

PERTEMUAN IV

MEMORI INTERNAL

Karakteristik Memori

1. LOKASI dan KAPASITAS

- ☐ LOKASI terdapat
 - ☐ CPU : yang mempunyai memori tersendiri dalam bentuk register
 - ☐ Internal : sering disebut **memori utama** adalah *dapat diakses secara langsung oleh prosesor*
 - ☐ External : adalah dapat diakses oleh prosesor melalui modul I/O terdiri dari perangkat storage peripheral (spt disk, pita dll)
- ☐ KAPASITAS dinyatakan dalam **byte** (1byte : 8 bit) *atau* **word** (umumnya panjangnya 8,16,32 bit)

2. SATUAN TRANSFER

- ☐ **Satuan transfer** = jumlah saluran data yang masuk ke dan keluar dari modul memori.
 - ☐ Usually governed by data bus width
- ☐ **Ada 3 konsep dalam satuan transfer:**
 - ☐ Word : Ukuran word biasanya = jumlah bit yg digunakan untuk representasi bilangan dan panjang intruksi
 - ☐ Addressable Unit : hubungan jumlah addressable unit (N) dg panjang (A) suatu alamat adalah $N = 2^A$
 - ☐ Unit of Transfer : *adalah* jumlah bit yg dibaca/ditulis ke dalam memori pada suatu saat

3. METODE AKSES

- ☐ **Sequential**
 - ☐ Memori diorganisasikan menjadi unit-unit data yang disebut **record**
 - ☐ Mekanisme baca/tulis digunakan bersama-sama (shared)
 - ☐ Waktu aksesnya tergantung pada lokasi data dan lokasi sebelumnya
 - ☐ Contoh: pita
- ☐ **Direct**
 - ☐ Meliputi *shared read/write*
 - ☐ Setiap blok dan record memiliki alamat-alamat yang unik berdasarkan lokasi fisik
 - ☐ Waktu aksesnya tergantung pada lokasi data dan lokasi sebelumnya
 - ☐ Contoh: disk

4. METODE AKSES

- ☐ **Random**
 - ☐ Setiap lokasi dapat dipilih secara random dan diakses dan dialamati secara langsung
 - ☐ Waktu akses : tidak tergantung pada lokasi sebelumnya dan bersifat konstan
 - ☐ Sistem memori utama merupakan akses random
- ☐ **Associative**
 - ☐ Merupakan jenis akses random
 - ☐ Membandingkan lokasi bit yang diinginkan didalam word secara simultan
 - ☐ Sebuah word dicari berdasarkan isinya bukan alamatnya.
 - ☐ Contoh: cache

5. KINERJA

- ☐ Access time
 - ☐ Pada RAM : merupakan waktu yang dibutuhkan untuk operasi BACA/TULIS atau waktu dari suatu alamat diberikan memori sampai pada saat disimpan atau digunakan kembali.
 - ☐ Pada NON RAM : merupakan waktu yg dibutuhkan melakukan mekanisme baca tulis pada lokasi tertentu
- ☐ Memory Cycle time
 - ☐ Time may be required for the memory to “recover” before next access
 - ☐ Cycle time is access + recovery
- ☐ Transfer Rate
 - ☐ Kecepatan data agar dapat ditransfer ke unit memori atau dari unit memori
 - ☐ Pada RAM : transfer rate = $1/\text{waktu siklus}$
 - ☐ Pada NON RAM : $TN = TA + N/R$ $TN = \text{wkt rata-rata baca/tulis } N \text{ bit}$ $TA = \text{wkt akses rata-rata}$, $N = \text{Jumlah bit}$, $R = \text{kec.transfer (bps)}$

6. TIPE FISIK dan KARAKTER FISIK

- ☐ Memori utama dikemas dalam sebuah chip
- ☐ Dua jenis yang umum digunakan pada saat ini adalah:
 - ☐ Memori semikonduktor : memakai teknologi LSI/VLSI
 - ☐ Memori Magnetik : disk atau pita
- ☐ **Karakter Fisik ada 2** yaitu
 - ☐ Memori Volatile : membutuhkan sumber daya yg terus menerus untuk menyimpan informasinya.
 - ☐ Contoh : SRAM dan DRAM
 - ☐ Memori Non Volatile : tidak membutuhkan sumber daya yang terus menerus untuk menyimpan informasinya
 - ☐ Contoh : ROM

