

LAPORAN AKHIR
PENELITIAN DOSEN PEMULA



**PEMANFAATAN ELEMEN PERANCANGAN *CACHE* YANG TEPAT UNTUK
MEMINIMALISASI KONSUMSI DAYA DAN MENINGKATKAN KINERJA *CACHE*
MEMORY PADA MICROPROCESSOR**

TIM PENGUSUL

Ketua	Tati Erlina, M.I.T.	0014047812
Anggota	Rahmi Eka Putri, MT	0023078402

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2017

HALAMAN PENGESAHAN

Judul : Pemanfaatan Elemen Perancangan Cache Yang Tepat Untuk Meminimalisasi Konsumsi Daya Dan Meningkatkan Kinerja Cache Memory Pada Microprocessor

Peneliti/Pelaksana

Nama Lengkap : Tati Erlina, MIT
NIDN : 0014047812
Jabatan Fungsional : Asisten Ahli
Program Studi : Sistem Komputer
Nomor HP : 085263907163
Alamat surel (e-mail) : tatiertina@fti.unand.ac.id

Anggota (1)

Nama Lengkap : Rahmi Eka Putri, MT
NIDN : 0023078402
Perguruan Tinggi : Universitas Andalas

Anggota (2)

Nama Lengkap : -
NIDN : -
Perguruan Tinggi : -

Institusi Mitra

Nama Institusi Mitra : -
Alamat : -
Penanggungjawab : -
Tahun Pelaksanaan : Tahun ke- 1 dari rencana 1 tahun
Biaya Tahun Berjalan : Rp. 12.500.000
Biaya Keseluruhan : Rp. 12.500.000

Padang, 13 November 2017

Mengetahui,
Ketua Jurusan



Ratna Aisuwarya, M.Eng
NIP. 198410302008122002

Ketua Peneliti,



Tati Erlina, MIT
NIP. 197804142002122003



Menyetujui,
Dekan Fakultas Teknologi Informasi



Dr. Ahmad Syafuruddin Indrapriyatna, MT
NIP. 196307071991031003

RINGKASAN

Menciptakan komputer yang berkinerja tinggi dan mengkonsumsi daya minimal merupakan salah satu fokus yang menjadi tren penelitian dalam bidang arsitektur computer belakangan ini. Diketahui bahwa sejumlah faktor berpengaruh dalam menentukan kinerja dan konsumsi daya dari setiap elemen pembangun komputer dan pada akhirnya mempengaruhi kinerja dan konsumsi daya sebuah komputer secara keseluruhan. Fokus penelitian ini adalah untuk mengetahui seberapa besar pengaruh dari jenis elemen perancangan yang diterapkan pada sebuah cache memory terhadap kinerja dan konsumsi daya dari cache memory tersebut dengan mensimulasikan berbagai macam konfigurasi elemen perancangan tersebut pada SMPCache dengan berbagai macam benchmark dan CACTI. Dari pengujian diketahui bahwa elemen-elemen perancangan cache memory seperti cache size, line size, dan mapping function yang digunakan berpengaruh besar terhadap kinerja cache memory seperti cache hit. Selain itu juga diketahui bahwa jenis pemetaan dan aturan penempatan serta ukuran cache dan yang digunakan dalam sebuah cache memiliki pengaruh terhadap konsumsi daya. Dimana nilainya cukup bervariasi seiring dengan meningkatnya asosiasi-fitas dari fungsi pemetaan dan ukuran technology cache yang digunakan.

Kata kunci:

Cache Memory, Konsumsi Daya, Kinerja, SMPCache, CACTI

DAFTAR ISI

DAFTAR ISI	i
DAFTAR TABEL	iii
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Asumsi	3
BAB 2	4
TINJAUAN PUSTAKA	4
2.1 Prinsip-prinsip Cache Memory	4
2.2 Elemen Perancangan Cache Memory.....	5
2.3 Software Simulator Cache Memory	8
BAB 3	9
TUJUAN DAN MANFAAT PENELITIAN.....	9
3.1 Tujuan Penelitian.....	9
3.2 Manfaat Penelitian.....	9
BAB 4	10
METODE PENELITIAN.....	10
3.1 Tahapan Penelitian	10
3.2 Lokasi Penelitian	10
3.3 Peubah Yang Diamati	10
3.4 Model yang Digunakan.....	11

3.5 Teknik Pengumpulan dan Analisis Data.....	12
3.6 Teknik Pengumpulan dan Analisis Data.....	12
BAB 5	14
HASIL DAN LUARAN YANG DICAPAI.....	14
BAB 6	25
RENCANA TAHAPAN BERIKUTNYA	25
BAB 7	26
KESIMPULAN DAN SARAN	26
DAFTAR PUSTAKA.....	28
LAMPIRAN.....	Error! Bookmark not defined.

DAFTAR TABEL

Tabel 1. Elemen-elemen Perancangan Cache Memory	5
---	---

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sejak beberapa puluh tahun yang lalu, kinerja microprocessor berkembang dengan sangat pesat. Hal ini dicapai diantaranya dengan tingginya *clock frequency* operasi microprocessor dan adanya pengintegrasian beberapa inti processor ke dalam satu chip. Akan tetapi, komponen utama pembangun komputer lainnya seperti main memory tidak mampu mengimbangi perkembangan kecepatan microprocessor tersebut, sehingga terdapat perbedaan yang sangat signifikan antara kecepatan *main memory* dan *microprocessor*. Hal ini menyebabkan adanya “*bottleneck*” pada kinerja processor. Oleh karena itu, untuk menyelesaikan masalah tersebut, maka digunakanlah *cache memory*.

Cache memory, disebut juga sebagai *Central Processing Unit (CPU) memory* adalah sebuah memory yang dapat diakses oleh microprocessor dengan lebih cepat dibanding main memory. Salah satu penyebabnya adalah cache memory diposisikan sedemikian rupa pada chip yang sama dengan microprocessor, sehingga waktu yang dibutuhkan microprocessor untuk mengakses cache memory menurun secara signifikan. Setiap saat, cache memory menyimpan salinan sebagian data yang tersimpan di main memory sehingga ketika processor perlu mengakses data dari main memory, maka processor dapat mengakses salinan data yang dibutuhkannya tersebut dari cache memory. Dengan demikian, keberadaan cache memory dapat mengurangi ketimpangan kecepatan antara main memory dan microprocessor dan secara langsung berkontribusi penting dalam meningkatkan kecepatan computer secara keseluruhan.

