**BAB 1**

**PENDAHULUAN**

1.1 Latar Belakang

Memori komputer bisa diibaratkan sebagai papan tulis, dimana setiap orang yang masuk kedalam ruangan bisa membaca dan memanfaatkan data yang ada dengan tanpa merubah susunan yang tersaji. Data yang diproses oleh komputer, sebenarnya masih tersimpan didalam memory, dan dalam hal ini komputer hanya membaca data dan kemudian memprosesnya. Satu kali data tersimpan didalam memory komputer, maka data tersebut akan tetap tinggal disitu selamanya. Setiap kali memory penuh, maka data yang ada bisa dihapus sebagian ataupun seluruhnya untuk diganti dengan data yang baru.

Walaupun konsepnya terasa sederhana, memori komputer memiliki aneka ragam jenis, teknologi, organisasi, unjuk kerja, dan biaya bagi sistem komputer. Tidak ada satu pun teknologi yang optimal dalam memuaskan kebutuhan memori suatu sistem komputer. Sebagai akibatnya, sistem komputer yang umum dilengkapi dengan hirarki subsistem-subsistem memori, yang sebagiannya bersifat internal terhadap sistem (dapat diakses secara langsung oleh processor) dan sebagian lagi bersifat eksternal (dapat diakses oleh processor melalui suatu modul I/O).

1.2 Rumusan Masalah

1. Apa dua interpretasi dari istilah memori akses acak?

2. Apa perbedaan antara DRAM dan SRAM dalam hal aplikasi?

3. Apa sajakah aplikasi ROM?

4. Apa perbedaan antara EPROM, EEPROM, dan memori flash?

5. Bagaimana SDRAM berbeda dari DRAM?

6. Apa perbedaan antara memori flash NAND dan NOR?

1.3 Tujuan

1. Menyajikan ikhtisar dari tipe-tipe utama dari memori utama semikonduktor.

2. Memahami pengoperasian kode dasar yang dapat mendeteksi dan memperbaiki kesalahan bit tunggal dalam kata-kata 8-bit.

3. Meringkas sifat-sifat organisasi DRAM DDR kontemporer.

4. Memahami perbedaan antara memori flash NOR dan NAND.

5. Memberikan gambaran umum tentang teknologi memori solid-state yang lebih baru dan tidak mudah menguap.

**BAB 2**

**PEMBAHASAN**

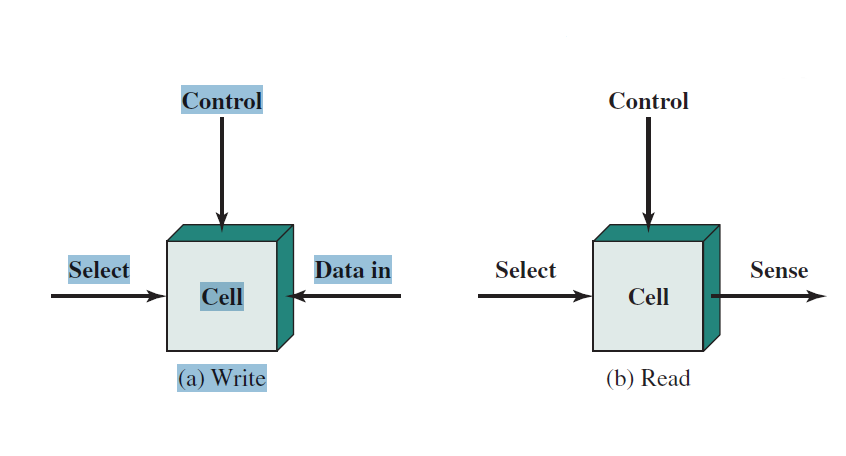
2.1 Memori Utama Semikonduktor

Di komputer sebelumnya, bentuk paling umum dari penyimpanan akses acak untuk komputer memori utama menggunakan array loop feromagnetik berbentuk donat yang dirujuk sebagai inti. Oleh karena itu, memori utama sering disebut sebagai inti, sebuah istilah yang terus berlanjut . Hari ini. Munculnya, dan keuntungan dari, mikroelektronika telah lama dikalahkan memori inti magnetik. Hari ini, penggunaan chip semikonduktor untuk memori utama hampir universal. Aspek-aspek kunci dari teknologi ini dieksplorasi di bagian ini.

Elemen dasar dari memori semikonduktor adalah sel memori. Meski beragam teknologi elektronik digunakan, semua sel memori semikonduktor berbagi tertentu properti:

1. Mereka menunjukkan dua negara stabil (atau semistable), yang dapat digunakan untuk mewakili biner 1 dan 0
2. Mereka mampu ditulis ke (setidaknya sekali), untuk mengatur negara.
3. Mereka mampu membaca untuk merasakan negara.

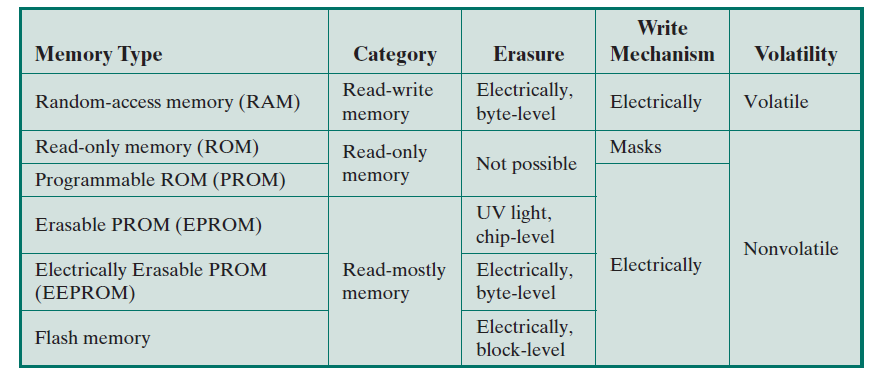
Ini memiliki tiga terminal fungsional yang mampu membawa sinyal listrik. Terminal pilih, seperti namanya, memilih sel memori untuk operasi baca atau tulis. Itu terminal kontrol menunjukkan membaca atau menulis. Untuk menulis, terminal lain menyediakan sinyal listrik yang mengatur keadaan sel menjadi 1 atau 0. Untuk membaca, terminal itu digunakan untuk menghasilkan status sel. Detail organisasi internal, berfungsi, dan waktu dari sel memori tergantung pada teknologi sirkuit terintegrasi yang spesifik digunakan dan berada di luar cakupan buku ini, kecuali untuk ringkasan singkat. Untuk tujuan kita, kita akan menganggapnya sebagai bahwa sel-sel individual dapat dipilih untuk dibaca dan menulis operasi.



DRAM dan SRAM

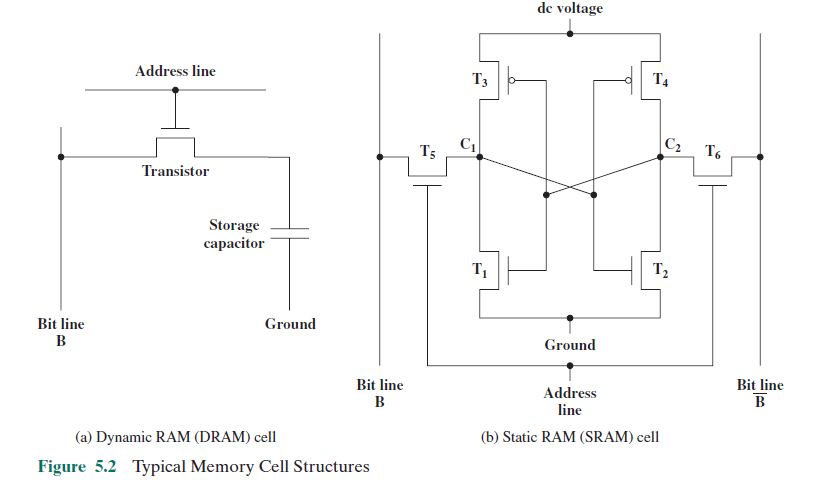
Semua jenis memori yang akan kita bahas dalam bab ini adalah akses acak. Bahwa adalah, kata-kata individu memori secara langsung diakses melalui pengalamatan kabel-in logika. Tabel 5.1 daftar jenis utama dari memori semikonduktor. Yang paling umum adalah disebut sebagai memori akses acak (RAM). Ini sebenarnya adalah penyalahgunaan istilah, karena semua jenis yang tercantum dalam tabel adalah akses acak. Satu yang membedakan Karakteristik dari memori yang ditetapkan sebagai RAM adalah memungkinkan untuk membaca data dari memori dan menulis data baru ke dalam memori dengan mudah dan cepat.

Baik membaca dan menulis dicapai melalui penggunaan sinyal listrik. Karakteristik lain yang membedakan RAM tradisional adalah sifatnya yang mudah menguap. RAM harus dilengkapi dengan catu daya konstan. Jika daya terganggu, maka datanya hilang. Jadi, RAM hanya dapat digunakan sebagai penyimpanan sementara. Keduanya bentuk RAM tradisional yang digunakan dalam komputer adalah DRAM dan SRAM. Formulir yang lebih baru RAM, dibahas pada Bagian 5.5, tidak mudah berubah.

   
 RAM, teknologi dinamis dibagi menjadi dua teknologi: dinamis dan statis. RAM dinamis (DRAM) dibuat dengan sel yang menyimpan data sebagai muatan kapasitor. Ada atau tidak adanya muatan dalam kapasitor ditafsirkan sebagai biner 1 atau 0. Karena kapasitor memiliki kecenderungan alami untuk keluar, dinamis .RAM membutuhkan penyegaran berkala untuk menjaga penyimpanan data. Syarat dinamis mengacu pada kecenderungan ini dari muatan yang disimpan, bahkan dengan kekuatan terus diterapkan.

Gambar diatas adalah struktur DRAM khas untuk sel individual yang menyimpan satu sedikit. Baris alamat diaktifkan ketika nilai bit dari sel ini harus dibaca atau tertulis. Transistor bertindak sebagai saklar yang tertutup (memungkinkan arus mengalir) jika a tegangan diterapkan ke garis alamat dan terbuka (tidak ada arus mengalir) jika tidak ada tegangan hadir di garis alamat. Untuk operasi tulis, sinyal tegangan diterapkan ke garis bit; tegangan tinggi mewakili 1, dan tegangan rendah mewakili 0. Sebuah sinyal kemudian diterapkan pada baris alamat, memungkinkan muatan ditransfer ke kapasitor. Untuk operasi baca, ketika garis alamat dipilih, transistor berubah dan muatan yang disimpan pada kapasitor dimasukkan ke dalam garis bit dan untuk suatu arti penguat. Penguat rasa membandingkan tegangan kapasitor ke nilai referensi dan menentukan apakah sel berisi logika 1 atau logika 0. Pembacaan dari sel membuang kapasitor, yang harus dipulihkan untuk menyelesaikan operasi. Meskipun DRAM cell digunakan untuk menyimpan bit tunggal (0 atau 1), pada dasarnya perangkat analog. Kapasitor dapat menyimpan nilai muatan apa pun dalam rentang; ambang batas nilai menentukan apakah muatan ditafsirkan sebagai 1 atau 0.

RAM statis Sebaliknya, RAM statis (SRAM) adalah perangkat digital yang menggunakan elemen logika yang sama yang digunakan dalam prosesor. Dalam SRAM, nilai-nilai biner disimpan menggunakan konfigurasi gerbang logika flip-flop tradisional . RAM statis akan menyimpan datanya selama daya dipasok ke sana.

 Empat transistor (T1, T2, T3, T4) saling terhubung dalam pengaturan yang menghasilkan logika stabil negara. Dalam keadaan logika 1, titik C1 tinggi dan titik C2 rendah; dalam keadaan ini, T1 dan T4 adalah mati dan T2 dan T3 menyala.1 Dalam keadaan logika 0, titik C1 rendah dan titik C2 tinggi; di dalam state, T1 dan T4 aktif dan T2 dan T3 mati. Kedua negara stabil selama yang langsung tegangan arus (dc) diterapkan. Berbeda dengan DRAM, tidak diperlukan penyegaran untuk mempertahankan data.

