif (35%)									Hasil Proyek (45%)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	1a	1b	1c	2a	2b	2c	2d	2e	3a	3b	1a	1b	2a	1c	2b	2c	3a	3b	4a	3c	4b	3d	4c	1	2	3	4
1	Sistem komputer	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
2	Input/Output	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	14	--	--	--
3	Sistem Bus	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
4	Organisasi memori	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
5	Cara kerja memori utama (RAM)	-	-	-		1	-	-	-	-	-	-	-	-	-	-		-	-	-		4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
6	Memori sekunder	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	--	26	--	--
7	Cara kerja *cache memory* (bag-1)	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
8	Cara kerja cache memory (bag-2)	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
9	Arsitektur SAP-1	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-				
10	Arsitektur SAP-2	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	2	1	-	-	-	-	-	-	-	-	-	--	--	30	--
11	Arsitektur SAP-3	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-	-	-	7	2		-	-	-	-	-	-				
12	Instruksi *Extended* dan *Indirect*	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	-	-	-	-	-	-	-				
13	Arsitektur MIPS	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-				
14	Instruksi MIPS	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1	-	-	-	-	--	--	--	30
15	Assembly MIPS (bag-1)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2	-	-				
16	Assembly MIPS (bag-2)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	7				

Part 1: Hirarki dan karakteristik memori?

Hirarki Memori

- Registers
- L1 Cache
- L2 Cache
- Main memory (RAM)
- Disk cache
- Disk (Harddisk)
- Optical (CD, DVD)
- Magnetic tape



→ **Biaya** per bit makin murah

→ **Kapasitas** makin besar

→ **Waktu akses** makin lama

→**Frekuensi** diakses oleh prosesor makin jarang

Karakteristik Memori (1)

Memory diklasifikasikan berdasarkan:

(1) Lokasi
(2) Kapasitas
(3) Satuan transfer
(4) Cara akses
(5) Performansi
(6) Jenis fisik
(7) Karakteristik fisik
(8) Organisasi memori

Karakteristik Memori (2)

(1) Lokasi:

- *Internal*: dapat diakses oleh prosesor tanpa melalui I/O
  - *Register*
  - *Cache memory*
  - *Main memory (RAM)*
- *External*: untuk mengaksesnya harus melalui I/O
  - *Harddisk, Diskette, Magnetic Tape*
  - *Flashdisk*
  - CDROM, dll

(2) Kapasitas:

- ***Adalah kemampuan menampung data dalam satuan tertentu (byte atau word)***
- Satu byte = 8 bit
- Satu word = 8, 16, atau 32 bit (tergantung pada pembuat prosesor, Intel: satu word = 16 bit, IBM 370: 32 bit), MIPS ???

Karakteristik Memori (3)

(3) Satuan transfer:

- Memori internal:
  - ***Adalah banyaknya bit yang dapat dibaca/ditulis dari/ke memori dalam setiap detik***
  - Adalah **setara** dengan banyaknya jalur data yang terhubung ke memori (lebar bus)
  - Biasanya sebanyak satu word, tetapi dapat lebih banyak lagi (misal: 32, 64, atau 128 bit)
- Memori eksternal
  - Digunakan satuan block yang ukurannya lebih dari satu word
- Satuan alamat: (*addressable unit*)
  - ***Adalah ukuran memori terkecil yang dapat diberi alamat tersendiri***
  - Besarnya tergantung pembuat prosesor (Intel: 1 byte atau 8 bit), MIPS???
  - *Cluster* di *harddisk*

Karakteristik Memori (4)

(4) Cara akses:

- *Sequential access*
  - Akses ke memori dilakukan secara berurutan (*searching, passing, rejecting*)
  - Digunakan mekanisme *shared read/write*
  - Waktu akses sangat variabel, bergantung pada lokasi data yang akan dituju dan data sebelumnya
  - Contoh: *Magnetic tape*
- *Direct access*
  - Akses ke memori langsung menuju ke lokasi terdekat, diteruskan dengan sedikit pencarian dan perhitungan
  - Setiap blok/*record* mempunyai alamat unik berdasarkan lokasi fisik
  - Digunakan mekanisme *shared read/write*
  - Waktu aksesnya variabel (berbeda-beda) dan bergantung pada lokasi data yang akan dituju dan lokasi data sebelumnya
  - Contoh: *harddisk*

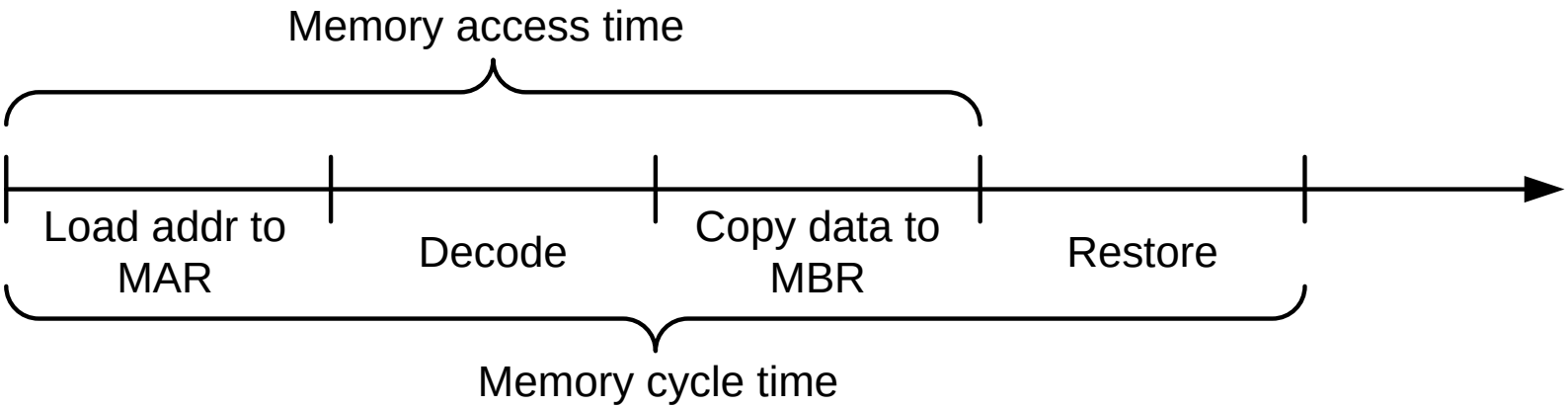
Karakteristik Memori (5)

- *Random access*
  - Akses ke memori dilakukan secara *random* langsung ke alamat yang dituju
  - Setiap alamat memori mempunyai alamat unik
  - Waktu aksesnya konstan dan tidak bergantung pada urutan akses sebelumnya
  - Contoh: *main memory*, beberapa sistem *cache*
- *Associative*
  - Pencarian data di memori dilakukan dengan membandingkan seluruh word secara bersamaan, tidak berdasarkan alamat
  - Waktu akses konstan dan tidak bergantung pada lokasi dan urutan akses sebelumnya
  - Contoh: *cache memory*

Karakteristik Memori (6)

(5) Performansi:

- Waktu akses (*latency*)
  - Waktu antara perintah akses (baca atau tulis) sampai didapatkannya data di MBR atau data dari MBR telah disalin ke lokasi memori tertentu
- Waktu siklus memori
  - Waktu dimulainya suatu operasi memori sampai memori siap melaksanakan operasi berikutnya (lebih penting)
  - Waktu akses + waktu untuk perubahan signal jalur data sebelum akses kedua



Karakteristik Memori (7)

– *Transfer rate*

- **Adalah waktu rata-rata perpindahan data**
- RAM: 1/waktu siklus
- Non-RAM:

$$T_N = T_A + N/R$$

T_N = Waktu rata-rata untuk baca/tulis sejumlah N bit

T_A = Rata-rata waktu akses

N = jumlah bit

R = transfer rate (bit per second)

Karakteristik Memori (8)

(6) Jenis fisik:

- Semikonduktor: RAM, flashdisk
- Magnetik: harddisk, *magnetic tape*
- Optik: CD, DVD

(7) Karakteristik fisik:

- *Volatile*: nilainya hilang bila tegangan listrik tidak ada → Semua internal memory???
- *Non-volatile*: nilainya TIDAK hilang (tetap ada) meskipun TIDAK ada tegangan listrik → Semua external memory???
- *Erasable*: nilainya dapat dihapus (semikonduktor, magnetik)
- *Non-erasable*: nilainya tidak dapat dihapus (ROM)

(8) Organisasi memori:

- Penyusunan bit untuk membentuk word

Part 2: Bagaimana cara kerja sel memori?

