

2021-Eg 211-Computer Architecture

Assignment 3

Submitted By Vihan Vashishth(IMT2021065), Vineet Priyadarshi(IMT2021018)

And Prasad Jore(IMT2021104)

Instructions to run the Python program files:

- Make sure you are in the directory which contains the text files and the codes.

IMT2021018_IMT2021065_IMT2021108<file_name>.py

&

<file_name>.trace

- Execute the following command in terminal:

```
python3 IMT2021018_IMT2021065_IMT2021108<file_name>.py
```

Cache Implementation

- Index bit size = (cache memory/block size)/number of ways
- Block offset = $\log_2(\text{block_size})$
- Tag bit size = 32 - (Index bit size) - (Block offset)

- We simulate a cache memory by defining a dictionary which stores 'index bits' as key and value as a list of list which contains the 'valid bit' and the 'tag bits'. We define two counters 'hit' and 'miss' to keep track of the hits and misses.
- If the index bits exists in the cache memory, we check for the tag bits. If the tag bit matches and the 'valid bit' equals 1, we report a hit. We increment the hit by 1.
- If the index bit exists but the tag bit doesn't match we report a miss. Even if the tag matches but the 'valid bit' is zero we report a miss and increment the miss counter by 1.
- But if the index itself doesn't match, we add the index as a new key in our dictionary and set an empty 2d list as it's value.

```
cache[b[15:30]].append([1,b[0:15]])  
else:                                     #if the index match fails we report a miss  
    miss=miss+1  
    cache[b[15:30]]=[[0,0000000000000000],[0,0000000000000000],[0,0000000000000000]] #here the first element of list o  
    cache[b[15:30]].append([1,b[0:15]])
```

- After each hit or miss, we implement our cache replacement policy. We have used LRU in our implementation.
- The 2d list which is the value to our keys of our dictionary has elements equal to number of ways in our cache.

LRU

- If it is a hit, we pop the hit element from our list of (valid, tag)'s which has index bit as key, and append it to the end of the list.

```
if b[15:30] in cache: # index match
    if [1,b[0:15]] in cache[b[15:30]]: # the index maps to a list of 4 terms consisting of (valid bit, tag bit,
        hit=hit+1 #a hit is reported
        #LRU
        cache[b[15:30]].pop(cache[b[15:30]].index([1,b[0:15]])) #the working of our LRU is explained in the
        cache[b[15:30]].append([1,b[0:15]])
```

- If it is a miss, we pop the first element which is the least used element and then append the miss element (valid, tag) at the end of the list.

```
else:
    miss=miss+1
    cache[b[15:30]].pop(0)
    cache[b[15:30]].append([1,b[0:15]])
```

This way the recently used element always remains at the end of the list of the respective index bit key.

Q1 a)

- 4-way Set associative cache with a cache size of 512kb and a block size of 4 bytes
- Number of lines used 32768 which is equal to 2^{15}
- This 4-way set associative cache is using LRU
- Byte offset is of 2 bits and tag size is 15 bits too.

```
Vihans-MacBook-Air:Assignment 3 Work$ python3 IMT2021065_IMT2021105_IMT2021018_1a.py
File to read: gcc.trace
```

```
-----
hits = 483893
misses = 31790
hit% = 93.83536009525231
miss% = 6.164639904747688
hit/miss = 15.221547656495753
```

```
File to read: gzip.trace
```

```
-----
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584
```

```
File to read: mcf.trace
```

```
-----
hits = 7508
misses = 719722
hit% = 1.03241065412593
miss% = 98.96758934587407
hit/miss = 0.01043180561383423
```

```
File to read: swim.trace
```

```
-----
hits = 280825
misses = 22368
hit% = 92.62252096849201
miss% = 7.377479031507988
hit/miss = 12.554765736766809
```

```
File to read: twolf.trace
```

```
-----
hits = 476844
misses = 5980
hit% = 98.76145344887578
miss% = 1.238546551124216
hit/miss = 79.73979933110368
```

Output table:

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	545683	483893	31790	93.83%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280825	22368	92.62%
5	twolf	482824	476844	5980	98.76%