Seperti pada DRAM, baris alamat SRAM digunakan untuk membuka atau menutup sebuah tombol. Baris alamat mengontrol dua transistor (T5 dan T6). Ketika sinyal diterapkan baris ini, dua transistor dinyalakan, memungkinkan operasi baca atau tulis. Untuk operasi tulis, nilai bit yang diinginkan diterapkan ke garis B, sementara komplemennya diterapkan ke baris B. Ini memaksa empat transistor (T1, T2, T3, T4) menjadi yang tepat negara. Untuk operasi baca, nilai bit dibaca dari baris B.

RAM statis dan dinamis bersifat volatil; yaitu, kekuasaan harus terus dipasok ke memori untuk mempertahankan nilai bit. Suatu dinamika sel memori lebih sederhana dan lebih kecil dari sel memori statis. Jadi, DRAM lebih padat (sel yang lebih kecil = lebih banyak sel per satuan luas) dan lebih murah daripada SRAM yang sesuai. Di sisi lain, DRAM membutuhkan dukungan memperbarui sirkuit. Untuk ingatan yang lebih besar, biaya tetap dari sirkuit penyegaran lebih banyak daripada dikompensasi oleh biaya variabel DRAM yang lebih kecil. Jadi, DRAM cenderung disukai untuk kebutuhan memori yang besar. Poin terakhir adalah bahwa SRAM adalah agak lebih cepat dari DRAM. Karena karakteristik relatif ini, SRAM adalah digunakan untuk memori cache (baik on dan off chip), dan DRAM digunakan untuk memori utama.

Jenis ROM

memori read-only (ROM) berisi pola permanen data yang tidak dapat diubah. ROM tidak mudah berubah, artinya, tidak ada sumber daya diperlukan untuk mempertahankan nilai bit dalam memori. Meskipun dimungkinkan untuk membaca ROM, itu tidak mungkin untuk menulis data baru ke dalamnya. Aplikasi penting dari ROM adalah microprogramming, dibahas dalam Bagian 4 Aplikasi potensial lainnya termasuk :

1. Subrutin Perpustakaan untuk fungsi yang sering dicari
2. Program sistem
3. Tabel fungsi

Untuk persyaratan ukuran sederhana, keuntungan ROM adalah data secara permanen dalam memori utama dan tidak perlu dimuat dari penyimpanan sekunder alat. ROM dibuat seperti chip sirkuit terintegrasi lainnya, dengan data sebenarnya ditransfer ke chip sebagai bagian dari proses fabrikasi. Ini menyajikan dua masalah :

1. Langkah penyisipan data termasuk biaya tetap yang relatif besar, baik satu atau ribuan salinan ROM tertentu dibuat.
2. Tidak ada ruang untuk kesalahan. Jika satu bit salah, seluruh batch ROM harus  
   dibuang.

Ketika hanya sejumlah kecil ROM dengan konten memori tertentu diperlukan, alternatif yang lebih murah adalah ROM yang dapat diprogram (PROM). Seperti ROM, PROM tidak mudah berubah dan dapat ditulis hanya satu kali. Untuk PROM, proses penulisan dilakukan secara elektrik dan dapat dilakukan oleh pemasok atau pelanggan pada waktu kemudian dari fabrikasi chip asli. Peralatan khusus diperlukan untuk proses penulisan atau "pemrograman". PROM menyediakan fleksibilitas dan kenyamanan. ROM tetap menarik untuk produksi volume tinggi. . Variasi lain pada memori read-only adalah kebanyakan memori yang dibaca, yaitu berguna untuk aplikasi yang operasi baca jauh lebih sering daripada menulis operasi tetapi untuk itu penyimpanan non-volatile diperlukan. Ada tiga yang umum bentuk-bentuk sebagian besar memori: EPROM, EEPROM, dan memori flash.Memori read-only yang dapat diprogram secara optis (EPROM) dibaca dan ditulis secara elektrik, seperti dengan PROM. Namun, sebelum operasi tulis, semua penyimpanan sel harus dihapus ke keadaan awal yang sama dengan pemaparan chip yang dikemas radiasi ultraviolet. Erasure dilakukan dengan menyinari sinar ultraviolet yang intens melalui jendela yang dirancang ke dalam chip memori. Proses penghapusan ini bisa terjadi dilakukan berulang kali; setiap penghapusan bisa memakan waktu hingga 20 menit untuk tampil. Demikian, EPROM dapat diubah beberapa kali dan, seperti ROM dan PROM, memegangnya data hampir tanpa batas. Untuk jumlah penyimpanan yang sebanding, EPROM lebih banyak mahal daripada PROM, tetapi memiliki keuntungan dari kemampuan pembaruan ganda.

Bentuk yang lebih menarik dari sebagian besar memori adalah elektrik yang dapat dihapus dapat diprogram read-only memory (EEPROM). Ini adalah sebagian besar memori yang dibaca dapat ditulis kapan saja tanpa menghapus konten sebelumnya; hanya byte atau byte dibahas diperbarui. Operasi tulis membutuhkan waktu lebih lama daripada membaca operasi, pada urutan beberapa ratus mikrodetik per byte. EEPROM menggabungkan keuntungan dari non-volatilitas dengan fleksibilitas yang dapat diupdate tempat, menggunakan kontrol bus biasa, alamat, dan jalur data. EEPROM lebih mahal dari EPROM dan juga kurang padat, mendukung bit lebih sedikit per keeping.

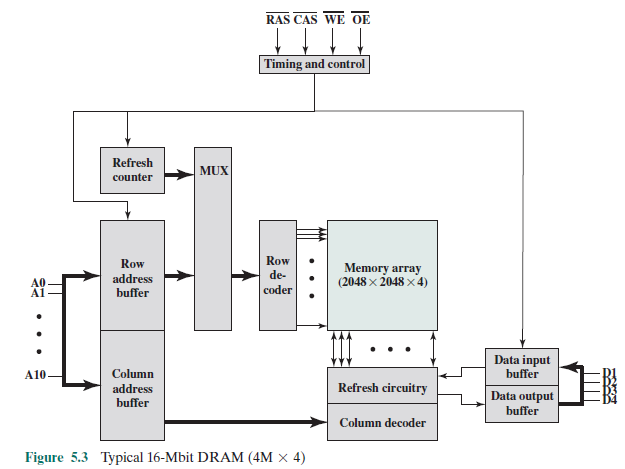
Bentuk lain dari memori semikonduktor adalah memori flash (dinamakan demikian karena kecepatan yang dapat diprogram ulang). Pertama kali diperkenalkan pada pertengahan 1980-an, flash memori adalah perantara antara EPROM dan EEPROM baik dalam biaya maupun fungsi. Seperti EEPROM, memori flash menggunakan teknologi penghapus listrik. Flash keseluruhan memori dapat dihapus dalam satu atau beberapa detik, yang jauh lebih cepat daripada EPROM.

Selain itu, dimungkinkan untuk menghapus hanya blok memori daripada seluruh chip. Flash memori mendapatkan namanya karena microchip diatur sedemikian rupa sehingga bagian dari memori Sel-sel dihapus dalam satu tindakan atau "flash." Namun, memori flash tidak menyediakan penghapusan tingkat byte. Seperti EPROM, memori flash hanya menggunakan satu transistor per bit, dan sebagainya mencapai kepadatan tinggi (dibandingkan dengan EEPROM) dari EPROM.

Chip Logic

Seperti produk sirkuit terpadu lainnya, memori semikonduktor dikemas keripik. Setiap chip berisi array sel memori. Dalam hirarki memori secara keseluruhan, kami melihat bahwa ada trade-off di antara kecepatan, kepadatan, dan biaya. Ini trade-off juga ada ketika kita mempertimbangkan organisasi sel memori dan logika fungsional pada sebuah chip. Untuk kenangan semikonduktor, salah satu masalah desain utama adalah jumlah bit data yang dapat dibaca / ditulis pada suatu waktu. Pada satu ekstrim adalah organisasi di mana pengaturan fisik sel dalam array sama dengan susunan logis (seperti yang dirasakan oleh prosesor) kata-kata dalam memori. Array diatur ke dalam kata-kata W dari bit B masing-masing.

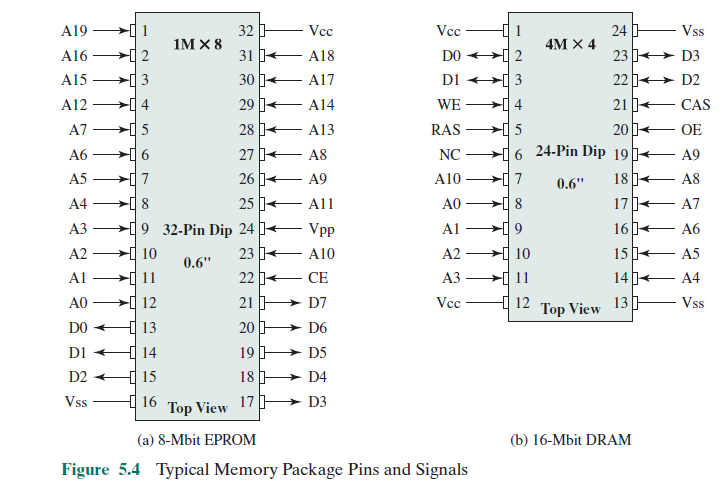
Misalnya, chip 16-Mbit dapat diatur sebagai kata-kata 1M 16-bit. Di sisi lain ekstrim adalah apa yang disebut organisasi 1-bit-per-chip, di mana data dibaca / ditulis sedikit demi sedikit. Kami akan menggambarkan organisasi chip memori dengan DRAM; ROM organisasi serupa, meskipun lebih sederhana. Secara logis, array memori diatur sebagai empat persegi array 2048 oleh 2048 elemen. Berbagai pengaturan fisik dimungkinkan. Di mana saja case, elemen-elemen array dihubungkan oleh horisontal (baris) dan vertikal (kolom) garis. Setiap garis horizontal terhubung ke terminal Pilih setiap sel di dalamnya baris; setiap garis vertikal terhubung ke terminal Data-In / Sense dari setiap sel di kolomnya.

 Baris alamat menyediakan alamat kata yang akan dipilih. Total log2 W garis diperlukan. Dalam contoh kami, 11 jalur alamat diperlukan untuk memilih salah satu dari 2048 baris. Ini 11 baris dimasukkan ke decoder baris, yang memiliki 11 baris input dan 2048 garis untuk output. Logika decoder mengaktifkan satu dari 2048 output tergantung pada pola bit pada 11 jalur input (211 = 2048). Tambahan 11 baris alamat pilih satu dari 2048 kolom 4 bit per kolom. Empat jalur data digunakan untuk input dan output dari 4 bit ke dan dari buffer data. Pada input (tulis), penggerak bit setiap bit line diaktifkan untuk 1 atau 0 sesuai dengan nilai dari garis data yang sesuai. Pada output (baca), nilai setiap bit line dilewatkan melalui penguat rasa dan disajikan ke garis data. Baris baris memilih baris sel man yang digunakan untuk membaca atau menulis.

Karena hanya 4 bit yang dibaca / ditulis ke DRAM ini, harus ada beberapa DRAM terhubung ke pengontrol memori untuk membaca / menulis kata data ke bus. Perhatikan bahwa hanya ada 11 baris alamat (A0 – A10), setengah dari jumlah Anda harapkan untuk array 2048 \* 2048. Ini dilakukan untuk menghemat jumlah pin. 22 jalur alamat yang diperlukan dilewatkan melalui logika pilih eksternal ke chip dan multiplexing ke 11 garis alamat. Pertama, 11 sinyal alamat dilewatkan ke chip untuk menentukan alamat baris dari array, dan kemudian 11 sinyal alamat lainnya disajikan untuk alamat kolom. Sinyal-sinyal ini disertai dengan alamat baris pilih (RAS) dan alamat kolom pilih (CAS) sinyal untuk menyediakan waktu ke chip. Pin write enable (WE) dan output enable (OE) menentukan apakah a menulis atau membaca operasi dilakukan. Dua pin lainnya, tidak ditunjukkan pada G, adalah tanah (Vss) dan sumber tegangan (Vcc).

