Al-Aqsa University Journal (Natural Sciences Series), Vol.14, No.1, Pages 69-80, Jan.2010
ISSN 2070-3155

# The Preferable Bank Number for Multibanked-Caches

**Dr. Aiman A. Abu Samra***
**Dr. Adwan F Abdel Fattah****

## الملخص

## عدد الضفاف الأمثل في الكاش متعدد الضفاف

حيث أن الفجوة تزداد بين سرعة المعالج وسرعة الذاكرة العشوائية، وحيث أن سـرعة المعالج تعتمد إلى حد كبير على سرعة الكاش بقد قام كثير من الباحثين بتقديم أفكار لزيادة سرعة الكاش. ومن هذه الأفكار الضفاف المتعددة في الكاش حيث أن هـذه الـضفاف المتعـددة تـدعم الوصول المتزامن للبيانات مما يزيد السرعة الكلية. وحيث أن بعض المعالجات تستخدم ضـفتين وبعضها ثلاث وبعضها أربع و البعض لا يستخدم تكنولوجيا الضفاف، فان هذه الورقـة تـدرس تأثير إختيار عدد معين من الضفاف على سرعة المعالج والطاقة المستهلكة والمساحة اللازمـة. وقد تم استخدام برنامجي محاكاة للحصول على النتائج.

## ABSTRACT

## The Preferable Bank Number for Multibanked-Caches

Since the gap between main memory access time and processor cycle time is continuously increasing, processor performance dramatically depends on the behavior of caches, many researchers worldwide have come forward to present performance of the current caches to decrease this gap. One of the used approaches is multi-banked caches, in which the caches is divided into independent banks that can support simultaneous accesses, two loads or two stores in parallel, rather than treat the cache as a single monolithic block acts accesses sequentially. Some processors use banking (AMD Opteron - two banks, Sun Niagara - four banks and IBM Power5 - three banks). But in the other hand some other processors does not use banking, like INTEL P4. This paper concerned on the effect of choosing a certain number of cache-banks on the performance, power and area of the cache. We used SimpleScalar with benchmarks spec95 to evaluate the performance of the cache with different banks-number. Also we use CACTI 5.3 and eCACTI tools to evaluate the power and area parameters corresponding to each banks number.

*Islamic University of Gaza.
**Arab American University

**Keywords:** Multibanked, Tradeoffs, Cache Power, Cache-Area, Cache-Performance.

# 1. INTRODUCTION

CPU caches are small memories on or close to the CPU chip, can be made faster than the much larger main memory. Most CPUs since the 1980s have used one or more caches, and modern general-purpose CPUs inside personal computers may have as many as half a dozen, each specialized to a different part of the task of executing programs. Modern CPUs have one overriding concern--keeping the processing pipelines filled with instructions and data, but this is becoming an increasingly difficult task because although CPU performance doubles every 18 to 24 months, the speed of the DRAM chips that constitute main memory increases by only a few percent each year [4]. As a result, the high-speed cache memory that acts as a buffer between main memory and the CPU is an increasingly significant factor in overall performance. In fact, upgrading your cache may give you more of a performance boost than upgrading your processor. Cache design and implementation can make or break the performance of your high-powered computer system. Larger caches avoid the need to go to memory for data as often but require a longer latency access time. Thus, caches in processors are commonly split into multiple banks to achieve large capacities while keeping access latency low [8]. One of the highly effective parameter in performance is miss rate, that is a rate of failed attempts to read or write a piece of data in the cache, which results in a main memory access with much longer latency. There are three kinds of cache misses: instruction read miss, data read miss, and data write miss. In order to lower cache miss rate, a great deal of analysis has been done on cache behavior in an attempt to find the best combination of size, associativity, block size, and so on.

Because the two most important parameters in architecture are power and area, the designers should keep track that the resulting cache gives a low miss rate, low power consumption and a small area.