Bersamaan dengan keuntungan penggunaan cache memory dalam perancangan microprocessor modern, cache memory juga memiliki kekurangan yang perlu ditangani, dimana cache memory merupakan salah satu komponen yang paling banyak mengkonsumsi daya dalam arsitektur Komputer[1]. Salah satu penyebabnya adalah karena cache biasanya merupakan struktur terbesar pembangun sebuah microprocessor, sehingga cache memory menjadi sumber

utama disipasi daya statis. Hal ini tentu saja tidak sejalan dengan fokus kalangan industri untuk menciptakan microprocessor yang memiliki kinerja tinggi dan mengkonsumsi daya yang lebih sedikit[2]. Menekan konsumsi daya ini menjadi sangat penting karena beberapa alasan[3], diantaranya yaitu konsumsi daya sangat berpengaruh pada pembatasan kerapatan elemen pembangun processor, kebutuhan pendinginan sistem, keandalan sistem serta masa hidup baterai terutama pada portable system.

Diketahui bahwa, terdapat banyak faktor yang mempengaruhi besarnya konsumsi daya dari sebuah cache memory. Salah satu hal yang berpengaruh besar yaitu pemilihan strategi yang diterapkan pada saat perancangan cache memory[4]. Elemen-elemen strategi perancangan yang dimaksud adalah seperti jenis mapping techniques yang digunakan serta ukuran memory cache. Dimana faktor-faktor tersebut tidak hanya berpengaruh terhadap kinerja sebuah cache memory, akan tetapi juga berpengaruh terhadap konsumsi daya sebuah microprocessor secara keseluruhan. Oleh karena itu, pada penelitian ini, penulis berencana untuk mengeksplorasi beberapa aspek perancangan cache memory dan bagaimana pengaruhnya terhadap kinerja serta konsumsi daya dari sebuah cache memory.

1.2 Perumusan Masalah

Berdasarkan pemaparan diatas, dapat dirumuskan permasalahan sebagai berikut: Bagaimana pengaruh dan kombinasi optimal aspek perancangan yang diterapkan terhadap kinerja dan konsumsi daya sebuah cache memory.

1.3 Batasan Masalah

Mengingat bahwa terdapat sejumlah aspek perancangan cache memory dan setiap aspek tersebut memiliki beberapa alternatif penerapan, maka dalam melaksanakan penelitian ini akan diseleksi setidaknya 3 (tiga) aspek perancangan cache memory yang diidentifikasi memiliki pengaruh lebih besar terhadap kinerja dan konsumsi daya sebuah cache memory untuk dieksplorasi.

1.4 Asumsi

Mengingat banyaknya waktu yang dibutuhkan untuk menjalankan software simulasi yang digunakan, maka diasumsikan bahwa penelitian yang akan dilakukan berada dalam konteks uniprocessor dan multiprocessor dengan jumlah processor yang sangat terbatas yaitu 2 dan 4.

BAB 2

TINJAUAN PUSTAKA

2.1 Prinsip-prinsip Cache Memory

Perangkat keras pembangun sistem komputer modern terdiri atas sejumlah komponen pembangun yang harus bekerja sama lain. Setiap komponen yang terlibat berkontribusi dalam menentukan kinerja komputer secara keseluruhan. Agar kinerja yang diharapkan dapat tercapai, maka setiap komponen harus dapat mengimbangi kinerja komponen lainnya. Akan tetapi, pada prakteknya, terjadi ketimpangan kecepatan yang cukup signifikan antara processor dengan main memory. Oleh karena itu, sejumlah strategi sudah dilakukan, salah satunya yaitu dengan memanfaatkan cache memory yang berfungsi sebagai perantara antar processor dan main memory.

Cache memory merupakan sebuah memory semiconductor yang merupakan tempat untuk menyimpan serangkaian data atau instruksi yang berasal dari memory lain yang relative lebih lambat[5]. Cache memory didesain untuk mengkombinasikan memory access time dari memory mahal berkecepatan tinggi dengan memory berukuran besar, berkecepatan lebih rendah dan berharga lebih murah. Dimana terdapat memory yang relative lambat dan berukuran besar bersamaan dengan cache memory yang lebih cepat dan lebih kecil. Cache berisi salinan bagian dari main memory. Ketika processor berusaha untuk membaca sebuah word dari memory, pengecekan dilakukan untuk memeriksa apakah word yang dibutuhkan berada dalam cache. Jika ya, maka word tersebut dikirim ke processor. Jika tidak, sebuah block main memory yang terdiri atas beberapa word yang berukuran tetap, dikirim ke cache dan kemudian word tersebut dikirim ke processor. Karena fenomena locality of reference, maka ada kemungkinan bahwa referensi selanjutnya ke lokasi memory yang sama atau word lain pada block tersebut.

2.2 Elemen Perancangan Cache Memory

Terdapat beberapa penelitian terdahulu yang meneliti tentang pengaruh penerapan berbagai elemen perancangan terhadap kinerja dari sebuah cache memory. Dimana yang dimaksud dengan kinerja cache memory dalam hal ini memiliki beberapa parameter, diantaranya seperti yang disebutkan pada[5] bahwa basis utama perhitungan kinerja cache memory adalah miss rate, miss penalty dan average access time. Miss rate adalah akses terhadap bagian main memory yang salinannya tidak dapat ditemukan pada cache memory, sebaliknya hit rate adalah akses terhadap bagian dari main memory yang salinannya dapat ditemukan di cache memory. Miss penalty adalah jumlah total CPU cycles yang terhenti akibat usaha untuk mengakses main memory.

Walaupun terdapat sejumlah implementasi cache, terdapat beberapa elemen perancangan dasar yang berfungsi untuk mengklasifikasikan dan membedakan arsitektur cache[6]. Elemen-elemen pentingnya dapat dilihat pada tabel 1.

Tabel 1. Elemen-elemen Perancangan Cache Memory

Cache Address <ul style="list-style-type: none">- Logical- Physical	Write Policy <ul style="list-style-type: none">- Write through- Write back
Cache Size	Line Size
Mapping Function <ul style="list-style-type: none">- Direct- Associative- Set Associative	Number of Caches <ul style="list-style-type: none">- Single or two level- Unified or split
Replacement Algorithm <ul style="list-style-type: none">- Least recently used (LRU)- First In First Out (FIFO)- Least Frequently Used (LFU)- Random	

Cache Address

Hampir semua nonembedded processor, and many embedded processor, support virtual memory. Pada dasarnya, virtual memory adalah sebuah fasilitas yang mengizinkan program untuk mengalami memory dari sudut pandang logika, tanpa tergantung pada jumlah main memory yang tersedia secara fisik. Ketika virtual memory digunakan, field alamat dari instruksi mesin berisi alamat virtual. Untuk membaca dan menulis dari memory, sebuah hardware memory management unit (MMU) menterjemahkan setiap alamat virtual menjadi alamat fisik dalam main memory.

