

MODUL VI

UNIPROCESSOR TRACE II

1.1 TUJUAN

-
1. Memperlihatkan pengaruh ukuran cache pada miss rate.
 2. Memperlihatkan pengaruh ukuran block pada miss rate.

1.2 PERSIAPAN

Pelajari kembali modul praktikum dan bahan kuliah yang berkaitan dengan *Uniprocessor Trace*. Kemudian kerjakan Tugas Pendahuluan dan kumpulkan sesuai ketentuan yang berlaku.

1.3 DASAR TEORI

1.3.1 SMP Cache

SMPCache adalah sebuah trace driven simulator yang digunakan untuk menganalisa dan mensimulasikan cache memory systems pada symmetric multiprocessors yang menggunakan bus yang berbasis shared memory.

1.4 PERCOBAAN

A. ALAT YANG DIGUNAKAN

1. SMP Cache

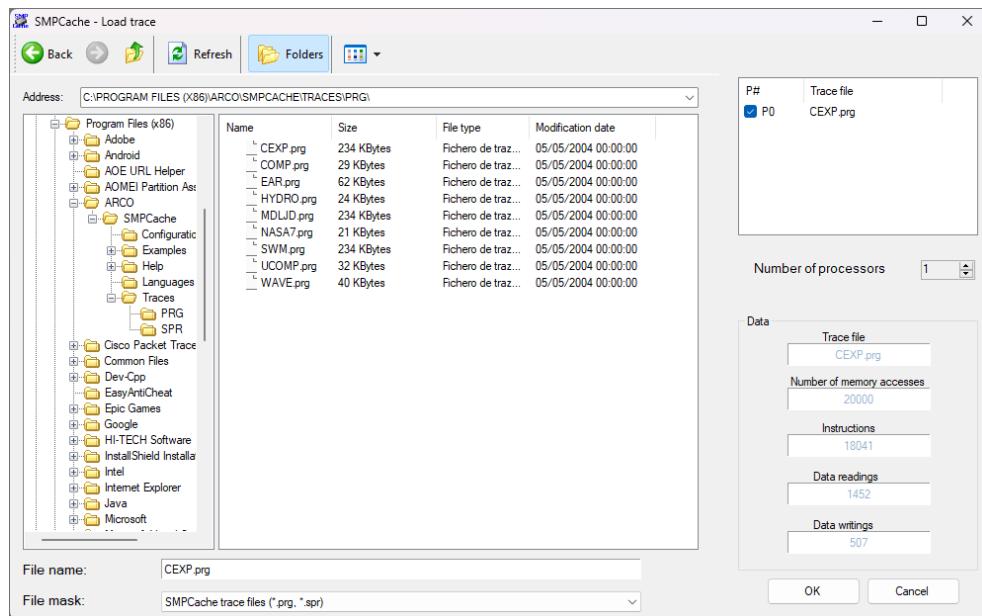
B. PROSEDUR PERCOBAAN

MEMULAI PERCOBAAN

1. PERCOBAAN 1: PENGARUH UKURAN CACHE

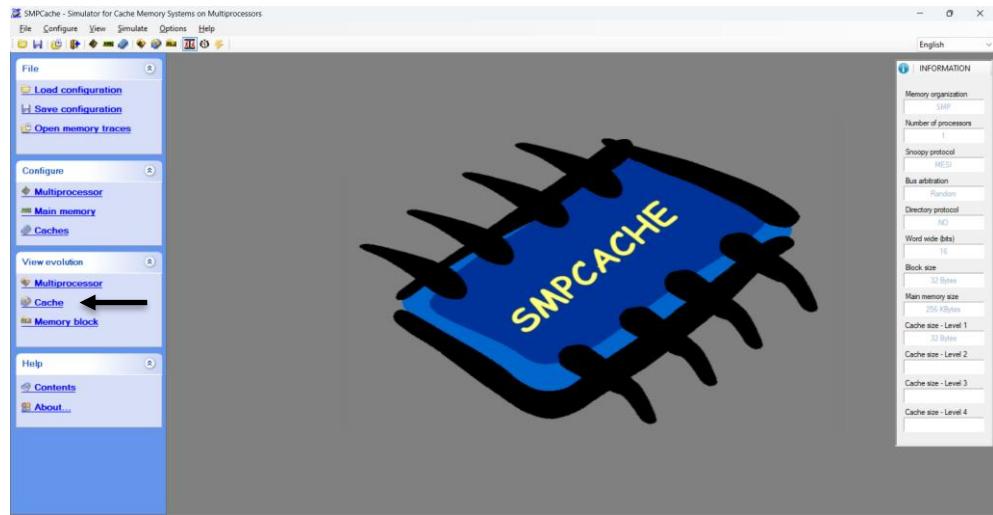
- a. Jalankan SMPCache kemudian konfigurasi sistem dengan karakteristik arsitektur sebagai berikut:
 - Processor in SMP = 1.
 - Cache coherence protocol = MESI.
 - Scheme for bus arbitration = Random.

