

BRIDGING CACHE MEMORY, KONSEP KINERJA PENINGKATAN MULTIPROSESOR

Fredy Susanto¹⁾, Teguh Avianto Nugroho²⁾

¹⁾Sistem Komputer, STMIK Raharja
Jl. Jend sudirman No 40 Cikokol Tangerang
fredy@raharja.info

²⁾Sistem Komputer, STMIK Raharja
Jl. Jend sudirman No 40 Cikokol Tangerang
teguhavianto@raharja.info

Abstrak – Pada bagian dari sistem arsitektur komputer, memori adalah memegang peranan yang sangat penting dan tidak bisa dipisahkan. Tujuan adanya memori secara garis besar adalah tempat penyimpanan data. Sehingga data bisa diolah serta diproses menjadi data keluaran atau data output. Tulisan ilmiah ini membahas mengenai memori terutama cache memory kegunaannya, macam-macamnya, dan fungsi kerjanya pada sistem arsitektur komputer. Cache memory adalah anggota hirarki memori yang mampu menyeimbangkan kecepatan pengolahan data pada prosesor dengan memori utama yaitu RAM. Beberapa bagian dan macam metode dari cache memory yaitu direct mapping, associate mapping dan set asosiate mapping yang akan dibahas ditulisan ini.

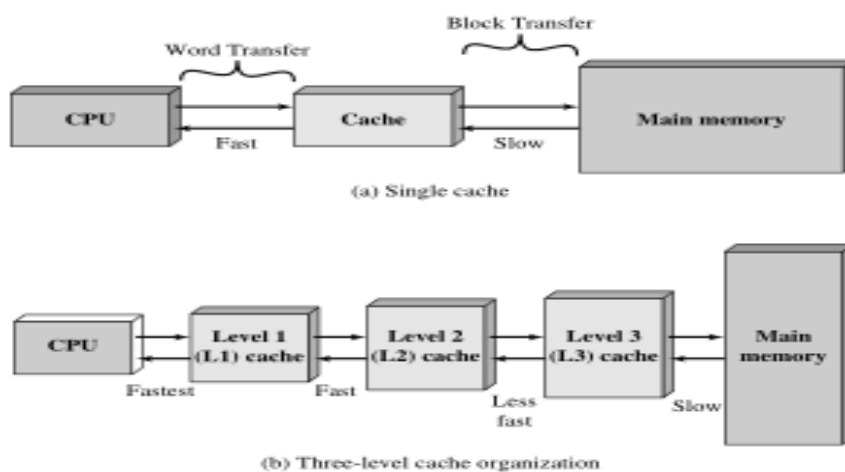
Kata kunci : cache memory, memori utama, prosesor

I. PENDAHULUAN

1.1 Definisi Cache Memory

Memori utama yang digunakan sistem komputer pada awalnya dirasakan masih lambat kerjanya dibandingkan dengan kinerja CPU, sehingga perlu dibuat sebuah memori yang dapat membantu kerja memori utama tersebut, sebagai perbandingan waktu akses memori cache lebih cepat 5-10 kali dibandingkan memori utama. Cache memory adalah memori yang memiliki kecepatan sangat tinggi yang

digunakan sebagai perantara antara RAM dan CPU. Memori ini mempunyai kecepatan yang lebih tinggi daripada RAM. Memori ini digunakan untuk menjembatani perbedaan kecepatan CPU yang sangat tinggi dengan kecepatan RAM yang jauh lebih rendah. Jika processor membutuhkan suatu data, pertama-tama ia akan mencarinya pada cache. Jika data ditemukan, processor akan langsung membacanya dengan delay yang sangat kecil. Tetapi jika data tidak ditemukan, processor akan mencarinya pada RAM.^[4]



Gambar 1. Diagram Cache Memory dan Level^[4]

Pada gambar diatas dijelaskan blok diagram dari kedudukan *cache memory*, *CPU* dan *memory utama (RAM)*.

Cache memory ada dua macam, yaitu:^[4]

1. *Cache memory* yang terdapat pada *internal processor*, *cache memory* jenis ini kecepatan aksesnya sangat tinggi dan harganya sangat mahal. Hal ini bisa terlihat pada *processor* yangberharga mahal seperti P4, P3, *AMD-Athlon*, dll. Semakin tinggi kapasitas L1,L2 *cache memory* maka semakin mahal dan semakin cepat *processor*.
2. *Cache memory* yang terdapatdi luar *processor*, yaitu berada pada *motherboard*, *memori* jenis ini kecepatan aksesnya sangat tinggi, meskipun tidak secepat *cache memory* jenis pertama (yang ada pada *internal processor*). Semakin besar kapasitasnya maka semakin mahal dan cepat. Hal ini bisa kita lihat pada *motherboard* dengan beraneka ragam kapasitas *cache memory*, yaitu 256 kb, 512 kb, 1 Mb, 2 Mb, dll.