Cache Size

Perancang cache menginginkan cache dengan ukuran yang cukup kecil sehingga dengan demikian harga rata-rata harga per bit mendekati harga main memory dan cukup besar sehingga waktu akses rata-rata secara keseluruhan mendekati cache itu sendiri. Terdapat beberapa motivasi untuk meminimalisasi ukuran cache. Dimana semakin besar cache, semakin besar jumlah gate yang terlibat dalam pengalamatan cache. Akibatnya, cache yang besar cenderung agak lebih lambat disbanding dengan cache yang lebih kecil –bahkan ketika dibangun dengan teknologi IC yang sama dan diletakkan pada tempat yang sama dalam chip dan papan sirkuit. Ketersediaan chip dan area papan juga membatasi ukuran cache. Karena kinerja cache sangat sensitive terhadap sifat alami dari beban kerjanya, maka tidaklah mungkin untuk mendapatkan ukuran cache yang optimum.

Mapping Function

Mapping techniques digunakan untuk memetakan main memory blocks yang jumlahnya sangat besar ke cache memory lines yang jumlahnya jauh lebih kecil serta untuk membuat tanda (tag) berupa bits pada setiap memory cache line untuk memastikan main memory blocks yang mana yang salinannya sedang tersedia pada cache memory line tertentu. Terdapat beberapa mapping techniques yang berperan menentukan organisasi dari sebuah cache, yaitu direct mapping, associative mapping dan set associative mapping.

Direct mapping merupakan teknik yang paling sederhana, memetakan setiap block main memory ke hanya satu cache line yang mungkin.

Associative mapping menyelesaikan kelemahan dari direct mapping dengan mengizinkan setiap block main memory dimuat ke setiap line yang ada pada cache. Dalam hal ini, cache control logic menginterpretasikan sebuah memory address hanya sebagai sebuah Tag and sebuah Word field. Tag field secara unik mengidentifikasi sebuah block main memory. Untuk mengetahui apakah sebuah block ada dalam cache, cache control logic harus secara simultan memeriksa setiap tag line untuk mengetahui yang sesuai.

Set associative mapping (disebut juga dengan k-way set associative) merupakan sebuah kompromi yang memperlihatkan kelebihan dari pendekatan direct dan associative dan sekaligus memperkecil kekurangannya. Dalam hal ini, cache terdiri atas serangkaian set, dimana masing-masingnya terdiri atas sejumlah lines.

Replacement Algorithms

Replacement algorithm memiliki peranan penting dalam perancangan cache memory karena replacement algorithm berperan untuk memutuskan cache memory line mana yang akan ditimpa oleh main memory block yang dibutuhkan. Terdapat 3 (tiga) alternatif replacement algorithm utama[7], yaitu First In First Out (FIFO), Least Frequently Used (LFU), Least Recently Used (LRU) dan Random.

Write Policy

Write policies berperan untuk menentukan bagaimana cara menjaga konsistensi antara isi cache memory line tertentu dengan main memory block terkait. Alternatif metode yang tersedia diantaranya adalah write back dan write through. Pada write back policy operasi penulisan hanya dilakukan pada cache memory line, sedangkan isi main memory block diperbaharui hanya ketika cache memory line yang terkait dengannya akan dihapus dari cache memory. Sedangkan pada write through policy operasi penulisan dilaksanakan pada main memory block bersamaan dengan penulisan pada cache memory line.

2.3 Software Simulator Cache Memory

Pemilihan nilai atau teknik yang berbeda pada parameter-parameter perancangan cache memory tersebut diatas, dapat mempengaruhi kinerja dan konsumsi daya pada cache memory[8]. Oleh karena itu untuk memilih konfigurasi terbaik dari sebuah cache maka diperlukan eksplorasi elemen-elemen perancangan cache. Penelitian[9] mempelajari tentang hubungan antara instruksi dan ukuran data cache memory dan kemudian menganalisa hubungannya dengan konsumsi daya. Diketahui bahwa menggunakan cache size yang optimal dapat menghasilkan konsumsi daya terendah walaupun tetap mempertimbangkan waktu eksekusi, daya dan pemanfaatan sumberdaya-sumberdaya yang terdapat dalam sebuah FPGA.

CACTI [10] adalah sebuah software simulasi yang digunakan untuk menganalisa daya, area dan timing untuk arsitektur yang berbasis memori. CACTI[11] merupakan sebuah model terintegrasi untuk mensimulasikan cache memory dan main memory access time, cycle time, area, leakage dan dynamic power. Simulator ini ditujukan untuk digunakan oleh computer architects untuk memahami kompromi kinerja dalam organisasi sistem memori.

SMPCache[12] adalah sebuah trace driven simulator yang digunakan untuk menganalisa dan mensimulasikan cache memory systems pada symmetric mutiprocessors yang menggunakan bus yang berbasis shared memory. Simulator ini beroperasi pada PC dengan sistem Windows dan menyediakan Graphic User Interface. Beberapa parameter pada cache memory dan multiprocessor yang dapat dipelajari menggunakan SMPCache adalah program locality, pengaruh jumlah processor, cache memory coherence protocol, bus arbitration scheme, mapping techniques, replacement policies, cache memory size, set associative caches dan main memory block.

BAB 3

TUJUAN DAN MANFAAT PENELITIAN

3.1 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengidentifikasi elemen perancangan yang tepat digunakan pada cache memory sehingga dapat meminimalisasi konsumsi daya dan meningkatkan kinerja cache memory pada microprocessor. Selain itu, hasil penelitian ini diharapkan dapat menjadi referensi dan bahan yang dapat memperkaya pemahaman pengampu matakuliah dan mahasiswa pada matakuliah Organisasi dan Arsitektur Komputer tentang konsep perancangan cache memory.

3.2 Manfaat Penelitian

Manfaat Penelitian ini bagi dosen pengampu matakuliah organisasi dan arsitektur komputer adalah lebih mengetahui dan memahami pengaruh elemen-elemen perancangan cache memory secara nyata dari spesifikasi komputer yang sudah ditetapkan. Selain itu dapat memperkaya bahan pengajaran dan praktikum matakuliah tersebut. Bagi mahasiswa, hasil penelitian ini akan dapat digunakan sebagai referensi untuk membantu memahami matakuliah dan praktikum matakuliah organisasi dan arsitektur komputer.