Sel Memori (1)

- Setiap memori tersusun dari rangkaian sel-sel memori:

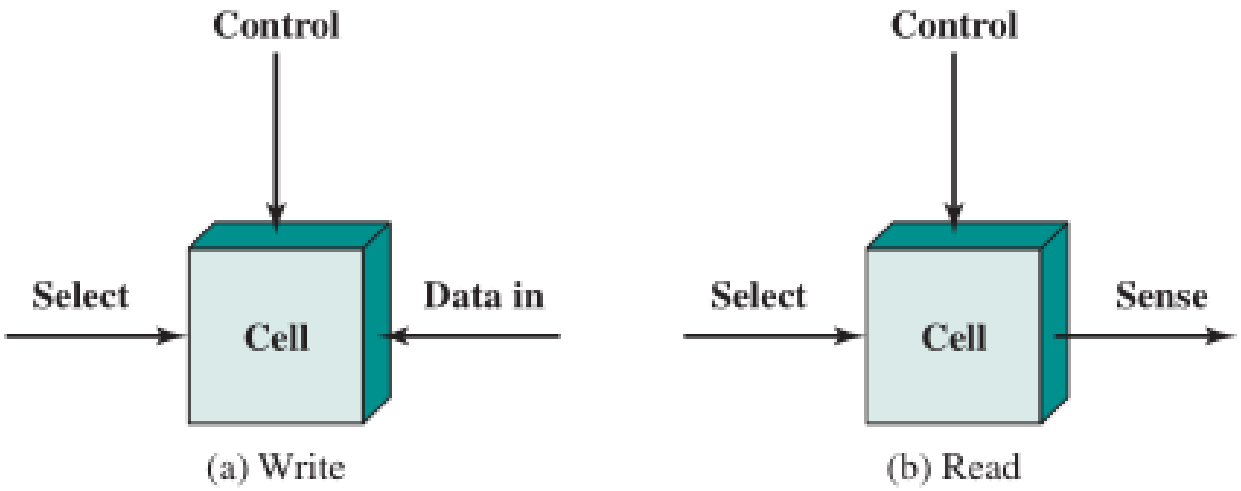


Figure 5.1 Memory Cell Operation

Sel Memori (2)

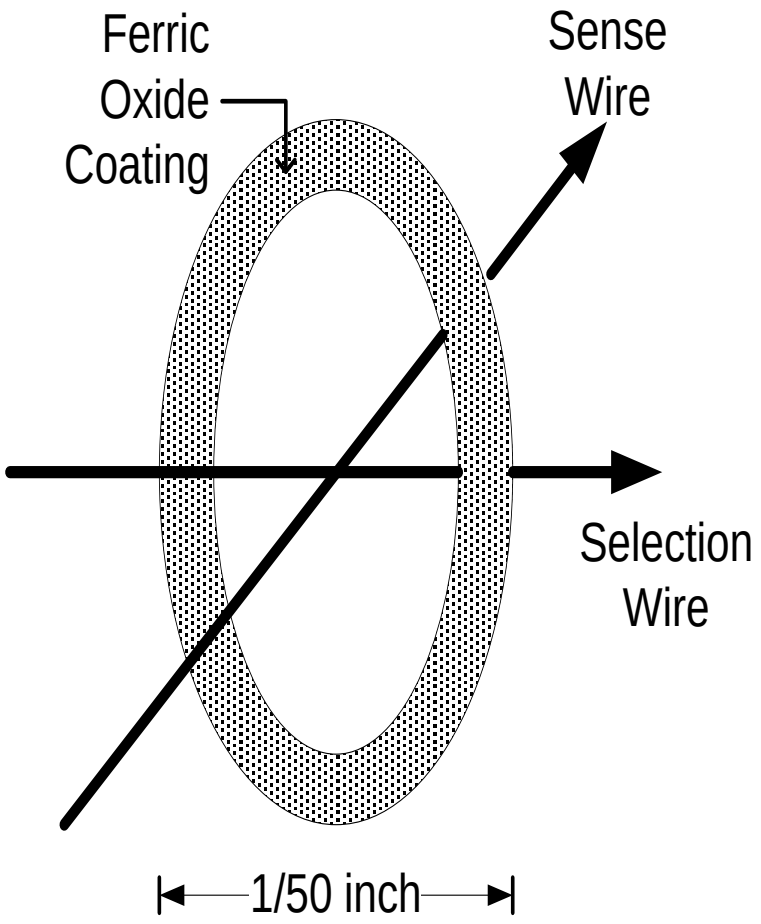
- Sifat-sifat sel memori:
  - Dapat memberikan 2 kondisi (1 atau 0)
  - Dapat ditulisi (minimal satu kali) dengan cara mengubah kondisi sel memori
  - Dapat dibaca
- Tiap sel terdiri dari 3 terminal:
  - Select: untuk memilih sel memori yang akan dibaca/ditulisi
  - Control: untuk menentukan jenis operasi *write/read*
  - Data:
    - Write: untuk mengubah kondisi sel dari 1 ke 0 atau sebaliknya
    - Read: untuk membaca kondisi sel memori

Sel Memori (3)

- Apa syarat *device* yang dapat digunakan untuk menyimpan data biner?
  - Hanya dapat menyimpan 2 macam nilai/kondisi (*true-false* atau 1-0) yang stabil
  - Mempunyai pembatas yang lebar yang dapat memisahkan kedua nilai/kondisi secara tegas
  - Mampu menangani perubahan nilai/kondisi dalam waktu yang tidak terbatas
  - Nilai/kondisinya tidak rusak pada saat dibaca

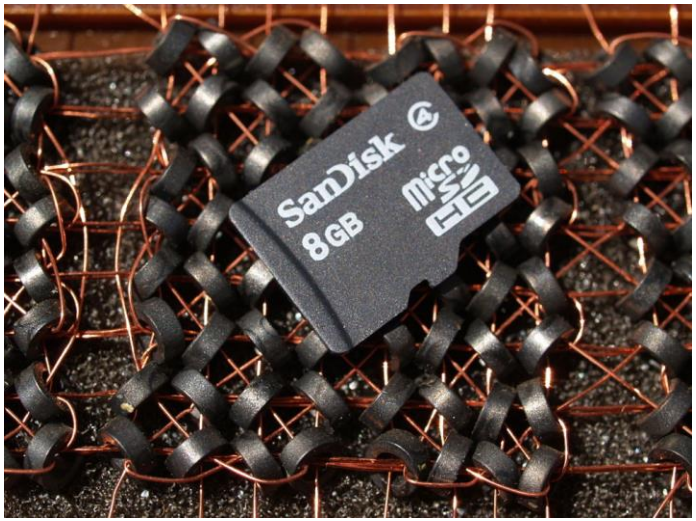
  Apakah saklar/switch memenuhi syarat???
- Implementasinya dengan apa?
  - *Magnetic core*
  - Semikonduktor:
    - Gabungan antara transistor dan kapasitor
    - Gabungan beberapa transistor

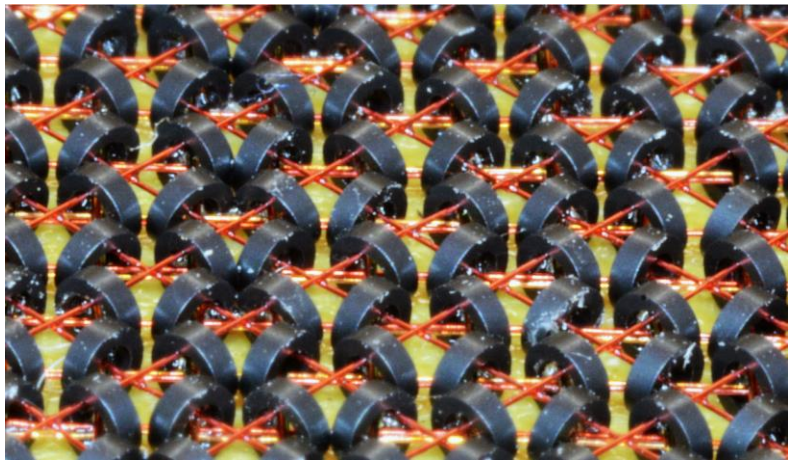
Magnetic core (1)

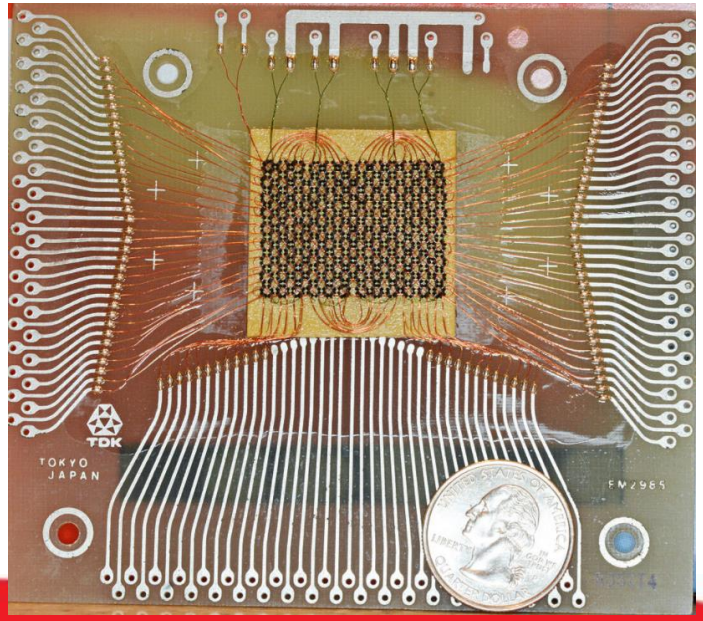


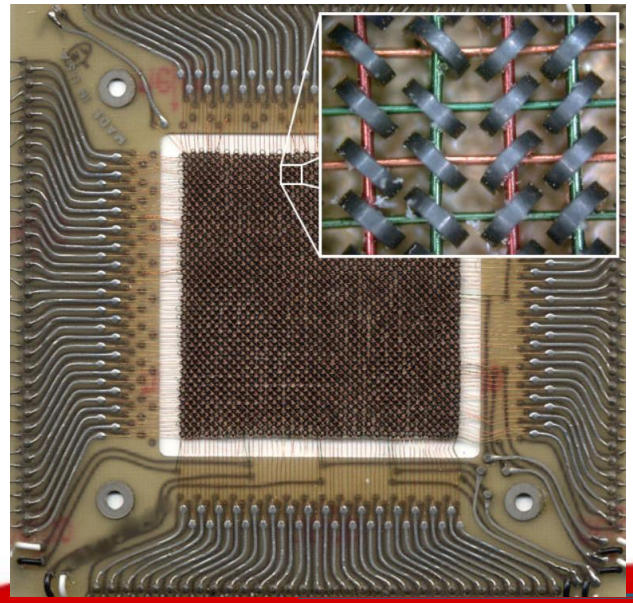
- *Ferric oxide coating*
  - Bahan yang mudah bersifat magnet
- *Selection wire:*
  - Untuk menentukan nilai medan magnet
  - Jika arah arus berbeda → medan magnet berbeda
- *Sense wire:*
  - Untuk membaca nilai bit yang disimpan
- *Cara kerja:*
  - Simpan data (write):
    - » Aliri *select wire* dengan arah arus sesuai dengan nilai bit data yang diinginkan

