

## Computer Architecture Assignment-2

Course Name: EGC 223, Computer Architecture Memory Design

Group Name : Chip Squad , Members : Kavya Gupta (IMT2023016), Pragya Rai (IMT2023529), Ananya Vundavalli (IMT2023537)

6 September 2024

### 1 Introduction

- In this assignment we are building a set associative cache. We have four parts included in this assignment.
- In our code, we have implemented a cache simulation using object oriented approach in python. We have made various classes.
- The BlockLine class represents a block in the cache which tracks last access time to implement LRU policy, valid bit and tag.
- The EachSet class represents a line or set in the cache. It contains a list of BlockLine objects. We have implemented methods get *lru\_index* and *each\_block\_access* to determine the least recently used block and for updating access time.
- The code initializes several trace files and loads them into memory.
- The user selects a part of the program to execute and the code includes a check for invalid user input, ensuring the user selects an appropriate part ('a', 'b', 'c', or 'd').
- Part a : We designed a 4-way set associative cache of size 1024 kilobytes, Block size: 4 bytes.  
Part b: In this part we are changing the cache size from 128kB to 4096 kB.  
Part c: Here we are changing the block size from 1 byte to 128 bytes.  
Part d: We are changing the number of ways from 1 way to 64 ways.
- Note: all the changes in the number of cache sizes, block sizes and associativities are done in powers of 2 between the provided ranges.
- We find miss and hit rates for each trace file while varying the parameters specified. We plot the graphs and tables as mentioned in the problem statement.

### 2 Parts

#### 2.1 a) 4-way Set-Associative cache

- We are given a cache size of 1024 kilobytes and block size of 4 bytes
- Number of bytes in one cacheLine= block size \* number of ways=4\*4=16 bytes
- number of cache lines= cache size(in bytes)/number of bytes in one cache line=1024 \*1024/16=65,536

```

Read the instructions below:
For the hit and miss rates of each trace file, enter: a
For results on varying cache size, enter: b
For results on varying block size, enter: c
For results on varying associativity, enter: d
Input= a

Results for Part A:
+-----+-----+-----+-----+
| Trace   | Config   | Hit Rate | Miss Rate |
+-----+-----+-----+-----+
| gcc.trace | Fixed Config | 93.8356 | 6.16445 |
+-----+-----+-----+-----+
| gzip.trace | Fixed Config | 66.8694 | 33.1306 |
+-----+-----+-----+-----+
| swim.trace | Fixed Config | 92.7343 | 7.26567 |
+-----+-----+-----+-----+
| twolf.trace | Fixed Config | 98.8868 | 1.11324 |
+-----+-----+-----+-----+
| mcf.trace | Fixed Config | 4.12345 | 95.8765 |
+-----+-----+-----+-----+
PS D:\sem3\ca_assgn>

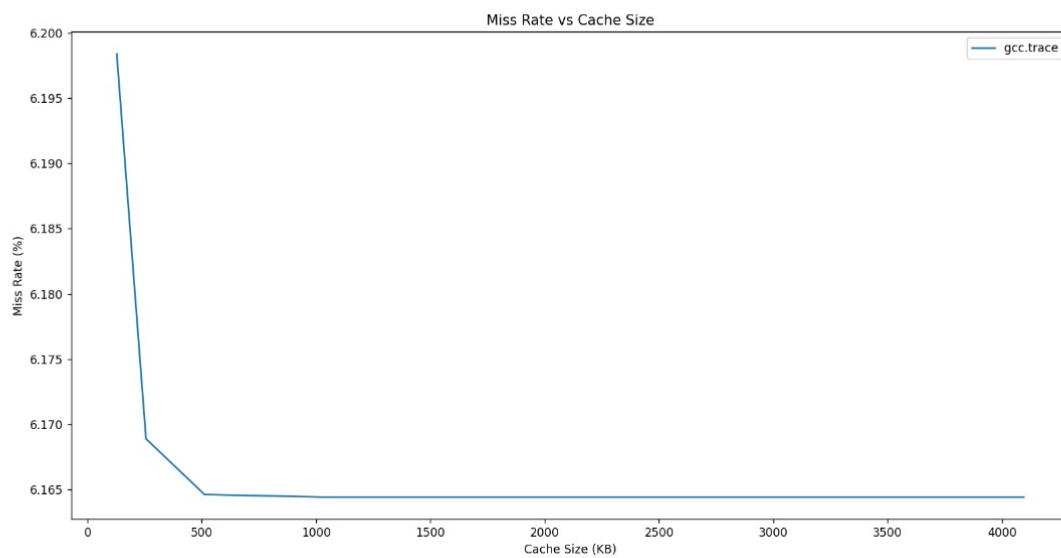
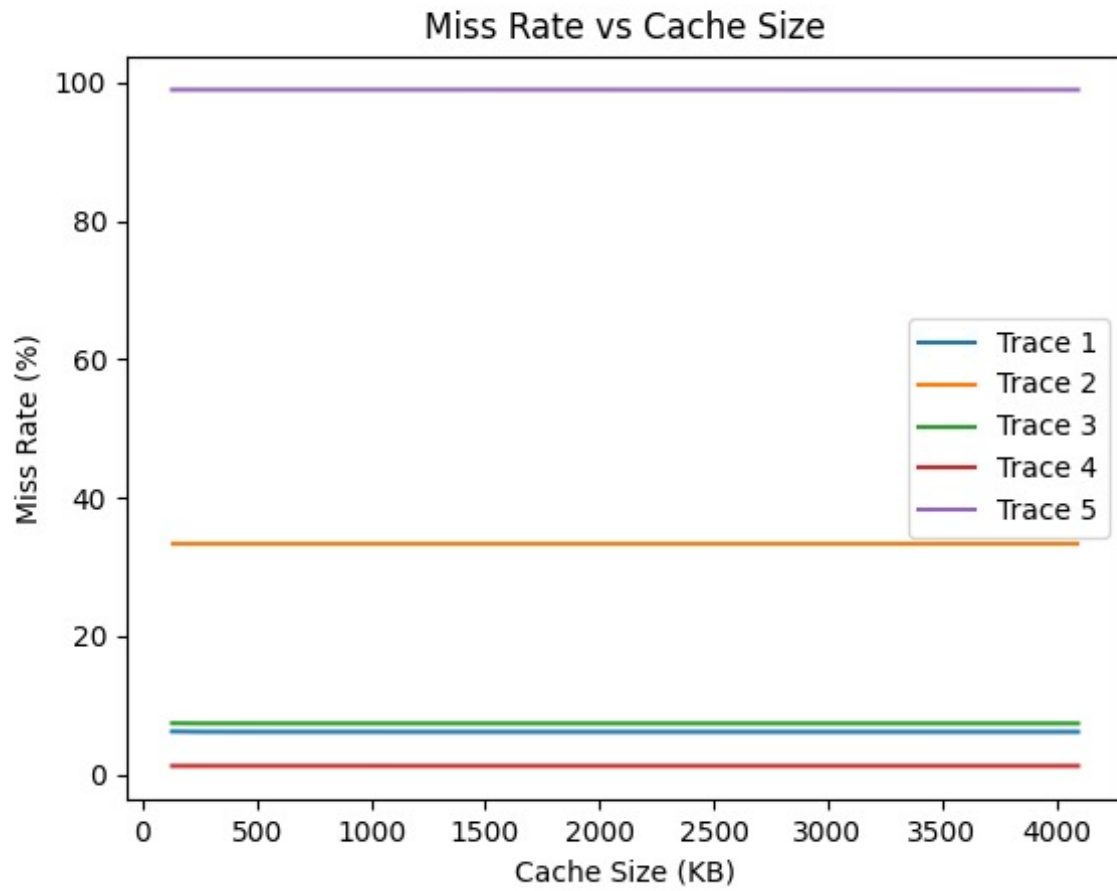
```