Q1 b)

- 4-way Set associative cache with a cache size of 2048kb and a block size of 4 bytes
- Number of lines used 131072 which is equal to 2^{17}
- Tag size is 13 bits
- block offset is 2bits
- This 4-way set associative cache is using LRU

```

Vihans-MacBook-Air:Assignment 3 work$ python3 IM12021065_IM12021105_IM12021018_1b.py
File to read: gcc.trace
-----
hits = 483894
misses = 31789
hit% = 93.83555401283347
miss% = 6.1644459871665305
hit/miss = 15.222057944572022

File to read: gzip.trace
-----
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584

File to read: mcf.trace
-----
hits = 7508
misses = 719722
hit% = 1.03241065412593
miss% = 98.96758934587407
hit/miss = 0.01043180561383423

File to read: swim.trace
-----
hits = 280825
misses = 22368
hit% = 92.62252096849201
miss% = 7.377479031507988
hit/miss = 12.554765736766809

File to read: twolf.trace
-----
hits = 476844
misses = 5980
hit% = 98.76145344887578
miss% = 1.238546551124216
hit/miss = 79.73979933110368

```

Output table:

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	545683	483894	31790	93.83%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280825	22368	92.62%
5	twolf	482824	476844	5980	98.76%

Observations:

- The hit/miss ratio is mostly same except for the gcc.trace file, an extra hit is reported in the b part of the question compared to the a part.

Q1 c)

For one byte block size:

- 4-way Set associative cache with a cache size of 512kb and a block size of 1 bytes
- Number of lines used 131072 which is equal to 2^{17} . Therefore index bit is 17.
- Tag size is 15 bits
- block offset is 0 bits
- This 4-way set associative cache is using LRU

1 byte

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	480610	35073	93.19%
2	gzip	481044	320875	160169	66.70%
3	mcf	727230	7451	719779	1.02%
4	swim	303193	280588	22605	92.54%
5	twolf	482824	475470	7354	98.47%

For four byte block size:

- 4-way Set associative cache with a cache size of 512kb and a block size of 4 bytes
- Number of lines used 32768 which is equal to 2^{15} . Therefore index bit is 15.
- Tag size is 15 bits
- block offset is 2 bits
- This 4-way set associative cache is using LRU

4 byte

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	483893	31790	93.83%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280825	22368	92.62%
5	twolf	482824	476844	5980	98.76%

For eight byte block size:

- 4-way Set associative cache with a cache size of 512kb and a block size of 8 bytes
- Number of lines used 16384 which is equal to 2^{14} . Therefore index bit is 14.
- Tag size is 15 bits
- block offset is 3 bits
- This 4-way set associative cache is using LRU

8 byte

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	494676	21007	95.92%
2	gzip	481044	320891	160153	66.70%
3	mcf	727230	7551	719679	1.03%
4	swim	303193	283377	19816	93.46%
5	twolf	482824	477319	5505	98.85%

For sixteen byte block size:

- 4-way Set associative cache with a cache size of 512kb and a block size of 16 bytes
- Number of lines used 8192 which is equal to 2^{13}
- Tag size is 15 bits
- block offset is 4 bits
- This 4-way set associative cache is using LRU

16 byte

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	504465	11218	97.82%
2	gzip	481044	321268	159776	66.78%
3	mcf	727230	367273	359957	50.50%
4	swim	303193	291770	11423	96.23%
5	twolf	482824	479869	2955	99.38%