Magnetic core (2)







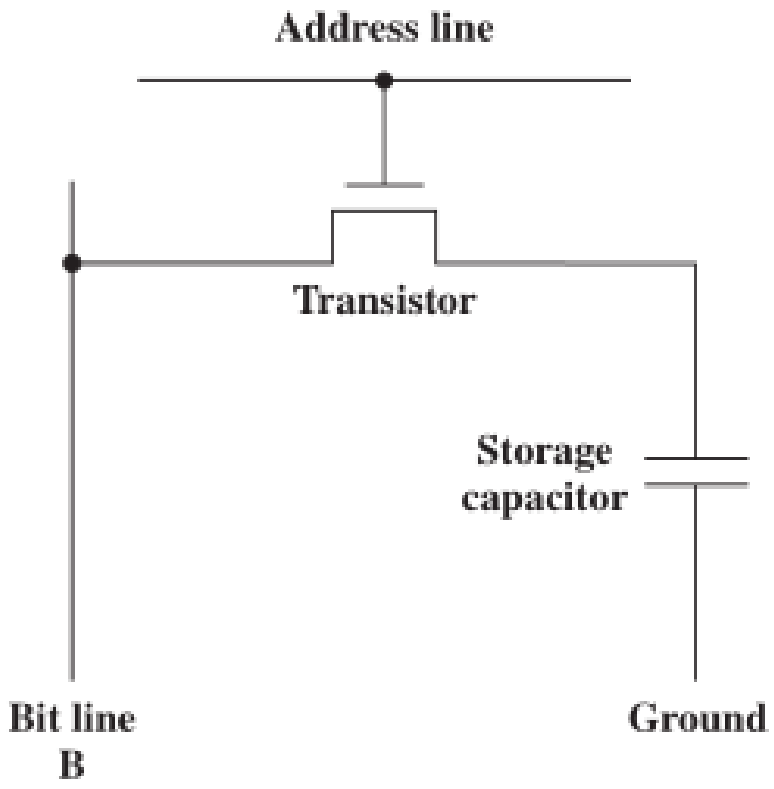


Magnetic Core (3)

- Cara kerja: (cont'd)
  - Baca data (*read*):
    - » *Selection wire* dialiri arus yang menghasilkan bit bernilai 0
    - » Jika bit yang sedang disimpan:
      - = 0 → Tidak ada tegangan induksi pada *sense wire*
        - → Bit bernilai 0
      - = 1 → Medan magnet berubah
        - → Data yang disimpan berubah dari 1 menjadi 0
        - → *Sense wire* terinduksi berarti data yang sedang disimpan bernilai 1
        - → Kembalikan nilai bit ke nilai semula (1) dengan cara aliri *select wire* dengan arus yang menghasilkan bit bernilai 1

Sel Memori Dinamis (1)

– Gabungan antara transistor dan kapasitor



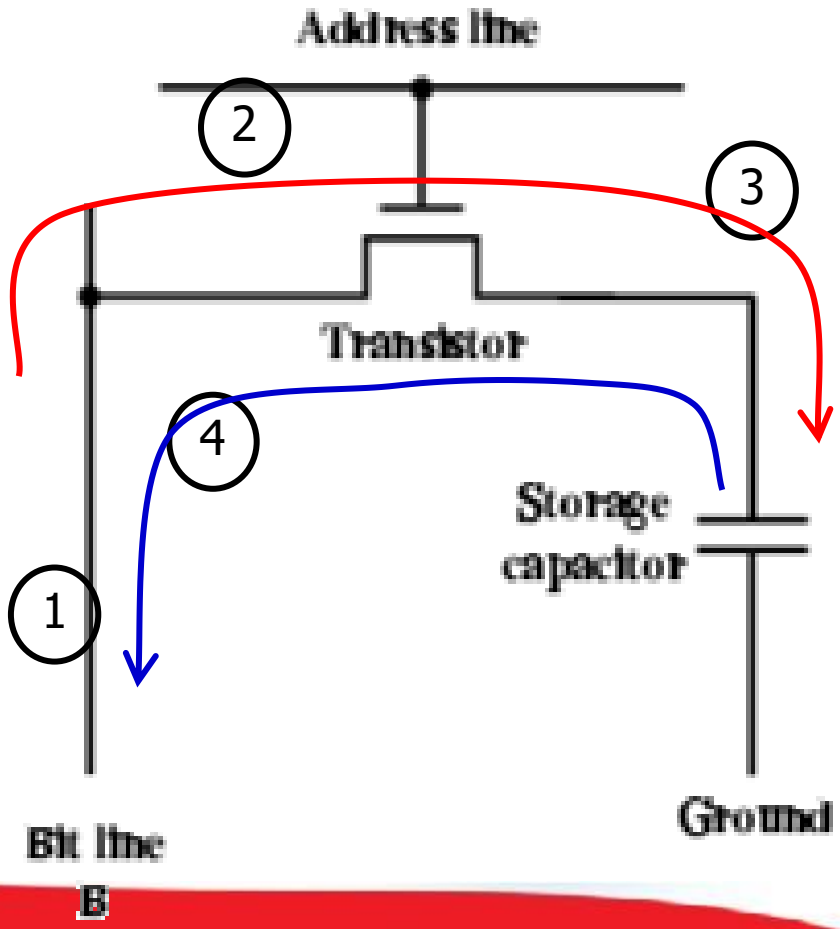
Capasitor:

- Untuk menyimpan nilai data

- *Address line*:
  - Untuk mengaktifkan sel memori (transistor)
- *Bit line*:
  - Untuk membaca/menulis data dari/ke sel memori

Sel Memori Dinamis (3)

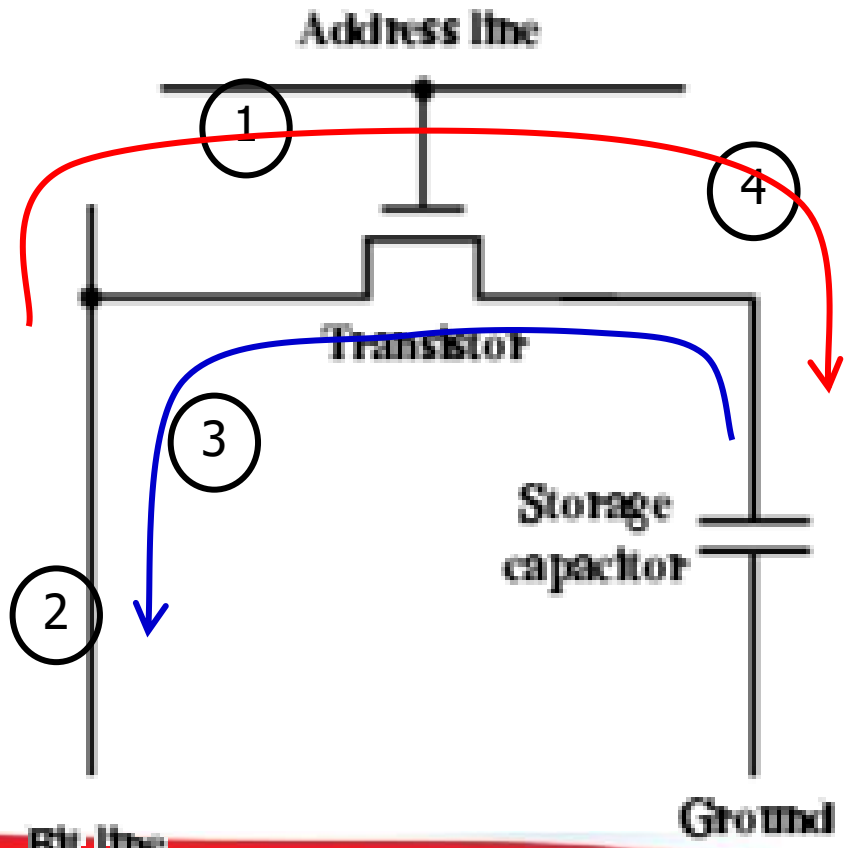
– Gabungan antara transistor dan kapasitor (memori dinamis) (cont'd)



- Cara kerja
  - Simpan data (*write*):
    1. Signal yang akan ditulis dihubungkan ke *bit line* (high voltage = 1, low voltage = 0)
    2. Signal diberikan ke *address line* → transistor on
    3. Untuk simpan bit 1 → arus mengalir masuk **ke** kapasitor →kapasitor diisi muatan
    4. Untuk simpan bit 0 → arus mengalir **dari** ke kapasitor → muatan kapasitor dikosongkan

Sel Memori Dinamis (4)

– Cara kerja: (cont'd)



