

## Project 2: Cache Performance

### Test Cases

To be able to test the simulator and keep track of all the data stored in it, we reduced its size and chose a smaller number on ways and a small number of lines. We made different combinations between different number of ways and different number of lines to test all the cases and all the possible types of misses: Compulsory (cold start), Capacity, and conflict. We also simulated the other types of caches (direct map cache and full associative cache) using this simulator.

#### **Test 1:**

##### **1 way set associative cache:**

Cache size: 512 bytes

Number of Ways: 1 -> Direct map cache

Cache block/line size: 16 bytes

Number of lines: 32

<b>Addresses</b>	<b>Tag</b>	<b>Index</b>	<b>H/M</b>	<b>Miss Type</b>
0010 1101 1011 0011 <sub>2</sub> 11699 <sub>10</sub>	0010110	11011 <sub>2</sub>	M	Compulsory
	0x16	0x1B		
0000 0110 1111 1100 <sub>2</sub> 1788 <sub>10</sub>	0000011 <sub>2</sub>	01111 <sub>2</sub>	M	Compulsory
	0x3	0xF		
0010 1101 1011 1000 <sub>2</sub> 11704 <sub>10</sub>	0010110	11011 <sub>2</sub>	H	-
	0x16	0x1B		
1010 1010 1010 1011 <sub>2</sub> 43691 <sub>10</sub>	1010101 <sub>2</sub>	01010 <sub>2</sub>	M	Compulsory
	0x55	0xA		

0000 0111 0000 0001 <sub>2</sub>	0000011 <sub>2</sub>	10000 <sub>2</sub>	M	Compulsory
1793 <sub>10</sub>	0x3	0x10		
1110 1101 1011 1000 <sub>2</sub>	1110110 <sub>2</sub>	11011 <sub>2</sub>	M	Conflict
60856 <sub>10</sub>	0x76	0x1B		
1110 1101 1011 1111 <sub>2</sub>	1110110 <sub>2</sub>	11011 <sub>2</sub>	H	-
60863 <sub>10</sub>	0x76	0x1B		
1010 1010 1010 1111 <sub>2</sub>	1010101 <sub>2</sub>	01010 <sub>2</sub>	H	-
43695 <sub>10</sub>	0x55	0xA		
1111 1010 1010 1111 <sub>2</sub>	1111101 <sub>2</sub>	01010 <sub>2</sub>	M	Conflict
64175 <sub>10</sub>	0x7D	0xA		
1111 1010 1010 1101 <sub>2</sub>	1111101 <sub>2</sub>	01010 <sub>2</sub>	H	-
64173 <sub>10</sub>	0x7D	0xA		
0000 0110 1111 1101 <sub>2</sub>	0000011 <sub>2</sub>	01111 <sub>2</sub>	H	-
1789 <sub>10</sub>	0x3	0xF		

Hit ratio: 45.45%

Miss ratio: 54.55%

The cache:

Index	Valid bit	Tag
0xA	1	0x7D
0xF	1	0x3
0x10	1	0x3

The rest indices of the cache are all empty and have 0 for the valid bit.

```
ranaelgahawy@Ranas-MacBook-Pro CACHE % ./output
Set associative Cache Simulator Testing
Enter addresses to test the cache
11699
0x00002db3 (Miss)
1788
0x000006fc (Miss)
11704
0x00002db8 (Hit)
43691
0x0000aaab (Miss)
1793
0x00000701 (Miss)
60856
0x0000edb8 (Miss)
60863
0x0000edbf (Hit)
43695
0x0000aaaf (Hit)
64175
0x0000faaf (Miss)
64173
0x0000faad (Hit)
1789
0x000006fd (Hit)

The Cache:
Index: 0      valid bit: 0      tag: 0x0
Index: 1      valid bit: 0      tag: 0x0
Index: 2      valid bit: 0      tag: 0x0
Index: 3      valid bit: 0      tag: 0x0
Index: 4      valid bit: 0      tag: 0x0
Index: 5      valid bit: 0      tag: 0x0
Index: 6      valid bit: 0      tag: 0x0
Index: 7      valid bit: 0      tag: 0x0
Index: 8      valid bit: 0      tag: 0x0
Index: 9      valid bit: 0      tag: 0x0
Index: a      valid bit: 1      tag: 0x7d
Index: b      valid bit: 0      tag: 0x0
Index: c      valid bit: 0      tag: 0x0
Index: d      valid bit: 0      tag: 0x0
Index: e      valid bit: 0      tag: 0x0
Index: f      valid bit: 1      tag: 0x3
Index: 10     valid bit: 1      tag: 0x3
Index: 11     valid bit: 0      tag: 0x0
Index: 12     valid bit: 0      tag: 0x0
Index: 13     valid bit: 0      tag: 0x0
Index: 14     valid bit: 0      tag: 0x0
Index: 15     valid bit: 0      tag: 0x0
Index: 16     valid bit: 0      tag: 0x0
Index: 17     valid bit: 0      tag: 0x0
Index: 18     valid bit: 0      tag: 0x0
Index: 19     valid bit: 0      tag: 0x0
Index: 1a     valid bit: 0      tag: 0x0
Index: 1b     valid bit: 1      tag: 0x76
Index: 1c     valid bit: 0      tag: 0x0
Index: 1d     valid bit: 0      tag: 0x0
Index: 1e     valid bit: 0      tag: 0x0
Index: 1f     valid bit: 0      tag: 0x0
Hit ratio = 45.4545%
Miss ratio = 54.5455%
ranaelgahawy@Ranas-MacBook-Pro CACHE %
```

## Test 2:

### 2 way set associative cache:

Cache size: 64 bytes

Number of Ways: 2

Cache block/line size: 2 bytes

Number of sets: 16

Addresses	Tag	Way	Index	M/H	Miss Type
-----------	-----	-----	-------	-----	-----------

0011 1110 0011 1011 <sub>2</sub>	0011110001 <sub>2</sub> 0x1FA	1	1101 <sub>2</sub> 0xD	M	Compulsory
1111 1110 0011 1011 <sub>2</sub>	111 1111 0001 <sub>2</sub> 0x7F1	0	1101 <sub>2</sub> 0xD	M	Compulsory
0011 1111 1000 1000 <sub>2</sub>	001 1111 1100 <sub>2</sub> 0x1FC	1	0100 <sub>2</sub> 0x4	M	Compulsory
0011 1110 0011 1010 <sub>2</sub>	001 1111 0001 <sub>2</sub> 0x1FA	1	1101 <sub>2</sub> 0xD	H	-
1111 0010 0011 1011 <sub>2</sub>	111 1001 0001 <sub>2</sub> 0x791	0/1	1101 <sub>2</sub> 0xD	M	Conflict/Capacity
0011 1111 1000 1001 <sub>2</sub>	001 1111 1100 <sub>2</sub> 0x1FC	1	0100 <sub>2</sub> 0x4	H	-
1111 1111 1000 1000 <sub>2</sub>	111 1111 1100 <sub>2</sub> 0x7FC	0	0100 <sub>2</sub> 0x4	M	Compulsory
1111 0010 0011 1010 <sub>2</sub>	111 1001 0001 <sub>2</sub> 0x791	0/1	1101 <sub>2</sub> 0xD	H	-
1110 0010 1000 1101 <sub>2</sub>	111 0001 0100 <sub>2</sub> 0x714	1	0110 <sub>2</sub> 0x6	M	Compulsory
1110 0010 1000 1100 <sub>2</sub>	111 0001 0100 <sub>2</sub> 0x714	1	0110 <sub>2</sub> 0x6	H	-

Hit ratio: 40%

Miss ratio: 60%

The cache:

Index	Valid bit 0	Tag 1	Valid bit 1	Tag 2
<b>0x4</b>	1	0x7FC	1	0x1FC
<b>0x6</b>	0	0	1	0x714
<b>0xD</b>	1	0x791/0x1FA/0x7F1	1	0x791/0x1FA/0x7F1

In set 0xD the tag **0x791** must exist in one of the two blocks, and one of the other two addresses in the other block