Screenshots for Q 1c):

```
Vihans-MacBook-Air:Assignment 3 Work$ python3 IM12021065_IM12021105_IM12021018_1c.py
File to read: gcc.trace
```

```
-----if block size is 1 byte:-----
hits = 480610
misses = 35073
hit% = 93.19872867633798
miss% = 6.8012713236620215
hit/miss = 13.703133464488353
```

```
-----if block size is 4 byte:-----
hits = 483893
misses = 31790
hit% = 93.83536009525231
miss% = 6.164639904747688
hit/miss = 15.221547656495753
```

```
-----if block size is 8 byte:-----
hits = 494676
misses = 21007
hit% = 95.9263733727891
miss% = 4.073626627210899
hit/miss = 23.54815061646118
```

```
-----if block size is 16byte:-----
hits = 504465
misses = 11218
hit% = 97.82463257466311
miss% = 2.175367425336887
hit/miss = 44.96924585487609
```

```
File to read: gzip.trace
```

```
-----if block size is 1 byte:-----
hits = 320875
misses = 160169
hit% = 66.70387739998836
miss% = 33.29612260001164
hit/miss = 2.003352708701434
```

```
-----if block size is 4 byte:-----
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584
```

```
-----if block size is 8 byte:-----  
hits = 320891  
misses = 160153  
hit% = 66.70720349905622  
miss% = 33.29279650094378  
hit/miss = 2.00365275705107
```

```
-----if block size is 16byte:-----  
hits = 321268  
misses = 159776  
hit% = 66.7855747083427  
miss% = 33.21442529165731  
hit/miss = 2.010740036050471
```

```
File to read: mcf.trace
```

```
-----if block size is 1 byte:-----  
hits = 7451  
misses = 719779  
hit% = 1.0245726936457518  
miss% = 98.97542730635425  
hit/miss = 0.010351788535092022
```

```
-----if block size is 4 byte:-----  
hits = 7508  
misses = 719722  
hit% = 1.03241065412593  
miss% = 98.96758934587407  
hit/miss = 0.01043180561383423
```

```
-----if block size is 8 byte:-----  
hits = 7551  
misses = 719679  
hit% = 1.0383235015057135  
miss% = 98.96167649849428  
hit/miss = 0.01049217776258582
```

```
-----if block size is 16byte:-----  
hits = 367273  
misses = 359957  
hit% = 50.50300455151741  
miss% = 49.49699544848259  
hit/miss = 1.0203246498887366
```

```

File to read: swim.trace
-----
----if block size is 1 byte:----
hits = 280588
misses = 22605
hit% = 92.54435293690817
miss% = 7.455647063091831
hit/miss = 12.41265206812652

----if block size is 4 byte:----
hits = 280825
misses = 22368
hit% = 92.62252096849201
miss% = 7.377479031507988
hit/miss = 12.554765736766809

----if block size is 8 byte:----
hits = 283377
misses = 19816
hit% = 93.46422905542015
miss% = 6.535770944579852
hit/miss = 14.300413807024627

----if block size is 16byte:----
hits = 291770
misses = 11423
hit% = 96.23243280682601
miss% = 3.7675671931739885
hit/miss = 25.542326884356125

File to read: twolf.trace
-----
----if block size is 1 byte:----
hits = 475470
misses = 7354
hit% = 98.47687770284824
miss% = 1.5231222971517582
hit/miss = 64.65460973619798

----if block size is 4 byte:----
hits = 476844
misses = 5980
hit% = 98.76145344887578
miss% = 1.238546551124216
hit/miss = 79.73979933110368

----if block size is 8 byte:----
hits = 477319
misses = 5505
hit% = 98.85983298261893
miss% = 1.1401670173810743
hit/miss = 86.70644868301544

----if block size is 16byte:----
hits = 479869
misses = 2955
hit% = 99.38797574271369
miss% = 0.6120242572863077
hit/miss = 162.3922165820643

```

Observations:

- As cache size remains the same, hit rate increases on increasing the block size for every file. This is the result of taking advantage of the spatial locality.
- The file 'mcf.trace' had an huge improvement in it's hit rate (1% → 50%) on increasing the block size to 16 bytes.