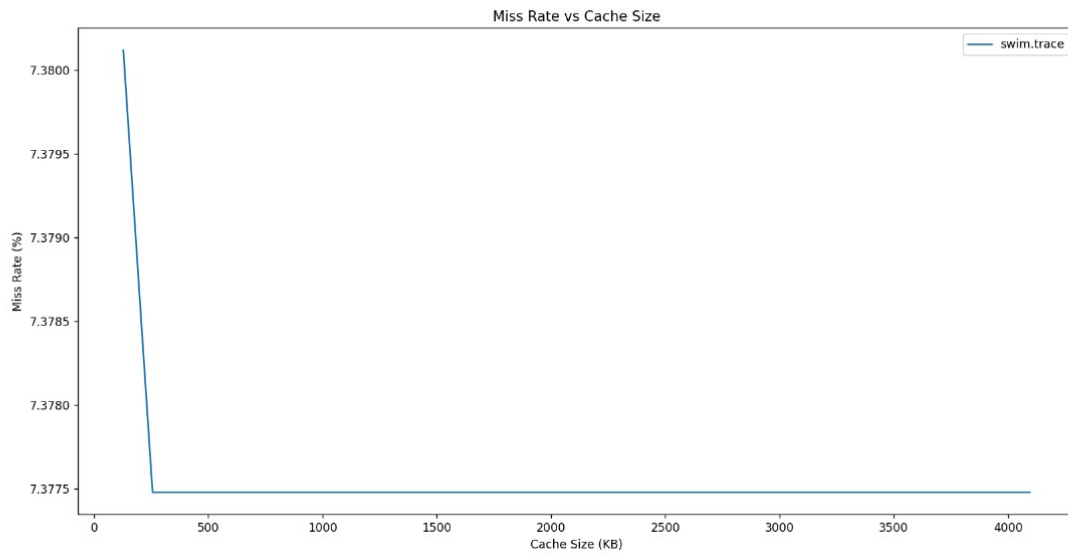
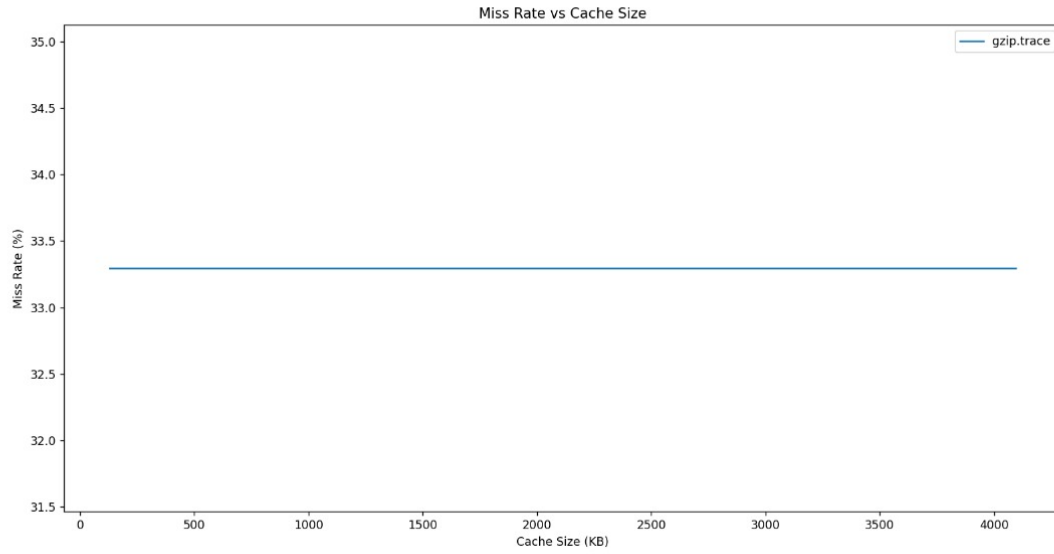
#### Observations:

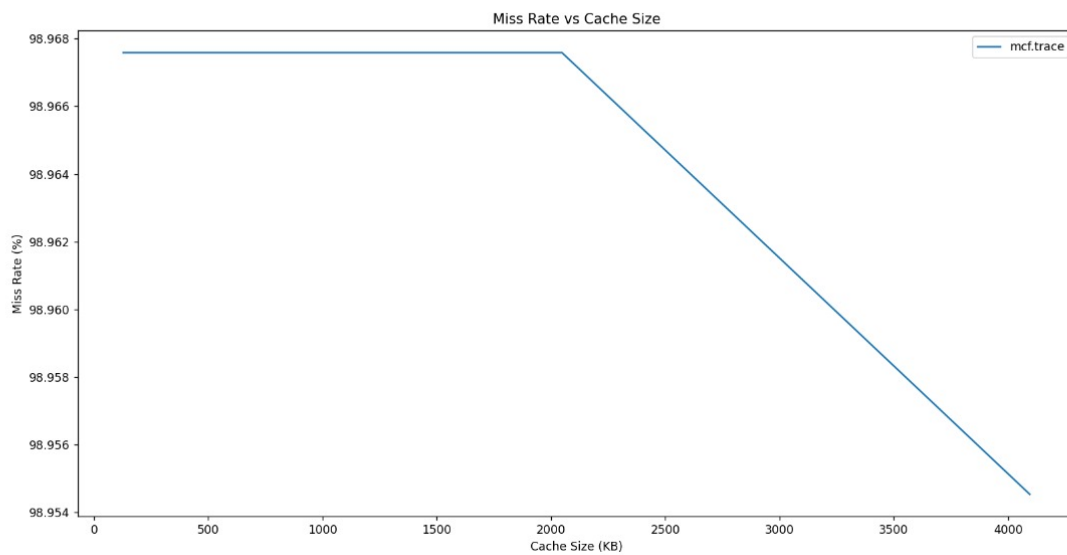
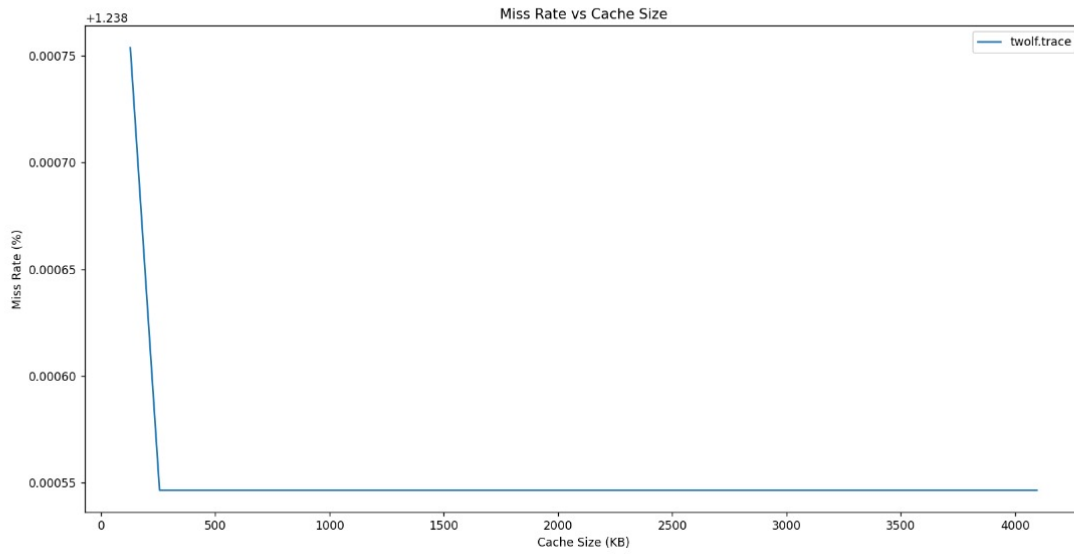
- The above table shows the hit and miss rates of each trace file.
- Descending order of hit rates: *twolf.trace* > *gcc.trace* > *swim.trace* > *gzip.trace* > *mcf.trace*)

## 2.2 b) 4-way set-associative cache with varying cache size

- In this experiment, the cache size was varied from 128 kB to 4096 kB to observe the effect on cache performance, specifically the miss rate.
- The goal was to analyze how increasing cache size impacts the cache's ability to store more data and reduce the number of misses.
- The simulation was run across multiple trace files, and the miss rate for each trace was recorded and plotted.







#### Observations:

- All the traces do not behave the same way. We can see that even after varying the cache size, changes in miss rates are very minute.
- This indicates that the cache is not exploiting temporal and spatial locality to a large extent.
- Hence, we have put individual graphs highlighting these minute differences in the miss rates.

Read the instructions below:

For the hit and miss rates of each trace file, enter: a

For results on varying cache size, enter: b

For results on varying block size, enter: c

For results on varying associativity, enter: d

Input= b

Results for Part B:

Trace	Config	Hit Rate	Miss Rate
gcc.trace	Cache Size: 128 KB	93.8016	6.19838
gcc.trace	Cache Size: 256 KB	93.8311	6.16891
gcc.trace	Cache Size: 512 KB	93.8354	6.16464
gcc.trace	Cache Size: 1024 KB	93.8355	6.16445
gcc.trace	Cache Size: 2048 KB	93.8355	6.16445
gcc.trace	Cache Size: 4096 KB	93.8355	6.16445
gzip.trace	Cache Size: 128 KB	66.7055	33.2945
gzip.trace	Cache Size: 256 KB	66.7055	33.2945
gzip.trace	Cache Size: 512 KB	66.7055	33.2945
gzip.trace	Cache Size: 1024 KB	66.7055	33.2945
gzip.trace	Cache Size: 2048 KB	66.7055	33.2945
gzip.trace	Cache Size: 4096 KB	66.7055	33.2945