Semua DRAM membutuhkan operasi penyegaran. Teknik sederhana untuk menyegarkan, pada dasarnya, untuk menonaktifkan DRAM chip sementara semua sel data di-refresh. Penghitung ulang langkah-langkah melalui semua dari nilai baris. Untuk setiap baris, garis output dari penghitung refresh disediakan ke decoder baris dan garis RAS diaktifkan. Data dibacakan dan ditulis kembali ke lokasi yang sama. Ini menyebabkan setiap sel di baris disegarkan.

Sirkuit terpadu dipasang pada paket itu berisi pin untuk koneksi ke dunia luar. Gambar 5.4a menunjukkan contoh paket EPROM, yang merupakan chip 8-Mbit diatur sebagai 1M \* 8. Dalam hal ini, organisasi diperlakukan sebagai satu kata-perchip paket. Paket termasuk 32 pin, yang merupakan salah satu paket chip standar ukuran. Pin mendukung garis sinyal berikut:

1. Alamat dari kata yang sedang diakses. Untuk kata 1M, total 20 (220 = 1M) pin diperlukan (A0 – A19).
2. Data yang akan dibaca, terdiri dari 8 baris (D0-D7).
3. Catu daya ke chip (Vcc).
4. Pin ground (Vss).
5. Pin pengaktifan chip (CE). Karena mungkin ada lebih dari satu chip memori, masing-masing terhubung ke bus alamat yang sama, pin CE digunakan untuk menunjukkan apakah alamat berlaku untuk chip ini atau tidak. Pin CE diaktifkan oleh logika yang terhubung ke bit urutan tinggi dari bus alamat (yaitu, alamat bit di atas A19). Penggunaan sinyal ini diilustrasikan saat ini.
6. Sebuah tegangan program (Vpp) yang disediakan selama pemrograman (operasi tulis).

Karena DRAM diakses oleh baris dan kolom, dan alamatnya multiplexing, hanya 11 pin alamat yang diperlukan untuk menentukan kombinasi baris / kolom 4M (211 \* 211 = 222 = 4M). Fungsi dari alamat baris pilih (RAS) dan kolom pin alamat pilih (CAS) telah dibahas sebelumnya. Akhirnya, tidak ada koneksi (NC) pin disediakan sehingga ada sejumlah pin.

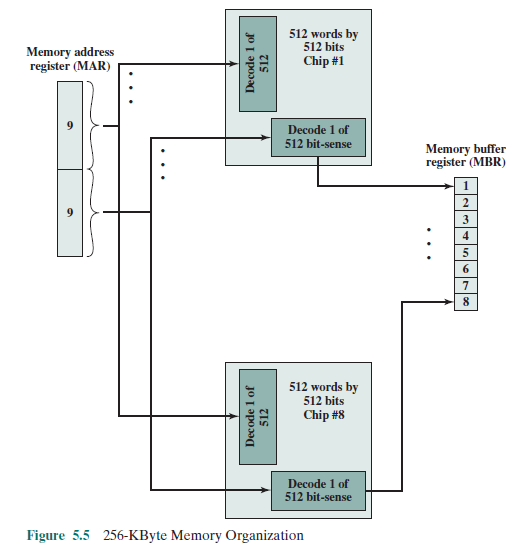
Module Organization

Jika sebuah chip RAM hanya berisi satu bit per kata, maka jelas kita akan membutuhkan setidaknya satu jumlah chip sama dengan jumlah bit per kata. Sebagai contoh, Gambar 5.5 menunjukkan bagaimana modul memori yang terdiri dari 256 ribu kata 8-bit dapat diatur. Untuk 256K kata-kata, alamat 18-bit diperlukan dan dipasok ke modul dari beberapa eksternal sumber (misalnya, baris alamat bus yang dilampirkan modul). Alamatnya adalah disajikan ke chip 8 256K \* 1 @ bit, yang masing-masing menyediakan input / output satu bit.

Organisasi ini berfungsi selama ukuran memori sama dengan jumlah bit per keping. Dalam kasus di mana memori yang lebih besar diperlukan, sebuah array chip dibutuhkan. Gambar 5.6 menunjukkan kemungkinan organisasi dari memori yang terdiri dari 1M kata dengan 8 bit per kata. Dalam hal ini, kami memiliki empat kolom chip, masing-masing kolom mengandung 256 ribu kata diatur seperti pada Gambar 5.5. Untuk kata 1M, 20 baris alamat dibutuhkan. 18 bit paling signifikan diarahkan ke 32 modul. Orde tinggi 2 bit adalah input ke grup pilih modul logika yang mengirim chip memungkinkan sinyal menjadi satu dari empat kolom modul.

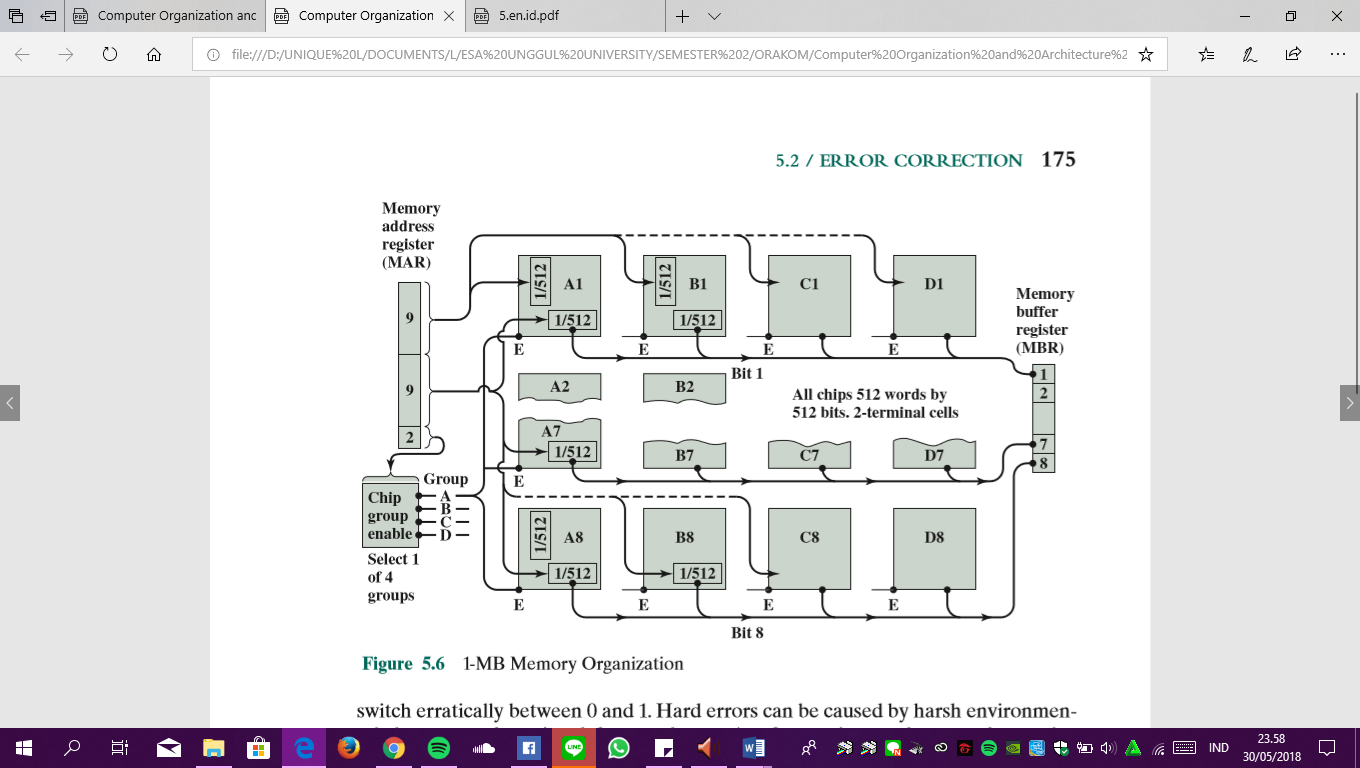
Interleaved Memory

Memori utama terdiri dari kumpulan chip memori DRAM. Sejumlah chip dapat dikelompokkan bersama untuk membentuk bank memori.



2.2 Koreksi Kesalahan

Sebuah sistem memori semikonduktor tunduk kesalahan. Ini dapat dikategorikan sebagai kegagalan keras dan kesalahan lembut. SEBUAH kegagalan hard adalah cacat fisik permanen sehingga sel memori atau sel yang terkena tidak dapat dipercaya SEBUAH kegagalan hard adalah cacat fisik permanen sehingga sel memori atau sel yang terkena tidak dapat dipercaya SEBUAH kegagalan hard adalah cacat fisik permanen sehingga sel memori atau sel yang terkena tidak dapat dipercaya menyimpan data tetapi menjadi terjebak pada 0 atau 1 atau beralih tak menentu antara 0 dan 1. kesalahan keras dapat disebabkan oleh penyalahgunaan lingkungan yang keras, cacat manufaktur, dan memakai. SEBUAH kesalahan lembut adalah acak, acara tak rusak yang mengubah isi keras, cacat manufaktur, dan memakai. SEBUAH kesalahan lembut adalah acak, acara tak rusak yang mengubah isi keras, cacat manufaktur, dan memakai. SEBUAH kesalahan lembut adalah acak, acara tak rusak yang mengubah isi dari satu atau lebih sel memori tanpa merusak memori. kesalahan yang lembut dapat disebabkan oleh masalah power supply atau partikel alpha. Partikel-partikel ini hasil dari peluruhan radioaktif dan sangat biasa terjadi karena inti radioaktif ditemukan dalam jumlah kecil di hampir semua bahan. Kedua kesalahan keras dan lunak jelas tidak diinginkan, dan sebagian besar sistem memori utama modern termasuk logika untuk kedua mendeteksi dan mengoreksi kesalahan.



Gambar diatas menggambarkan secara umum bagaimana proses dilakukan. Ketika data yang akan ditulis ke dalam memori, perhitungan, digambarkan sebagai fungsi f, dilakukan pada data untuk menghasilkan ditulis ke dalam memori, perhitungan, digambarkan sebagai fungsi f, dilakukan pada data untuk menghasilkan ditulis ke dalam memori, perhitungan, digambarkan sebagai fungsi f, dilakukan pada data untuk menghasilkan kode. Kedua kode dan data disimpan. Dengan demikian, jika M bit kata data akan disimpan dan kode ini kode. Kedua kode dan data disimpan. Dengan demikian, jika M bit kata data akan disimpan dan kode ini kode. Kedua kode dan data disimpan. Dengan demikian, jika M bit kata data akan disimpan dan kode ini panjang K bit, maka ukuran sebenarnya dari kata yang tersimpan M + K bit.panjang K bit, maka ukuran sebenarnya dari kata yang tersimpan M + K bit.panjang K bit, maka ukuran sebenarnya dari kata yang tersimpan M + K bit.panjang K bit, maka ukuran sebenarnya dari kata yang tersimpan M + K bit.panjang K bit, maka ukuran sebenarnya dari kata yang tersimpan M + K bit.