Cache memory biasanya^[4] mempunyai beberapa *level* yang menunjukkan tingkat kedekatannya dengan *microprocessor*. Contoh, L1 *cache* ada pada *chip* yang sama dengan *microprocessor (built-in)*, sedangkan L2 *cache* adalah *cache memory* yang merupakan *chip* tersendiri yang terpisah dari *microprocessor*.

II. LANDASAN TEORI

Sebagai penunjang penelitian dan makalah ini maka perlu adanya perbandingan dengan *paper-paper* dan makalah-makalah lainnya yang gunanya apakah *paper* ini sudah ada yang membuat atau meneruskan hasil *paper* yang lain sebagai perbaikan. Maka dibandingkan dengan *paper* dalam maupun luar negeri sebagai acuan. Diantaranya adalah

1. “*Virtual memory simulator equipped by cache using PIN*”jurnal yang dibuat oleh Stella Maris, Yudi Satria Gondokaryono, Jurnal Sarjana ITB bidang Teknik Elektro dan Informatika, volume 1 No 1 (2012), mengenai perkembangan yang cepat dan besar dalam teknologi memori, baik dalam hardware atau teknik pengembangan tersebut harus diimbangi dengan kinerja yang baik dalam perangkat lunak sebagai urutan instruksi dilaksanakan oleh hardware. Pin adalah alat instrumentasi, diperkenalkan oleh Intel untuk menganalisis program. Dengan demikian, simulator simulasi sistem memori virtual dengan penambahan *cache* menggunakan Pin dibuat, disebut alat memori pin virtual. Alat ini akan digunakan untuk mendapatkan kinerja mengakses memori virtual dan *cache* oleh program dieksekusi. Alat ini dibuat dengan menentukan informasi yang ingin digali dari program dan jenis instrumentasi. Bagian dari simulator dibuat satu per satu dan terakhir digabungkan membuat satu imulator dan ditambah dengan Pin API, membuat alat pin. Alat

ini kemudian diuji dengan program benchmarking beberapa. Dari hasil, kinerja memori virtual dan *cache*, dan kesalahan diperoleh. Selain itu, efek dari unsur membangun organisasi dari cahe untuk kinerja dapat dianalisis. Walaupunada kesalahan yang dihadapi saat alat ini diuji. Ketika kapasitas *cache* meningkat, memori Pin untuk melaksanakan program ini adalah keluar dari batas. Dalam kesimpulan, alat Pin dibuat masih memiliki beberapa keterbatasan.

2. “*Memory Map: A Multiprocessor Cache Simulator Nowaday*” jurnal yang dibuat oleh Shaily dan Nitin, Journal of Electrical and Computer Engineering Volume 2012 (2012), *Multiprocessor System-on-Chip (MPSoC) architectures are mainly focused on by manufacturers to provide increased concurrency, instead of increased clock speed, for embedded systems. However, managing concurrency is a tough task. Hence, one major issue is to synchronize concurrent accesses to shared memory. An important characteristic of any system design process is memory configuration and data flow management. Although, it is very important to select a correct memory configuration, it might be equally imperative to choreograph the data flow between various levels of memory in an optimal manner. Memory map is a multiprocessor simulator to choreograph data flow in individual caches of multiple processors and shared memory systems. This simulator allows user to specify cache reconfigurations and number of processors within the application program and evaluates cache miss and hit rate for each configuration phase taking into account reconfiguration costs. The code is open source and in java.*
3. “*C-Pack: Cache Compression for Microprocessor Performance*”, jurnal yang ditulis oleh T. Tarasimhulu, pada jurnal International Journal Of Power System Operation and Energy Management (IJPSOEM) Volume-1, Issue-1, 2011. *Computer systems and micro architecture researchers have proposed using hardware data compression units within the memory hierarchies of microprocessors in order to improve performance, energy efficiency, and functionality. However, most past work, and all work on cache compression, has made unsubstantiated assumptions about the performance, power consumption, and area overheads of the proposed compression algorithms and hardware. In this work, I present a lossless compression algorithm that has been designed for fast on-line data compression, and cache compression in particular. The algorithm has a number of novel features tailored for this application, including combining pairs of compressed lines into one cache line and allowing parallel compression of multiple words while using a single dictionary and without degradation*

in compression ratio. We reduced the proposed algorithm to a register transfer level hardware design, permitting performance, power consumption, and area estimation.

III. PEMBAHASAN

Penyimpanan dan pemetaan atau mapping pada sistem memori utama dan sistem cache memory berbeda, pada memory utama berbentuk word block dan cache memory berbentuk block –block atau lines.