The rest indices of the cache are all empty and have 0 for the valid bit.

```

Miss ratio = 70%
ranaelgahawy@Ranas-MacBook-Pro CACHE % ./output
Set associative Cache Simulator Testing
Enter addresses to test the cache
15931
0x00003e3b (Miss)
65083
0x0000fe3b (Miss)
16264
0x00003f88 (Miss)
15930
0x00003e3a (Hit)
62011
0x0000f23b (Miss)
16265
0x00003f89 (Hit)
65416
0x0000ff88 (Miss)
62010
0x0000f23a (Hit)
57997
0x0000e28d (Miss)
57996
0x0000e28c (Hit)

The Cache:

Index: 0      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: 1      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: 2      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: 3      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: 4      valid bit: 1      tag: 0x7fc     valid bit: 1      tag: 0x1fc
Index: 5      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: 6      valid bit: 0      tag: 0x0      valid bit: 1      tag: 0x714
Index: 7      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: 8      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: 9      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: a      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: b      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: c      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: d      valid bit: 1      tag: 0x7f1     valid bit: 1      tag: 0x791
Index: e      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Index: f      valid bit: 0      tag: 0x0      valid bit: 0      tag: 0x0
Hit ratio = 40%
Miss ratio = 60%
ranaelgahawy@Ranas-MacBook-Pro CACHE %

```

### Test 3:

#### 4 way set associative cache:

Cache size: 128 bytes

Number of Ways: 4

Cache block/line size: 8 bytes

Number of sets: 4

Addresses	Tag	W	H/M	Miss Type
1111 0000 1111 0000 <sub>2</sub>	111 1000 0111 <sub>2</sub> 0x787	3	M	Compulsory
61680 <sub>10</sub>				
1111 0000 1111 0110 <sub>2</sub>	111 1000 0111 <sub>2</sub> 0x787	3	H	-
61686 <sub>10</sub>				
1111 0000 1110 0110 <sub>2</sub>	111 1000 0111 <sub>2</sub> 0x787	3	M	Compulsory
61670 <sub>10</sub>				
1111 1110 1110 0110 <sub>2</sub>	111 1111 0111 <sub>2</sub> 0x7F7	2	M	Compulsory
65254 <sub>10</sub>				
1100 1110 1110 0110 <sub>2</sub>	110 0111 0111 <sub>2</sub> 0x677	1	M	Compulsory
52966 <sub>10</sub>				
0000 1110 1110 0110 <sub>2</sub>	000 0111 0111 <sub>2</sub> 0x77	0	M	Compulsory
3814 <sub>10</sub>				
1111 0000 1110 0010 <sub>2</sub>	111 1000 0111 <sub>2</sub> 0x787	3	H	-
61666 <sub>10</sub>				
1111 0000 1110 0011 <sub>2</sub>	111 1000 0111 <sub>2</sub>	3	H	-

$61671_{10}$	0x787			
$1100\ 1110\ 1110\ 0000_2$	$110\ 0111\ 0111_2$	1	H	-
$52960_{10}$	0x677			
$0000\ 0010\ 1110\ 0010_2$	$000\ 0001\ 0111_2$	0/1/2/3	M	Capacity/ Conflict
$738_{10}$	0x17			
$1110\ 0010\ 1110\ 0101_2$	$111\ 0001\ 0111_2$	3	M	Compulsory
$58090_{10}$	0x717			
$0000\ 1110\ 1111\ 1000_2$	$000\ 0111\ 0111_2$	3	M	Compulsory
$3832_{10}$	0x77			
$1100\ 0010\ 1110\ 0101_2$	$110\ 0001\ 0111_2$	2	M	Compulsory
$49899_{10}$	0x617			
$0001\ 1010\ 1011\ 1111_2$	$000\ 1101\ 0101_2$	2	M	Compulsory
$6847_{10}$	0xD5			
$1110\ 0010\ 1110\ 0101_2$	$111\ 0001\ 0111_2$	3	H	-
$58091_{10}$	0x717			
$0001\ 1010\ 1011\ 1001_2$	$000\ 1101\ 0101_2$	2	H	-
$6841_{10}$	0xD5			
$1100\ 0010\ 1110\ 0101_2$	$110\ 0001\ 0111_2$	2	H	-
$49899_{10}$	0x617			
$0000\ 0010\ 1110\ 0010_2$	$000\ 0001\ 0111_2$	0/1/2/3	H	-
$738_{10}$	0x17			
$0000\ 0010\ 1110\ 0011_2$	$000\ 0001\ 0111_2$	0/1/2/3	H	-