BAB 4

METODE PENELITIAN

3.1 Tahapan Penelitian

Untuk melaksanakan penelitian ini, maka akan dilaksanakan tahapan-tahapan sebagai berikut :

1. Menentukan jumlah processor yang akan digunakan sebagai basis penentuan kinerja cache.
2. Mengidentifikasi parameter-parameter output yang akan digunakan untuk menguji kinerja sebuah cache (cache hit, power consumption)
3. Mengidentifikasi elemen perancangan cache yang akan digunakan sebagai input yang dapat mempengaruhi kinerja cache (ukuran cache, jumlah level cache, ukuran line, metode pemetaan, metode penulisan, metode penimpaan, cache coherence protocol yang digunakan, konsumsi daya yang dibutuhkan).
4. Menentukan metode pengujian yang akan dilakukan.
5. Pengujian dan Pengumpulan data.
6. Analisa Hasil.
7. Penulisan Laporan

3.2 Lokasi Penelitian

Pelaksanaan penelitian ini tidak terikat pada satu lokasi tertentu, selama pada lokasi tersebut tersedia spesifikasi perangkat komputer yang dibutuhkan serta terpasangnya perangkat yang lunak yang dapat mendukung terlaksananya penelitian ini, khususnya adalah perangkat lunak SMPCache 3.0 dan CACTI.

3.3 Peubah Yang Diamati

Pertama, peubah yang akan diamati dalam penelitian ini adalah besarnya cache hit sebagai salah satu peubah yang dijadikan paramemter kinerja dari sebuah

cache memory yang dihasilkan dari berbagai macam kombinasi input yang diberikan dan berbagai macam teknik dari elemen perancangan yang diterapkan.

Kedua, peubah yang akan diamati adalah daya yang dikonsumsi sebuah cache memory. Hal ini juga dipengaruhi oleh elemen perancangan cache memory. Dimana penerapan sebuah teknik yang berbeda dapat memberikan pengaruh terhadap besarnya konsumsi daya dari suatu cache memory.

3.4 Model yang Digunakan

Terdapat setidaknya 2 (dua) model (perangkat simulator) yang digunakan dalam penelitian ini.

1. SMPCache simulator
2. CACTI Cache simulator

Kedua perangkat lunak tersebut akan dikonfigurasi dengan jangkauan sebagai berikut :

Number of Processors = [1, 2, 3, 4, 5, 6, 7 atau 8]

Cache Coherence Protocol = [MSI, MESI, DRAGON]

Bus Arbitration = [Random, LRU, LFU]

Word Wide (bits) = [8, 16, 32, 64]

Blocks in Main Memory = [1, 2, 4, 8, 16, 32, 64, 128, 256]

Line Size = [1, 2, 4, 8, 16, 32, 64, 128]

Maximum Main Memory Size = 32 GB

Blocks in Cache = [1, 2, 4, 8, 16, 32, 64, 128]

Maximum Cache Size = 16 bits

Mapping = [Direct, Set-Associative, Full-Associative]

Writing Strategy = Write-Back

Replacement Policies = [Random, FIFO, LFU, LRU]

Maximum Block Size = 8 KB

3.5 Teknik Pengumpulan dan Analisis Data

Pengumpulan data dilakukan dengan cara :

1. Mengkonfigurasi SMPCCache dengan menggunakan elemen perancangan seperti jumlah processor, cache coherence protocol, bus arbitration, word wide, blocks in main memory, line size, block in cache, mapping, writing strategy dan replacement policy tertentu, menjalankan SMPCCache dengan konfigurasi tersebut dan merekam tingkat cache hit (atau cache miss) yang dihasilkan sebagai salah satu parameter kinerja cache memory (langkah ini diulangi dengan menggunakan setiap kemungkinan kombinasi konfigurasi yang ada)
2. Mengkonfigurasi CACTI cache simulator dengan menggunakan konfigurasi yang sama dengan yang digunakan pada SMPCCache dan menjalankan setiap kombinasi aspek perancangan yang digunakan untuk mengetahui besarnya konsumsi daya dari setiap konfigurasi yang digunakan.
3. Menganalisa hubungan antara keluaran SMPCCache dan CACTI cache simulator yang menggunakan konfigurasi yang sama, kemudian membandingkan keluaran tersebut dengan keluaran kedua simulator tersebut saat menggunakan kombinasi konfigurasi lainnya, kemudian menganalisa data secara keseluruhan.

3.6 Teknik Pengumpulan dan Analisis Data

Perangkat yang digunakan dalam penelitian ini terbagi dua yaitu perangkat keras dan perangkat lunak, dengan rincian sebagai berikut :

A. Perangkat keras

Komputer dengan spesifikasi sebagai berikut :

Motherboard ASUS B250F

Processor Intel core i5-7400

Samsung DDR4 4GB
HDD Seagate 1TB
Monitor Led LG 20"
DVD Player LG
Casing Power Logic Ultima x2000

B. Perangkat Lunak

Perangkat lunak yang digunakan adalah :

1. Sistem Operasi Windows 10 64 bit
2. Windows Subsystem for Linux
3. SMPCache Simulator
4. CACTI Simulator

BAB 5

HASIL DAN LUARAN YANG DICAPAI

5.1 Hasil

Dari konfigurasi dan pengujian-pengujian yang telah dilakukan dari penelitian ini adalah terkait dengan pengaruh beberapa aspek perancangan cache memory terhadap kinerjanya yang terangkum dalam poin-poin berikut :

Uniprocessor

1. Pengaruh Block Size pada Cache Sizes yang berbeda

Processors in SMP = 1.

Cache coherence protocol = MESI.

Scheme for bus arbitration = Random.

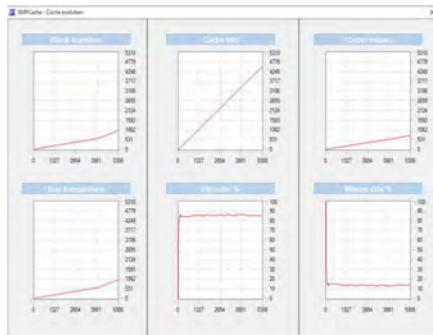
Word wide (bits) = 32.

Main memory size = 1024 KB

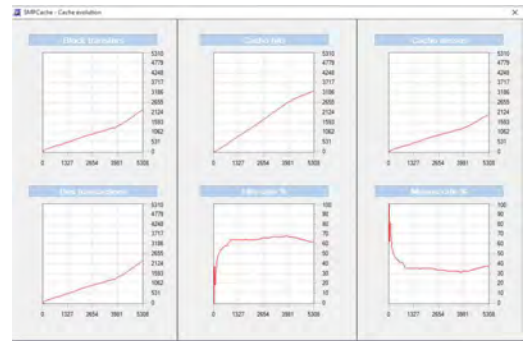
Mapping = Fully-Associative.

Replacement policy = LRU.