Q1 d)

For 1-way set associative cache:

- 1-way Set associative cache with a cache size of 512kb and a block size of 4 bytes
- Number of lines used 131072 which is equal to 2^{17} . Therefore index bit is 17.
- Tag size is 13 bits
- block offset is 2 bits
- This 4-way set associative cache is using LRU

1 way SA

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	483894	31789	93.83%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280825	22368	92.62%
5	twolf	482824	476844	5980	98.76%

For 2-way set associative cache:

- 2-way Set associative cache with a cache size of 512kb and a block size of 4 bytes
- Number of lines used 65536 which is equal to 2^{16} . Therefore index bit is 16.
- Tag size is 14 bits
- block offset is 2 bits
- This 4-way set associative cache is using LRU

2 way SA

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	483894	31789	93.83%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280825	22368	92.62%
5	twolf	482824	476844	5980	98.76%

For 4-way set associative cache:

- 4-way Set associative cache with a cache size of 512kb and a block size of 4 bytes
- Number of lines used 32768 which is equal to 2^{15} . Therefore index bit is 15.
- Tag size is 15 bits
- block offset is 2 bits
- This 4-way set associative cache is using LRU

4 way SA

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	483893	31790	93.83%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280825	22368	92.62%
5	twolf	482824	476844	5980	98.76%

For 8-way set associative cache:

- 8-way Set associative cache with a cache size of 512kb and a block size of 4 bytes
- Number of lines used 16384 which is equal to 2^{14} . Therefore index bit is 14.
- Tag size is 16 bits
- block offset is 2 bits
- This 4-way set associative cache is using LRU

8 way SA

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	483871	31812	93.83%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280825	22368	92.62%
5	twolf	482824	476844	5980	98.76%

For 16-way set associative cache:

- 16-way Set associative cache with a cache size of 512kb and a block size of 4 bytes
- Number of lines used 8192 which is equal to 2^{13} . Therefore index bit is 13.
- Tag size is 17 bits
- block offset is 2 bits
- This 4-way set associative cache is using LRU

**16 way
SA**

Sr no	TRACE	INSTRUCTIONS	HITS	MISSES	HIT RATE
1	gcc	515683	483719	31964	93.80%
2	gzip	481044	320883	160161	66.70%
3	mcf	727230	7508	719722	1.03%
4	swim	303193	280817	22376	92.61%
5	twolf	482824	476843	5981	98.76%

Screenshots for Q 1d):

```
Vihans-MacBook-Air:Assignment 3 Work$ python3 IMT2021065_IMT2021105_IMT2021018_1d.py
File to read: gcc.trace
```

```
-----
---1-way Set Associative:---
hits = 483844
misses = 31839
hit% = 93.82585813377598
miss% = 6.174141866224019
hit/miss = 15.196582807248971
```

```
---2-way Set Associative:---
hits = 483878
misses = 31805
hit% = 93.83245133153507
miss% = 6.167548668464931
hit/miss = 15.213897185977048
```

```
---4-way Set Associative:---
hits = 483893
misses = 31790
hit% = 93.83536009525231
miss% = 6.164639904747688
hit/miss = 15.221547656495753
```

```
---8-way Set Associative:---
hits = 483894
misses = 31789
hit% = 93.83555401283347
miss% = 6.1644459871665305
hit/miss = 15.222057944572022
```

```
---16-way Set Associative:---
hits = 483894
misses = 31789
hit% = 93.83555401283347
miss% = 6.1644459871665305
hit/miss = 15.222057944572022
```

```
File to read: gzip.trace
```

```
-----
---1-way Set Associative:---
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584
```

```
---2-way Set Associative:---
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584
```

```

---4-way Set Associative:---
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584

---8-way Set Associative:---
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584

---16-way Set Associative:---
hits = 320883
misses = 160161
hit% = 66.70554044952229
miss% = 33.29445955047771
hit/miss = 2.003502725382584

File to read: mcf.trace
-----

---1-way Set Associative:---
hits = 7505
misses = 719725
hit% = 1.031998129890131
miss% = 98.96800187010987
hit/miss = 0.010427593872659697

---2-way Set Associative:---
hits = 7507
misses = 719723
hit% = 1.0322731460473304
miss% = 98.96772685395267
hit/miss = 0.010430401696208125

---4-way Set Associative:---
hits = 7508
misses = 719722
hit% = 1.03241065412593
miss% = 98.96758934587407
hit/miss = 0.01043180561383423

---8-way Set Associative:---
hits = 7508
misses = 719722
hit% = 1.03241065412593
miss% = 98.96758934587407
hit/miss = 0.01043180561383423

---16-way Set Associative:---
hits = 7508
misses = 719722
hit% = 1.03241065412593
miss% = 98.96758934587407
hit/miss = 0.01043180561383423