- Word wide (bits) = 16.
 - Words by block = 16 (block size = 32 bytes).
 - Blocks in main memory = 8192 (main memory size = 256 KB).
 - Mapping = Fully-Associative.
 - Replacement policy = LRU.
- b. Konfigurasi blok dalam cache menggunakan konfigurasi berikut: 1 (ukuran cache = 0,03 KB), 2, 4, 8, 16, 32, 64, 128 tidak, 256, dan 512 (ukuran cache = 16 KB).
- c. Lakukan **screenshot** pada setiap arsitektur yang telah dikonfigurasikan.
- d. Buka *open memory traces* pada SMP Cache, arahkan direktori ke folder trace PRG dan lakukan **screenshot**.



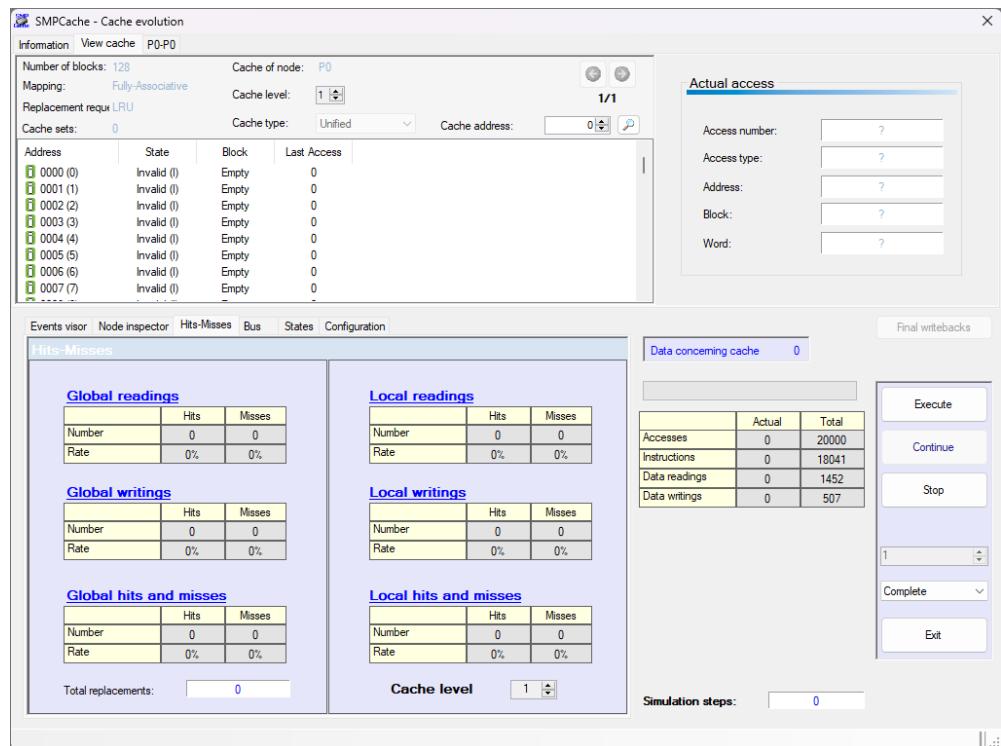
Gambar 1. 1 *Open Memory Traces*

- e. Lakukan load trace file dengan memilih trace file ekstensi .prg : CEXP.prg dan HYDRO.prg.
- f. Buka Cache pada bagian View Evolution.