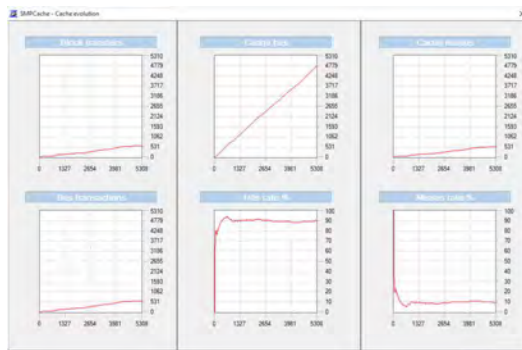
Ketika block size meningkat maka miss rate-nya akan turun, akan tetapi kita tidak bisa untuk tetap terus menambah ukuran dari block atau block size, hal ini disebabkan dengan semakin besarnya ukuran block akan memakan waktu lama untuk mengambil data dari block tersebut. Hal ini bisa terlihat pada ukuran cache yang hanya 4KB dan word by block nya 1024 :



Word by block 512 Cache size 4



Word by block 1024 cache size 4



Word by block 32 Cache size 4

2. Pengaruh Mapping pada Cache Sizes yang berbeda

Konfigurasi :

Processors in SMP = 1.

Cache coherence protocol = MESI.

Scheme for bus arbitration = Random.

Word wide (bits) = 32.

Words by block = 64 (block size = 256 bytes).

Blocks in main memory = 4096 (main memory size = 1024 KB).

Replacement policy = LRU.

1. Apakah miss rate meningkat seiring tingkat associativity yang meningkat ?

Kenapa?

Tidak (Miss Rate Menurun) Hits rate meningkat, Karena dengan tingkat associativity yang meningkat sehingga membuat jumlah block yang dicopy ke cache secara bersamaan menjadi makin besar, sehingga kemungkinan miss menjadi menurun.

2. Apa yang terjadi dengan conflict misses memperbesar grade asosiatif?

Ketika tingkat asosiatif naik atau diperbesar, conflict misses yang terjadi adalah menurun.

3. Apakah tingkat associativity memiliki pengaruh ketika ukuran cache diperbesar? Kenapa?

Iya, karena semakin besar ukuran cache maka jumlah block yang dicopy secara bersamaan akan semakin banyak sehingga pada tahap ini merupakan pengaruh dari tingkat asosiatif. Pada tingkat two-way set asosiatif ukuran cache menjadi lebih kecil dibandingkan pada four-way dan eight-way.

Multiprocessor

1. Pengaruh Cache Size pada miss-rate

Secara keseluruhan dengan meningkatnya ukuran cache maka performa akan meningkat atau dalam artian cache miss atau missrate akan menurun, dan hits rate akan meningkat meskipun tidak menunjukkan perubahan atau peningkatan yang terlalu signifikan. Akan tetapi peningkatan ukuran cache tidak menjadi mempercepat proses, ini dikarenakan karena prinsipnya cache digunakan sebagai perantara atau jembatan bagi suatu proses dengan main memory, semakin besar ukuran cache maka semakin banyak block yang dapat ditampung, data semakin banyak, sehingga waktu yang dibutuhkan untuk menemukan data yang dibutuhkan akan semakin lama. Dengan kata lain, waktu yang dibutuhkan untuk cache hits akan semakin panjang, ditambah apabila data yang dibutuhkan tidak ada hingga harus mengakses memori utama.

Jika ukuran block cache yang diperbesar maka akan mengurangi jumlah block– block yang akan menempati cache, dikarenakan ukuran per blocknya yang

semakin besar, ini mengakibatkan update atau pergantian block menjadi lebih cepat, sehingga block yang ada dalam cache sebelumnya berkemungkinan besar untuk ditimpa ketika ada permintaan baru. Semakin besar ukuran block maka semakin panjang word dan data yang ada dalam cache, sehingga sekali lagi pencarian akan membutuhkan waktu yang lebih lama, atau tidak secepat yang diharapkan.

- a. Apakah peningkatan atau penurunan ini terjadi pada semua benchmark atau apakah ia tergantung pada nilai locality yg berbeda ?

Ya, peningkatan dan penurunan ini umumnya terjadi pada semua benchmark dan berdasarkan locality yang berbeda. Ketika lokalitas diganti atau berbeda maka semua benchmarks umumnya mengalami peningkatan pada Cache hits, dan mengalami penurunan terhadap miss rate nya.

- b. Semakin besar nilai cache nya, semakin stabil nilai miss rate nya.

Hal ini dikarenakan, semakin banyak data yang disimpan pada cache sehingga data yang dibutuhkan umumnya telah berada dalam cache sehingga akses ke main memory utama juga akan semakin jarang, oleh karena itu cache miss atau miss rate seiring diperbesar cache akan stabil.

- c. Perbedaan nilai miss rate yang tinggi seiring peningkatan ukuran cache dikarenakan, semua data atau kebutuhan yang diperlukan telah berada dalam cache sehingga, pengaksesan ke memori utama berkurang. Perbedaan nilai miss rate tidak muncul pada point yang sama, ini diakibatkan proses dari tiap –tiap program berbeda, data yang dibutuhkan masing-masing program berbeda sehingga miss rate tidak muncul pada point yang sama.

- d. Apakah peningkatan ukuran cache meningkatkan performa sistem multiprocessor? Ya secara umum ini meningkatkan performa dari sistem multiprocessor, ini dikarenakan semakin besar ukuran block cache, maka semakin banyak data dalam satu block yang dapat ditampung, sehingga akses ke main memori utama berkurang karena itulah ukuran cache yang semakin diperbesar dapat meningkatkan peroforma dari sistem multiprocessor.

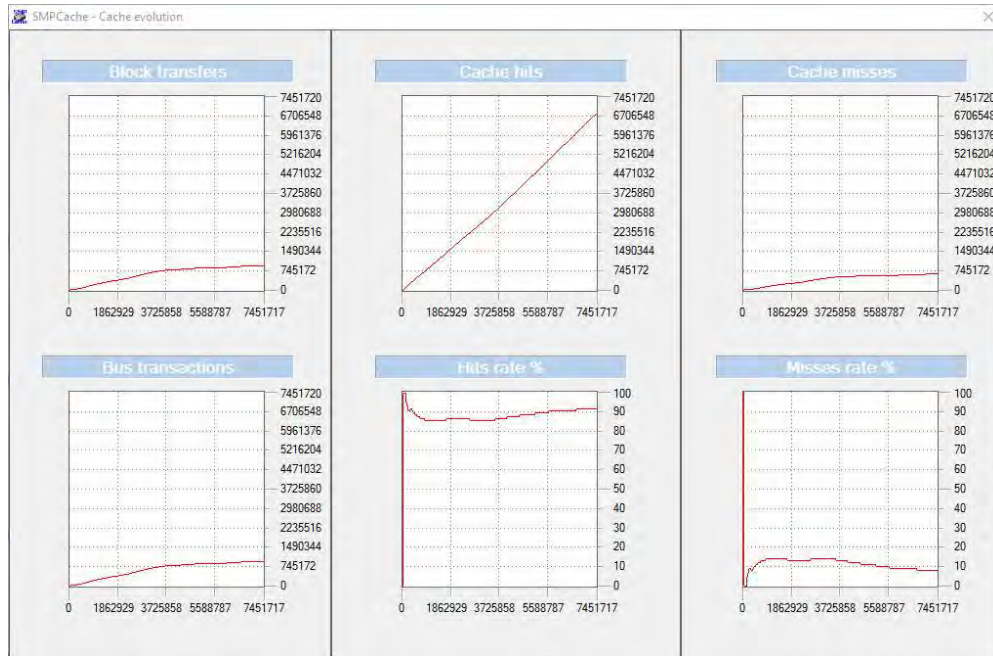
2. Pengaruh banyaknya processor terhadap Miss Rate