```

```

---8-way Set Associative:---
hits = 476844
misses = 5980
hit% = 98.76145344887578
miss% = 1.238546551124216
hit/miss = 79.73979933110368

```

```

---16-way Set Associative:---
hits = 476844
misses = 5980
hit% = 98.76145344887578
miss% = 1.238546551124216
hit/miss = 79.73979933110368

```

```
Vihans-MacBook-Air:Assignment 3 Work$
```

File to read: swim.trace

---1-way Set Associative:---

hits = 280738
misses = 22455
hit% = 92.59382637461947
miss% = 7.406173625380532
hit/miss = 12.502248942329103

---2-way Set Associative:---

hits = 280825
misses = 22368
hit% = 92.62252096849201
miss% = 7.377479031507988
hit/miss = 12.554765736766809

---4-way Set Associative:---

hits = 280825
misses = 22368
hit% = 92.62252096849201
miss% = 7.377479031507988
hit/miss = 12.554765736766809

---8-way Set Associative:---

hits = 280825
misses = 22368
hit% = 92.62252096849201
miss% = 7.377479031507988
hit/miss = 12.554765736766809

---16-way Set Associative:---

hits = 280825
misses = 22368
hit% = 92.62252096849201
miss% = 7.377479031507988
hit/miss = 12.554765736766809

File to read: twolf.trace

---1-way Set Associative:---

hits = 476771
misses = 6053
hit% = 98.74633406790052
miss% = 1.25366593209948
hit/miss = 78.76606641334875