Also to achieve low miss rates, modern processors employ banked-caches rather than treating the cache as a single monolithic block. In banked-caches the cache is divided into independent banks that can support simultaneous accesses. Banks were originally used to improve performance of main memory and are now used inside modern DRAM chips as well as with caches. Clearly, banking works best when the accesses naturally spread themselves across the banks, so the mapping of addresses to banks affects the behavior of the memory system. A simple mapping that works well is to spread the addresses of the block sequentially across the banks, called sequential interleaving. For example, if there are four banks, bank 0 has all

blocks whose address modulo 4 is 0; bank 1 has all blocks whose address modulo 4 is 1; and so on. Figure 1 shows this interleaving.

The following sections are organized as follows (section 2) states the nature of the problem, (section 3) mentions the previous related work, (section 4) gives a brief discussion about some related parameters used in calculating cache performance, (section 5) is a assigned for simulation and evaluation, and finally we present our conclusions and future directions in (Section 7).

## 2. STATEMENT OF PROBLEM

As what mentioned before, the parameters that affects the cache miss rate are cache size, associativity, block size , banks-number and some other parameters. In this paper we concentrate on the effect of banks-number in miss rate. The problem is, we discover that increasing the number of banks lead to a decrease in the miss rate but unfortunately this also lead to an increase in cache power dissipation and cache area. Then our work is to make some tradeoffs between these parameters to find the preferable banks-number that gives the best miss rate with a low power dissipation and a small area.

## 3. RELATED WORK

It is very interesting to search and work in the range of cache banking, there are a plenty of resources in the cache power/performance tradeoffs for example the study to determine the factors and parameters that are critical for minimizing the power dissipation in caches without significantly affecting cache performance [10]. But unfortunately the available resources about cache power/performance tradeoffs including banking are almost negligible, despite the decrease in the resources we are insisted to search this range.

## 4. CALCULATION OF CACHE PERFORMANCE PARAMETERS

There are a huge number of parameters that affect the cache performance but we will mention only the used parameters in our paper that are miss rate, access time, power and area.

### 4.1. Cache miss rate

One measure of the benefits of different cache organizations is miss rate. Miss rate is simply the fraction of cache accesses that result in a miss—that is, the number of accesses that miss divided by the number of accesses.

$$\text{Miss rate} = \frac{\frac{\text{Misses}}{1000}\ /1000}{\frac{\text{Memory accesses}}{\text{Instructions}}} \quad [5]$$

Miss rates can be measured with cache simulators that take an address trace of the instruction and data references, simulate the cache behavior to determine which references hit and which miss, and then report the hit and miss totals. Many microprocessors today provide hardware to count the number of misses and memory references, which is a much easier and faster way to measure miss rate.

## 4.2. Cache access time

Average memory access time (A.M.A.T.) is the time interval between the instant at which an instruction control unit initiates a call for data or a request to store data, and the instant at which delivery of the data is completed or the storage is started. While multi-banked-caches incur fewer misses than single-banked caches, they typically have slower access times.

The average memory access time can be calculated using the following formula:

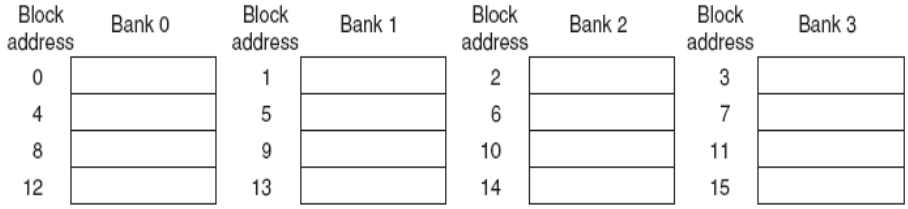


**Figure 1:** Four-way interleaved cache banks using block addressing. Assuming 64 bytes per blocks, each of these addresses would be multiplied by 64 to get byte addressing.

**A.M.A.T. = Hit time + Miss rat × Miss penalty[5]**

Average memory access time is still an indirect measure of performance; although it is a better measure than miss rate, it is not a substitute for execution time [5].

## 4.3. Cache power dissipation

While multi-banked-caches incur fewer misses than single-banked caches, they typically have extra dynamic power consumption due to increased bank accesses[9]. The power dissipation can be calculated using the following formula [2]:

$$\text{Dynamic Power(W)} = \frac{\text{Dynamic Energy (J)}}{\text{Random Cycle Time(S)}}$$

**The Preferable Bank Number for…**

The motivation for splitting a cache into multiple banks is that long word-lines and bit-lines have high capacitance which lead to both long delays as well as high power consumption [8].