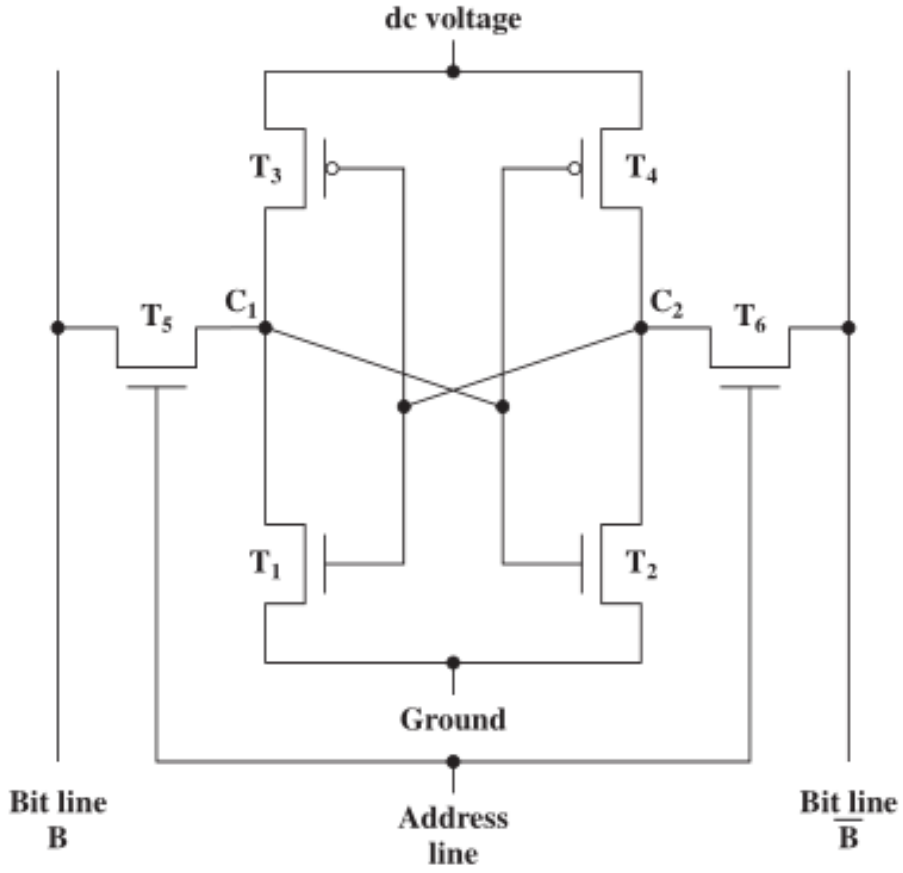
– Baca data (*read*):

1. *Address line* diberi signal → transistor on
2. Muatan kapasitor dialirkan ke *bit line/sense amplifier*
3. *Sense amplifier* membandingkan muatan kapasitor dengan tegangan reference untuk menentukan apakah data bernilai 1 atau 0
4. Muatan kapasitor berkurang → perlu di-*refresh* → disebut ***memori dinamis***

Sel Memori Statis (2)

Fakultas Informatika
School of Computing
Telkom University

– Gabungan antara beberapa transistor (memori statis)



(b) Static RAM (SRAM) cell

- Status T1 selalu berlawanan dengan T3, tetapi selalu sama dengan T4
- Status T2 selalu berlawanan dengan T4, tetapi selalu sama dengan T3
- Nilai bit line $\bar{B}$ berlawanan dengan nilai bit line B
- Signal ***high***: (pd bit line B)
  - T1=off, T3=on → titik C1=high
    - → bit line B = **HIGH**
  - T2=on, T4=off → titik C2 = LOW
- Signal ***low***:
  - T1=on, T3=off → titik C1=low
    - → bit line B = **LOW**
  - T2=off, T4=on → titik C2 = HIGH

Magnetic core vs Semikonduktor

Fakultas Informatika
School of Computing
Telkom University

Magnetic core

(-) Lambat (*switching*: 0.5-3 μS)

(-) Untuk luasan yang sama jml memori lebih sedikit

(+) *Non-volatile*

Semikonduktor

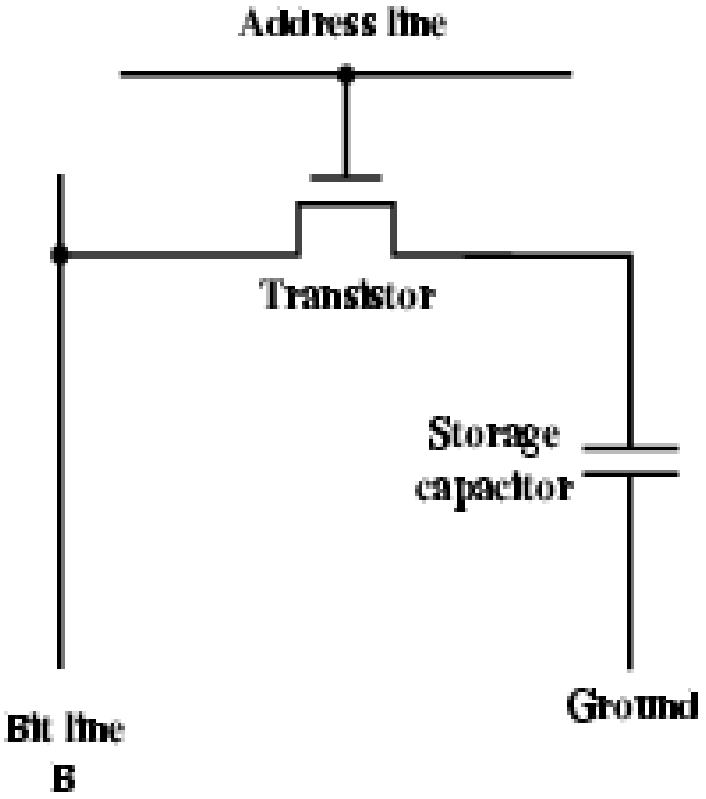
(+) Cepat (*switching*: 10-200 nS)

(+) Untuk luasan yang sama jml memori jauh lebih banyak

(-) *Volatile*

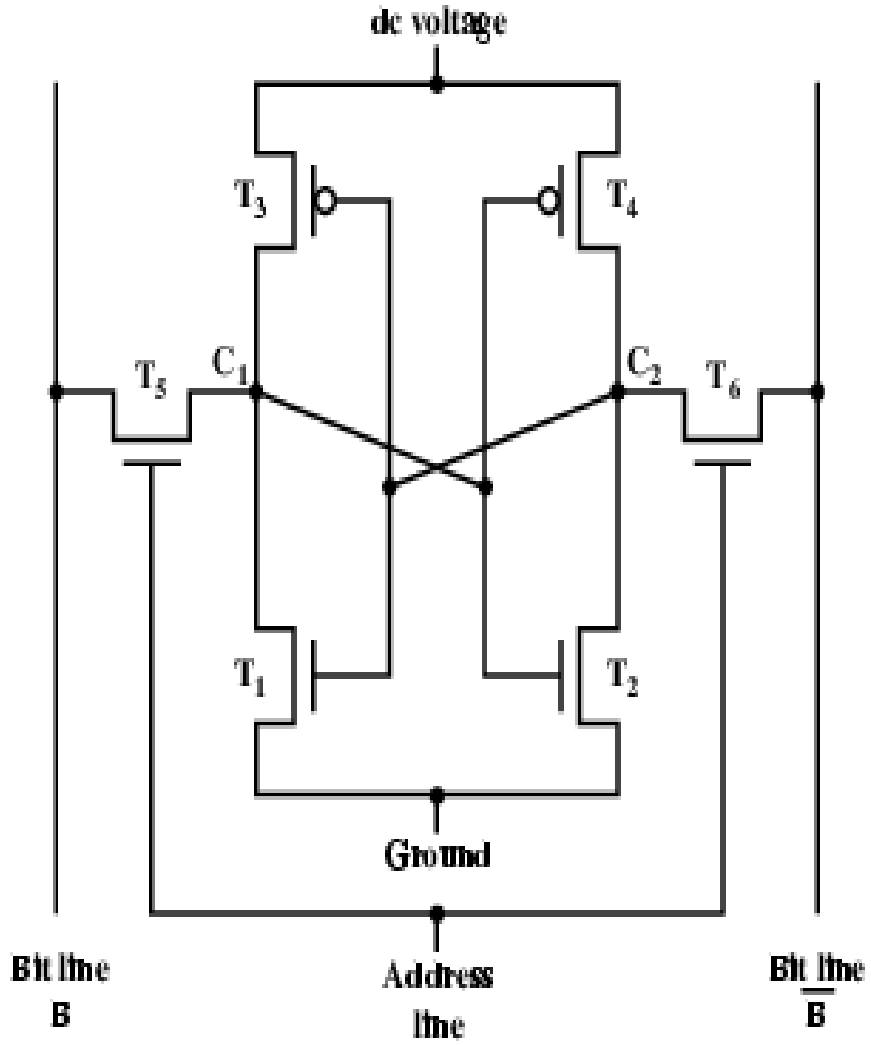
Memori Dinamis (DRAM)

- Dynamic RAM (DRAM)
  - Sel DRAM:
    - Data berupa muatan listrik yang disimpan di kapasitor
  - Mengapa disebut dynamic RAM?
    - Muatan listrik yang disimpan di kapasitor cenderung mengalami kebocoran, sehingga harus selalu di-*refresh*
  - DRAM digunakan untuk *main memory*



Memori Statis (SRAM)

- Static RAM (SRAM)
  - Sel SRAM:
    - Disusun dari beberapa transistor (*flip-flop*)
  - Mengapa disebut static RAM?
    - Selama masih ada listrik, maka data yang disimpan tidak hilang, sehingga tidak perlu di-*refresh*
  - SRAM digunakan untuk *cache memory*



DRAM vs SRAM

Fakultas Informatika
School of Computing
Telkom University

DRAM	SRAM
(+) sederhana	(-) lebih kompleks
(+) dimensi lebih kecil	(-) dimensi lebih besar
(+) murah	(-) lebih mahal
(+) kapasitas besar	(-) kapasitas kecil
(-) perlu rangkaian *refresh*	(+) tidak perlu di-*refresh*
(-) biaya rangkaian *refresh* memori berukuran besar lebih mahal daripada biaya memori itu sendiri	(+) tidak perlu rangkaian *refresh*
(-) lebih lambat	(+) lebih cepat

- Apa kesamaannya???

