MC206: Computer Organization and Architecture

Cache memory : Performance Issues

Aneesh Panchal 2K20MC21

Ayushi Sagar 2K20MC35

For the project we have analyzed these research paper.

1. **Cache memory: An Analysis on Performance Issue**
2. **Cache memory: An Analysis on Optimization Techniques**
3. **Enhancing Cache Performance based on Improved Average Access Time**
4. **Cache memory: An Analysis on Replacement Algorithms and Optimization Techniques**
5. **Design strategy of cache memory for computer performance improvement**

CACHE MEMORY: AN ANALYSIS ON PERFORMANCE ISSUE

1. Reducing the miss rate using victim cache
2. Reducing the penalty ratio of cache miss
3. Decreasing memory access time

CONCLUSION :

In this work, different methods are proposed to improve the performance of cache memory. The main redemptions that we draw out from this study :

- conflict miss rate can be decreased after taking bigger block size, more cache size is required.
- Usage of bigger block size can rise penalty ratio of misses, decrease the time of the hit in addition to decreasing the power dissipation.
- Larger cache results in slow access time and more cost. More associatively results in fast access time and consequently less cycle time or lesser number of cycles.
- Victim cache always decreases the rate of misses but at a higher cost in contrast to look aside miss cache.

CACHE MEMORY: AN ANALYSIS ON OPTIMIZATION TECHNIQUES

1. REPLACEMENT ALGORITHMS
2. CACHE ALGORITHMS
3. DESIGN BASED OPTIMIZATION

CONCLUSION

In this paper, we discussed and evaluated the performance of different techniques and algorithms used for cache optimization.

- the rate of conflict misses is reduced by using larger block size, larger cache and way prediction methods. using larger block size may increase miss penalty, reduced hit time and power consumption.
- larger cache produces slow access time and high cost. It is associated with cache coherence problem. Higher associativity produce fast access time but they have low cycle time.
- Victim cache reduces miss rate at a high cost comparing to cache miss look aside.
- Lr + 5lf is a very good technique for reducing cache misses at a high rate comparing to some of the other techniques like LRU, FIFO, LFU . It is more complex then all of these methods.

ENHANCING CACHE PERFORMANCE BASED ON IMPROVED AVERAGE ACCESS TIME

1. DIRECT MAPPING
2. FULLY ASSOCIATIVE MAPPING
3. SECTOR MAPPING CACHE

CONCLUSION

- In this paper, we presented optimum hardware cache architecture to enhance the performance of the cache based on high hit ratio.
- Fewer cache lines than the main memory blocks, the cache line’s tags can be come onto unequally two groups of tags.
- The proposed EOT approach exploited the LSB of the tag field in the main memory address to distinguish between the match tags and miss match tags in the cache. The even-or odd comparator compares only the LSBS of the cache line’s tag and the memory field tag and rejects any miss match cache line in very low time.
- Only the matched cache line’s tags are passed to the main cache comparator for a long and complete c-character comparison. In this way a lot of miss match line’s tags are grasped and rejected quickly in the even-or odd comparator and they do not need to go to the main cache comparator. Building on that, there is no waste of time and a minimum time for cache line hitting is reached.

CACHE MEMORY: AN ANALYSIS ON REPLACEMENT ALGORITHMS AND OPTIMIZATION TECHNIQUES

REPLACEMENT ALGORITHM

1. LRU (LEAST RECENTLY USED) ALGORITHM
2. HLFU (HISTORY LEAST RECENTLY USED) ALGORITHM
3. LFU (LEAST FREQUENTLY USED) ALGORITHM
4. GDS (GREEDY DUAL SIZE)
5. CAR (CLOCK WITH ADAPTIVE REPLACEMENT) ALGORITHM
6. ARC (ADAPTIVE REPLACEMENT CACHE) ALGORITHM
7. RR (RANDOM REPLACEMENT) ALGORITHM
8. SLRU (SEGMENTED LRU) ALGORITHM
9. LR+5LF ALGORITHM
10. FIFO ALGORITHM
11. LLF (LOWEST LATENCY FIRST) ALGORITHM

OPTIMIZATION TECHNIQUES

- One way to minimize the gap between memory latency and CPU cycle is the use of a multilevel cache.
- Miss rate of L1 (first level) cache can be reduced by introducing L2 cache. Another technique called ULCC(user level cache control) has been implemented through which user can control space allocation in cache.
- This technique is hard to implement but it produces less hit rate and also reduce cache pollution.
- Cache optimization is compiler based optimization in which loops are optimizing through compiler. To set the accessed data in cache loops must be reduced to smaller size. In this way all the tasks will be executed consecutively which will be using same data from cache.

CONCLUSION :

By comparing different techniques we analyze

- cache miss rate is reduced by using larger block size, larger cache, and way prediction method.
- Larger block size reduces hit time, increases miss penalty and consume high power.
- Higher associativity produce fast access time but they have low cycle time.
- ULCC have fast access time and also used to eliminate cache pollution. Pipelined cache reduced miss penalty.
- Multilevel cache has very less miss penalty but yields high cycle time and have high power consumption.

DESIGN STRATEGY OF CACHE MEMORY FOR COMPUTER PERFORMANCE IMPROVEMENT

- In this paper, it is tried to show how can the speed and performance can be improved. Following are the many factors that affect the performance if computer such as processor speed, the size of RAM and the weakness of the cache memory.
- Cache operation at high level

  The entire cache line is loaded to L1d whenever memory content is needed by the processor. Let’s consider a 64 bytes cache line so it means lower 6 bits are zero. The remaining bits are in some cases used to locate the line in the cache and as the tag.

- The simulation process basically depends on normal parameters and ram size in order to evaluate the effect of multi different cache levels on both processor speed and overall computer performance.