### 4.4. Cache Area

Multi-banked-caches typically have larger area than single-banked caches. And dual ported cache requires approximately twice the area of a single ported cache. The larger area corresponds to slower access times as well as increased power usage [8] Therefore, our goal is to reduce the area required by a multi-banked caches without sacrificing performance.

## 5. SIMULATION AND EVALUATION

To evaluate the performance and power consumption of the multi-banked cache system on real-world workloads, we used the SimpleScalar3.0b simulator for performance
results and used the CACTI5.3 and eCACTI cache models to evaluate the power consumption and access latency of our tested banked-cache.

### 5.1 Simulation Environment

Table 1 shows the parameters used for CPU parameters default values [1]. And our tested CACHE parameters values are shown in Table 2.

| Table 1: CPU parameter values | |
|---|---|
| **PROCESSOR** | |
| Fetch Width | 4 (instructions) |
| Branch Misprediction Latency | 3(cycles) |
| Decode Width | 4 (instructions) |
| Capacity of the load/store queue | 16(instructions) |
| Number of L1 cache ports | 2 (ports) |

| Table 2:CACHE parameter values | |
|---|---|
| CACHE | |
| Cache size | 512(Kbyte) |
| Line size | 64(byte) |
| Associativity | 1 |
| Bank number | 1,2,4,8, and 16 |
| Technology node | 80 (ns) |
| Replacement policy | LRU |

## 5.2 Simulation Results

In this section we decide to use some testing tools and make our tests to compare the performance between some chosen cache patterns.
To achieve this goal we have been used the SimpleScalar version 3.0b installed on Linux SUSE platform using some int-benchmarks from SPEC95 benchmark suite, these benchmarks are the most known game benchmark named GO.ss and 3 other benchmarks are compress95.outorderO0.gcc.100M.ss, compress95.outorderO0.mirv.100M.ss and compress95.outorderO1.gcc.100M.ss.
And for every stated benchmark we vary the cache bank-number from 1 to 16 through 2,4 and 8, without changing the other cache parameters as mentioned in table 2. And for each one of these patterns we record the performance parameter like miss rate, number of hits, number of accesses, and number of misses. And other parameters like Access Time, Dynamic Power consumption and Cache Area we have been used the eCACTI and CACTI 5.3 cache model that is available in two forms: a web-based version [2] and a C++ source code version. And since the two versions are the same , we decided to use the web based one for simplicity.

### 5.2. 1. Results using Go.ss benchmark

Figure 2 shows the enhancement of one of the performance parameters, miss rate, corresponding to an increase of cache's bank number. It is clear that using multi-banked caches is more better than using single-banked caches with a percentage of 56.25 % when the banks number is 4 and any other increase in the banks number is useless.

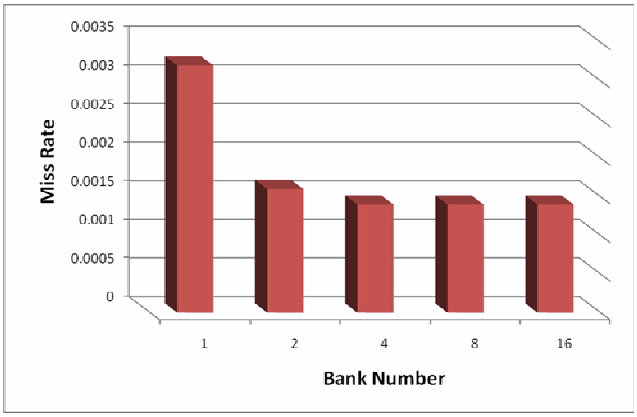


**Figure 2:** Miss Rate improvement result for various cache's bank number-GO.

### 5.2. 2. Results using the benchmark Compress95.outorderO0.gcc.100M.ss

Figure 3 shows the enhancement of the miss rate corresponding to an increase of cache's bank number. It is clear that using multi-banked caches is more better than using single-banked caches with a percentage of 5.26% when the banks number is 4 and any other increase in the banks number is useless. This decrease in the percentage of miss rate improvement with respect to GO.ss percentage is due to the nature of the used benchmark, such that GO.ss is more complicated and hence has a larger number of

