**CS203 Fall 2016**

**LAB 3**

**ANSWERS**

**Lalitha Dwarapudi (SID:861310053)**

1.

In 512 KB 4-way set associative cache with 16B block size:

Cache size = 524288 B

Block size= 16 B

Number of ways= 4

**The number of bits the tag has = 15**

Number of sets= 8192

Total Cache size = 16×4×8192 = 524288 B

Total tag size= 15×4×8192= 491520 bits = 61440 B

The total size of the cache including tag bits, in bytes = 524288+61440

= 585728 B= 572 KB

**The total size of the cache including tag bits is 572 KB**

2.

Cache size = 524288 B

Block size= 16 B

Number of lines=32768

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trace | Direct(Miss rate, tag bits, index bits, offset bits) | 2-way | 4-way | Fully Assoc |
| gcc-10K | **5.47**, 13, 15, 4 | **5.46**, 14, 14, 4 | **5.46**, 15, 13, 4 | **5.46**, 28, 0, 4 |
| gcc-1M | **0.8374**, 13, 15, 4 | **0.7015,** 14, 14, 4 | **0.6977**, 15, 13, 4 | **0.6977,** 28, 0, 4 |

3.

Cache size = 262144 B

Block size= 16 B

Number of lines=16384

Number of sets= Number of lines/Number of ways

Number of offset bits= log2Block\_size

Number of index bits= log2Number\_of\_sets

Number of tag bits= 32- offset bits - index bits

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trace | Direct(Tag,index,offset) | 2-way | 4-way | Fully Assoc |
| gcc-10K | (14,14,4) | (15,13,4) | (16,12,4) | (28,0,4) |
| gcc-1M | (14,14,4) | (15,13,4) | (16,12,4) | (28,0,4) |

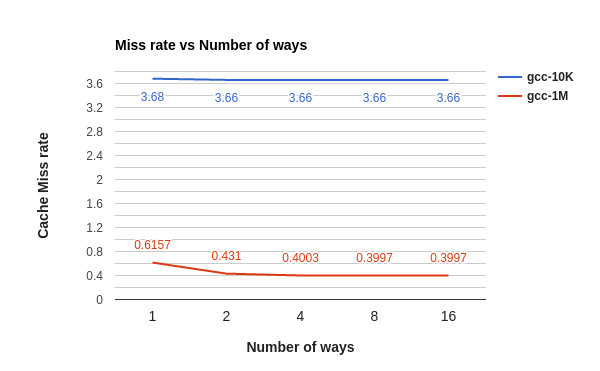
4.

Cache size = 262144 B

Block size= 32B

A **highly-associative cache will have a lower miss rate** because each set has more blocks, so there’s less chance of a conflict between two addresses. This will remove some of the conflict misses in Direct mapped caches. So, increasing the number of ways will reduce the number of misses which in turn decreases the miss rate.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Trace | 1-way | 2-way | 4-way | 8-way | 16-way |
| gcc-10K | 3.68 | 3.66 | 3.66 | 3.66 | 3.66 |
| gcc-1M | 0.6157 | 0.431 | 0.4003 | 0.3997 | 0.3997 |



Miss rate decreased from 1-way to 2-way and then becomes almost constant for both the traces. There is small difference in miss rate when we go from 1-way to 2-way because the conflict misses are reduced by increasing the associativity. There is not much difference in miss rate when we increase associativity from 2-way to 16-way because the conflict misses that are reduced is very less. In gcc-10K, there are 6 conflict misses in 1-way and 0 in 2-way. So, the miss rate is constant after 2-way because no conflict misses are reduced after 2-way.

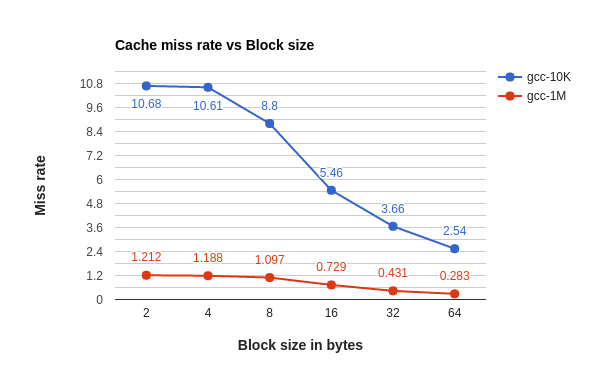
5.

Cache size = 256KB

2-way set associative cache

Increasing the block size decreases the cache miss rate. This is because it reduces the number of cold misses. Larger blocks take advantage of spatial locality. Larger block size means that fewer blocks will be in cache and this increases capacity misses and conflict misses.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Trace | 2B | 4B | 8B | 16B | 32B | 64B |
| gcc-10K | 10.68 | 10.61 | 8.80 | 5.46 | 3.66 | 2.54 |
| gcc-1M | 1.212 | 1.188 | 1.097 | 0.729 | 0.431 | 0.283 |



Miss rate is decreased when block size is increased from 2B to 64B. Since most of the misses are cold misses, the miss rate is reduced significantly because increasing the block size decreases the number of cold misses.Since there are very less conflict misses that are increased from 2B to 64B, it doesn’t affect the miss rate. In gcc-1M, the miss rate is very less because the input file has more addresses and the number of cold misses are less. So, the miss rate in gcc-1M decreases slightly when block size is increased from 2B to 64B.

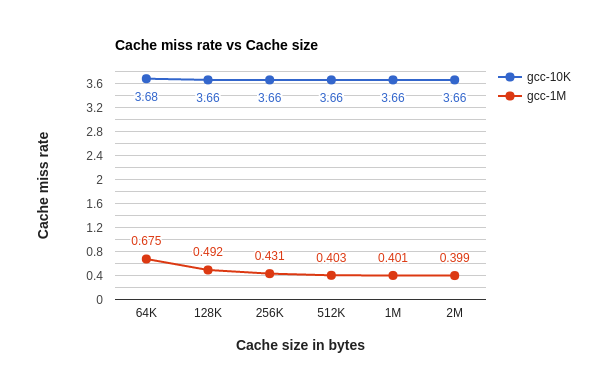
6.

Block size= 32B

2-way set associative cache

In a larger cache there’s less chance that there will be of a conflict. This means the miss rate decreases if the cache size is increased. This reduces number of capacity misses.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Trace | 64K | 128K | 256K | 512K | 1M | 2M |
| gcc-10K | 3.68 | 3.66 | 3.66 | 3.66 | 3.66 | 3.66 |
| gcc-1M | 0.675 | 0.492 | 0.431 | 0.403 | 0.401 | 0.399 |



As the Cache size is increased from 64KB to 2MB, the miss rate is decreased very slightly in both the traces. This is because the Cache is never full and there are no capacity misses to reduce the miss rate by increasing the Cache size. Moreover, in both the traces, the miss rate is very less because the input file has more addresses and there are very less number of misses. So, even after increasing the Cache size to 2M, the miss rate decreases very slightly.

**Extra Credit:**

Calculate the cold, capacity, and conflict misses for the two traces using a 512KB 4-way set associative cache with 16B block size

Cache size= 524288 B

Number of ways= 4

Block size= 16B

**For gcc-10K.memtrace:**

Cold misses= 546

Conflict misses= 0

Capacity misses= 0

Total misses= 546

**For gcc-1M.memtrace:**

Cold misses= 10706

Conflict misses= 17

Capacity misses= 0

Total misses= 10723