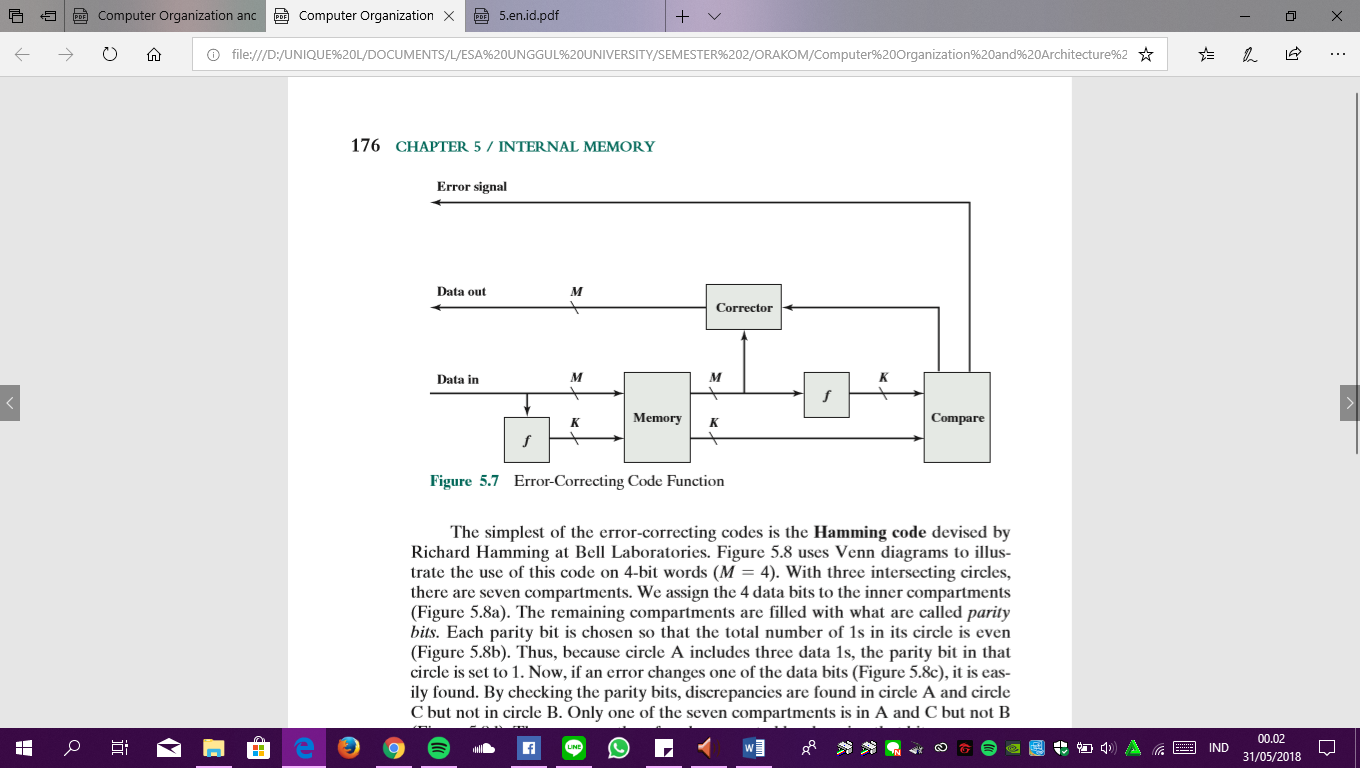
Ketika kata yang tersimpan sebelumnya dibacakan, kode yang digunakan untuk mendeteksi dan mungkin memperbaiki kesalahan. Sebuah set baru K kode bit yang dihasilkan dari M bit data dan dibandingkan dengan bit memperbaiki kesalahan. Sebuah set baru K kode bit yang dihasilkan dari M bit data dan dibandingkan dengan bit memperbaiki kesalahan. Sebuah set baru K kode bit yang dihasilkan dari M bit data dan dibandingkan dengan bit memperbaiki kesalahan. Sebuah set baru K kode bit yang dihasilkan dari M bit data dan dibandingkan dengan bit memperbaiki kesalahan. Sebuah set baru K kode bit yang dihasilkan dari M bit data dan dibandingkan dengan bit kode diambil. perbandingannya satu dari tiga hasil:

■ Tidak ada kesalahan yang terdeteksi. The diambil bit data dikirim keluar.

■ Kesalahan terdeteksi, dan adalah mungkin untuk memperbaiki kesalahan. Bit data ditambah koreksi kesalahan bit dimasukkan ke korektor, yang menghasilkan set dikoreksi koreksi kesalahan bit dimasukkan ke korektor, yang menghasilkan set dikoreksi M bit untuk dikirim keluar. M bit untuk dikirim keluar.

■ Kesalahan terdeteksi, tetapi tidak mungkin untuk memperbaikinya. Kondisi ini dilaporkan.

Kode ditandai dengan Kode yang beroperasi dalam mode ini disebut sebagai Kesalahan-kode koreksi. Kode ditandai dengan jumlah kesalahan bit dalam sebuah kata yang dapat memperbaiki dan mendeteksi.



Yang paling sederhana dari kode error-correcting adalah kode Hamming dirancang oleh Richard Hamming di Yang paling sederhana dari kode error-correcting adalah kode Hamming dirancang oleh Richard Hamming di Yang paling sederhana dari kode error-correcting adalah kode Hamming dirancang oleh Richard Hamming di Bell Laboratories. Gambar 5.8 menggunakan diagram Venn untuk menggambarkan penggunaan kode ini pada kata-kata 4-bit ( M = 4). Dengan tiga lingkaran berpotongan, ada tujuh kompartemen. Kami menetapkan 4 bit data ke kata-kata 4-bit ( M = 4). Dengan tiga lingkaran berpotongan, ada tujuh kompartemen. Kami menetapkan 4 bit data ke kata-kata 4-bit ( M = 4). Dengan tiga lingkaran berpotongan, ada tujuh kompartemen. Kami menetapkan 4 bit data ke kompartemen bagian dalam (Gambar 5.8a). Kompartemen sisanya dipenuhi dengan apa yang disebut paritas bit. Setiap kompartemen bagian dalam (Gambar 5.8a). Kompartemen sisanya dipenuhi dengan apa yang disebut paritas bit. Setiap kompartemen bagian dalam (Gambar 5.8a). Kompartemen sisanya dipenuhi dengan apa yang disebut paritas bit. Setiap bit paritas dipilih sehingga jumlah total 1s dalam lingkaran yang bahkan (Gambar 5.8b).

Dengan demikian, karena lingkaran A mencakup tiga 1s data, bit paritas dalam lingkaran yang diatur ke 1. Sekarang, jika kesalahan perubahan salah satu bit data (Gambar 5.8c), itu mudah ditemukan. Dengan memeriksa paritas bit, perbedaan yang ditemukan dalam lingkaran A dan lingkaran C tapi tidak dalam lingkaran B. Hanya satu dari tujuh kompartemen di A dan C tetapi tidak B (Gambar 5.8d). Kesalahan karena itu dapat diperbaiki dengan mengubah sedikit itu.

Untuk memperjelas konsep-konsep yang terlibat, kami akan mengembangkan kode yang dapat mendeteksi dan benar single-bit kesalahan dalam kata-kata 8-bit. Untuk memulai, mari kita menentukan berapa lama kode harus. Mengacu pada Gambar 5.7, logika perbandingan menerima sebagai masukan dua K nilai-nilai bit. Sebuah perbandingan bit-by-bit dilakukan dengan perbandingan menerima sebagai masukan dua K nilai-nilai bit. Sebuah perbandingan bit-by-bit dilakukan dengan perbandingan menerima sebagai masukan dua K nilai-nilai bit. Sebuah perbandingan bit-by-bit dilakukan dengan mengambil eksklusif-OR dari dua input. Hasilnya disebut sindrom kata. Dengan demikian, setiap bit dari sindroma adalah mengambil eksklusif-OR dari dua input. Hasilnya disebut sindrom kata. Dengan demikian, setiap bit dari sindroma adalah mengambil eksklusif-OR dari dua input. Hasilnya disebut sindrom kata. Dengan demikian, setiap bit dari sindroma adalah mengambil eksklusif-OR dari dua input. Hasilnya disebut sindrom kata. Dengan demikian, setiap bit dari sindroma adalah mengambil eksklusif-OR dari dua input. Hasilnya disebut sindrom kata. Dengan demikian, setiap bit dari sindroma adalah 0 atau 1 sesuai jika ada atau tidak cocok dalam posisi bit untuk dua input.

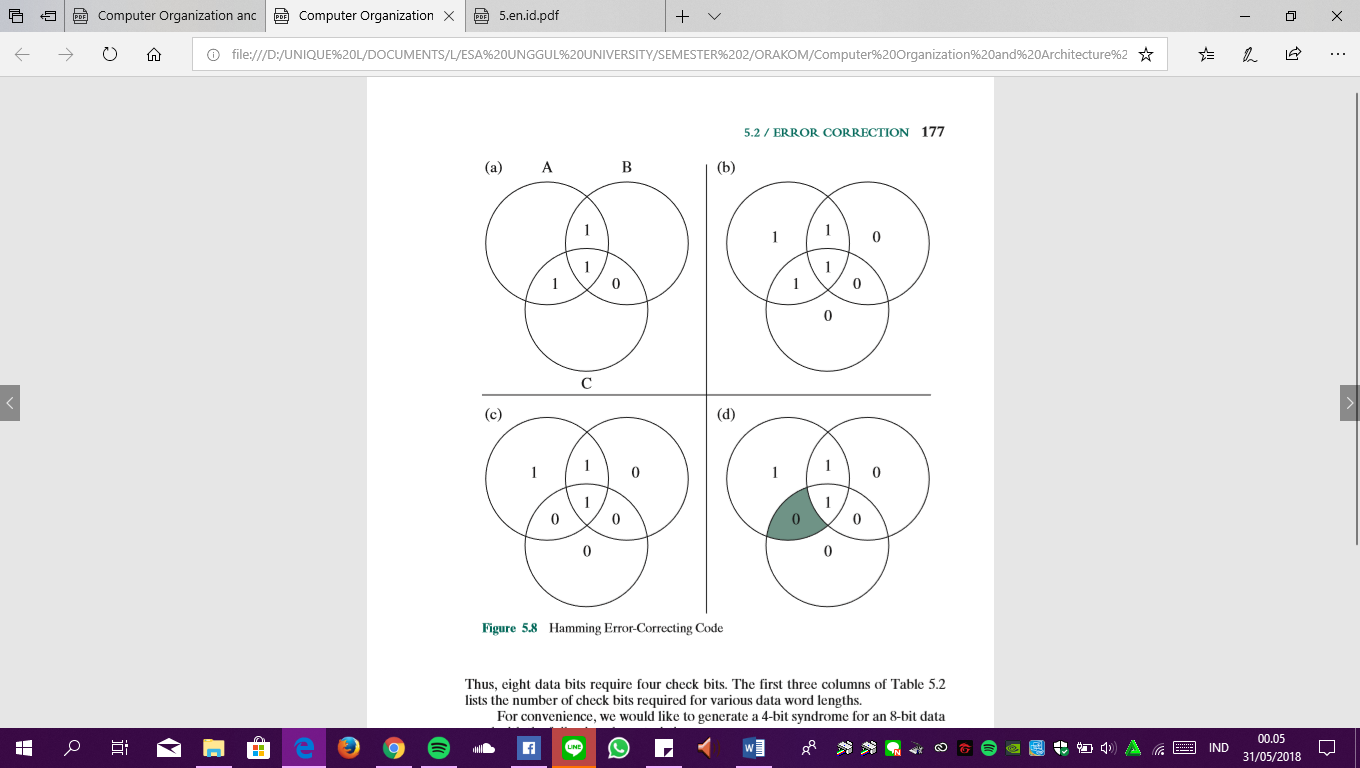
Oleh karena itu kata syndrome adalah K bit lebar dan memiliki jangkauan antara 0 dan 2 K 1. Nilai 0 menunjukkan Oleh karena itu kata syndrome adalah K bit lebar dan memiliki jangkauan antara 0 dan 2 K 1. Nilai 0 menunjukkan Oleh karena itu kata syndrome adalah K bit lebar dan memiliki jangkauan antara 0 dan 2 K 1. Nilai 0 menunjukkan Oleh karena itu kata syndrome adalah K bit lebar dan memiliki jangkauan antara 0 dan 2 K 1. Nilai 0 menunjukkan Oleh karena itu kata syndrome adalah K bit lebar dan memiliki jangkauan antara 0 dan 2 K 1. Nilai 0 menunjukkan bahwa tidak ada kesalahan terdeteksi, meninggalkan 2 K 1 nilai untuk menunjukkan, jika ada kesalahan, yang sedikit adalah bahwa tidak ada kesalahan terdeteksi, meninggalkan 2 K 1 nilai untuk menunjukkan, jika ada kesalahan, yang sedikit adalah bahwa tidak ada kesalahan terdeteksi, meninggalkan 2 K 1 nilai untuk menunjukkan, jika ada kesalahan, yang sedikit adalah keliru. Sekarang, karena kesalahan bisa terjadi pada salah satu M Data bit atau K memeriksa bit, kita harus memilikikeliru. Sekarang, karena kesalahan bisa terjadi pada salah satu M Data bit atau K memeriksa bit, kita harus memilikikeliru. Sekarang, karena kesalahan bisa terjadi pada salah satu M Data bit atau K memeriksa bit, kita harus memilikikeliru. Sekarang, karena kesalahan bisa terjadi pada salah satu M Data bit atau K memeriksa bit, kita harus memilikikeliru. Sekarang, karena kesalahan bisa terjadi pada salah satu M Data bit atau K memeriksa bit, kita harus memiliki