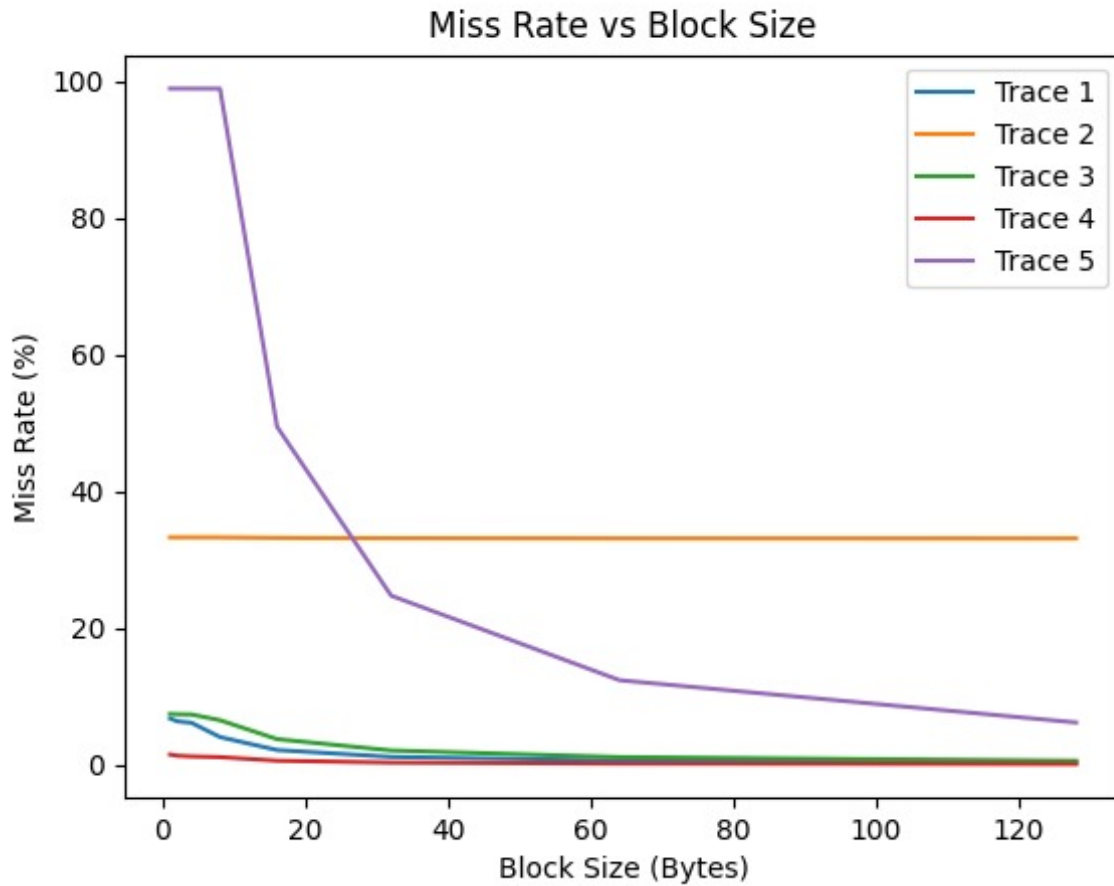
```

| swim.trace | Cache Size: 128 KB | 92.6199 | 7.38012 |
+-----+-----+-----+-----+
| swim.trace | Cache Size: 256 KB | 92.6225 | 7.37748 |
+-----+-----+-----+-----+
| swim.trace | Cache Size: 512 KB | 92.6225 | 7.37748 |
+-----+-----+-----+-----+
| swim.trace | Cache Size: 1024 KB | 92.6225 | 7.37748 |
+-----+-----+-----+-----+
| swim.trace | Cache Size: 2048 KB | 92.6225 | 7.37748 |
+-----+-----+-----+-----+
| swim.trace | Cache Size: 4096 KB | 92.6225 | 7.37748 |
+-----+-----+-----+-----+
| twolf.trace | Cache Size: 128 KB | 98.7613 | 1.23875 |
+-----+-----+-----+-----+
| twolf.trace | Cache Size: 256 KB | 98.7614 | 1.23855 |
+-----+-----+-----+-----+
| twolf.trace | Cache Size: 512 KB | 98.7614 | 1.23855 |
+-----+-----+-----+-----+
| twolf.trace | Cache Size: 1024 KB | 98.7614 | 1.23855 |
+-----+-----+-----+-----+
| twolf.trace | Cache Size: 2048 KB | 98.7614 | 1.23855 |
+-----+-----+-----+-----+
| twolf.trace | Cache Size: 4096 KB | 98.7614 | 1.23855 |
+-----+-----+-----+-----+
| mcf.trace | Cache Size: 128 KB | 1.03241 | 98.9676 |
+-----+-----+-----+-----+
| mcf.trace | Cache Size: 256 KB | 1.03241 | 98.9676 |
+-----+-----+-----+-----+
| mcf.trace | Cache Size: 512 KB | 1.03241 | 98.9676 |
+-----+-----+-----+-----+
| mcf.trace | Cache Size: 1024 KB | 1.03241 | 98.9676 |
+-----+-----+-----+-----+
| mcf.trace | Cache Size: 2048 KB | 1.03241 | 98.9676 |
+-----+-----+-----+-----+
| mcf.trace | Cache Size: 4096 KB | 1.04547 | 98.9545 |
+-----+-----+-----+-----+
PS D:\sem3\ca_assgn>

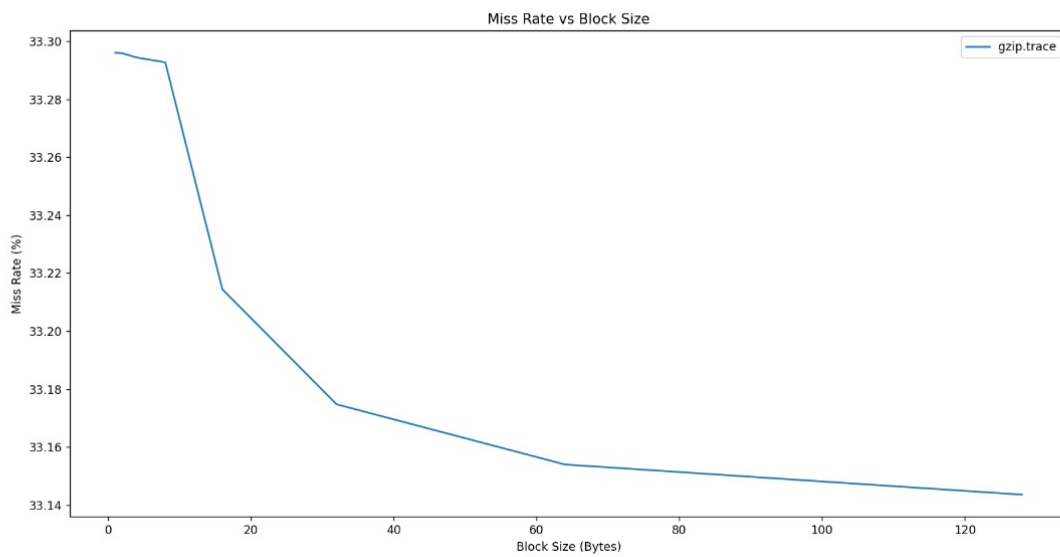
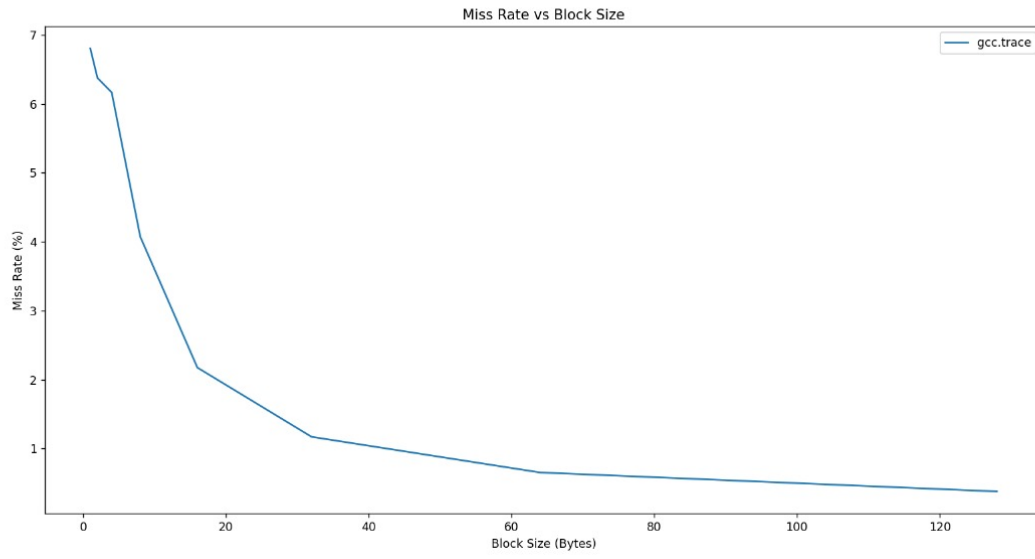
```