MEMORI SEMIKONDUKTOR

- ☐ **RAM**
 - ☐ Nama lain untuk semua memori semikonduktor adalah random access
 - ☐ Read/Write dg menggunakan signal-signal listrik
 - ☐ Volatile ☐ harus diberi suplai daya yg konstan
 - ☐ Temporary storage
 - ☐ Dibagi menjadi 2 : Static or dynamic
 - ☐ *DynamicRAM* : disusun oleh sel-sel yg menyimpan data sbg muatan listrik pada kapasitor (bilangan biner 0 dan 1) dan memerlukan pengisian muatan listrik secara priodik untuk memelihara penyimpanan data, sel memori dinamik lebih sederhana (lebih kecil), lebih rapat (sel lebih kecil = lebih banyak sel per satuan luas) dan lebih murah, memerlukan rangkaian pengosong muatan, cenderung digunakan untuk kebutuhan memori yg besar. Contoh main memori
 - ☐ *StaticRAM* : nilai-nilai biner disimpan dg menggunakan konfigurasi gerbang logika (on/off switch) dan dapat menampung data sepanjang daya listrik disediakan untuknya, lebih mahal, lebih besar dan lebih cepat. Contoh Cache

☐ ROM

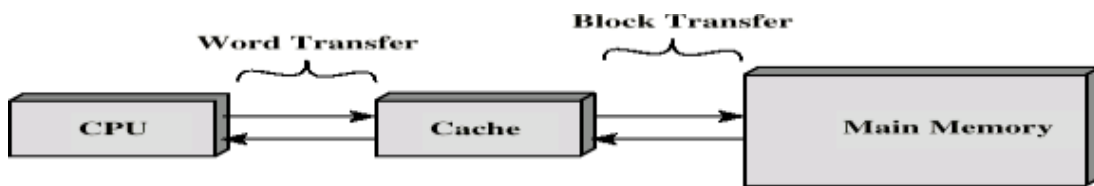
- ☐ Permanen storage
- ☐ Aplikasi : Mikroprogramming, Library subroutines, System Programs (BIOS), table-tabel fungsi
- ☐ Dibuat seperti halnya IC
- ☐ Bersifat Non volatile
- ☐ Tipe-tipe ROM
 - ☐ *ProgrammableROM (PROM)* : hanya bisa ditulisi sekali saja dan proses penulisannya dibentuk secara elektrik, memerlukan peralatan khusus untuk proses penulisan, fleksibilitas dan kemudahan, sangat menarik untuk diproduksi secara besar.
 - ☐ *Erasable Programmable ROM (EPROM)* : dibaca dan ditulis secara elektrik, sebelum operasi menulis seluruh sel harus dihapus untuk mendapatkan keadaan awal yang sama, penghapusan dapat dilakukan berulang-ulang oleh UV(+/- 20 menit), kemampuan menampung data *tidak* terhingga, lebih mahal dari PROM (krn memiliki kemampuan beberapa kali update)
 - ☐ *Electrically Erasable Programmable ROM (EEPROM)* : merupakan read only memori yg dapat ditulis kapan saja tanpa menghapus isi sebelumnya, hanya byte yang beralamat yang dapat diupdate, operasi write memerlukan waktu lebih lama dibanding operasi read (ratusan milidetik per byte), menggabungkan kelebihan non volatile dg fleksibilitas kemampuan di update dg menggunakan bus kontrol, alamat dan saluran data yang biasa, lebih mahal dibanding EPROM
 - ☐ *Flash memory (1980-an)* : kecepatannya yg dapat diprogram ulang, harga diantara EPROM dan EEPROM, menggunakan teknologi penghapusan elektrik, dapat dihapus dlm 1 atau beberapa detik saja, jauh lebih cepat, memungkinkan penghapusan blok memori tetapi tidak menyediakan penghapusan byte.