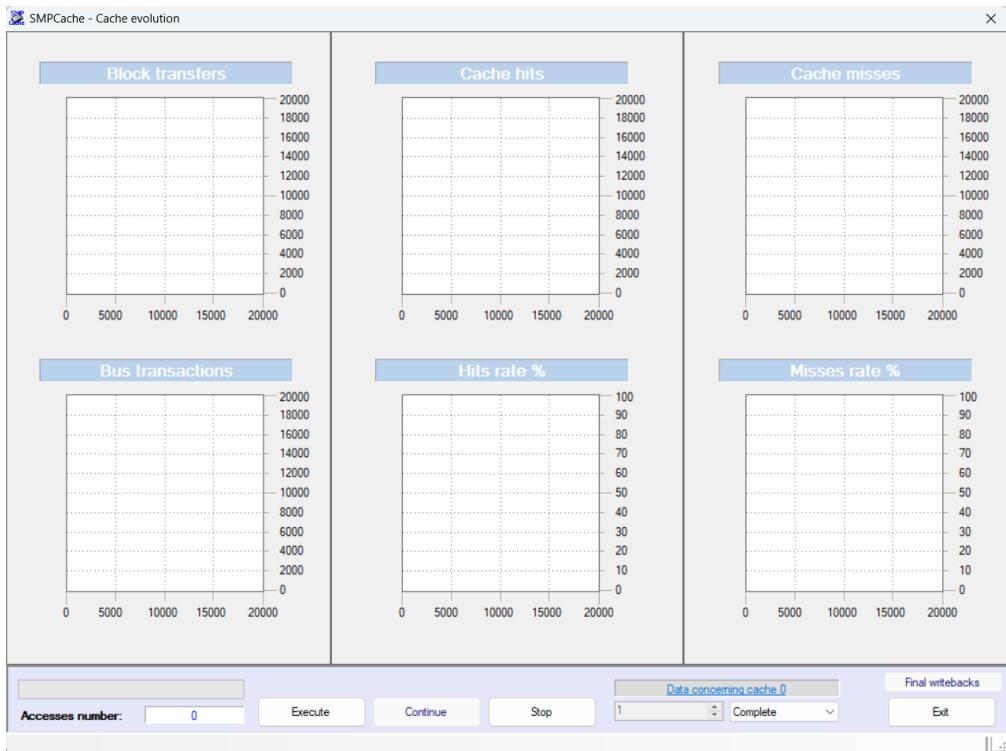
Gambar 1. 2 Pilih cache pada *View Evolution*

- g. Pilih bagian teks lalu pada tab Hits-Misses execute trace file dengan complete step untuk memperoleh miss rate memory traces dan lakukan screenshot



Gambar 1.2 Hits-Misses *memory traces*

- h. Pilih bagian graphic lalu execute trace file dengan complete step untuk memperoleh miss rate memory traces berbentuk grafik dan lakukan screenshot