---2-way Set Associative:---

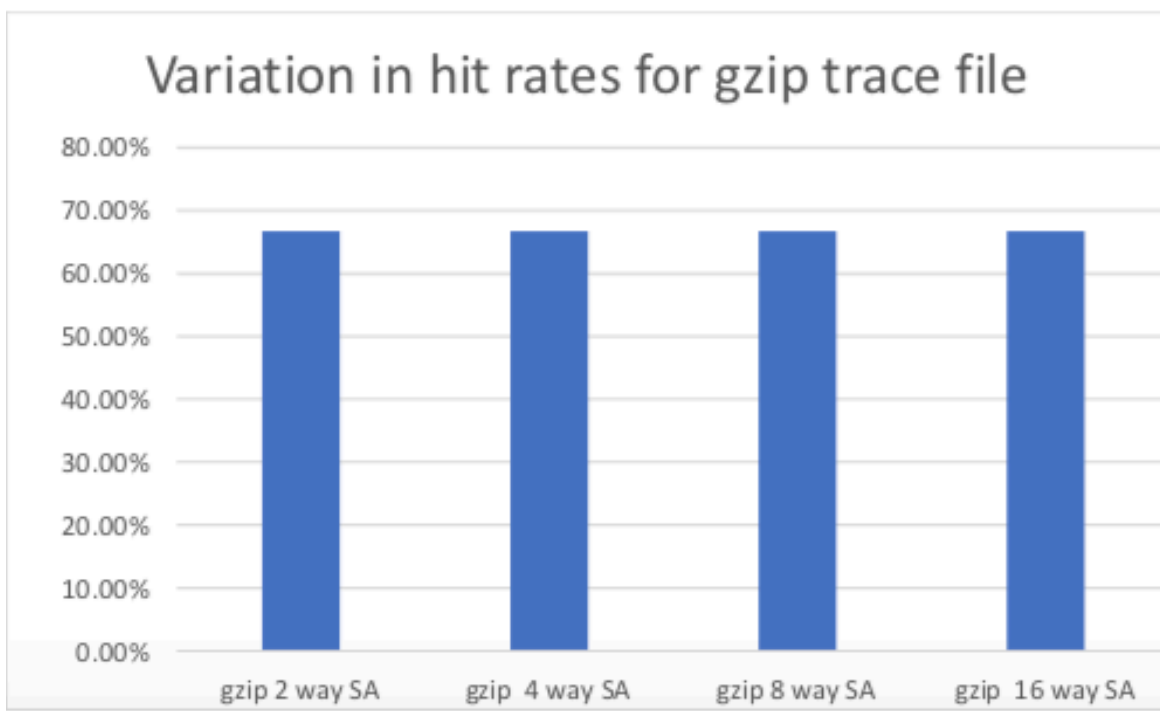
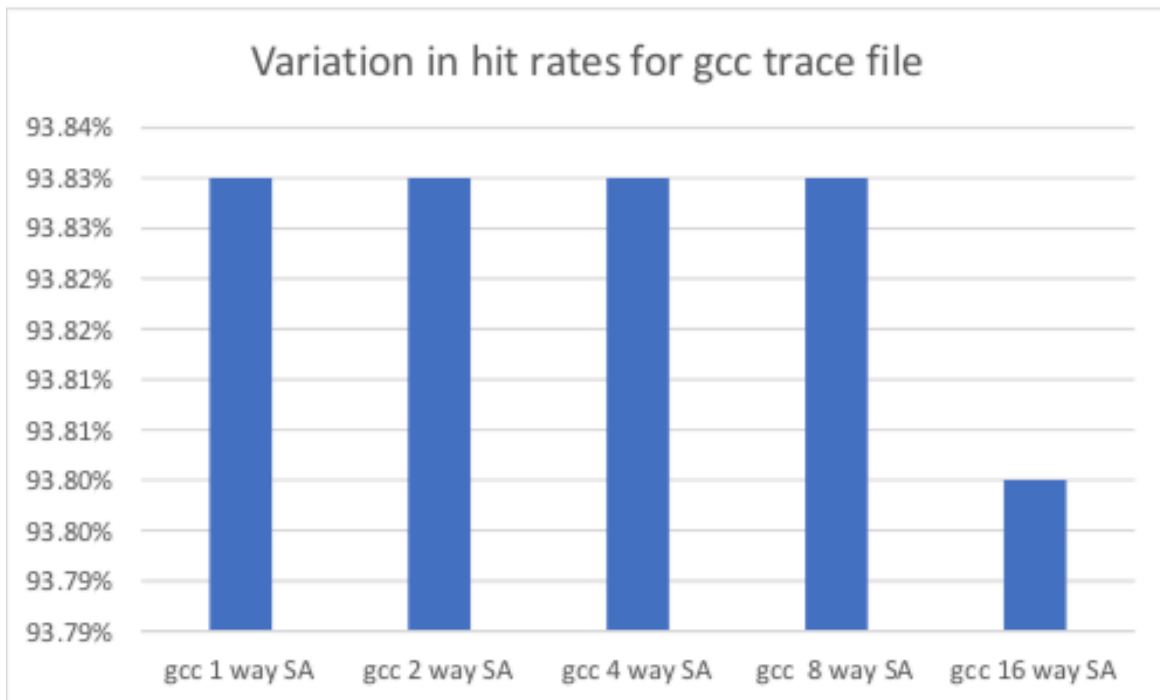
hits = 476841
misses = 5983
hit% = 98.76083210445215
miss% = 1.2391678955478511
hit/miss = 79.69931472505432