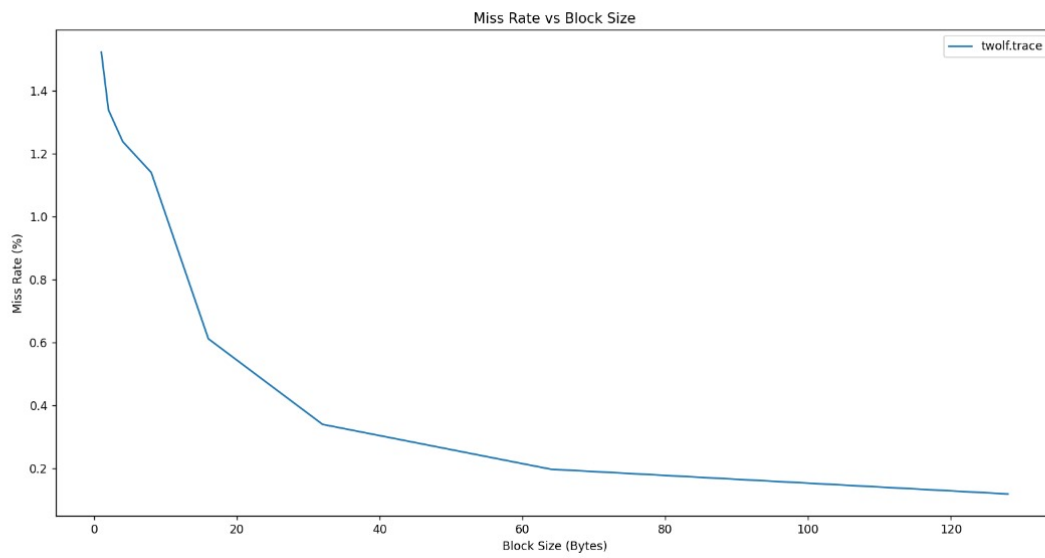
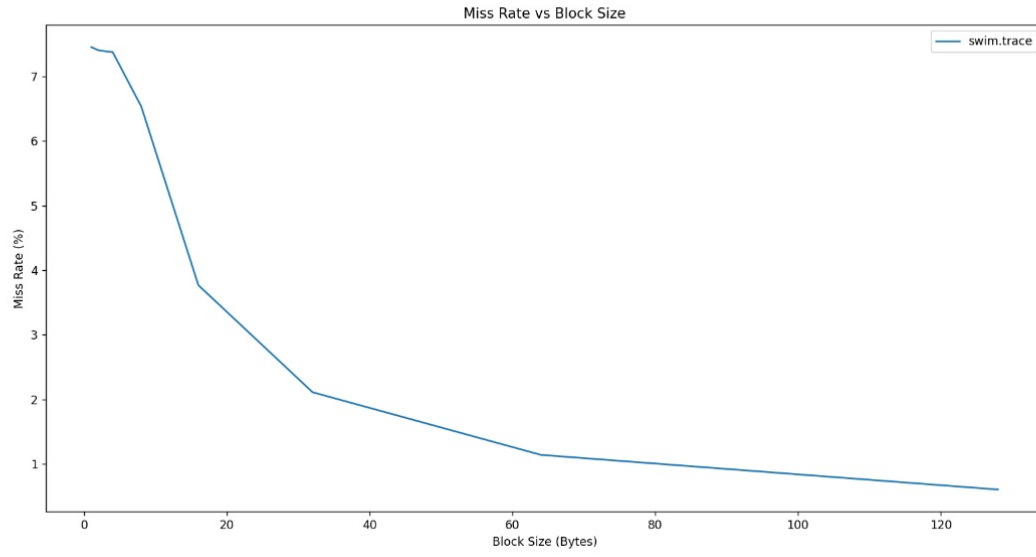
### 2.3 c) 4-way set-associative cache with varying blockSize

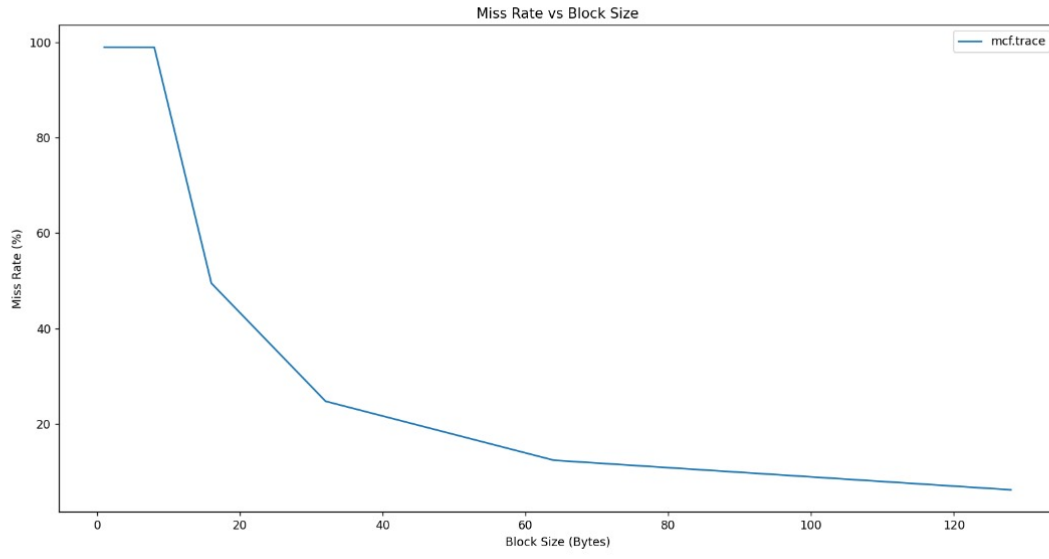
- In this experiment, the **cache size was fixed at 1024 kB**, and the block size was varied from **1 byte to 128 bytes**.
- The purpose was to study how changing the block size affects the miss rate, considering that larger block sizes reduce the number of cache lines available.
- The simulation was repeated for all trace files, and the miss rates were plotted against block size for each trace.