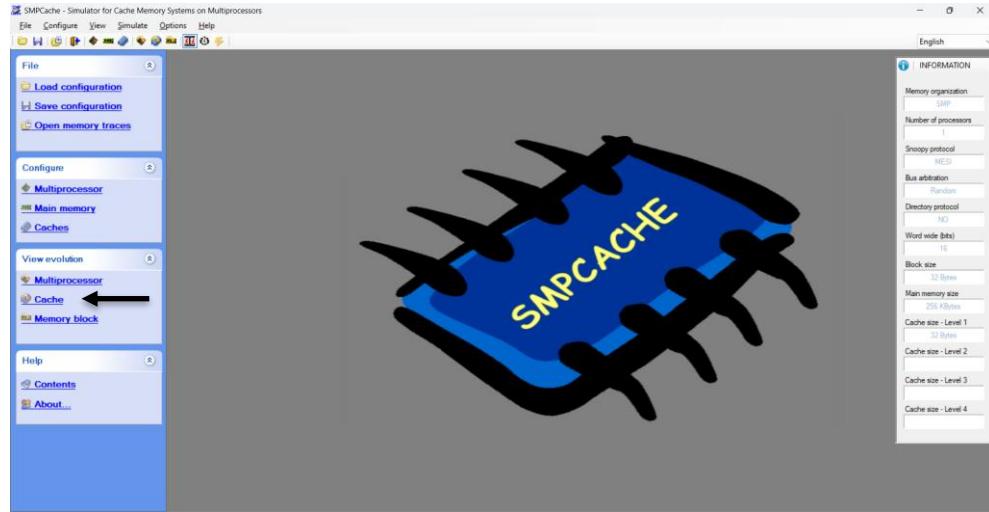
Gambar 1. 3 Grafik *memory traces*

2. PERCOBAAN 1: PENGARUH UKURAN BLOCK

- a. Jalankan SMPCache kemudian konfigurasi sistem dengan karakteristik arsitektur sebagai berikut:
 - Processor in SMP = 1.
 - Cache coherence protocol = MESI.
 - Scheme for bus arbitration = Random.
 - Word wide (bits) = 16.
 - Main memory size = 256 KB (the number of blocks in main memory will vary).
 - Cache size = 4 KB (the number of block in cache will vary).
 - Mapping = Fully-Associative.
 - Replacement policy = LRU.
- b. Konfigurasi words dengan block menggunakan konfigurasi berikut: 4 (block size = 8 bytes), 8, 16, 32, 64, 128, 256, 512, dan 1024 (ukuran block = 2048 bytes). Untuk masing-masing konfigurasi, dapatkan miss rate dengan

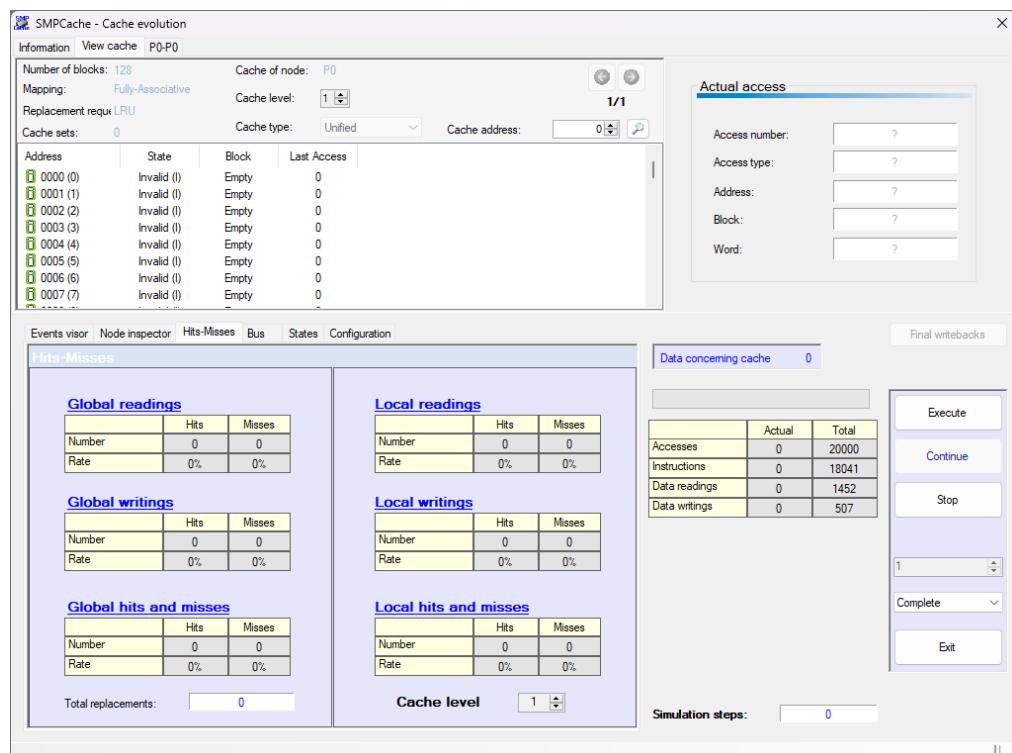
menggunakan trace file (extension ".prg"): Swm dan UComp.

- c. Lakukan **screenshot** pada setiap arsitektur yang telah dikonfigurasikan
- d. Lakukan load trace file dengan memilih trace file ekstension .prg untuk Swm.prg dan UComp.prg
- e. Buka *cache* pada bagian View Evolution