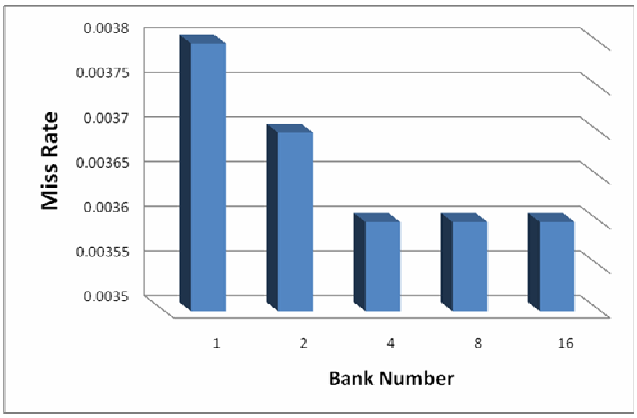


**Figure 3:** Miss Rate improvement result for various cache's bank number-Compress95gcc0.

cache accesses so it is more vulnerable to an extra miss rate than the Compress95 benchmarks.

### 5.2. 3. Results using the benchmark Compress95.outorderO0.mirv.100M.ss

Figure 4 shows the enhancement of the miss rate corresponding to an increase of cache's bank number. It is clear that using multi-banked caches is more better than using single-banked caches with a percentage of 25% when the banks number is 4 and any other increase in the banks number is useless.

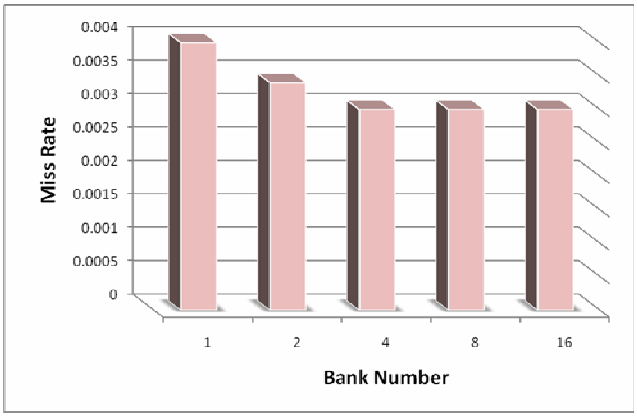


**Figure 4:** Miss Rate improvement result for various cache's bank number-Compress95mirv.

### 5.2. 4. Results using the benchmark Compress95.outorderO1.gcc.100M.ss

Figure 5 shows the enhancement of the miss rate corresponding to an increase of cache's bank number. It is clear that using multi-banked caches is more better than using single-banked caches with a percentage of 5.41% when the banks number is 4 and any other increase in the banks number is useless.

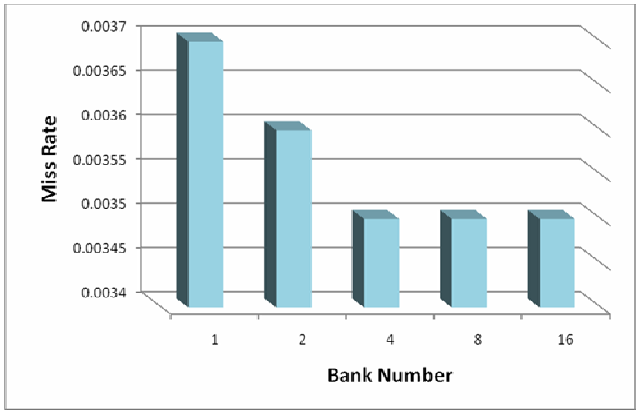


**Figure 5:** Miss Rate improvement result for various cache's bank number-Compress95gcc1.

### 5.2. 5. Cache access time results

Like what is shown it Figure 6 bellow and according to the access time calculation equation shown above, although the miss rate is decreased

it is a natural result to have an increase in access time when there is an increase in the cache's banks number because the hit time is increased, and the miss penalty does not changed.