Read the instructions below:

For the hit and miss rates of each trace file, enter: a

For results on varying cache size, enter: b

For results on varying block size, enter: c

For results on varying associativity, enter: d

Input= c

Results for Part C:

Trace	Config	Hit Rate	Miss Rate
gcc.trace	Block Size: 1 Bytes	93.1989	6.80108
gcc.trace	Block Size: 2 Bytes	93.6248	6.37523
gcc.trace	Block Size: 4 Bytes	93.8355	6.16445
gcc.trace	Block Size: 8 Bytes	95.9266	4.07343
gcc.trace	Block Size: 16 Bytes	97.825	2.17498
gcc.trace	Block Size: 32 Bytes	98.8289	1.17107
gcc.trace	Block Size: 64 Bytes	99.3459	0.65408
gcc.trace	Block Size: 128 Bytes	99.6209	0.37911

Enter Caption

gzip.trace	Block Size: 1 Bytes	66.7039	33.2961	
gzip.trace	Block Size: 2 Bytes	66.7041	33.2959	
gzip.trace	Block Size: 4 Bytes	66.7055	33.2945	
gzip.trace	Block Size: 8 Bytes	66.7072	33.2928	
gzip.trace	Block Size: 16 Bytes	66.7856	33.2144	
gzip.trace	Block Size: 32 Bytes	66.8253	33.1747	
gzip.trace	Block Size: 64 Bytes	66.8461	33.1539	
gzip.trace	Block Size: 128 Bytes	66.8565	33.1435	
swim.trace	Block Size: 1 Bytes	92.5443	7.45565	
swim.trace	Block Size: 2 Bytes	92.5935	7.4065	
swim.trace	Block Size: 4 Bytes	92.6225	7.37748	
swim.trace	Block Size: 8 Bytes	93.4642	6.53577	
swim.trace	Block Size: 16 Bytes	96.2324	3.76757	
swim.trace	Block Size: 32 Bytes	97.8905	2.10955	
swim.trace	Block Size: 64 Bytes	98.8611	1.13888	
swim.trace	Block Size: 128 Bytes	99.3977	0.60226	

Enter Caption



twolf.trace	Block Size: 1 Bytes	98.4769	1.52312
+-----+		+-----+	+-----+
twolf.trace	Block Size: 2 Bytes	98.6608	1.3392
+-----+		+-----+	+-----+
twolf.trace	Block Size: 4 Bytes	98.7614	1.23855
+-----+		+-----+	+-----+
twolf.trace	Block Size: 8 Bytes	98.8598	1.14017
+-----+		+-----+	+-----+
twolf.trace	Block Size: 16 Bytes	99.388	0.61202
+-----+		+-----+	+-----+
twolf.trace	Block Size: 32 Bytes	99.6599	0.34008
+-----+		+-----+	+-----+
twolf.trace	Block Size: 64 Bytes	99.8024	0.19759
+-----+		+-----+	+-----+
twolf.trace	Block Size: 128 Bytes	99.8809	0.11909
+-----+		+-----+	+-----+
mcf.trace	Block Size: 1 Bytes	1.02457	98.9754
+-----+		+-----+	+-----+
mcf.trace	Block Size: 2 Bytes	1.0287	98.9713
+-----+		+-----+	+-----+
mcf.trace	Block Size: 4 Bytes	1.03241	98.9676
+-----+		+-----+	+-----+
mcf.trace	Block Size: 8 Bytes	1.03832	98.9617
+-----+		+-----+	+-----+
mcf.trace	Block Size: 16 Bytes	50.503	49.497
+-----+		+-----+	+-----+
mcf.trace	Block Size: 32 Bytes	75.2378	24.7622
+-----+		+-----+	+-----+
mcf.trace	Block Size: 64 Bytes	87.6081	12.3919
+-----+		+-----+	+-----+
mcf.trace	Block Size: 128 Bytes	93.7955	6.2045
+-----+		+-----+	+-----+
PS D:\sem3\ca_assgn> █			

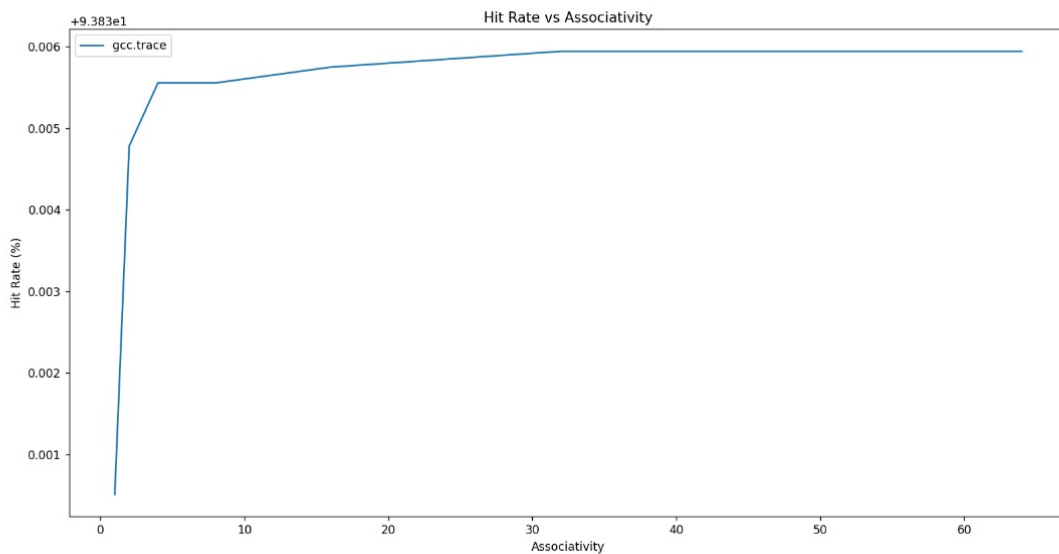
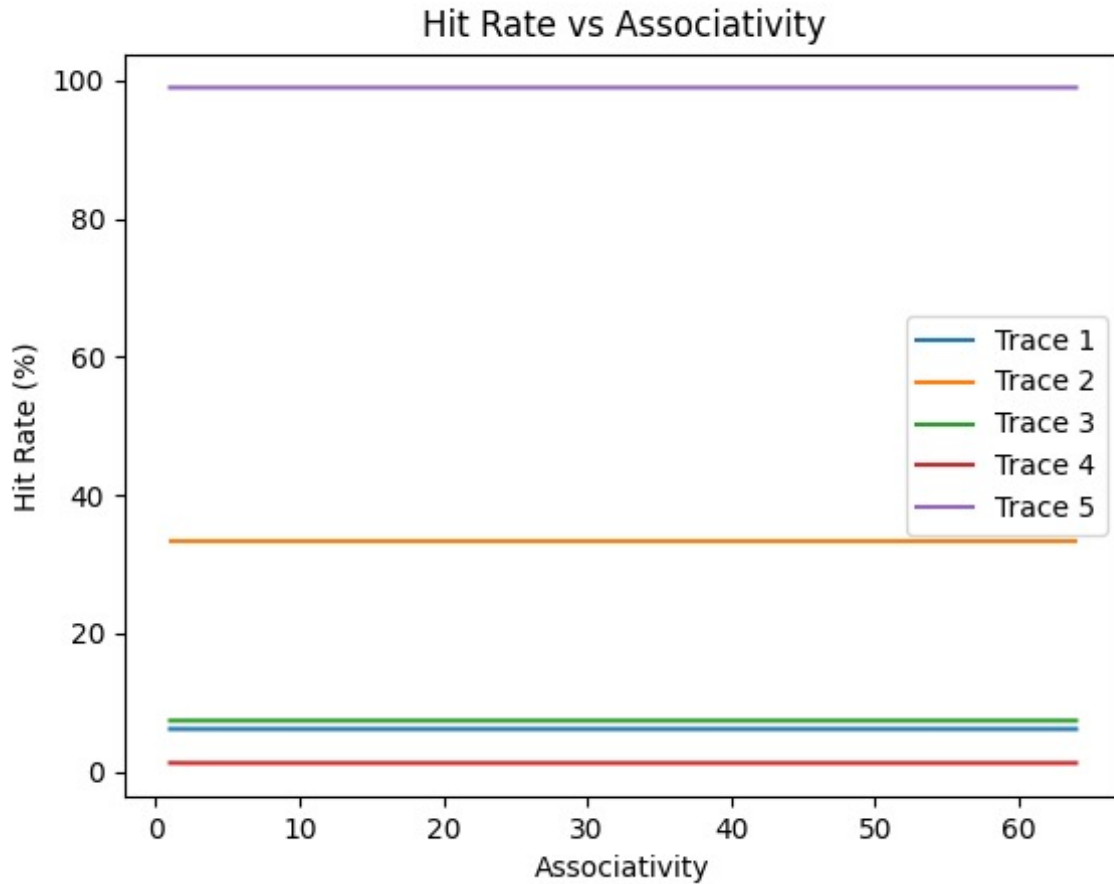
#### Observations:

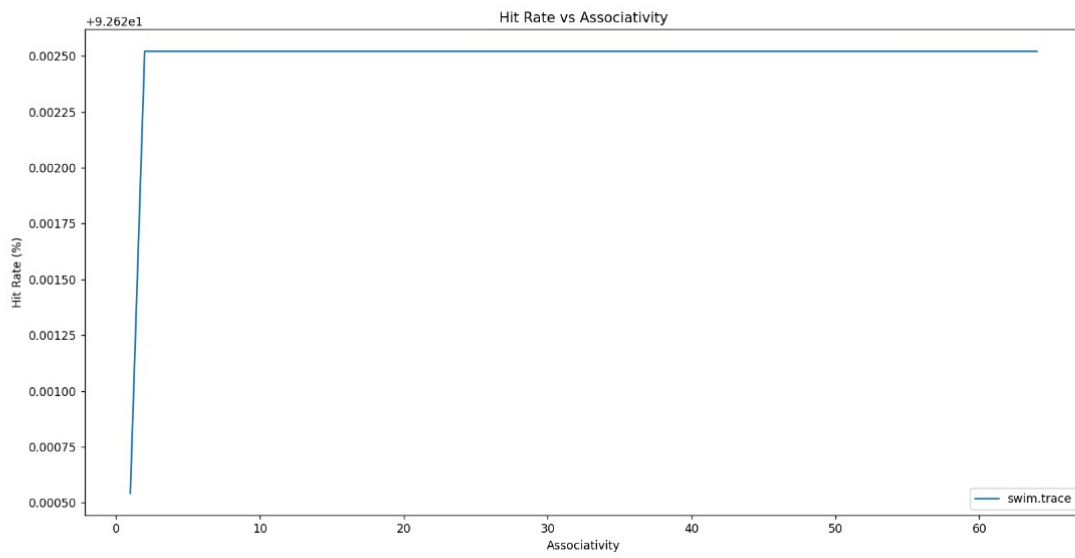
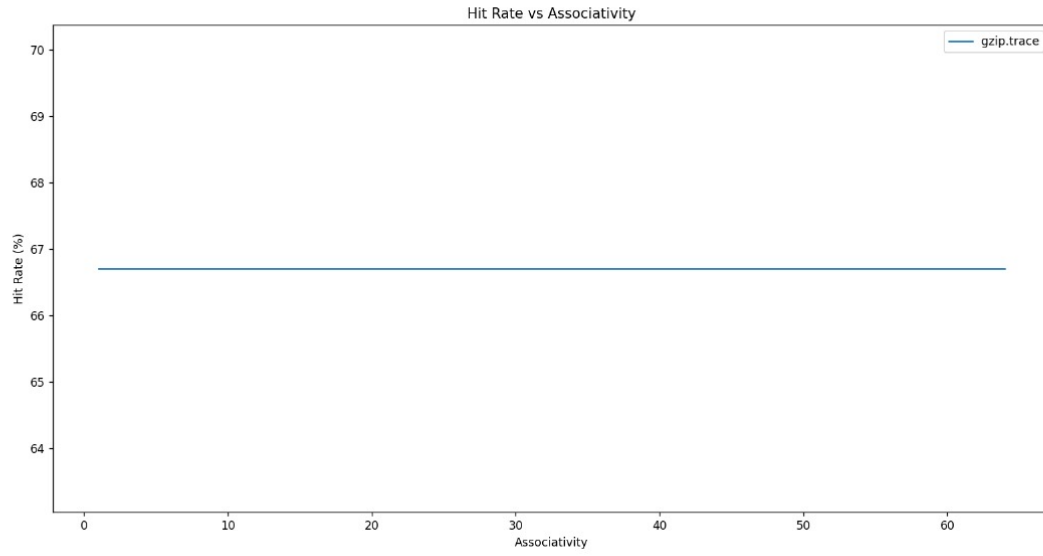
- The traces behave the same way and the miss rates decrease as we increase block size.
- Thus by increasing the block size we are taking advantage of spatial locality (which means consecutive bytes need not be fetched again)
- From the table we infer that hit rates increase for a particular trace file very slightly