2 K 1 Ú M + K 2 K 1 Ú M + K 2 K 1 Ú M + K 2 K 1 Ú M + K 2 K 1 Ú M + K

Ketimpangan ini memberikan jumlah bit yang diperlukan untuk memperbaiki kesalahan bit tunggal dalam kata yang mengandung M bit data. Sebagai contoh, untuk sebuah kata dari 8 bit data ( M = 8), kita memilikimengandung M bit data. Sebagai contoh, untuk sebuah kata dari 8 bit data ( M = 8), kita memilikimengandung M bit data. Sebagai contoh, untuk sebuah kata dari 8 bit data ( M = 8), kita memilikimengandung M bit data. Sebagai contoh, untuk sebuah kata dari 8 bit data ( M = 8), kita memilikimengandung M bit data. Sebagai contoh, untuk sebuah kata dari 8 bit data ( M = 8), kita memiliki

■ K = 3: 2 3- 1 6 8 + 3 K = 3: 2 3- 1 6 8 + 3 K = 3: 2 3- 1 6 8 + 3 K = 3: 2 3- 1 6 8 + 3 ■ K = 3: 2 3- 1 6 8 + 3 K = 3: 2 3- 1 6 8 + 3

■ K = 4: 2 4- 1 7 8 + 4

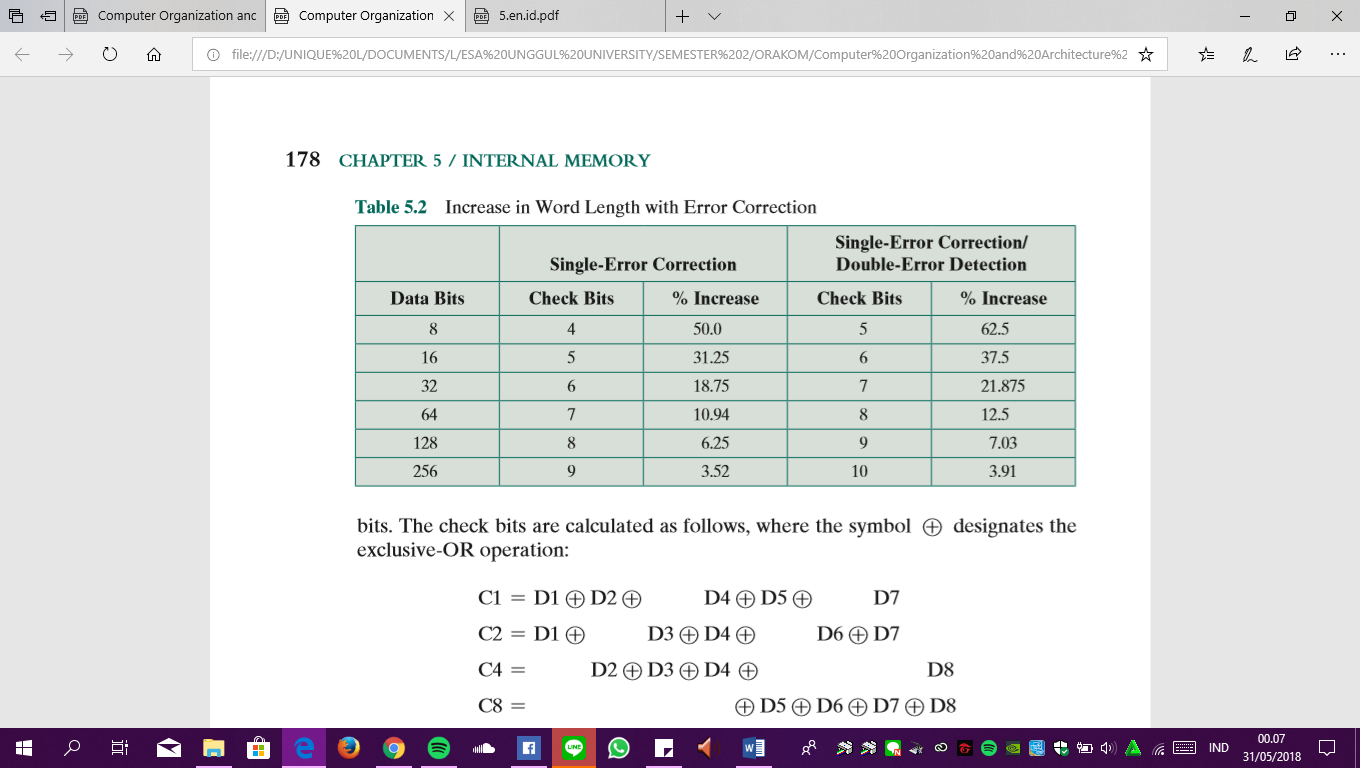


Dengan demikian, delapan bit data memerlukan empat bit cek. Tiga kolom pertama dari Tabel 5.2 daftar jumlah cek bit yang diperlukan untuk berbagai panjang word data. Untuk kenyamanan, kami ingin menghasilkan sindrom 4-bit untuk kata data 8-bit dengan karakteristik sebagai berikut:

■ Jika sindrom berisi semua 0s, tidak ada kesalahan telah terdeteksi.

■ Jika sindrom berisi satu dan hanya satu bit set ke 1, maka telah terjadi kesalahan dalam salah satu dari 4 bit check. Tidak ada koreksi yang diperlukan.

■ Jika sindrom berisi lebih dari satu bit set ke 1, maka nilai numerik dari sindrom menunjukkan posisi bit data dalam kesalahan. bit data ini terbalik untuk koreksi.



C1 = D1 ⊕ D2 ⊕ D4 ⊕ D5 ⊕ D7

C2 = D1 ⊕ D3 ⊕ D4 ⊕ D6 ⊕ D7

C4 = D2 ⊕ D3 ⊕ D4 ⊕ D8

C8 = ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D8

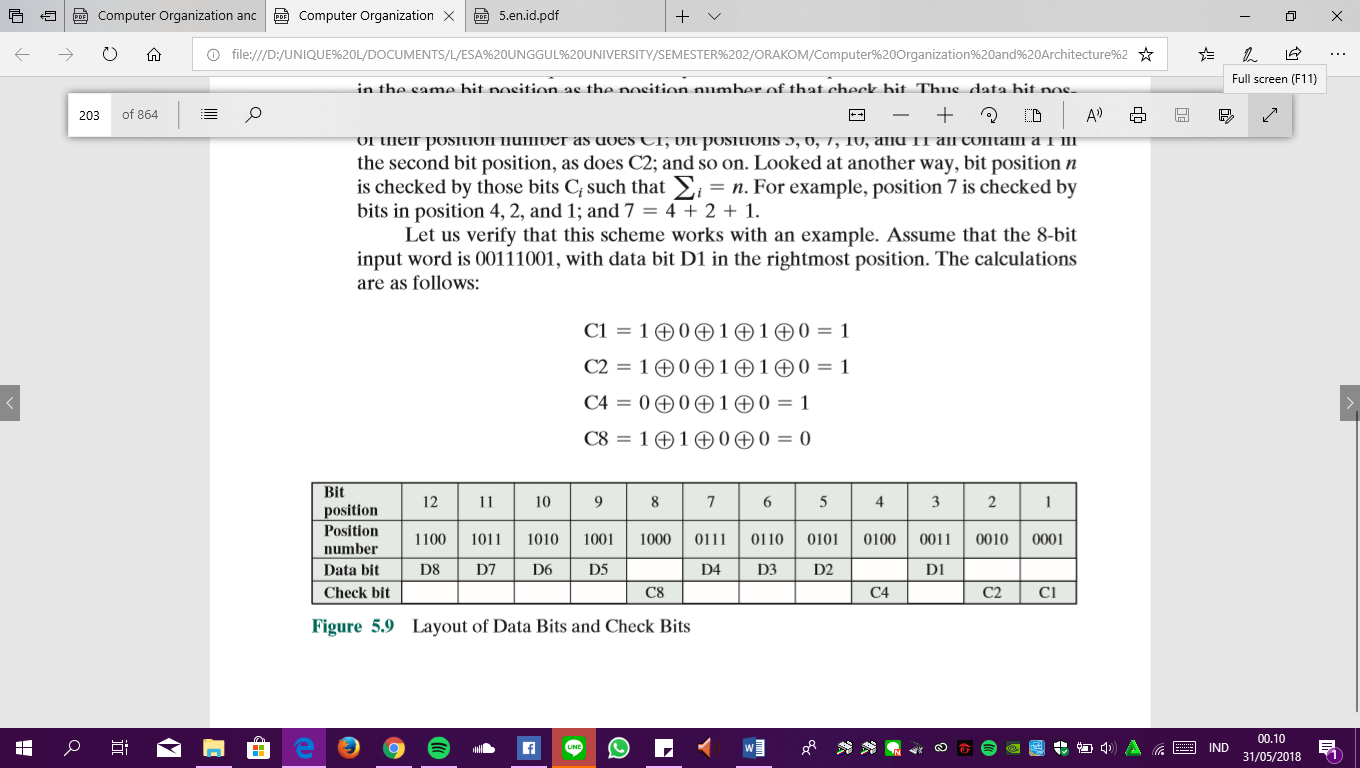
Setiap bit cek beroperasi pada setiap bit data yang nomor posisinya berisi 1 dalam posisi bit sama dengan jumlah posisi yang sedikit cek. Dengan demikian, data bit posisi 3, 5, 7, 9, dan 11 (D1, D2, D4, D5, D7) semua mengandung 1 di bit paling signifikan dari nomor posisi mereka seperti halnya C1; posisi bit 3, 6, 7, 10, dan 11 semua mengandung 1 di posisi bit kedua, seperti halnya C2; dan seterusnya. Memandang cara lain, posisi bit n posisi bit n diperiksa oleh orang-bit C saya seperti yang Sebuah i = n. Misalnya, posisi 7 diperiksa oleh bit di posisi 4, 2, dan diperiksa oleh orang-bit C saya seperti yang Sebuah i = n. Misalnya, posisi 7 diperiksa oleh bit di posisi 4, 2, dan diperiksa oleh orang-bit C saya seperti yang Sebuah i = n. Misalnya, posisi 7 diperiksa oleh bit di posisi 4, 2, dan diperiksa oleh orang-bit C saya seperti yang Sebuah i = n. Misalnya, posisi 7 diperiksa oleh bit di posisi 4, 2, dan diperiksa oleh orang-bit C saya seperti yang Sebuah i = n. Misalnya, posisi 7 diperiksa oleh bit di posisi 4, 2, dan diperiksa oleh orang-bit C saya seperti yang Sebuah i = n. Misalnya, posisi 7 diperiksa oleh bit di posisi 4, 2, dan diperiksa oleh orang-bit C saya seperti yang Sebuah i = n. Misalnya, posisi 7 diperiksa oleh bit di posisi 4, 2, dan 1; dan 7 = 4 + 2 + 1. 1; dan 7 = 4 + 2 + 1. Mari kita memverifikasi bahwa skema ini bekerja dengan sebuah contoh. Asumsikan bahwa kata input 8-bit adalah 00.111.001, dengan data bit D1 di posisi paling kanan. Perhitungan adalah sebagai berikut:

C1 = D1 ⊕ D2 ⊕ D4 ⊕ D5 ⊕ D7

C2 = D1 ⊕ D3 ⊕ D4 ⊕ D6 ⊕ D7

C4 = D2 ⊕ D3 ⊕ D4 ⊕ D8

C8 = ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D8



Misalkan sekarang bahwa data bit 3 mendukung kesalahan dan diubah dari 0 ke 1. Ketika bit cek dihitung ulang, kita harus