Konfigurasi :

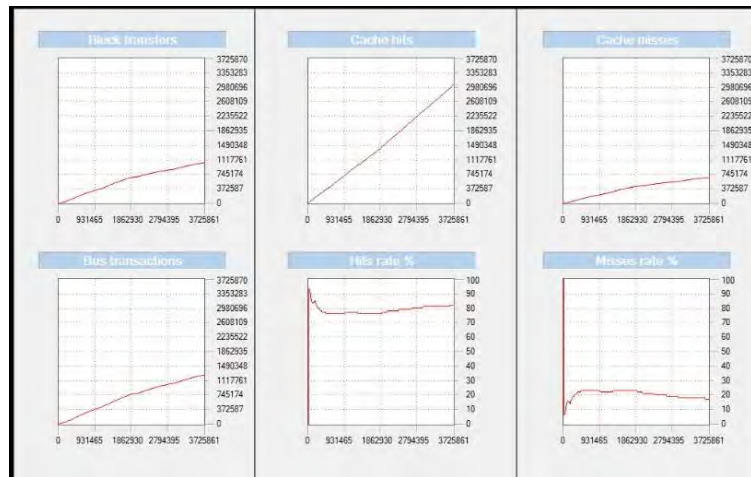
- Cache coherence protocol = MESI.
- Scheme for bus arbitration = LRU.
- Word wide (bits) = 16.
- Words by block = 32 (block size = 64 bytes).
- Blocks in main memory = 524288 (main memory size = 32 MB).
- Blocks in cache = 256 (cache size = 16 KB).
- Mapping = Set-Associative.
- Cache sets = 64 (four-way set associative caches).
- Replacement policy = LRU.

Miss Rate (persentase tidak ditemukannya data dalam cache) secara umum dari percobaan ini meningkat, seiring dengan ditambahkan jumlah processor. Rata – Rata peningkatan dari miss rate ini terjadi pada semua processor. SMP atau Symmetric Multiprocessing melibatkan arsitektur multiprocessor perangkat keras komputer dimana dua tau lebih processor identik terhubung ke satu shared memory utama dan dikendalikan oleh OS tunggal. Arsitektur SMP berlaku untuk core, diperlakukan sebagai processor yang terpisah, akan tetapi tetap saling berhubungan menggunakan bus, switch, crossbar atau on chip jaringan Mesh.

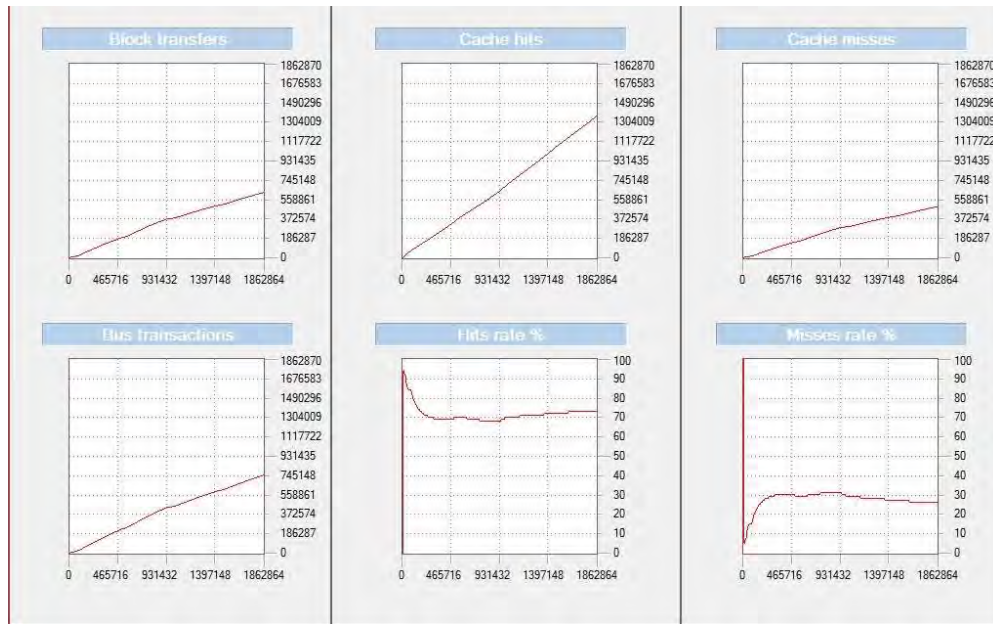
Coherence Miss mungkin akan meningkat seiring dengan ditingkatkannya jumlah processor, maksud dari koherensi disini adalah konsistennya suatu data dalam memory. Ini terjadi dikarenakan pada SMP menggunakan satu memory dan jalur secara bersama – sama, ketika kita menggunakan banyak processor otomatis hasil dari processor akan disimpan ke memori utama dan pada prinsip koherensi sendiri dimana satu tingkatan memory harus sama, dalam arti jika kita memasukkan suatu informasi makan satu tingkat memory akan diupdate, data bisa jadi tidak konsisten dikarenakan banyaknya data yang diupdate ke dalam memory dari banyak processor sedangkan jalur yang digunakan sama.