$743_{10}$	0x17			
0000 1110 111 <b>1 1011</b> <sub>2</sub>	000 0111 0111 <sub>2</sub>	3	H	-
$3835_{10}$	0x77			

Hit ratio: 50%

Miss ratio: 50%

The cache:

Index	V0	Tag 0	V1	Tag 1	V2	Tag 2	V3	Tag 3
<b>0x0</b>	1	0x17/0x677/0x787/0x7F	1	0x17/0x677/0x787/0x7F	1	0x17/0x677/0x787/0x7F	1	0x17/0x677/0x787/0x7F
<b>0x1</b>	0	0	0	0	1	0x617	1	0x717
<b>0x2</b>	0	0	0	0	0	0	1	0x787
<b>0x3</b>	0	0	0	0	1	0xD5	1	0x77

In set 0x0 the tag **0x17** must exist in one of the 4 blocks, and three of the other four addresses in the other three blocks.

```
ranaelgahawy@Ranas-MacBook-Pro CACHE % ./output
Set associative Cache Simulator Testing
Enter addresses to test the cache
61680
0x0000f0f0 (Miss)
61686
0x0000f0f6 (Hit)
61670
0x0000f0e6 (Miss)
65254
0x0000fee6 (Miss)
52966
0x0000cee6 (Miss)
3814
0x00000ee6 (Miss)
6166
0x0000f0e2 (Hit)
61671
0x0000f0e7 (Hit)
52960
0x0000cee0 (Hit)
738
0x000002e2 (Miss)
58090
0x0000e2ea (Miss)
3832
0x00000ef8 (Miss)
49899
0x0000c2eb (Miss)
6847
0x00001abf (Miss)
58091
0x0000e2eb (Hit)
6841
0x00001ab9 (Hit)
49899
0x0000c2eb (Hit)
738
0x000002e2 (Hit)
743
0x00000027 (Hit)
3835
0x000000efb (Hit)

The Cache:
Index: 0      valid bit: 1    tag: 0x17            valid bit: 1    tag: 0x677      valid bit: 1    tag: 0x7f7      valid bit: 1    tag: 0x787
Index: 1      valid bit: 0    tag: 0x0             valid bit: 0    tag: 0x0         valid bit: 1    tag: 0x617      valid bit: 1    tag: 0x717
Index: 2      valid bit: 0    tag: 0x0             valid bit: 0    tag: 0x0         valid bit: 0    tag: 0x0         valid bit: 1    tag: 0x787
Index: 3      valid bit: 0    tag: 0x0             valid bit: 0    tag: 0x0         valid bit: 1    tag: 0xd5       valid bit: 1    tag: 0x77

Hit ratio = 50%
Miss ratio = 50%
ranaelgahawy@Ranas-MacBook-Pro CACHE %
```

#### Test 4:

We made another experiment to test the effect of the number of ways on the hit ratio if we fixed the cache size and the addresses references (an array from 0 to 256), but we used a small cache size to be able to keep track of the cache. We called the function of the simulator of this array two times to test the spatial and temporal locality.

```
int testing [256];

for (int i = 0; i < 256; i++)
{
    testing[i] = i;
```

First:

Cache size: 128 bytes

Number of Ways: 1

Cache block/line size: 16 bytes

The miss ratio was 93.75%

#### The Cache:

```
Index: 0      valid bit: 1      tag: 0x1
Index: 1      valid bit: 1      tag: 0x1
Index: 2      valid bit: 1      tag: 0x1
Index: 3      valid bit: 1      tag: 0x1
Index: 4      valid bit: 1      tag: 0x1
Index: 5      valid bit: 1      tag: 0x1
Index: 6      valid bit: 1      tag: 0x1
Index: 7      valid bit: 1      tag: 0x1
Hit ratio = 93.75%
Miss ratio = 6.25%
```