KOREKSI ERROR

- ☐ Dikategorikan menjadi : *kegagalan berat dan kegagalan ringan* Hard Failure
 - ☐ Kerusakan berat (Hard Failure)
 - ☐ Kerusakan fisik yg permanen
 - ☐ Disebabkan oleh kesalahan penggunaan atau dari pabrik
 - ☐ Kerusakan Ringan (Soft Error)
 - ☐ Kejadian yang random-non destructive ☐ tanpa merusak memori (No permanent)
 - ☐ Disebabkan oleh masalah catu daya atau partikel-partikel alpha (= hasil dari peluruhan radioaktif)
- ☐ Dideteksi dengan menggunakan *kode Hamming* (yang diciptakan oleh Richard Hamming di Lab Bell)

Memori Cache

- ☐ Lebih kecil dan lebih cepat
- ☐ Terletak diantara memori utama dan CPU



Operasi Cache

- ☐ Cache berisi salinan sebagian memori utama
- ☐ Pada saat CPU membaca sebuah word memori
- ☐ Cache mengecek data tersebut
- ☐ Jika ada maka word akan dikirim ke CPU If present, get from cache (fast)
- ☐ Jika tidak ada, membutuhkan blok memori utama yang terdapat sejumlah word dan akan dibaca ke dalam cache
- ☐ Kemudian dikirimkan dari cache ke CPU
- ☐ Cache terdiri C buah slot yg masing-masing slotnya mengandung K word, jumlah slots jauh lebih sedikit dibandingkan dg jumlah blok memori utama ($C \ll M$) dengan jumlah blok (M) = $2^n / K$ blok

Elemen Rancangan Cache (Cache Design)

1. UKURAN CACHE

- ☐ Disesuaikan kebutuhannya dalam membantu kerja memori utama
- ☐ Semakin besar ukuran cache, maka semakin besar pula jumlah gerbang yang terdapat dalam pengalamatan cache, yang mengakibatkan cache berukuran besar akan lebih lambat dari cache yang berukuran kecil
- ☐ Ukuran cache antara 1 K sampai 512 K

2. Fungsi Pemetaan (Mapping)

- ☐ Karena saluran cache lebih sedikit dibandingkan memori utama ☐ diperlukan algoritma untuk pemetaan blok-blok memori utama ke dalam saluran cache dan diperlukan juga alat untuk menentukan blok memori mana yang sedang memakai saluran cache. Untuk itu diperlukan Pemilihan fungsi Mapping.
- ☐ Elemen-elemen : Cache dapat menampung 64kByte dengan Cache block berukuran 4 byte ini berarti cache diorganisasi sebagai $16K = 2^{14}$ masing-masing besarnya 4 byte. Memori utama terdiri dari 16MByte yg masing-masing dialamati 24 alamat ($2^{24}=16M$)
- ☐ Dibagi menjadi :
 1. PEMETAAN LANGSUNG
 - Memetakan masing-masing blok memori utama hanya ke sebuah saluran cache saja.
 - Fungsi pemetaan mudah diimplementasikan dengan menggunakan alamat
 - Cache diakses dengan menggunakan alamat memori utama dianggap terdiri tiga field yaitu tag, line dan word
 - Kekurangan : terdapat lokasi cache yang tetap bagi sembarang blok-blok yang diketahui (dkl: apabila suatu program berulang-ulang melakukan word referensi dari 2 blok yg berbeda memetakan ke saluran yg sama maka blok tersebut secara terus menerus akan di-swap ke cache)

2. PEMETAAN ASOSIATIF

- ☐ Mengizinkan setiap blok memori utama untuk dimuatkan ke sembarang saluran cache.
- ☐ Cache control logic menginterpretasikan alamat memori hanya sebagai sebuah field tag dan field word.
- ☐ Terdapat fleksibilitas penggantian blok ketika sebuah blok di baca ke dalam cache
- ☐ Kekurangannya adalah kompleksitas rangkaian yang diperlukan untuk menguji tag seluruh saluran cache secara paralel