Gambar. FFT dengan 1 processor SMP



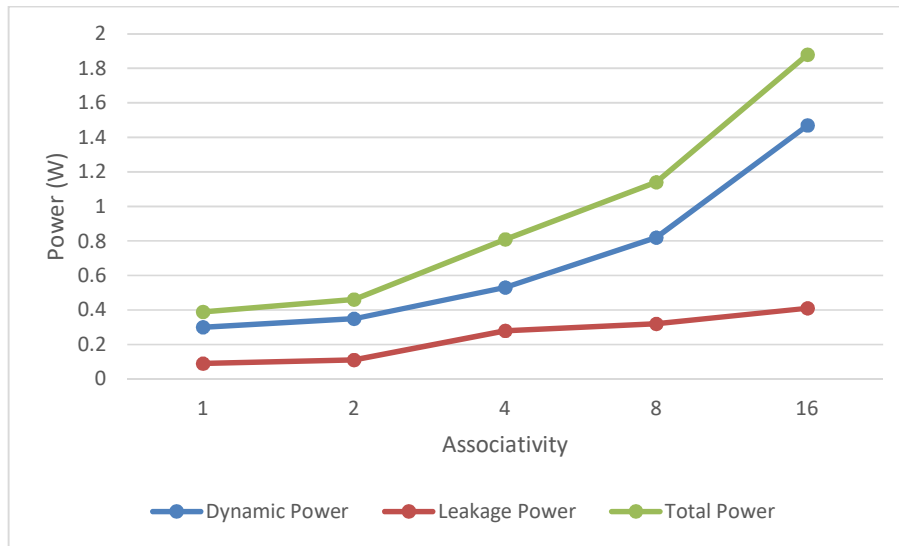
Gambar. FFT dengan 2 Processor SMP



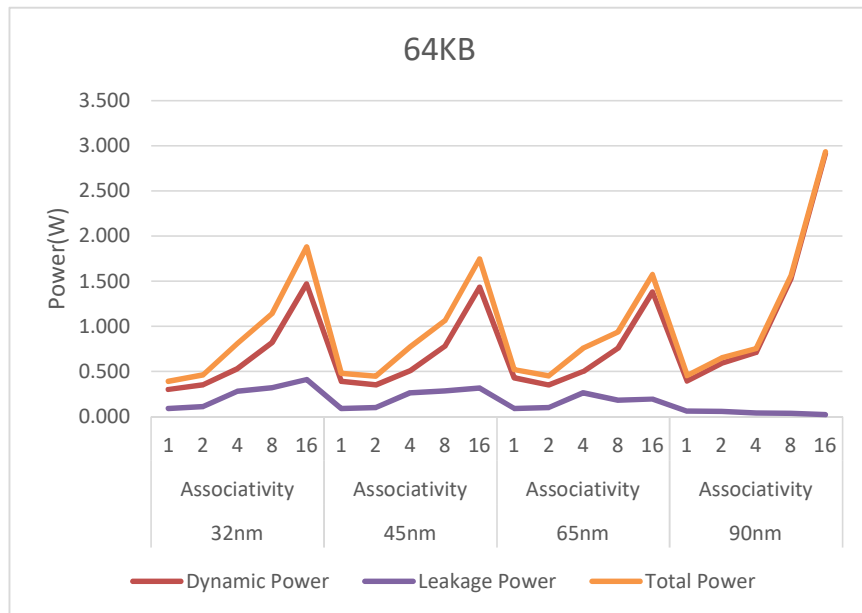
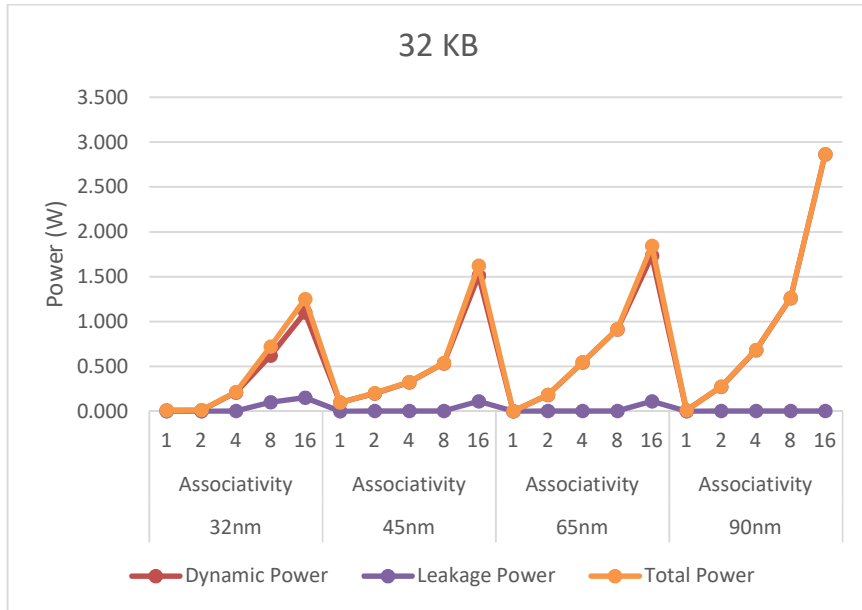
Gambar. FFT dengan 4 processor SMP

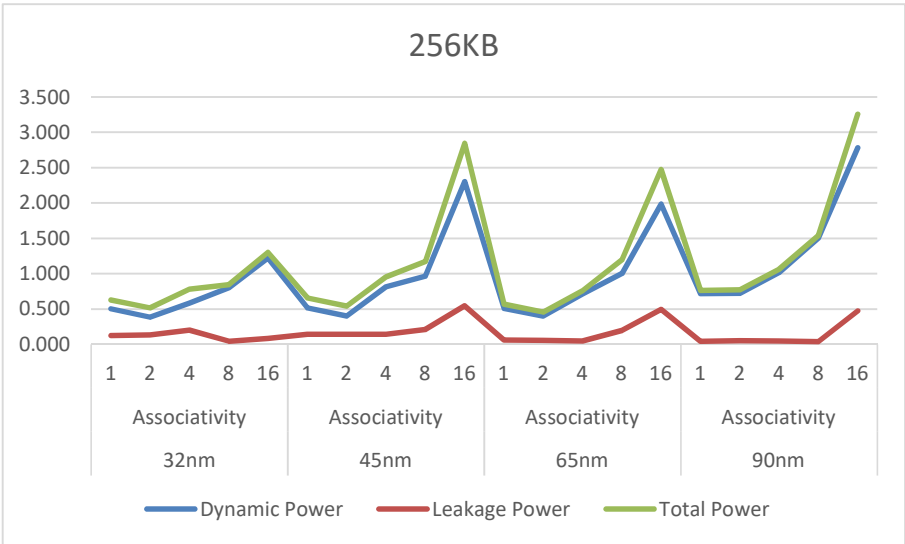
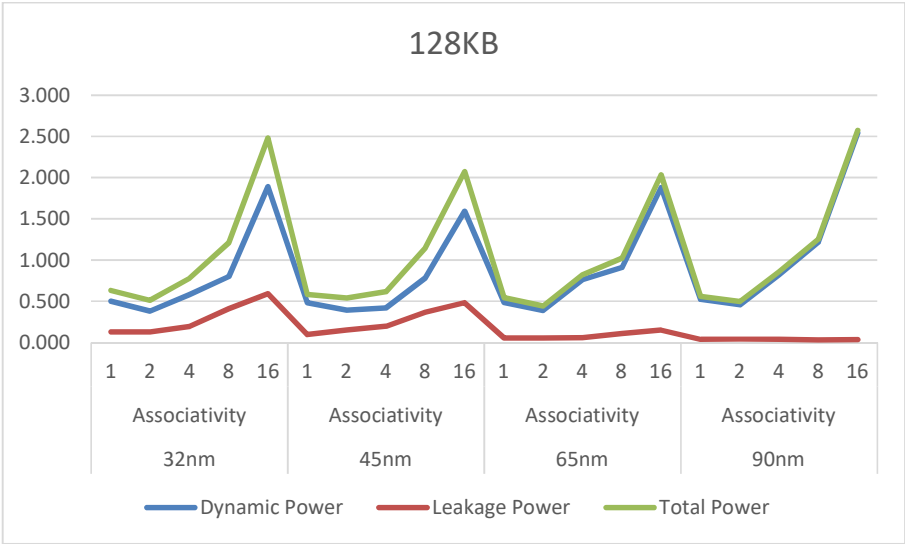
PENGUJIAN MENGGUNAKAN CACTI