  Sama-sama *volatile*

Part 3: Perkembangan RAM

Advanced DRAM (1)

- *Synchronous DRAM (SDRAM)*
  - Pertukaran data didasarkan pada *signal clock* eksternal tanpa *wait state*
  - Kecepatan sesuai dengan kecepatan prosesor atau bus memori
  - Selama proses pencarian data, CPU dapat melakukan tugas lain (tidak perlu menunggu, karena CPU tahu kapan data sudah tersedia)

Advanced DRAM (2)

- Contoh: *SDRAM read timing*
  - *Burst length = 4 = T3-T6, CAS latency = 2 = T1-T2*

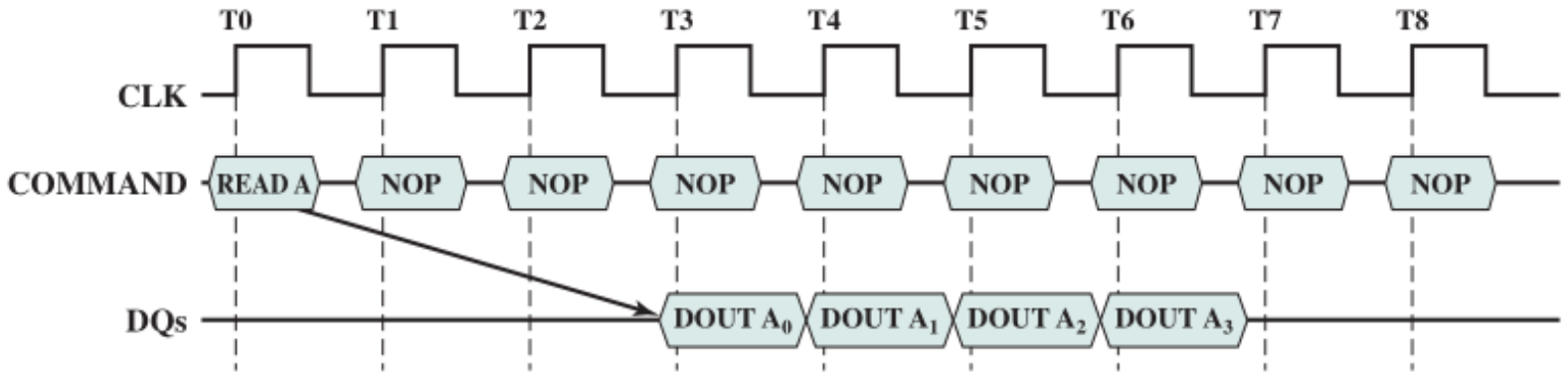
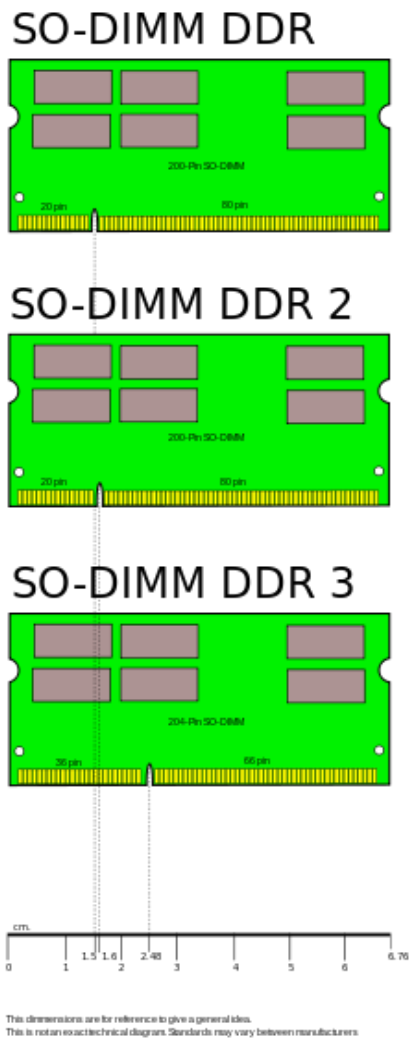


Figure 5.13 SDRAM Read Timing (burst length = 4, $\overline{\text{CAS}}$ latency = 2)

- *Burst length* = banyaknya *clock* untuk mengirimkan data (1, 2, 4, 8, *full page*)
- *Latency* = banyaknya *clock* yang diperlukan untuk persiapan sebelum data dikirimkan

- *Double Data Rate – SDRAM (DDR-SDRAM)*
  - Sama seperti SDRAM, tetapi dapat mengirimkan data 2x dalam satu *clock*

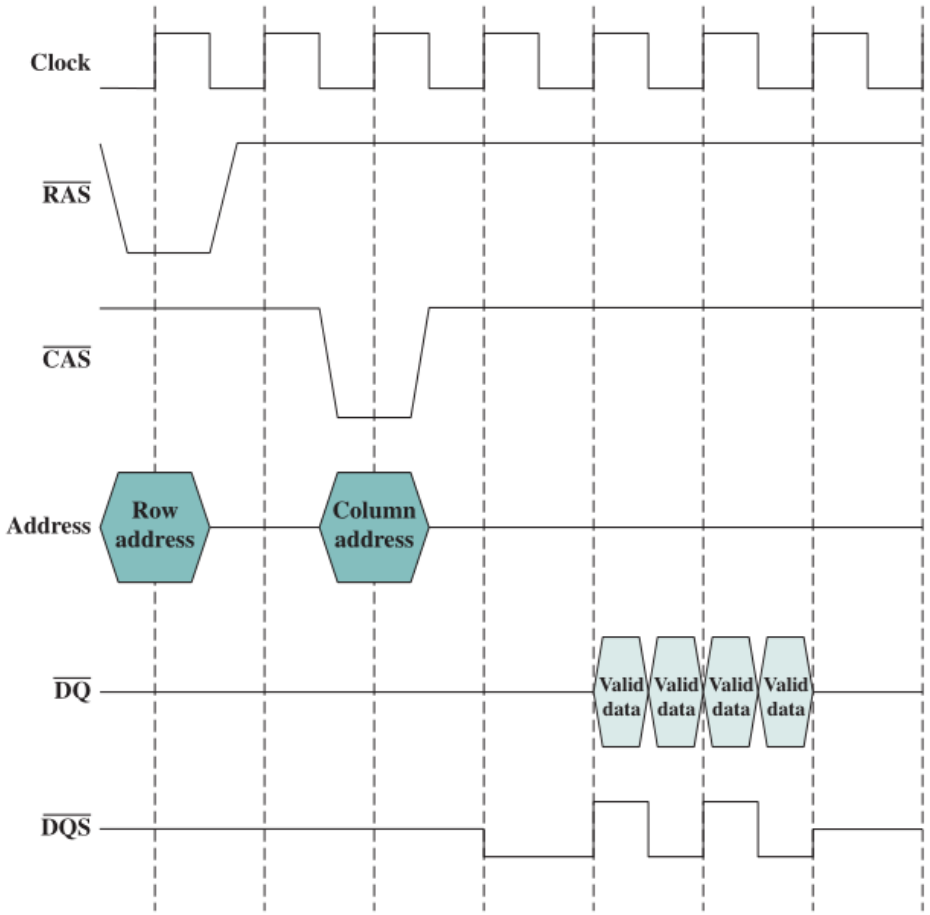
Advanced DRAM (3)



- Synchronized DRAM (SDRAM)
  - Menggunakan Clock (salah satu edge)
  - Tidak ada waktu tunggu
- Double Data Rate (DDR) SDRAM
  - Menggunakan kedua *edge clock* (*rising edge* dan *falling edge*) untuk data transfer
  - DDR 2 dan DDR 3 mempunyai frekuensi lebih tinggi dan *buffer* lebih besar
- Small Outline-Dual Inline Memory Module (SO-DIMM)
  - *Dual Inline Memory Module (DIMM)* punya pin berbeda di kedua sisi sehingga meningkatkan ukuran bus dan kecepatan
  - SO-DIMM adalah versi DIMM yang lebih kecil yang digunakan di Laptop

http://abhaycopi.blogspot.com/2014/04/advanced-dram-organization.html

Advanced DRAM (4)