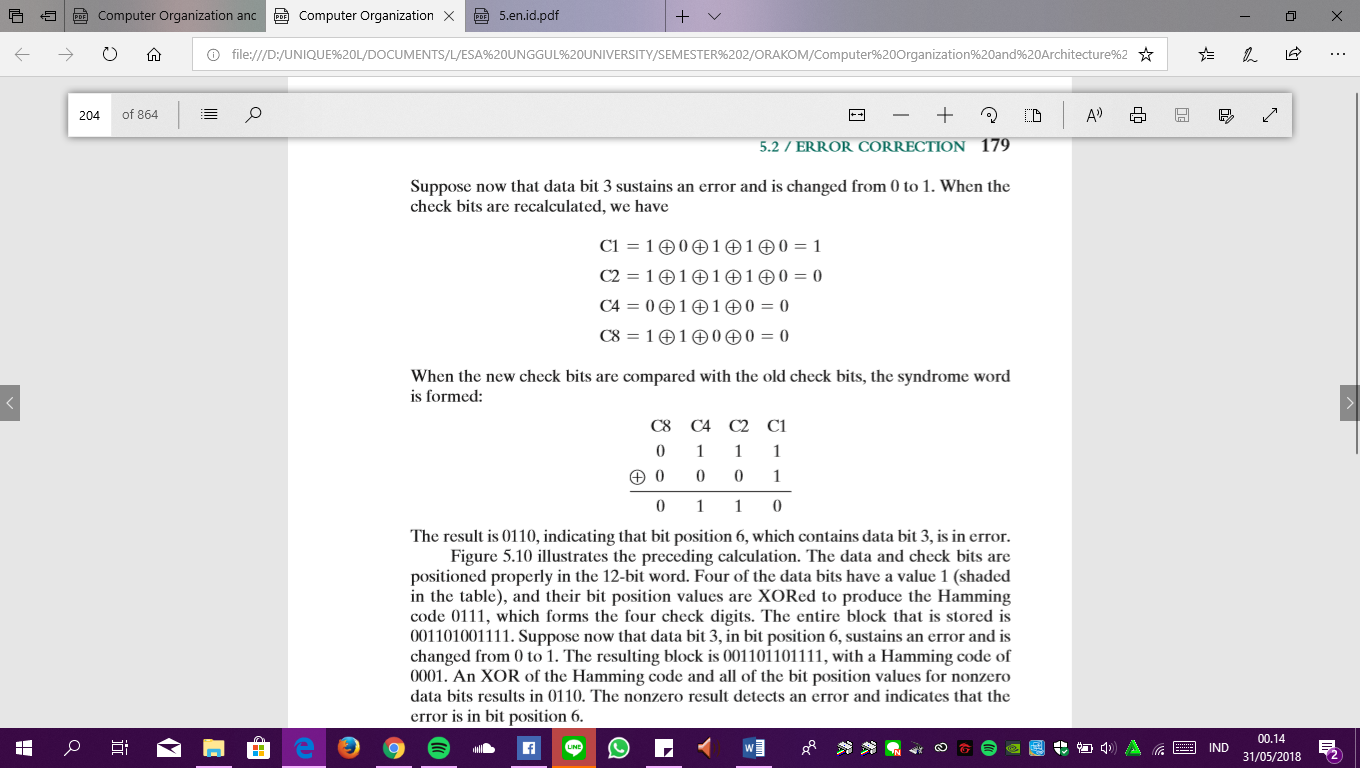
C1 = 1 ⊕ 0 ⊕ 1 ⊕ 1 ⊕ 0 = 1

C2 = 1 ⊕ 1 ⊕ 1 ⊕ 1 ⊕ 0 = 0

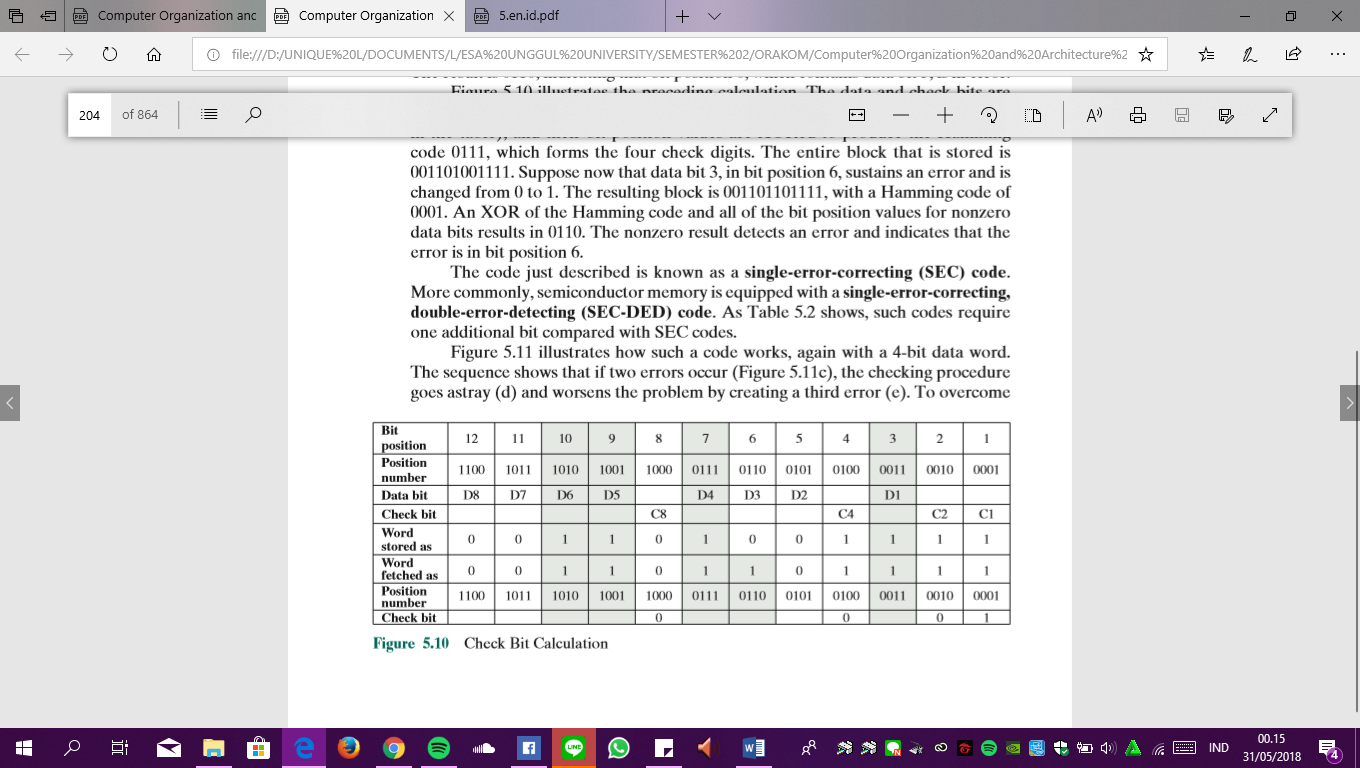
C4 = 0 ⊕ 1 ⊕ 1 ⊕ 0 = 0

C8 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0

Ketika cek bit baru dibandingkan dengan bit check tua, kata sindrom terbentuk:



Hasilnya adalah 0110, menunjukkan bahwa posisi bit 6, yang berisi data bit 3, adalah kesalahan. Gambar 5.10 mengilustrasikan perhitungan sebelumnya. Data dan cek bit diposisikan dengan benar dalam kata 12-bit. Empat dari data bit memiliki nilai 1 (diarsir dalam tabel), dan nilai-nilai posisi bit mereka XOR untuk menghasilkan kode Hamming 0111, yang membentuk empat digit cek. Seluruh blok yang disimpan adalah 001101001111. Misalkan sekarang bit data 3, di bit posisi 6, menopang kesalahan dan berubah dari 0 ke 1. blok yang dihasilkan adalah 001101101111, dengan kode Hamming dari 0001. Sebuah XOR dari kode Hamming dan semua bit nilai posisi nol hasil data bit di 0110. Hasil nol mendeteksi kesalahan dan menunjukkan bahwa kesalahan adalah dalam sedikit posisi 6. kode yang baru saja dijelaskan dikenal sebagai single-error-correcting (SEC) kode. kode yang baru saja dijelaskan dikenal sebagai single-error-correcting (SEC) kode. Lebih umum, memori semikonduktor dilengkapi dengan single-error-correcting, double-kesalahan-mendeteksi Lebih umum, memori semikonduktor dilengkapi dengan single-error-correcting, double-kesalahan-mendeteksi (SEC-DED) kode. Seperti Tabel 5.2 menunjukkan, kode tersebut memerlukan satu bit tambahan dibandingkan dengan (SEC-DED) kode. Seperti Tabel 5.2 menunjukkan, kode tersebut memerlukan satu bit tambahan dibandingkan dengan kode SEC. Gambar 5.11 menggambarkan bagaimana kode tersebut bekerja, lagi dengan kata data yang 4-bit. urutan menunjukkan bahwa jika dua kesalahan terjadi (Gambar 5.11c), prosedur pengecekan tersesat (d) dan memperburuk masalah dengan membuat kesalahan ketiga (e). Untuk mengatasi masalah, bit kedelapan ditambahkan yang telah ditetapkan, sehingga jumlah total 1s dalam diagram bahkan. Bit paritas ekstra menangkap kesalahan (f). Kode error-correcting meningkatkan keandalan memori pada biaya tambahan kompleksitas. Dengan organisasi 1-bit-per-chip, kode SEC-DED umumnya dianggap memadai. Sebagai contoh, implementasi 30XX IBM menggunakan sebuah 8-bit SEC-DED kode untuk setiap 64 bit data di memori utama. Dengan demikian, ukuran memori utama sebenarnya sekitar 12% lebih besar daripada yang jelas bagi pengguna. Komputer VAX menggunakan 7-bit SEC-DED untuk setiap 32 bit memori, untuk overhead 22%. sistem DRAM kontemporer mungkin memiliki mana saja dari 7% menjadi 20% di atas kepala [SHAR03].



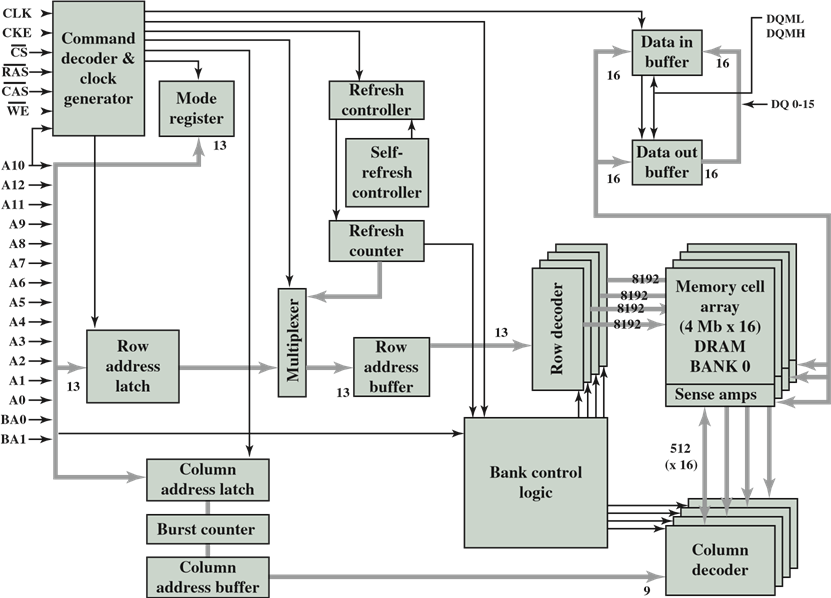


2.3 DDR DRAM

Salah satu hambatan sistem paling kritis ketika menggunakan prosesor berkinerja tinggi adalah antarmuka ke memori utama internal, yang mana antarmuka ini merupakan jalur paling penting di seluruh sistem komputer. Chip DRAM tradisional dibatasi baik oleh arsitektur internalnya dan oleh antarmukanya ke bus memori prosesor. Serangan yang paling berpengaruh terhadap masalah kinerja memori utama DRAM adalah memasukkan satu atau lebih tingkat penyembunyian SRAM berkecepatan tinggi antara memori utama DRAM dan prosesor. Kekurangan SRAM yaitu SRAM jauh lebih mahal daripada DRAM.