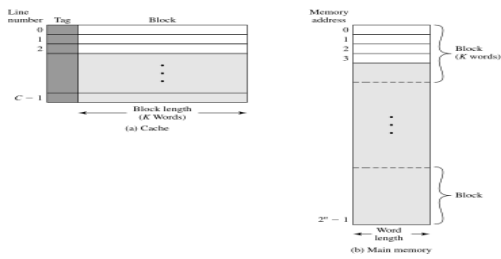


Figure 4.4 Cache/Main Memory Structure

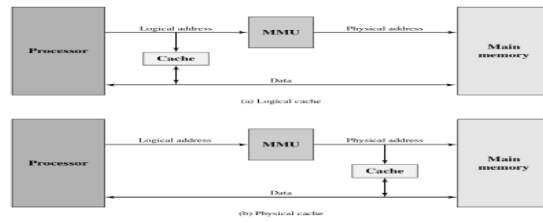
Gambar 2. Mapping Cache dan Main Memory^[2]

Adanya sifat lokalitas referensi menyebabkan nilai ukuran blok sangatlah penting. Apabila blok berukuran besar ditransfer ke cache akan menyebabkan hit ratio mengalami penurunan karena banyaknya data yang dikirim di sekitar referensi. Tetapi bila terlalu kecil, dimungkinkan memori yang akan dibutuhkan CPU tidak tercakup. Apabila blok berukuran besar ditransfer ke cache, maka akan terjadi:

1. Blok-blok yang berukuran lebih besar mengurangi jumlah blok yang menempati cache. Karena setiap pengambilan blok menumpangkan isi cache yang lama, maka sejumlah kecil blok akan menyebabkan data menjadi tertumpangkan setelah blok itu diambil.
2. Dengan meningkatnya ukuran blok maka jarak setiap word tambahan menjadi lebih jauh dari word yang diminta, sehingga menjadi lebih kecil kemungkinannya digunakan dengan cepat.

Dalam tahapan selanjutnya terdapat virtual memory yang didalamnya terdapat juga virtual address yang berfungsi mengatur alokasi dari cache antara MMU dan prosesor. Untuk tujuan membaca dan menulis dari memory utama MMU menerjemahkan dari virtual address ke physical address didalam memory utama.

Logical cache yang disebut juga virtual cache menyimpan data menggunakan virtual address. Prosesor mengakses cache memory langsung tanpa melewati MMU. Keuntungan logical cache adalah lebih cepat daripada physical address, cache logical dapat merespon tanpa melewati MMU.



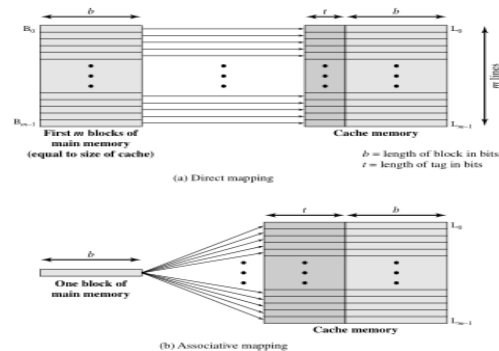
Gambar 3. Blok Diagram Akses Prosesor Dengan Main Memori^[2]

MAPPING (PEMETAAN)

Terdapat tiga metode yang digunakan yaitu:^[2]

1. Pemetaan Langsung (Direct Mapping)

Pemetaan langsung adalah teknik yang paling sederhana, yaitu teknik ini memetakan blok memori utama hanya ke sebuah saluran cache saja. Jika suatu blok ada di cache, maka tempatnya sudah tertentu. Keuntungan dari direct mapping adalah sederhana dan murah. Sedangkan kerugiannya adalah suatu blok memiliki lokasi yang tetap (jika program mengakses 2 blok yang di map ke line yang sama secara berulang-ulang, maka cache-tertinggal sangat tinggi).



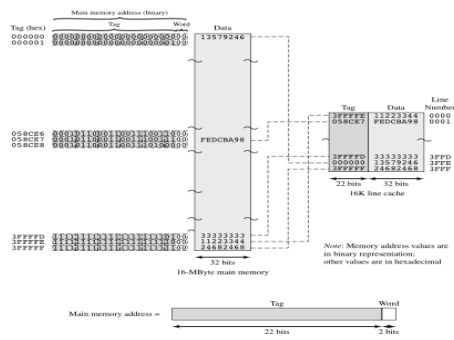
Gambar 4. Direct Mapping dan Associate Mapping^[2]

Contoh pemetaan direct mapping:

Persamaan blok nomor pengaksesan alamat adalah 000000, 010000, FF0000.... Mempunyai tag number 00,01,FF..... yang mempunyai nomor secara langsung masing-masing.