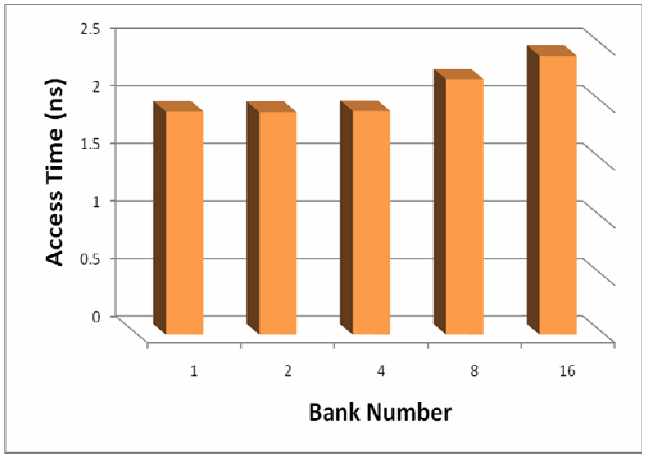


**Figure 6:** The effect on access time for various cache's bank number.

**5.2. 6. Cache dynamic power results**

For the CPU to hit a block in the cache, multi-banked cache dissipates extra power than the single banked cache, that is for both read and write operations. This extra power dissipation results from the power dissipated in routing to bank especially when our cache contains a large number of banks. These power dissipation results are shown in figue 7 bellow.

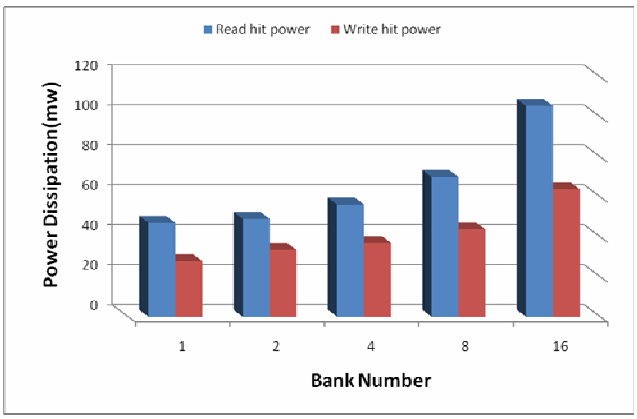


**Figure 7:** Read and write hitt power results for various cache's bank number.

And when the CPU misses a block, either in read or write operationes, in the cache there is also an extra power dissipation in multi-banked caches than the power dissipation in the single banked caches, that is also because the power dissipated in routing to bank during the searching operation for the LRU block. These power dissipation results are shown in figue 8 bellow.

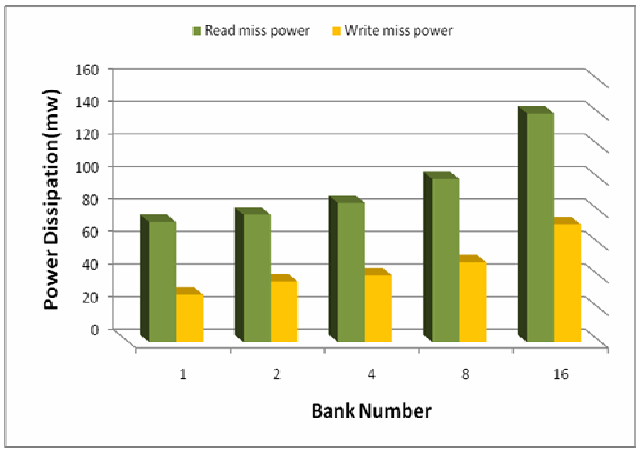


**Figure 8**: Read and write miss power results for various cache's bank number.

**5.2. 7. Cache area results**

When the number of cache banks is increased this leads to an increase in the size of the cache architecture and hence this leads to an increase in the cache area. The area results are shown in figure 9 bellow.