Second:

Cache size: 128 bytes

Number of Ways: 8

Cache block/line size: 16 bytes

Miss Ratio: 95.1172%

But when increasing the number of ways, the hit ratio increased by 1.3672 which but seem a small increase but compared to the number of accesses and the cache size it is quite large if we used a larger amount of data and a bigger cache size the change would be more significant.

```
The Cache:  
Index: 0   valid bit: 1   tag: 0x15      valid bit: 1   tag: 0x6      valid bit: 1   tag: 0x5      valid bit: 1   tag: 0x4      valid bit: 1   tag: 0x3  
valid bit: 1   tag: 0x2      valid bit: 1   tag: 0x1      valid bit: 1   tag: 0x0  
Hit ratio = 95.1172%  
Miss ratio = 4.88281%
```

### Test 5:

We made another experiment to test the effect of the line size on the hit ratio if we fixed the cache size and the addresses references (an array from 0 to 256), but we used a small cache size to be able to keep track of the cache as above. We called the function of the simulator of this array two times to test the spatial and temporal locality.

```
int testing [256];  
  
for (int i = 0; i < 256; i++)  
{  
    testing[i] = i;  
}
```

First:

Cache size: 128 bytes

Number of Ways: 1

Cache block/line size: 4 bytes

The miss ratio was 75%

The Cache:

Index: 0	valid bit: 1	tag: 0x1
Index: 1	valid bit: 1	tag: 0x1
Index: 2	valid bit: 1	tag: 0x1
Index: 3	valid bit: 1	tag: 0x1
Index: 4	valid bit: 1	tag: 0x1
Index: 5	valid bit: 1	tag: 0x1
Index: 6	valid bit: 1	tag: 0x1
Index: 7	valid bit: 1	tag: 0x1
Index: 8	valid bit: 1	tag: 0x1
Index: 9	valid bit: 1	tag: 0x1
Index: 10	valid bit: 1	tag: 0x1
Index: 11	valid bit: 1	tag: 0x1
Index: 12	valid bit: 1	tag: 0x1
Index: 13	valid bit: 1	tag: 0x1
Index: 14	valid bit: 1	tag: 0x1
Index: 15	valid bit: 1	tag: 0x1
Index: 16	valid bit: 1	tag: 0x1
Index: 17	valid bit: 1	tag: 0x1
Index: 18	valid bit: 1	tag: 0x1
Index: 19	valid bit: 1	tag: 0x1
Index: 20	valid bit: 1	tag: 0x1
Index: 21	valid bit: 1	tag: 0x1
Index: 22	valid bit: 1	tag: 0x1
Index: 23	valid bit: 1	tag: 0x1
Index: 24	valid bit: 1	tag: 0x1
Index: 25	valid bit: 1	tag: 0x1
Index: 26	valid bit: 1	tag: 0x1
Index: 27	valid bit: 1	tag: 0x1
Index: 28	valid bit: 1	tag: 0x1
Index: 29	valid bit: 1	tag: 0x1
Index: 30	valid bit: 1	tag: 0x1
Index: 31	valid bit: 1	tag: 0x1

Hit ratio = 75%

Miss ratio = 25%

Second:

Cache size: 128 bytes

Number of Ways: 1

Cache block/line size: 16 bytes

Miss Ratio: 93.75%

But when increasing the line size, the hit ratio increased by 18.75 % which is a significant increase and if we used a larger amount of data and a bigger cache size the change would be even more significant.

```
The Cache:  
Index: 0      valid bit: 1      tag: 0x1  
Index: 1      valid bit: 1      tag: 0x1  
Index: 2      valid bit: 1      tag: 0x1  
Index: 3      valid bit: 1      tag: 0x1  
Index: 4      valid bit: 1      tag: 0x1  
Index: 5      valid bit: 1      tag: 0x1  
Index: 6      valid bit: 1      tag: 0x1  
Index: 7      valid bit: 1      tag: 0x1  
Hit ratio = 93.75%  
Miss ratio = 6.25%
```