3. Pemetaan Asosiatif Set

- ☐ Memanfaatkan kelebihan-kelebihan pendekatan pemetaan langsung dan pemetaan asosiatif
- ☐ Cache dibagi menjadi v buah set yang terdiri dari k saluran
 - ☐ $m = v \times k$, $i = j \text{ modulus } v$
 - ☐ i = nomor set cache, j = nomor blok memori utama, m = jumlah saluran pada cache
 - ☐ $v = 2^d$ set, d = jumlah bit set
 - ☐ Jika $v = m$, $k = 1$ teknik asosiatif mengurangi pemetaan langsung
 - ☐ Jika $v = 1$, $k = m$ teknik asosiatif mengurangi pemetaan asosiatif
- ☐ Penggunaan 2 saluran per set ($v = m/2$, $k = 2$) merupakan organisasi asosiatif set yang paling umum
- ☐ Contoh : Nomor set 13 bit mengidentifikasi set unik 2 saluran didalam cache, maka jumlah blok didalam memori utama modulus 213 sehingga blok 000000,00A000,00B000, ..., FF1000 pada memori utama dipetakan pada set 0 cache

3. ALGORITMA PENGGANTIAN

- ☐ Digunakan untuk menentukan blok mana yang harus dikeluarkan dari cache untuk menyiapkan tempat bagi blok baru
- ☐ Ketika sebuah blok baru dibawa ke cache, salah satu blok yg ada harus digantikan.
- ☐ Untuk Direct Map : hanya hanya 1 kemungkinan slot untuk sembarang blok (No choice), masing-masing blok hanya memetakan pada satu saluran
- ☐ Untuk Asosiatif Map dan asosiatif set Map : diperlukan algoritma penggantian yang diimplementasikan dalam hardware.
 - ☐ Algoritma LRU (Least Recently Used) : pengganti blok yang berada dalam set yg telah berada paling lama di dalam cache dgn tidak memiliki referensi
 - ☐ First in first out (FIFO) : menggantikan blok didalam set yg telah berada pada cache dalam waktu yg terlama.
 - ☐ Least frequently used (LFU) : menggantikan blok didalam set yg mengalami referensi paling sedikit
 - ☐ Random
- ☐ Ada 2 metode
 1. Write- through
 2. Write-Back

1. Write-Through

- ☐ Seluruh operasi write dibuat untuk memori utama dan cache
- ☐ Modul cache CPU dapat memonitor lalu lintas ke memori utama untuk menjaga konsistensi didalam cachanya sendiri
- ☐ Keunggulannya salinan data di memori utama dan cache tetap
- ☐ Kelemahannya : menghasilkan lalu lintas memori yg sangat besar

2. Write-Back

- ☐ Update hanya dilakukan didalam cache saja
- ☐ Bila update terjadi, maka bit UPDATE yg berkaitan dgn slot disetel
- ☐ Kelemahannya bagian-bagian memori utama tidak valid maka akses oleh modul I/O hanya diijinkan bila melalui cache
- ☐ Jika blok diganti maka blok tersebut dituliskan ke memori utama bila update bit sudah di setel

Teknologi RAM

- ☐ Dasar DRAM sama dengan sejak awal chip RAM
- ☐ Enhanced DRAM (EDRAM)
 - ☐ Dibuat oleh Ramtron
 - ☐ Mencakup SRAM yang kecil (Cache SRAM menyimpan seluruh isi pembacaan baris terakhir yg berisi 2048 bit)
- ☐ Cache DRAM (CDRAM)
 - ☐ Dibuat oleh Mitsubishi
 - ☐ Mencakup cache SRAM yang lebih besar
 - ☐ Mode cache CDRAM untuk access random ke memori
- ☐ Synchronous DRAM (SDRAM)
 - ☐ Akses synchronous dengan signal clock external
 - ☐ Address is presented to RAM
 - ☐ RAM finds data (CPU waits in conventional DRAM)
 - ☐ Selama SDRAM memindahkan data dibawah kontrol waktu sistem, maka CPU akan tahu ketika data telah siap
 - ☐ CPU tidak mengenal keadaan wait
 - ☐ Mode Burst sejumlah bit data dapat diwaktu dengan cepat setelah bit pertama diakses