---4-way Set Associative:---

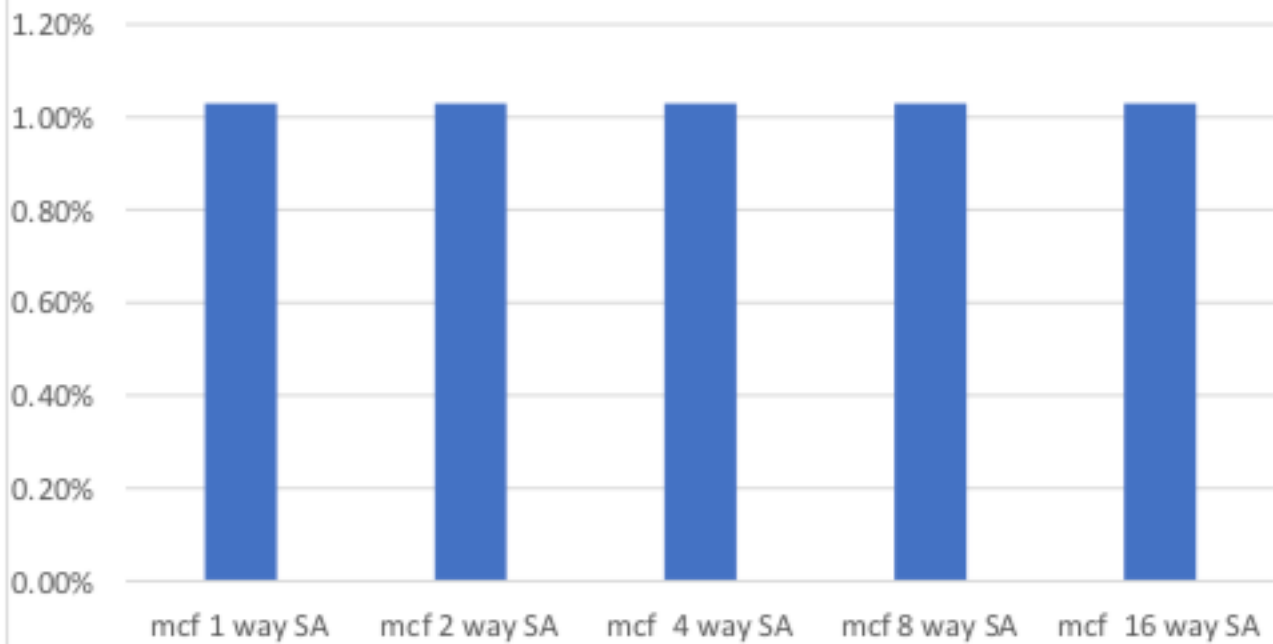
hits = 476844
misses = 5980
hit% = 98.76145344887578
miss% = 1.238546551124216
hit/miss = 79.73979933110368

Graph showing hit rate variations:

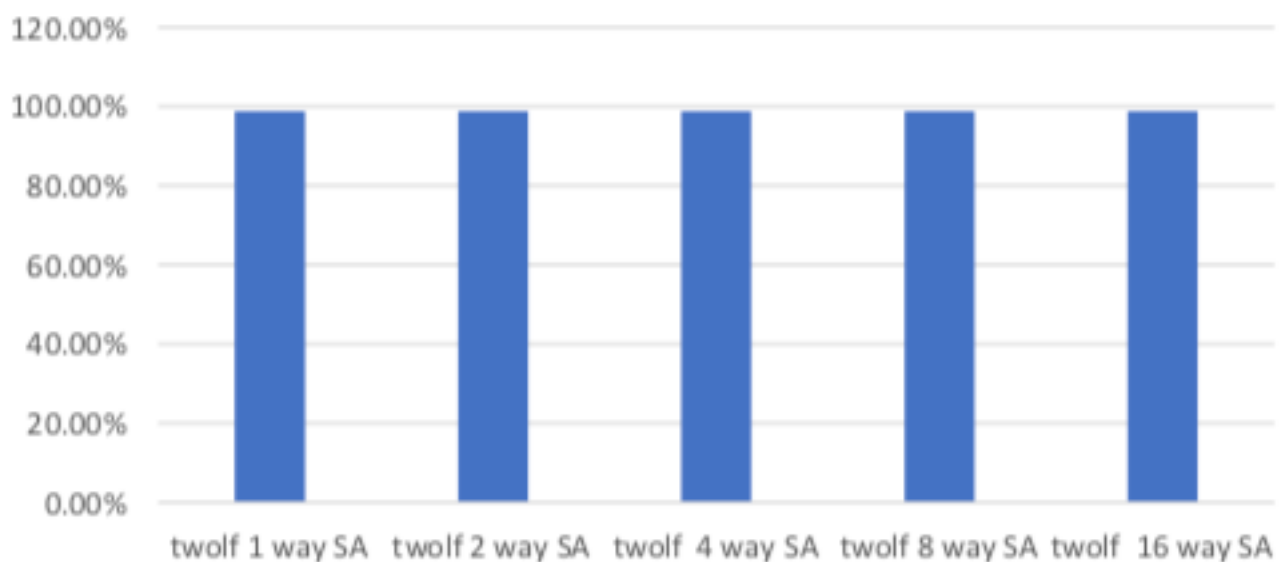
- Hit rates are along the y axis and the associativity is along x axis

