# **Assignment 2 - Cache Evaluation**

## Abstract

In this assignment, we are aiming to achieve a simulator to evaluate the effects of different structures of cache on a system’s performance. Various combinations of cache size, associativity and words per block affect are considered. Other perspectives such as hit time, RAS latency, CAS latency, DRAM structure are chosen in sense of more realistic simulation of the real world situation.

## Simulation specification

Cache size: 8kB, 32kB, 512kB.

Associativity: direct mapping (1 way), 4 way set associative.

Words per block: 2 or 8

Hit time is chosen according to various cache size.

Miss time: 2 clock cycles plus DRAM access latency.

Combinations as follow:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Simulation No. | Associativity | Cache size (kB) | Words per block | Hit time |
| 1 | Direct mapping | 8 | 2 | 1 |
| 2 | Direct mapping | 32 | 2 | 2 |
| 3 | Direct mapping | 512 | 2 | 3 |
| 4 | Direct mapping | 8 | 8 | 1 |
| 5 | Direct mapping | 32 | 8 | 2 |
| 6 | Direct mapping | 512 | 8 | 3 |
| 7 | 4 way set associative | 8 | 2 | 1 |
| 8 | 4 way set associative | 32 | 2 | 2 |
| 9 | 4 way set associative | 512 | 2 | 3 |
| 10 | 4 way set associative | 8 | 8 | 1 |
| 11 | 4 way set associative | 32 | 8 | 2 |
| 12 | 4 way set associative | 512 | 8 | 3 |

## Latency source

Comparing to short latency (1-3 clock cycle) by accessing data in cache, the main source of latency is access data in DRAM. There are two type of DRAM access latency, RAS and CAS. RAS latency as 72 clock cycles is much higher than a CAS latency of 24 clock cycles. The occurrence of RAS is less than CAS, because of temporal and/or spatial locality. DRAM is specified to have a 64bit wide bus, 2 words per column, and 8 columns per row. In this simulation cache write through is adapted.

Here is considering situation when cache miss, needs access to main memory. For a reading activity, if it is accessing data on same raw as previous activity, it will be a RAS only. Otherwise, if it is access data on different row, clock delay will be both one RAS and one CAS. For a writing activity, because of cache write through, two CAS miss if access same row of previous activity, otherwise two CAS plus one RAS for accessing different row.

## Simulation result analysis

### Hit rate:

There are three factors affecting hit rate: cache size, associativity and words per block.

#### Effect by Cache size

These are the improvement of hit rate by changing cache size. All improvement calculated based on hit rate of 8kB cache size with same situation.

The main trend is obviously larger cache size will have higher hit rate. More data are store in the cache, there are more change of cache hit. Such improvement in hit rate will decrease yet still positive when either block size or associativity increases. If we have larger block size, spatial locality boost the hit rate even cache size is small. By having 4 ways associativity, ping pong effect due to conflict misses is solved, hence boost the hit rate even cache size is small.

#### Effect by block size

These are the improvement of hit rate by changing block size. All improvement calculated based on hit rate of 2 word per block with same situation.

By having more word per block, it load more data next to requested data, hence takes the advantage of spatial locality. The theory, when the block size becomes a significant proportion of cache size, it can drag down the hit rate. However, the block size used in simulation is small even for 8kB cache size. Result also shows improvement by increase block size is more significant when cache size is large and when set associative is used, this is because the data range within each block becomes wider.

#### Effect by associative

These are the improvement of hit rate by changing block size. All improvement calculated based on hit rate of direct mapping with same situation.

The overall improvement is about 5% of hit rate comparing to direct mapping. By set associative, for each index, there is now more than one tag available in the cache. Each index can hold the data from any address inside the mapped block. With 4-way set associative, the problems (collision and loop) with direct mapping are solved. Hence improves the system performance.

As result above also shows the improvement by having higher associative is more significant when cache size is small. When cache size is small, there is more change that data in cache is replace due to cache miss. In direct mapping, all data in the index will be replace, but fore set associative, only one of them will be replace, hence remaining data can be hit in the future, increasing system performance. When cache size is larger, cache will have more lines, such improvement will be less than that for smaller cache size.

#### Effect by matrix size

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ways | Data\_Size\_**kB** | Words\_Per\_Bock | Hit\_Time | Sets | Index\_Size | Tag\_Size | Hit\_Time | |
| 1 | 8 | 2 | 1 | 1024 | 10 | 19 | 1 |

### Clock cycle

## Best combination