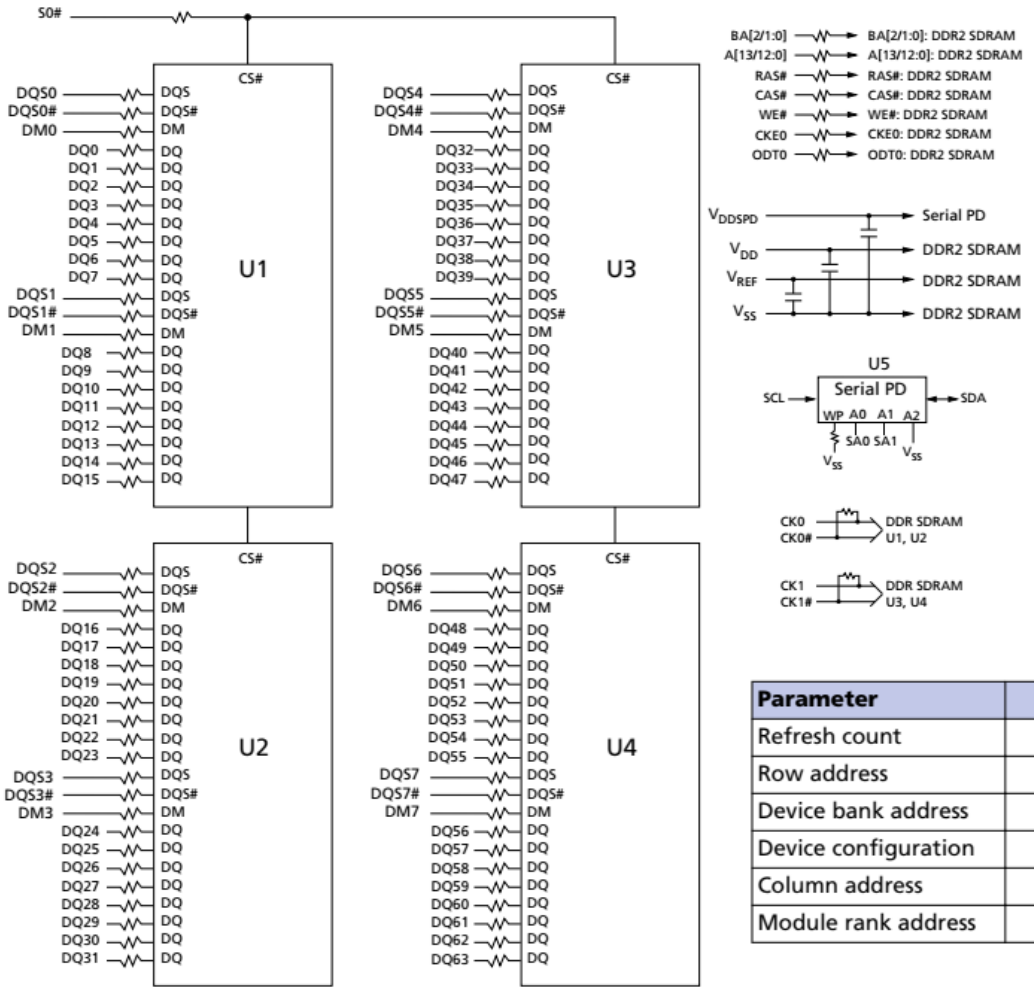


RAS = Row address select
CAS = Column address select
DQ = Data (in or out)
DQS = DQ select

Figure 5.15 DDR SDRAM Road Timing

Advanced DRAM (5)

Fakultas Informatika
School of Computing
Telkom University



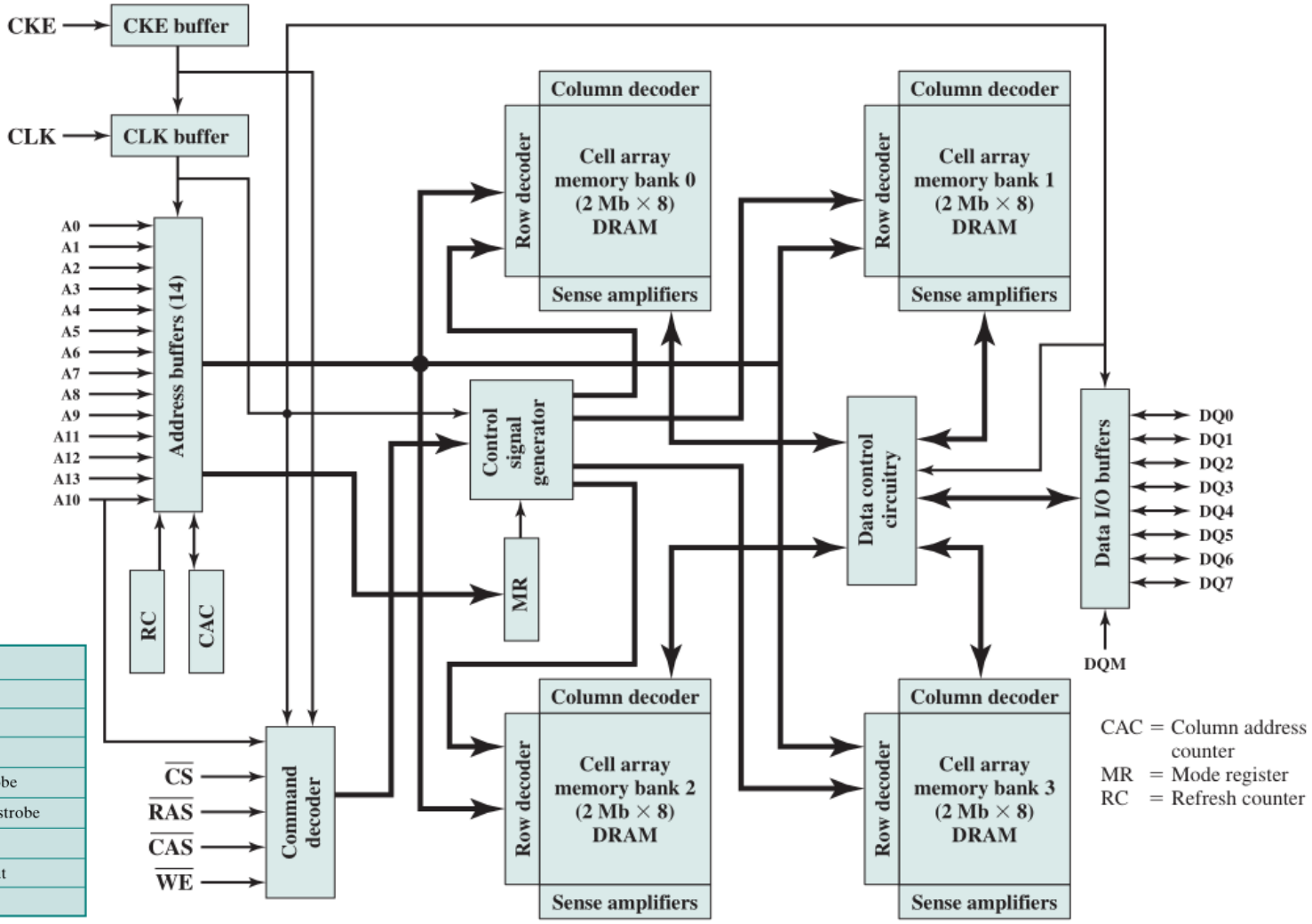
256 MB DDR2 SO-DIMM

- Address:
  - Row = 13 bit
  - Column = 10 bit
  - Bank = 2 bit
- Data:
  - DQ pin = 64 bit
- Kapasitas:
  - $2^{13+10+2}$ x 64 = 256 MB

Parameter	256MB
Refresh count	8K
Row address	8K A[12:0]
Device bank address	4 BA[1:0]
Device configuration	512Mb (32 Meg x 16)
Column address	1K A[9:0]
Module rank address	1 S0#

Advanced DRAM (6)

Contoh:
IBM 64 Mb SDRAM



Just info

A0 to A13	Address inputs
CLK	Clock input
CKE	Clock enable
$\overline{CS}$	Chip select
$\overline{RAS}$	Row address strobe
$\overline{CAS}$	Column address strobe
$\overline{WE}$	Write enable
DQ0 to DQ7	Data input/output
DQM	Data mask

Figure 5.12 Synchronous Dynamic RAM (SDRAM)

Advanced DRAM (7)

- *Rambus DRAM (RDRAM)*
  - Diadopsi oleh Intel untuk Pentium dan Itanium
  - Pesaing model SDRAM
  - Bus alamat *support* s.d. 320 chip RDRAM
  - Transfer data s.d. 1,6 GBps (2 x 800 MBps)
    - *Impedance, clock*, dan *signal* didefinisikan secara tepat
  - Jalur data sebanyak 18 bit (16 data, 2 parity)
  - Jalur RC (8 bit) digunakan untuk alamat dan signal control
  - *Memory request* tidak *menggunakan RAS, CAS, R/W atau CE, tetapi melalui high* speed bus yang memuat informasi:
    - alamat yang diinginkan
    - jenis operasi
    - jumlah byte dalam operasi
  - Pengiriman data secara *synchronous*

Advanced DRAM (8)