DRAM sinkron (SDRAM), adalah salah satu bentuk DRAM yang paling banyak digunakan. Berbeda dengan DRAM tradisional (tidak sinkron), SDRAM menukar data dengan prosesor yang disinkronkan ke sinyal clock eksternal dan berjalan pada kecepatan penuh dari bus prosesor / memori tanpa memaksakan status menunggu. Perbedaan DRAM biasa dengan SDRAM yaitu :

* Dalam DRAM biasa, prosesor menyajikan alamat dan tingkat kontrol ke memori, yang menunjukkan bahwa satu set data di lokasi tertentu dalam memori harus dibaca dari atau ditulis ke dalam DRAM. Setelah penundaan, waktu akses, DRAM menulis atau membaca data. Selama penundaan waktu akses, DRAM melakukan berbagai fungsi internal, seperti mengaktifkan kapasitansi tinggi dari baris baris dan kolom, merasakan data, dan routing data keluar melalui buffer output. Prosesor harus menunggu penundaan ini, memperlambat kinerja sistem.
* Dengan akses sinkron, DRAM memindahkan data masuk dan keluar di bawah kendali jam sistem. Prosesor mengeluarkan instruksi dan informasi alamat, yang dikunci oleh DRAM. DRAM kemudian merespon setelah sejumlah siklus jam tertentu.



* Gambar di atas menunjukkan logika internal SDRAM tipe 256-Mb dari tipe organisasi SDRAM.

|  |  |
| --- | --- |
| A0 to A13 | Address inputs |
| BA0, BA1 | Bank address lines |
| CLK | Clock input |
| CKE | Clock enable |
| CS | Chip select |
| RAS | Row address strobe |
| CAS | Column address strobe |
| WE | Write enable |
| DQ0 to DQ7 | Data input/output |
| DQM | Data mask |

Tabel tersebut mendefinisikan berbagai macam tugas pin.

SDRAM menggunakan mode burst untuk menghilangkan waktu pengaturan alamat dan baris dan waktu baris precharge setelah akses pertama. Dalam mode burst, serangkaian bit data dapat cepat habis setelah bit pertama telah diakses. Mode ini berguna ketika semua bit yang akan diakses berurutan dan di baris yang sama dari array sebagai akses awal. Mode register dan logika kontrol yang terkait adalah fitur kunci lain yang membedakan SDRAM dari DRAM konvensional. Register mode menentukan panjang burst, yang merupakan jumlah unit data yang terpisah yang secara sinkron dimasukkan ke dalam bus. Selain itu, register juga memungkinkan programmer untuk menyesuaikan latensi antara penerimaan permintaan yang dibaca dan permulaan transfer data. Kinerja SDRAM terbaik adalah ketika mentransfer sekumpulan besar data secara berurutan, seperti untuk aplikasi seperti pengolah kata, spreadsheet, dan multimedia.

DDR SDRAM, SDRAM masih memiliki kekurangan yang tidak perlu membatasi I / O data rate yang dapat dicapai. Untuk mengatasi kekurangan ini, versi lebih baru dari SDRAM disebut sebagai DRAM data ganda (DDR DRAM) menyediakan beberapa fitur yang secara dramatis meningkatkan laju data. DDR DRAM dikembangkan oleh JEDEC Solid State Technology Association, badan semikonduktor-rekayasa standardisasi Aliansi Industri Elektronik. DDR mencapai tingkat data yang lebih tinggi dalam tiga cara, yaitu :

1. Transfer data disinkronkan ke tepi naik dan turunnya waktu, bukan hanya tepi yang naik. Ini menggandakan penilaian data; maka diistilahkan sebagai laju data ganda.
2. DDR menggunakan penilaian waktu yang lebih tinggi di bus untuk meningkatkan kecepatan transfer.
3. Skema penyangga digunakan, seperti yang dijelaskan selanjutnya.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **DDR1** | **DDR2** | **DDR3** | **DDR4** |
| Prefetch buffer (bits) | 2 | 4 | 8 | 8 |
| Voltage level (V) | 2.5 | 1.8 | 1.5 | 1.2 |
| Front side bus data rates (Mbps) | 200—400 | 400—1066 | 800—2133 | 2133—4266 |

**1N Memory array (100–150 MHz) I/O (100–150 MHz) SDRAM**

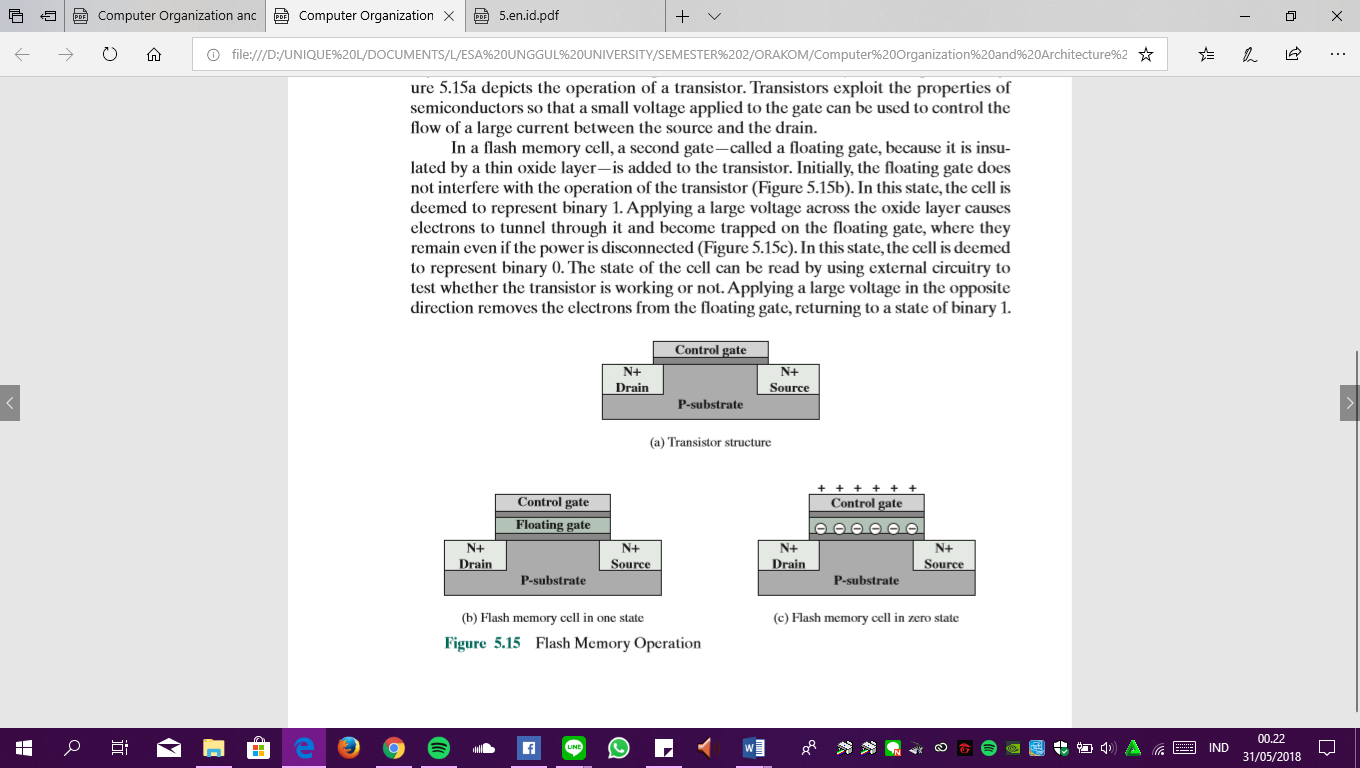
Seperti yang tercantum dalam tabel di atas, JEDEC sejauh ini telah menetapkan empat generasi teknologi DDR Versi DDR awal menggunakan buffer prefetch 2-bit. Buffer prefetch adalah penyimpanan memori yang terletak di chip SDRAM. Hal ini memungkinkan chip SDRAM ke bit preposisi untuk ditempatkan pada bus data secepat mungkin. Bus DDR I / O menggunakan penilaian waktu yang sama dengan chip memori, tetapi karena ia dapat menangani dua bit per siklus, maka ia mencapai kecepatan data yang menggandakan laju waktu. Buffer prefetch 2-bit memungkinkan chip SDRAM untuk mengikuti bus I / O.

Pengoperasian buffer prefetch dilihat dari sudut pandang transfer kata. Ukuran buffer prefetch menentukan berapa banyak kata data yang diambil (di beberapa chip SDRAM) setiap kali perintah kolom dilakukan dengan memori DDR. Karena inti DRAM jauh lebih lambat daripada antarmuka, perbedaannya dijembatani dengan mengakses informasi secara paralel dan kemudian membuat serialisasi antarmuka melalui multiplexor (MUX). Dengan demikian, DDR mengambil dua kata, yang berarti bahwa setiap kali operasi baca atau tulis dilakukan, maka dilakukan pada dua kata data, dan burst keluar, atau ke dalam SDRAM melalui satu siklus waktu pada kedua sisi waktu untuk total dua operasi berturut-turut. Akibatnya, antarmuka DDR I / O dua kali lebih cepat dari inti SDRAM. Kekurangan dari prefetch yaitu secara efektif menentukan panjang minimum burst SDRAM.

2.4 Flash Memory

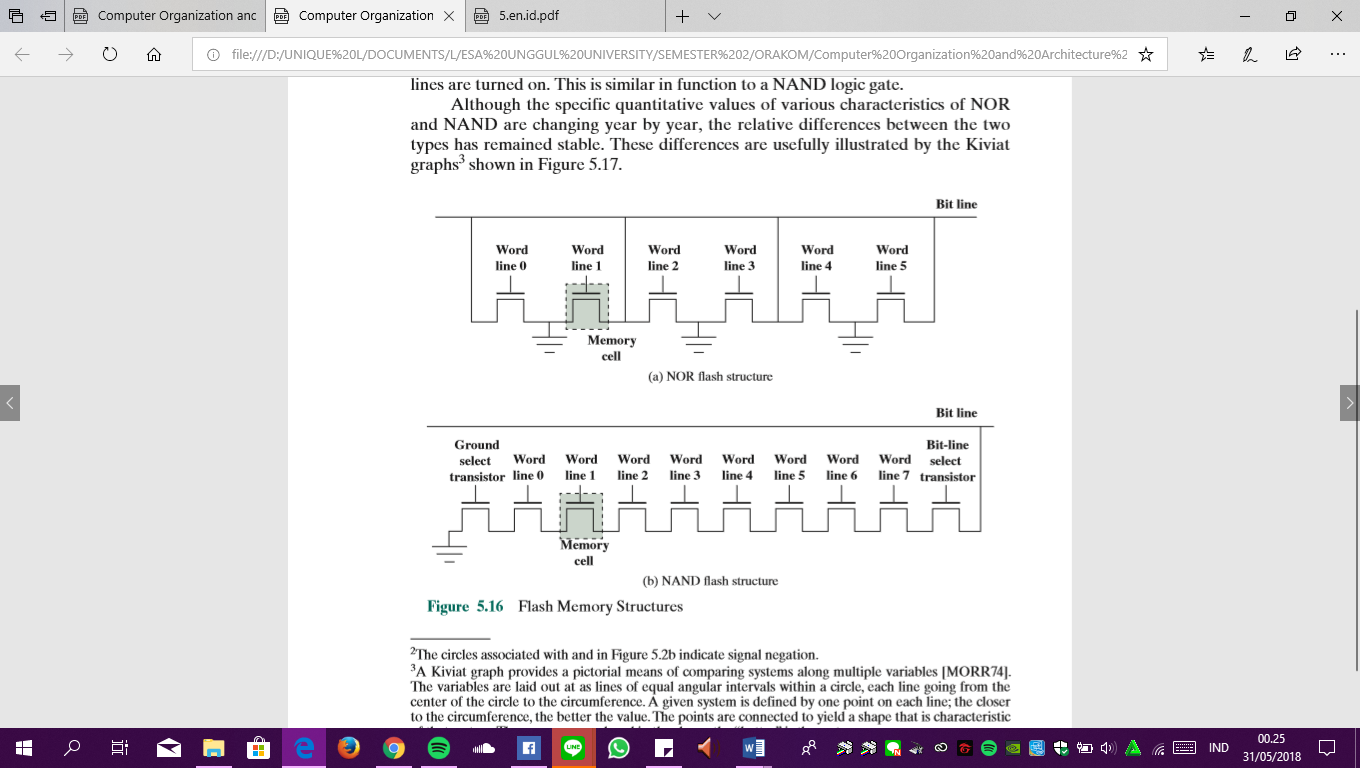
Bentuk lain dari memori semikonduktor adalah memori flash. Flash memory digunakan baik untuk memori internal dan aplikasi memori eksternal. Di sini, kami memberikan gambaran teknis dan melihat penggunaannya untuk memori internal. Pertama kali diperkenalkan pada pertengahan 1980-an, memori flash adalah penengah antara EPROM dan EEPROM dalam biaya dan fungsionalitas. Seperti EEPROM, flash memori menggunakan teknologi menghapus listrik. Seluruh memori flash dapat dihapus dalam satu atau beberapa detik, yang jauh lebih cepat daripada EPROM. Selain itu, adalah mungkin untuk menghapus hanya blok memori daripada seluruh chip yang. memori flash mendapatkan namanya karena microchip ini disusun sehingga bagian sel memori akan terhapus dalam aksi tunggal atau “flash.” Namun, memori flash tidak menyediakan penghapusan bytelevel. Seperti EPROM, memori flash hanya menggunakan satu transistor per bit, sehingga mencapai kepadatan tinggi (dibandingkan dengan EEPROM) dari EPROM.