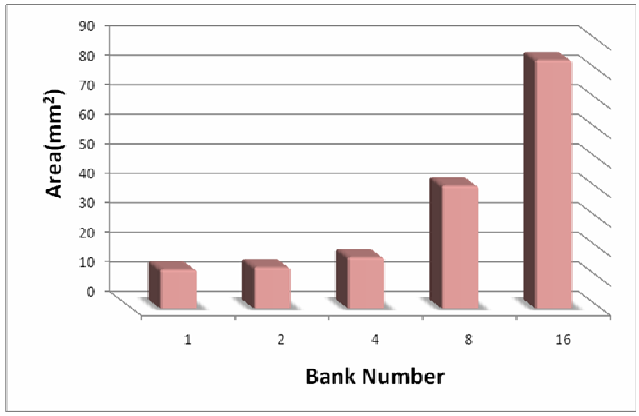


**Figure 9:** Area results for various cache's bank number.

## 5.2 Evaluation of the results

The most important parameters that the cache designers give an importance are miss rate, access time, power and area [3] then when we need to choose the preferable bank number we should take care and make a tradeoffs between these parameters. Our results showed that the best miss rate is achieved using cache bank number equal or greater than 4, Hence to avoid an extra power and area, we should exclude banks number greater than 4. For the cache banks number 1 and 2 the decrease in power and area parameters are negligible with respect to cache bank number 4. Then the best multi-banked cache's bank number is 4.

We also should mention that the performance improvement when moving from single-banked caches to multi-banked caches is not so that large and the use of banked caches gives an increase in power and area that is the reason that Intel Pentium 4 excluded the use of multi-banked caches and use only single-banked cache [5]. So we concluded that if designers are not concerned with banks they should use single banked caches else they should use a 4-banks banked-cache but they must use some available architecture techniques to reduce the power dissipation generated by this number of banks, one architecture they can use is the low power cache architecture [6], which is based on separating the in-coming cache data, data is separated in two different banks, bank1 that mostly contains 1's data, and bank0 that mostly contain 0's data. This separation reduces transistor switching activity inside the on-chip cache and hence dynamic power consumption. Another cache architecture they can use is the architecture that uses a technique utilizes cache line address locality to determine (rather than predict) the cache bank prior to the cache access, it thus allows only the desired bank to be accessed for both tags and data [7]. And there is a plenty of techniques that can be used.

# 6. CONCLUSION

Multi-banked caches are useful for cache designers who concentrate on decreasing the miss rate, one parameter that share others in the calculation of the whole cache performance, as possible. But these designers are still suffering from an increase in power dissipation and larger area. In this paper we study the effect of changing cache's bank number on the miss rate, access time ,power, and area. The simulation results shows an improvement percentage in miss rate of 23% ,degradation in power dissipation of 20.85% and area of 24.07% compared to single-banked caches. These improvements appeared with a multi-banked cache with banks number equal to 4 or greater. We finally concluded that the preferable

bank number in the multi-banked caches is 4 as a tradeoff between miss rate from one side and power, area from the other side.

## 7. References


1. Burger D., Austin T., "*The SimpleScalar Tool Set"*, United States, 2000.
2. CACTI model, http://quid.hpl.hp.com:9081/cacti.
3. C. Su and A. Despain, "*Cache design tradeoffs for power and performance optimization*: *A case study*," in International Symposium on Low Power Electronics and Design,2000.
4. David F. Bacon, "cache advantage", *IBM T. J. Watson Research Center in Hawthorne, New York,2003*.
5. Hennessy J., Patterson D., *Computer Architecture - A Quantitative Approach 4ed*, United States, 2007.
6. N. Mouna and A. Nadia, "*Low Power Cache Architecture" , Sharjah, UAE,* 2006.
7. V. Alex, N Alex and N. Dan, "*Reducing Power Consumption for High Associativity Data Caches in Embedded Processors", USA,* 2004.
8. F-O. John, "*Putting Bank Conflicts Behind BARS*", *University of California, San Diego, USA,* 2007.
9. P. Foglia, A. Bardine, G. Gabrielli, C.A. Prete, "*Evaluating Power Consumption of D-NUCA caches", University of Pisa, Italy,* 2008.
10. S. Bandypadhyay, C. Baker , "*Energy Based Analysis of Cache Design", University of California, Berkeley,USA*,2004.