- Struktur RDRAM

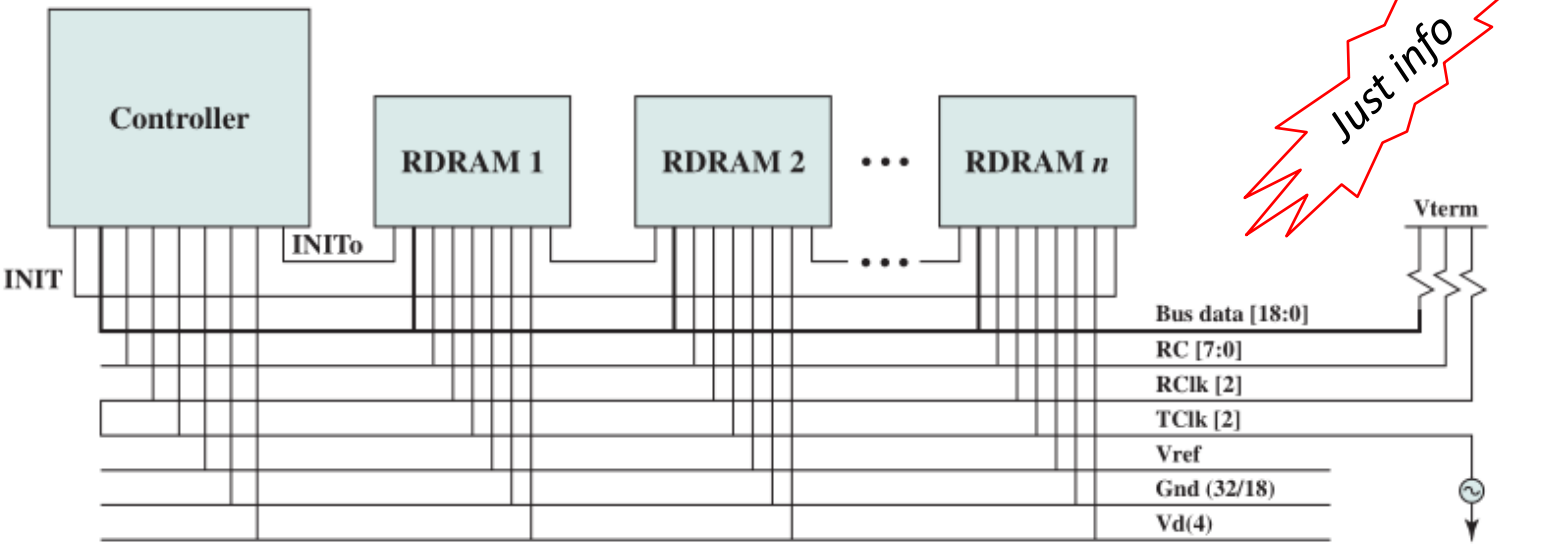
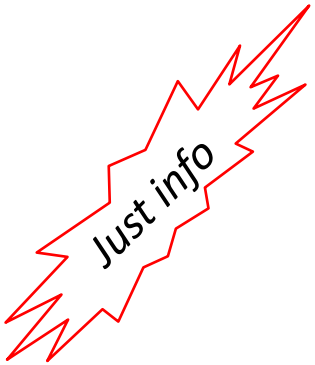


Figure 5.14 RDRAM Structure

	Clock Frequency (MHz)	Transfer Rate (GB/s)	Access Time (ns)	Pin Count
SDRAM	166	1.3	18	168
DDR	200	3.2	12.5	184
RDRAM	600	4.8	12	162

Advanced DRAM (9)

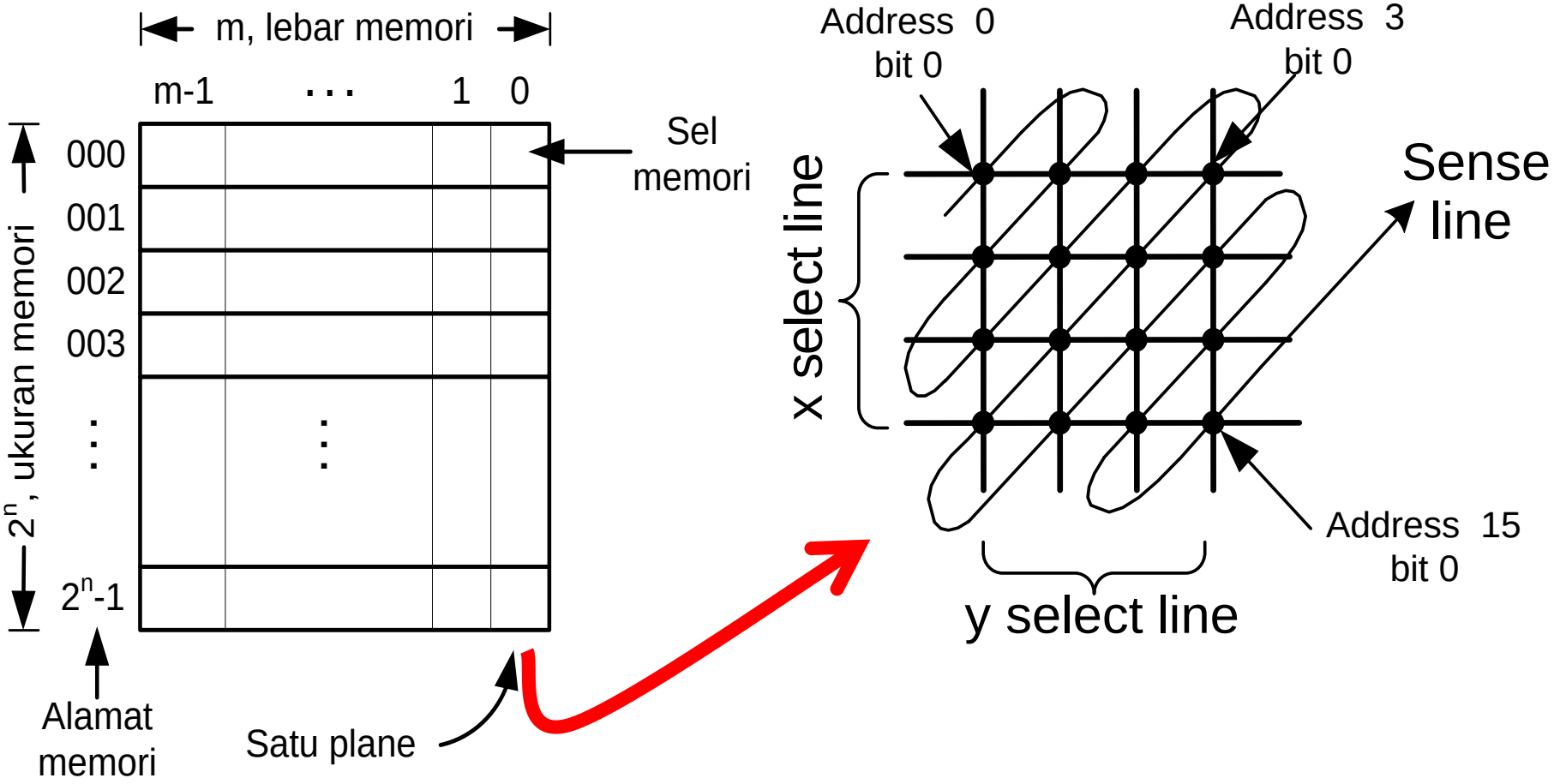
- *Cache DRAM (CDRAM)*
  - Penggabungan sejumlah kecil SRAM (16 kbit) ke dalam chip DRAM
  - Tujuan:
    - Sebagai *cache* seperti *cache* pada prosesor yang terdiri dari 64 line
    - Sebagai *buffer* akses blok data serial, misal untuk *refresh screen*



Part 4: Bagaimana sel-sel memori dirangkai?

Organisasi Memori (1)

- Bagaimana sel-sel memori disusun?



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:

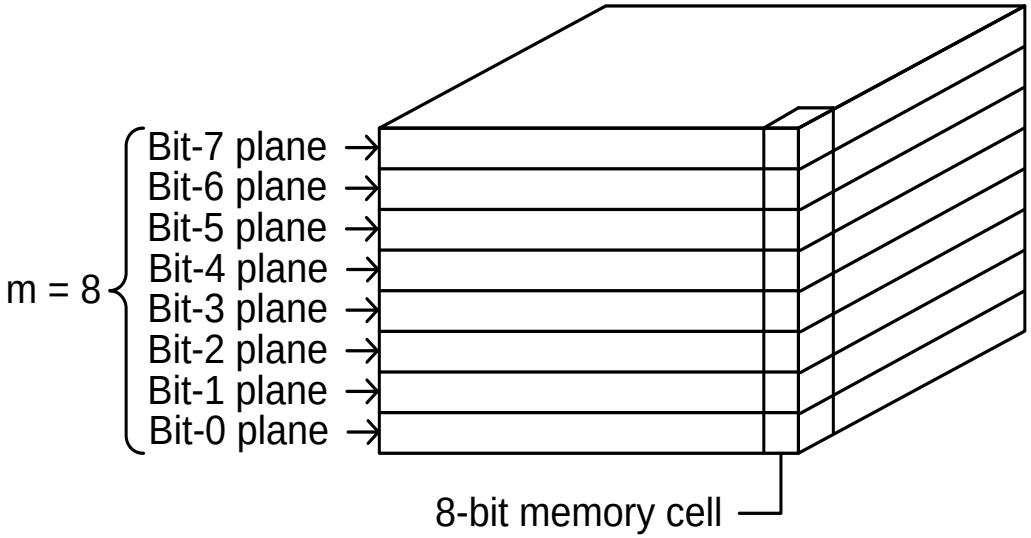
  – Jumlah jalur setiap *select line* = 1 dan 32, 2 dan 16, 4 dan 8, atau 8 dan 4, atau 16 dan 2

  – 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 di bawah ini, 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?