Operasi Gambar 5.15 mengilustrasikan operasi dasar dari memori flash. Sebagai perbandingan, Gambar 5.15a menggambarkan operasi transistor. Transistor mengeksploitasi sifat semikonduktor sehingga tegangan kecil diterapkan ke pintu gerbang dapat digunakan untuk mengontrol aliran arus yang besar antara sumber dan sia-sia. Dalam sel memori flash, kedua gerbang-disebut gerbang mengambang, karena terisolasi oleh oksida tipis lapisan-ditambahkan ke transistor. Awalnya, gerbang mengambang tidak mengganggu operasi dari transistor (Gambar 5.15b). Dalam keadaan ini, sel dianggap mewakili biner 1. Menerapkan tegangan besar di lapisan oksida menyebabkan elektron untuk terowongan melalui itu dan menjadi terjebak di gerbang mengambang, di mana mereka tetap bahkan jika daya terputus (Gambar 5.15c). Dalam keadaan ini, sel dianggap mewakili biner 0. Keadaan sel dapat dibaca dengan menggunakan sirkuit eksternal untuk menguji apakah transistor bekerja atau tidak. Menerapkan tegangan besar dalam arah yang berlawanan menghilangkan elektron dari gerbang mengambang, kembali ke keadaan biner 1.

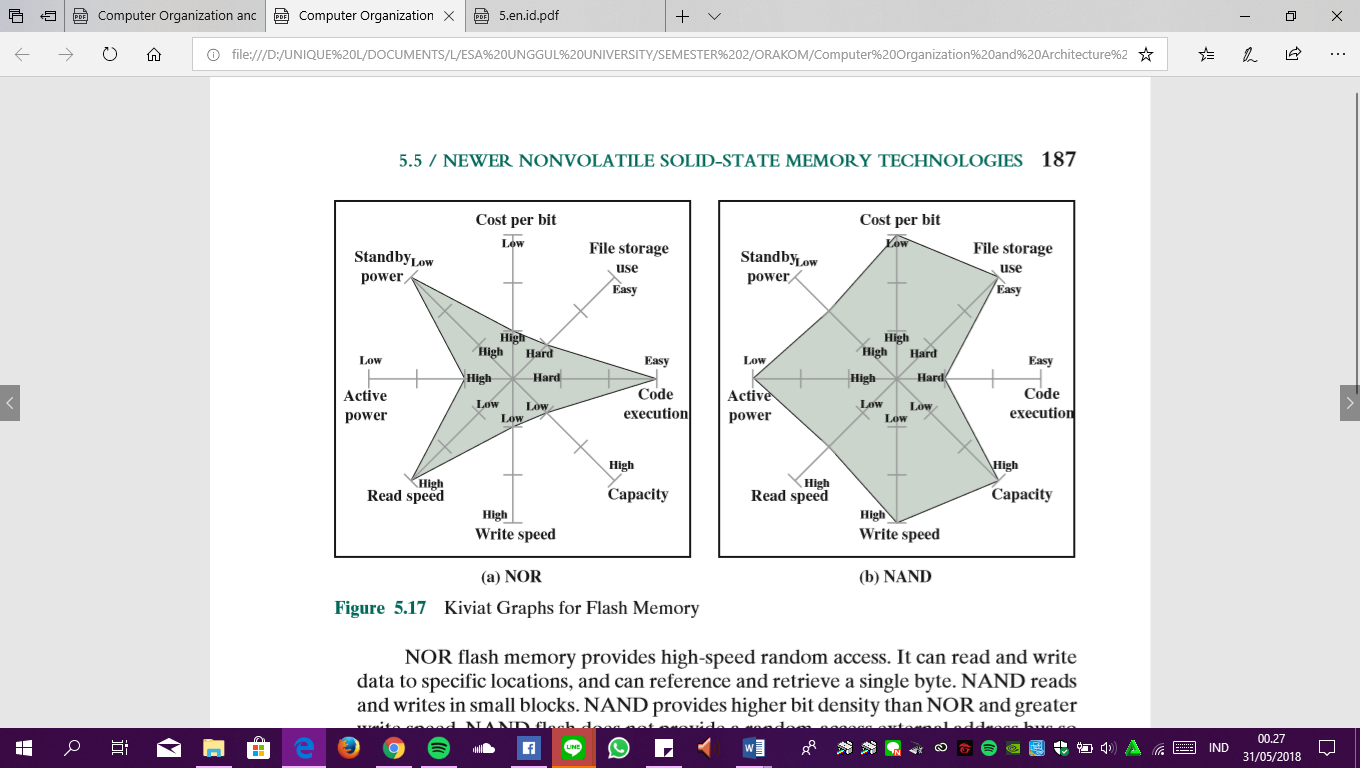


NOR dan NAND Flash Memory

Ada dua jenis khas dari memori flash, ditunjuk sebagai NOR dan NAND (Gambar 5.16). Di NOR flash Ada dua jenis khas dari memori flash, ditunjuk sebagai NOR dan NAND (Gambar 5.16). Di NOR flash memory, unit dasar akses sedikit, disebut sebagai memory, unit dasar akses sedikit, disebut sebagai sel memori. Sel-sel di NOR flash terhubung secara paralel dengan garis bit sehingga setiap sel dapat membaca / sel memori. Sel-sel di NOR flash terhubung secara paralel dengan garis bit sehingga setiap sel dapat membaca / menulis / terhapus secara individual. Jika ada sel memori dari perangkat dihidupkan oleh garis kata yang sesuai, garis bit pergi rendah. Hal ini mirip dengan fungsi NOR gerbang logika. 2 garis bit pergi rendah. Hal ini mirip dengan fungsi NOR gerbang logika

NAND flash memory diselenggarakan di array transistor dengan 16 atau 32 transistor dalam seri. Bit NAND flash memory diselenggarakan di array transistor dengan 16 atau 32 transistor dalam seri. Bit baris pergi rendah hanya jika semua transistor dalam garis kata yang sesuai diaktifkan. Hal ini mirip dengan fungsi gerbang NAND logika. Meskipun nilai kuantitatif spesifik berbagai karakteristik NOR dan NAND berubah dari tahun ke tahun, perbedaan relatif antara dua jenis tetap stabil. Perbedaan-perbedaan ini berguna diilustrasikan oleh grafik Kiviat 3 ditunjukkan pada Gambar 5.17.

2 Lingkaran yang berhubungan dan pada Gambar 5.2b menunjukkan sinyal negasi. 2 Lingkaran yang berhubungan dan pada Gambar 5.2b menunjukkan sinyal negasi. 3 Sebuah grafik Kiviat menyediakan sarana bergambar membandingkan sistem bersama beberapa variabel [MORR74]. Variabel diletakkan 3 Sebuah grafik Kiviat menyediakan sarana bergambar membandingkan sistem bersama beberapa variabel [MORR74]. Variabel diletakkan pada sebagai garis interval sudut yang sama dalam lingkaran, setiap baris pergi dari pusat lingkaran ke keliling. Sebuah sistem yang diberikan didefinisikan oleh satu titik pada setiap baris; lebih dekat ke lingkar, semakin baik nilai. Poin-poin yang terhubung untuk menghasilkan bentuk yang karakteristik dari sistem itu. Semakin banyak daerah tertutup di bentuk, “lebih baik” adalah sistem.



NOR flash memory menyediakan akses random kecepatan tinggi. Hal ini dapat membaca dan menulis data ke lokasi tertentu, dan dapat referensi dan mengambil satu byte. NAND membaca dan menulis di blok kecil. NAND memberikan sedikit kepadatan lebih tinggi dari NOR dan kecepatan tulis yang lebih besar. Flash NAND tidak menyediakan akses-acak eksternal bus alamat sehingga data harus dibaca secara blockwise (juga dikenal sebagai akses halaman), di mana setiap blok memegang ratusan hingga ribuan bit.

Untuk memori internal dalam sistem embedded, memori flash NOR secara tradisional telah disukai. memori NAND telah membuat beberapa terobosan, tetapi NOR tetap teknologi yang dominan untuk memori internal. Ini cocok untuk mikrokontroler di mana jumlah kode program relatif kecil dan sejumlah data aplikasi tidak bervariasi. Sebagai contoh, memori flash pada Gambar 1.16 adalah NOR memori.

Memori NAND lebih cocok untuk memori eksternal, seperti USB flash drive, kartu memori (di kamera digital, MP3 player, dll), dan dalam apa yang dikenal sebagai disk solid-state (SSD). Kami membahas SSD dalam Bab 6.

**BAB 3**

**KESIMPULAN**

Sifat utama dari memori semikonduktor yaitu menunjukkan dua keadaan stabil yang dapat diwakili oleh 0 atau 1, dan mampu ditulis untuk mengatur keadaan tersebut, serta dapat dibaca untuk merasakan keadaannya. Memori akses acak adalah memori yang mudah menguap yang mana data dapat dibaca dan ditulis dengan mudah dan cepat. Perbedaan antara DRAM dan SRAM dalam hal aplikasi yaitu DRAM digunakan untuk memori utama, dan SRAM digunakan untuk memori cache. Aplikasi ROM berupa microprogramming, library subroutines untuk fungsi yang sering dicari, sistem program, dan tabel fungsi.

EPROM adalah memori yang dapat diprogram yang bisa dihapus yang mana hanya bisa dibaca tetapi bisa dihapus dengan sinar ultraviolet, memungkinkan memori untuk kemudian ditulis ulang. EEPROM seperti EPROM, tetapi dapat dihapus dengan muatan listrik, sementara memori flash adalah perantara antara EPROM dan EEPROM. Perbedaan SDRAM dengan DRAM yaitu SDRAM disinkronkan ke sinyal clock eksternal dan berjalan dengan kecepatan penuh dari bus prosesor / memori tanpa memaksakan status menunggu, sementara DRAM tidak sinkron. Lalu perbedaan antara memori flash NAND dan NOR yaitu NOR dapat memberikan kecepatan baca lebih cepat dan kemampuan akses acak, sehingga cocok untuk penyimpanan kode di perangkat seperti PDA dan ponsel. Sedangkan, NAND dapat memberikan kemampuan tulis / hapus yang cepat dan lebih lambat daripada NOR di area kecepatan baca. Bagaimanapun, NAND lebih bagus untuk sebagian besar aplikasi konsumen seperti video digital, musik atau penyimpanan data.