2. Pemetaan Asosiatif (Associative Mapping)

Pemetaan asosiatif mengatasi kekurangan pemetaan langsung dengan cara mengizinkan setiap blok memori utama untuk dimuatkan ke sembarang saluran cache. Dengan pemetaan asosiatif, terdapat fleksibilitas penggantian blok ketika blok baru dibaca ke dalam cache. Kekurangan pemetaan asosiatif yang utama adalah kompleksitas rangkaian yang diperlukan untuk menguji tag seluruh saluran cache secara parallel, sehingga pencarian data di cache menjadi lama.



Gambar 5, Asosiatif Mapping^[2]

Contoh pemetaan asosiatif mapping adalah: Alamat tag memory 22 bit menentukan 24 bit alamat memory utama.

3. Pemetaan Asosiatif Set (Set Associative Mapping)

Pada pemetaan ini, cache dibagi dalam sejumlah sets. Setiap set berisi sejumlah line. Pemetaan asosiatif set memanfaatkan kelebihan-pendekatan pemetaan langsung dan pemetaan asosiatif.

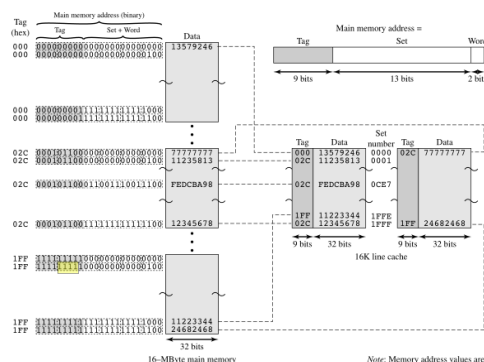


Figure 4.15 Two-Way Set Associative Mapping Example

Gambar 6, set asosiatif mapping. ^[2]

Contoh set asosiatif mapping adalah: 9 bit untuk alamat tag 13 bit set dan 2 bit word menganalogikan alamat memory utama.^[2]

IV. KESIMPULAN

Dari pembahasan diatas dapat disimpulkan bahwa *cache memory* adalah *system memory* untuk membantu dalam menyeimbangkan kecepatan prosesor yang mempunyai kecepatan tinggi dalam mengolah data terhadap memori utama yang kecepatan menyimpan (*store*) data dan alamat lebih rendah. Didalam blok diagram pengaksesan dari *prosesor cache memory* dan *memory utama* terdapat *logical access* serta *psikal access* yang semua nya tercakup dalam *virtual memory* yang akan diterjemahkan oleh *MMU (Memory Management*

Unit) ke dalam *main memory*. Dalam metode pengaksesan ke *memory* utama terdapat 3, yaitu *direct mapping*, *asosiatif mapping* serta *set asosiatif maaping* yang masing-masing mempunyai contoh masing-masing.

DAFTAR REFERENSI

- [1] Maris Stella, Satria Gondokarya Yudi (2012) "Virtual memory simulator equipped by cache using PIN", Jurnal Sarjana ITB bidang Teknik Elektro dan Informatika, volume 1 No 1
- [2] Shaily dan Nitin (2012) "Memory Map: A Multiprocessor Cache Simulator Nowaday" Journal of Electrical and Computer Engineering Volume 2012
- [3] Tarasimhulu T (2011) "C-Pack: Cache Compression for Microprocessor Performance", Journal Of Power System Operation and Energy Management (IJPSOEM) Volume-1, Issue-1
- [4] Stallings William, "COMPUTER ORGANIZATION AND ARCHITECTURE, DESAIN AND PERFORMANCE " EIGHTH EDITION Prentice Hall
- [5] Heri Andrianto.. *Pemrograman Mikrokontroler AVR Atmega16A + CD*. Informatika, Computer 2008
- [6] Winoto, Ardi. *Mikrokontroler AVR ATmega8/16/32/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. Informatika, Bandung. Mikrokontroler ATmega8535, Andi, Yogyakarta. 2008.

- [7] Sudjadi. *Teori dan aplikasi Mikrokontroler.*, Penerbit : Graha Ilmu, 2009

Biodata Penulis

Fredy susanto, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Komputer STMIK Jakarta STI&K, lulus tahun 2002. Sedang menempuh pendidikan Strata 2 Magister Komputer (M.Kom) Program Pasca Sarjana Magister Ilmu Komputer Universitas Budi Luhur Jakarta. Mempunyai sertifikasi pada bidang Nertwork yaitu CCNA dan MTCNA . Saat ini menjadi Dosen di Perguruan Tinggi Raharja sekaligus kepala devisi IT .

Teguh avianto nugroho, saat ini sedang menempuh pendidikan sarjana Komputer pada STMIK Raharja Tangerang