64KB caches with 64-byte cache lines and 32 nm node technology			
Associativity	Dynamic Power	Leakage Power	Total Power
1	0.3	0.09	0.39
2	0.35	0.11	0.46
4	0.53	0.28	0.81
8	0.82	0.32	1.14
16	1.47	0.41	1.88



Gambar. Hasil Pengujian 64KB caches menggunakan 64-byte cache lines dan 32 nm technology.





5.2 Luaran

Luaran yang dicapai adalah berupa draf makalah yang akan disubmit pada jurnal jitter.

BAB 6

RENCANA TAHAPAN BERIKUTNYA

Terkait dengan pelaksanaan dan hibah penelitian ini, maka langkah selanjutnya yang akan diambil adalah men-submit makalah yang dihasilkan ke Jurnal JITTER seperti yang sudah dinyatakan dalam proposal. Hasil penelitian ini diharapkan dapat menjadi pijakan untuk mengeksplorasi aspek-aspek lainnya yang mempengaruhi kinerja dan konsumsi daya sebuah cache memory. Dengan demikian diharapkan, peneliti dapat melanjutkan penelitian ini melalui skema-skema penelitian lain sehingga hasil ini dapat lebih bermanfaat dan lebih berkontribusi baik dalam bidang industri perangkat keras computer secara umum maupun dapat dijadikan sebagai bahan yang dapat membantu pengayaan pemahaman mahasiswa terhadap topik-topik matakuliah terkait Organisasi dan Arsitektur Komputer.

BAB 7

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Dari pengujian-pengujian yang dilakukan dapat diambil kesimpulan bahwa:

1. Elemen-elemen perancangan cache memory seperti cache size, line size, dan mapping function yang digunakan berpengaruh besar terhadap kinerja cache memory seperti cache hit.
2. Jenis pemetaan dan aturan penimpanan serta ukuran cache dan yang digunakan dalam sebuah cache memiliki pengaruh terhadap konsumsi daya. Dimana nilainya cukup bervariasi seiring dengan meningkatnya asosiasi-fitas dari fungsi pemetaan yang digunakan dan ukuran technology cache yang digunakan.

7.2 Saran

Berdasarkan pelaksanaan penelitian ini, maka dapat disarankan beberapa hal berikut :

1. Pengujian sebaiknya dilakukan pada computer berspesifikasi tinggi sehingga eksplorasi dapat dilakukan pada aspek yang lebih beragam dan melibatkan jumlah core processor yang lebih banyak.
2. Konfigurasi cache memori yang digunakan sebaiknya berdasarkan spesifikasi nyata sebuah cache memory tertentu sehingga dapat memberikan gambaran yang lebih jelas ketika hasil pengujian didapatkan.
3. Sebaiknya menggunakan software simulasi yang sudah umum digunakan sehingga dapat meminimalisasi waktu yang digunakan untuk bertanya di forum untuk mendapatkan solusi

yang dihadapi ketika mendapat masalah dalam melaksanakan simulasi.

DAFTAR PUSTAKA

- [1] K. Saito, R. Kobayashi, and H. Shimada, "Reduction of Cache Energy by Switching between L1 High Speed and Low Speed Cache under application of DVFS," in *International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, 2016.
- [2] D. P. Kothari, "A Study on Factors Influencing Power Consumption in Multithreaded and Multicore CPUs," *WSEAS Trans. Comput.*, vol. 10, no. 3, pp. 93–103, 2011.
- [3] Z. Zhu, "Power Efficient Designs." [Online]. Available: <http://home.eng.iastate.edu/~zzhang/courses/cpre585-f04/slides/Lecture24.pdf>. [Accessed: 22-Feb-2017].
- [4] J. Abella and A. Gonzalez, "Power efficient data cache designs," in *Proceedings 21st International Conference on Computer Design*, 2003, no. 3, pp. 8–13.
- [5] S. Kumar and U. Pradesh, "An Overview of Modern Cache and Performance Analysis of Replacement Policies," in *IEEE International Conference on Engineering and TEchnology (ICETECH)*, 2016, no. March, pp. 210–214.
- [6] W. Stallings, *Computer Organization and Architecture: Designing for Performance*, Ninth Edit. Pearson, 2013.
- [7] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, vol. 4th, no. 0. 2009.
- [8] B. R. Upadhyay and S. TSB, "Design Space Exploration of Cache Memory – A Survey," in *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 2294–2297.
- [9] D. M. Cambre, E. Boemo, and E. Todorovich, "Energy Evaluation in the Nios II Processor as a Function of Cache Sizes," in *Southern Conference on Programmable Logic*, 2008, pp. 55–61.
- [10] S. Thoziyoor, J. H. Ahn, M. Monchiero, J. B. Brockman, and N. P. Jouppi, "A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies," *Proc. - Int. Symp. Comput.*

Archit., pp. 51–62, 2008.

- [11] HP Labs, “CACTI - An integrated cache and memory access time, cycle time, area, leakage, and dynamic power model,” 2008. [Online]. Available: <http://www.hpl.hp.com/research/cacti/>. [Accessed: 23-Feb-2017].
- [12] M. Á. Vega Rodríguez, J. M. Sánchez Pérez, and J. A. Gómez Pulido, “An educational tool for testing caches on symmetric multiprocessors,” *Microprocessors and Microsystems*, 2001. .