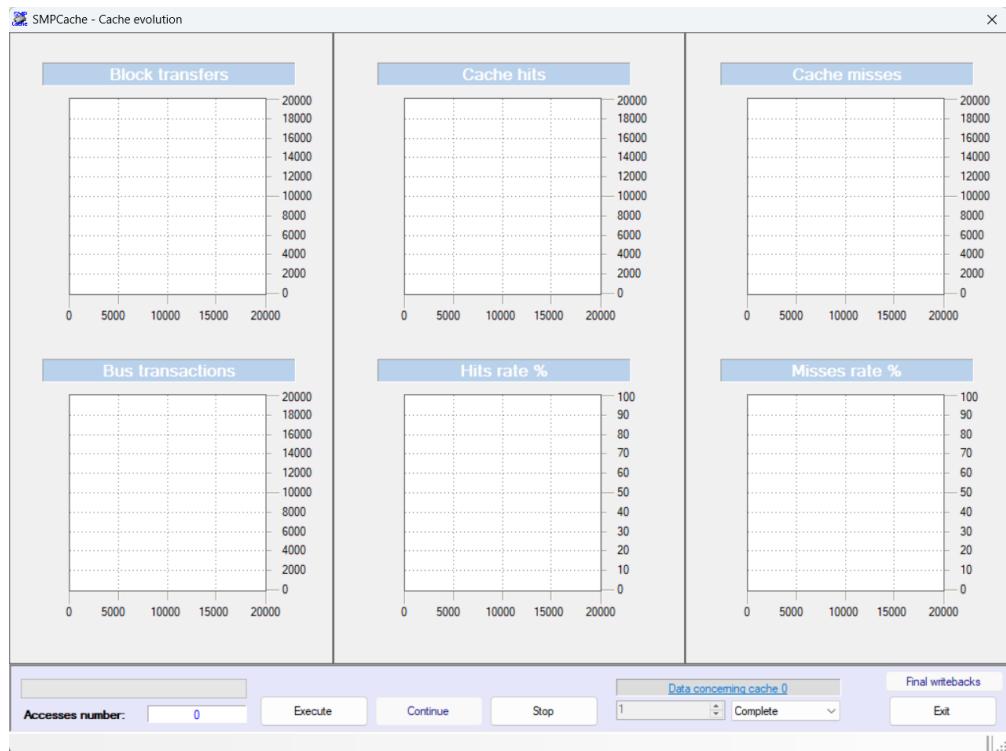
Gambar 1. 4 Pilih *cache* pada *View Evolution*

- f. Pilih bagian teks lalu pada tab Hits-Misses execute trace file dengan complete step untuk memperoleh miss rate memory traces dan lakukan screenshot



Gambar 1. 5 Hits-Misses *memory traces*

- g. Pilih bagian graphic lalu execute trace file dengan complete step untuk memperoleh miss rate memory traces berbentuk grafik dan lakukan screenshot



Gambar 1. 6 Grafik *memory traces*

1.5 PERTANYAAN JURNAL PRAKTIKUM

1.5.1 PERCOBAAN 1: PENGARUH UKURAN CACHE

- Apakah *miss rate* meningkat atau menurun saat ukuran cache meningkat? Mengapa? Apakah kenaikan atau penurunan ini terjadi untuk semua tolok ukur atau apakah itu bergantung pada nilai lokalitas yang berbeda?
- Apa yang terjadi dengan kapasitas dan konflik (*collision*) yang meleset saat Anda memperbesar cache? Apakah ada konflik yang dilewatkan dalam eksperimen ini? Mengapa?
- Dalam percobaan ini, dapat diamati bahwa untuk ukuran cache yang bagus, *miss rate* distabilkan. Mengapa?
- Apa yang menjadi indikasi perbedaan besar dari *miss rate* untuk peningkatan ukuran cache secara konkret? Apakah perbedaan besar dari *miss rate* ini muncul pada titik yang sama untuk semua program? Mengapa?
- Kesimpulannya, apakah peningkatan ukuran cache meningkatkan kinerja

sistem?

1.5.2 PERCOBAAN 2: PENGARUH UKURAN BLOCK

- a. Apakah *miss rate* meningkat atau menurun seiring dengan meningkatnya ukuran block? Mengapa ?
- b. Apakah kenaikan atau penurunan ini terjadi untuk semua tolok ukur atau apakah itu bergantung pada nilai lokalitas yang berbeda?
- c. Apa yang terjadi dengan *compulsory misses* saat Anda memperbesar ukuran block?
- d. Apa yang dimaksud dengan *pollution point* ? Apakah itu muncul dalam eksperimen ini?
- e. Kesimpulannya, apakah peningkatan ukuran block meningkatkan kinerja sistem?

1.6 PERTANYAAN ANALISA PRAKTIKUM

Jawab pertanyaan berikut sebagai analisa tambahan pada Laporan Akhir Praktikum Selama pengembangan percobaan, Anda dapat mengamati secara grafis secara umum, bagaimana miss rate menurun ketika pelaksanaan program berjalan maju. Mengapa? Berikan Alasannya?