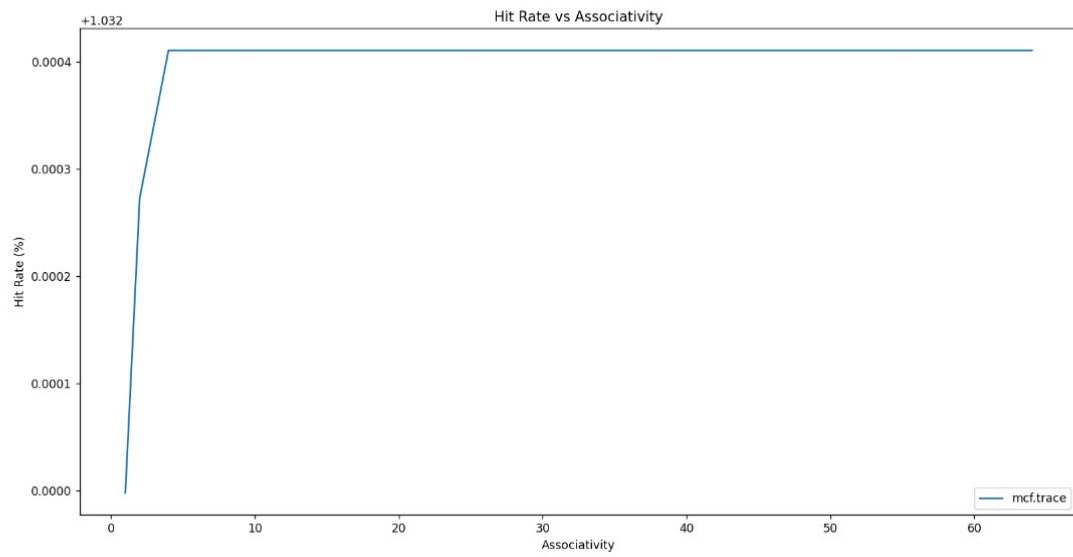
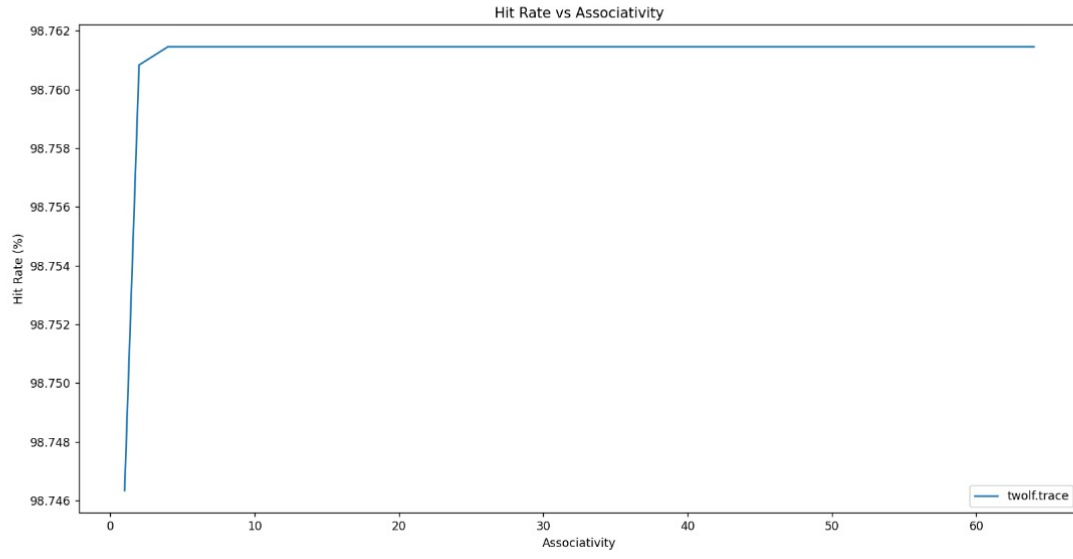


#### 2.4 d) set-associative Cache with varying number of ways :

- To analyze the effect of cache associativity on the performance of the cache system, we fixed the cache size at 1024kB and systematically varied the associativity levels from 1-way (direct-mapped) to 64-way associative.







Read the instructions below:

For the hit and miss rates of each trace file, enter: a

For results on varying cache size, enter: b

For results on varying block size, enter: c

For results on varying associativity, enter: d

Input= d

Results for Part D:

Trace	Config	Hit Rate	Miss Rate
gcc.trace	Associativity: 1	93.8305	6.16949
gcc.trace	Associativity: 2	93.8348	6.16522
gcc.trace	Associativity: 4	93.8355	6.16445
gcc.trace	Associativity: 8	93.8355	6.16445
gcc.trace	Associativity: 16	93.8358	6.16425
gcc.trace	Associativity: 32	93.8359	6.16406
gcc.trace	Associativity: 64	93.8359	6.16406
gzip.trace	Associativity: 1	66.7055	33.2945
gzip.trace	Associativity: 2	66.7055	33.2945
gzip.trace	Associativity: 4	66.7055	33.2945
gzip.trace	Associativity: 8	66.7055	33.2945
gzip.trace	Associativity: 16	66.7055	33.2945
gzip.trace	Associativity: 32	66.7055	33.2945
gzip.trace	Associativity: 64	66.7055	33.2945

swim.trace	Associativity: 1	92.6205	7.37946	
swim.trace	Associativity: 2	92.6225	7.37748	
swim.trace	Associativity: 4	92.6225	7.37748	
swim.trace	Associativity: 8	92.6225	7.37748	
swim.trace	Associativity: 16	92.6225	7.37748	
swim.trace	Associativity: 32	92.6225	7.37748	
swim.trace	Associativity: 64	92.6225	7.37748	
twolf.trace	Associativity: 1	98.7463	1.25367	
twolf.trace	Associativity: 2	98.7608	1.23917	
twolf.trace	Associativity: 4	98.7614	1.23855	
twolf.trace	Associativity: 8	98.7614	1.23855	
twolf.trace	Associativity: 16	98.7614	1.23855	
twolf.trace	Associativity: 32	98.7614	1.23855	
twolf.trace	Associativity: 64	98.7614	1.23855	



mcf.trace	Associativity: 1	1.032	98.968	
+-----+	+-----+	+-----+	+-----+	+-----+
mcf.trace	Associativity: 2	1.03227	98.9677	
+-----+	+-----+	+-----+	+-----+	+-----+
mcf.trace	Associativity: 4	1.03241	98.9676	
+-----+	+-----+	+-----+	+-----+	+-----+
mcf.trace	Associativity: 8	1.03241	98.9676	
+-----+	+-----+	+-----+	+-----+	+-----+
mcf.trace	Associativity: 16	1.03241	98.9676	
+-----+	+-----+	+-----+	+-----+	+-----+
mcf.trace	Associativity: 32	1.03241	98.9676	
+-----+	+-----+	+-----+	+-----+	+-----+
mcf.trace	Associativity: 64	1.03241	98.9676	
+-----+	+-----+	+-----+	+-----+	+-----+
PS D:\sem3\ca_assgn>				

#### Observations:

- gzip.trace input file graph obtained is different from other four graphs.
- Generally hit rates increases while increasing associativity but it depends on different input files.
- Thus the file gzip.trace is behaving differently due to its memory access patterns that does not cause many conflicts between indexes.