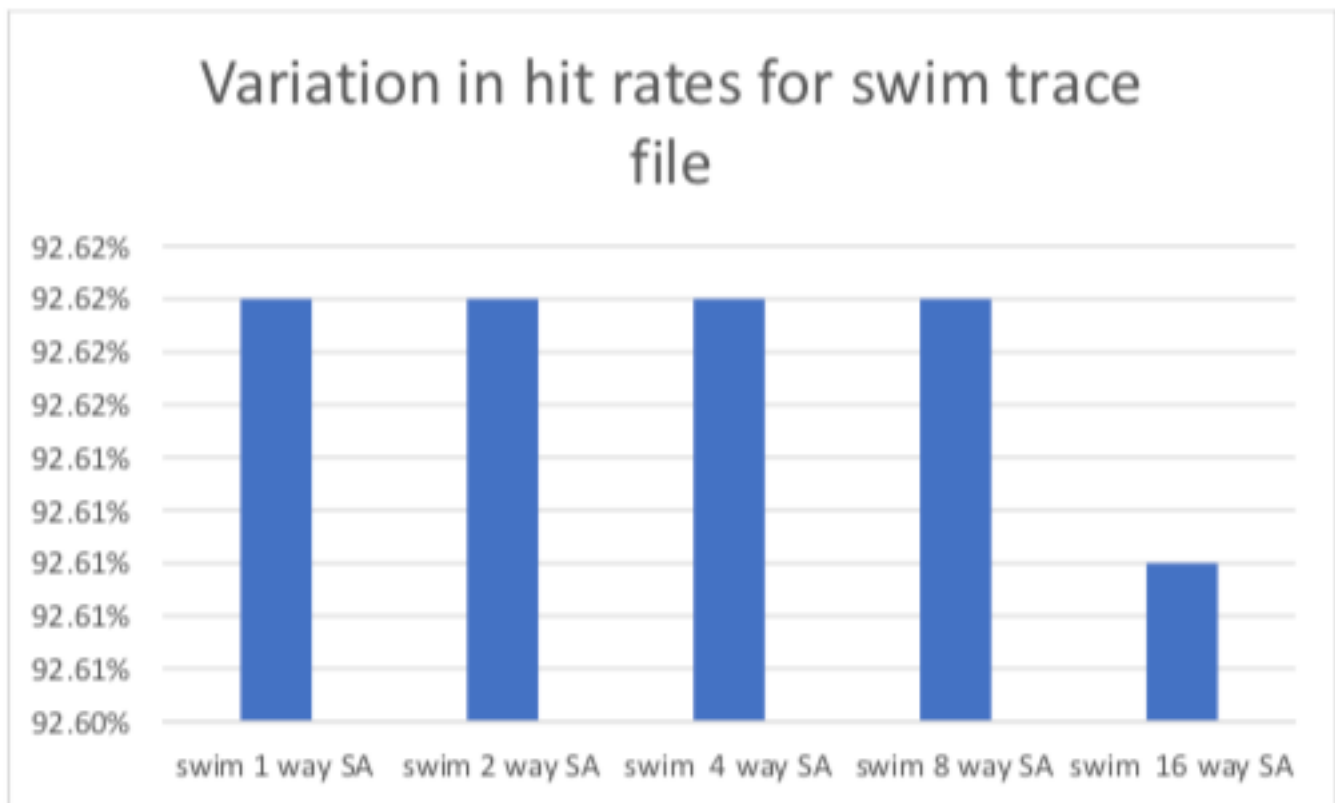


Variation in hit rates for mcf trace file



Variation in hit rates for twolf trace file





Observations:

- Hit rate increases or remains the same as we increase the associativity from 1 way to 16 way.