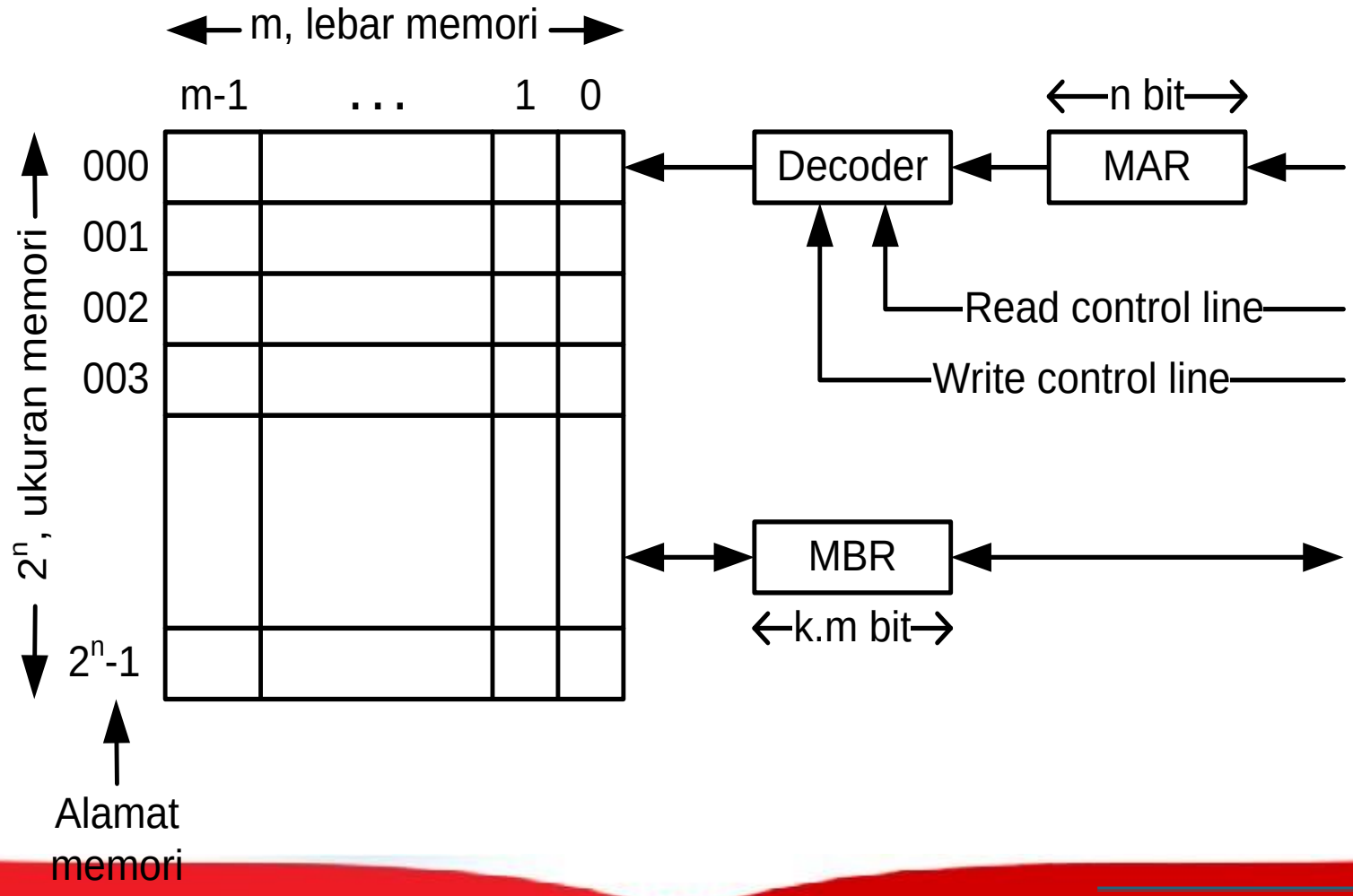
Jawab:

- 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)

Fakultas Informatika
School of Computing
Telkom University

- 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

      ke-0 0 – 15 0000000 – 0001111

      ke-1 16 – 31 0010000 – 0011111

      ... ...

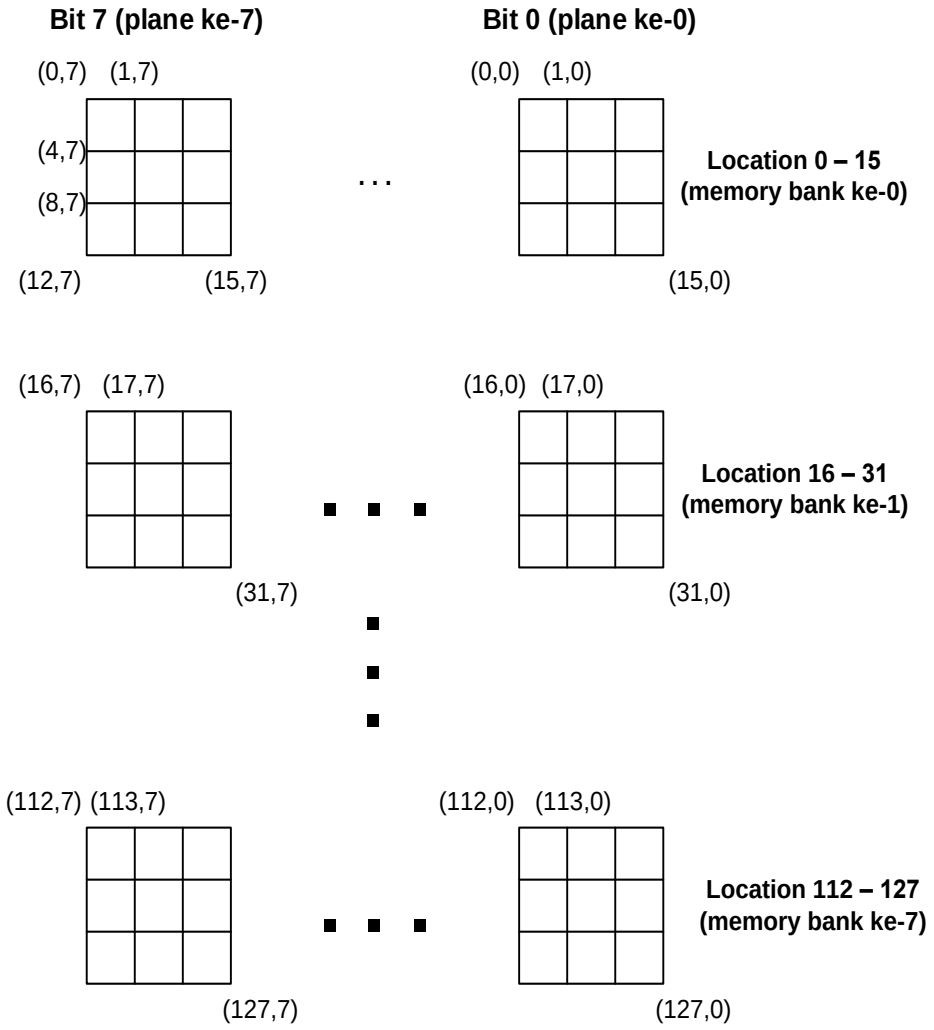
      ke-7 112 – 127 1110000 – 1111111

      - 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)

Fakultas Informatika
School of Computing
Telkom University

Distribusi lokasi memori sebanyak 128 alamat



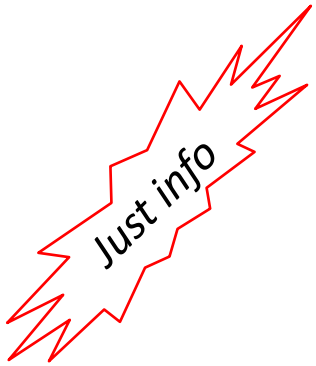
* (a,b) = alamat ke-a, posisi bit ke-b

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:

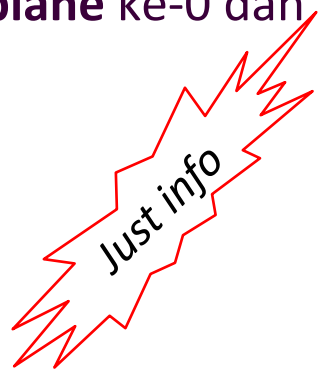
Alamat	Dalam biner
4	000 01 00
5	000 01 01
6	000 01 10
7	000 01 11

- 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



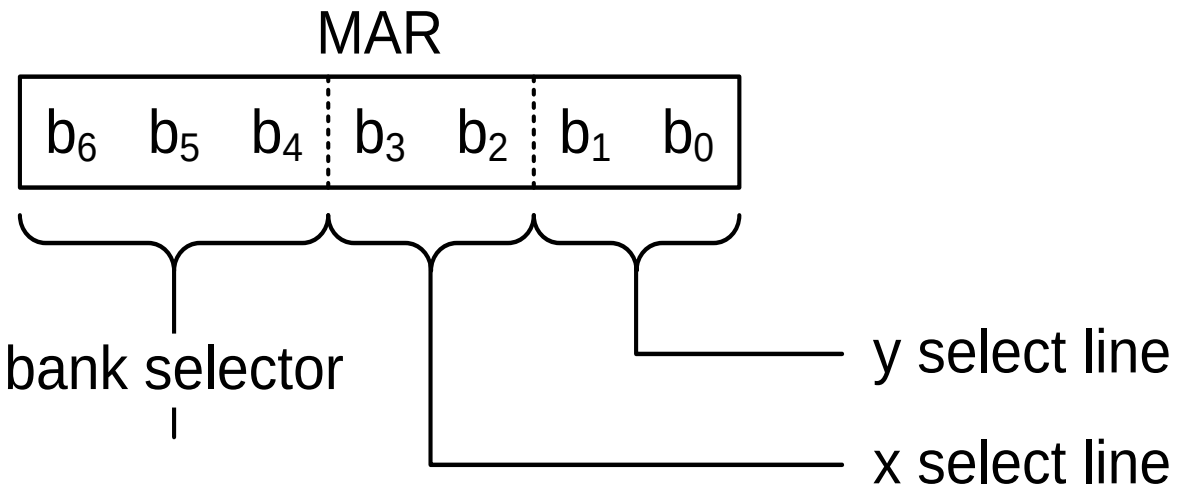
- Alamat yang terletak pada **kolom kedua** pada **plane** ke-0 dan pada **bank** ke-0 adalah sbb:

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:



Referensi

- [SCH85] Schneider, Michael G. 1985. "*The Principle of Computer Organization".* 1st edition. John Wiley & Sons. Canada.
- [STA19] Stalling, William. 2019. "*Computer Organization and Architecture: Designing for Performance*". 11th edition. Prentice Hall.
- https://en.wikipedia.org/wiki/Magnetic-core_memory#/media/File:Magnetic-core_memory,_at_angle.jpg