Assignment 1

Part 1

Problem 1

1. Block size = 4 Byte
   Line = 8 lines
   Memory address = 8 bits
   Tag: 3 bits
   Line number: 3 bits
   Byte number: 2 bits
2. 0001 1011: Line = (110) = 6
   0011 0100: Line = (101) = 5
   1101 0000: Line = (100) = 4
   1010 1010: Line = (010) = 2
3. 1010 0001 is stored along with: 1010 0000
   1010 0001
   1010 0010
   1010 0011
4. 8 lines x 4 bytes = 32 bytes
5. Tag is used to identify the address because there a too many addresses in the memory. To avoid mess up, the system needs to check tag after verified the address bits.

Problem 2

1. The following code is the example of spatial locality.

```
cout << "Hello World";
 cin >> a;
```

   Because when the programmer is keep typing on the keyboard, the data will store in the first element of the array, and then the array elements will move forward. Furthermore, when the string "Hello World" is outputted, the system scans the first element and visits the next address.

2. This code is the example of temporal locality.

```
 for(i = 0; i < 50; i++)
cout<<i;
```

   Because this is a loop, although the value of the variable i keep changing, the address never changes.

Part 3

A[0]	0x0	B[0]	0x100
A[1]	0x4	B[1]	0x104
A[2]	0x8	B[2]	0x108
A[3]	0xc	B[3]	0x10c
A[4]	0c10	B[4]	0x110
A[5]	0x14	B[5]	0x114
A[6]	0x18	B[6]	0x118
A[7]	0x1c	B[7]	0x11c

1）

Paper simulation

block size = 16

cache size = 256

block size = 16

way = 4

set = 4

tag = 18

set_bit = 2

offset_bit = 4

A[0] to A[3] & B[0] to B[3] -> Set0

A[4] to A[7] & B[4] to B[7] -> Set1

14 hits 2 miss

Hit rate: 0.875

Set0 Way0	Set0 Way1	Set0 Way2	Set0 Way3	Set1 Way0	Set1 Way1	Set1 Way2	Set1 Way3	Address	Outcome
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x0	Hit
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x100	Miss
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x4	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x104	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x8	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x108	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0xc	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x10c	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x10	Hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x110	miss
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x14	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x114	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x18	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x118	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x1c	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x11c	hit

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 4
0KB 4-way associative cache:
Block size = 16 bytes
Number of [sets, blocks] = [4,16]
Extra space for tag storage = 36 bytes(12.33%)
Bits for [tag,index,offset]=[18,2,4]= 24
write plicy = write-through
Hex address      Binary address          Tag     Set     Blk     Way     UWay    Read    Write
=============================================================================================

     0R         000000000000            0       0       0       0      -1       0       0

   100R         000100000000            4       0       0      -1       1       1       0

     4R         000000000100            0       0       4       0      -1       0       0

   104R         000100000100            4       0       4       1      -1       0       0

     8R         000000001000            0       0       8       0      -1       0       0

   108R         000100001000            4       0       8       1      -1       0       0

     cR         000000001100            0       0      12       0      -1       0       0

   10cR         000100001100            4       0      12       1      -1       0       0

    10R         000000010000            0       1       0       0      -1       0       0

   110R         000100010000            4       1       0      -1       1       1       0

    14R         000000010100            0       1       4       0      -1       0       0

   114R         000100010100            4       1       4       1      -1       0       0

    18R         000000011000            0       1       8       0      -1       0       0

   118R         000100011000            4       1       8       1      -1       0       0

    1cR         000000011100            0       1      12       0      -1       0       0

   11cR         000100011100            4       1      12       1      -1       0       0

  5601                  344         0     1       -1       2       0       0
nref= 16, nread = 16, nwrite = 0
hit: 14, hit_rate:0.88
miss: 2, miss_rate:0.12
```

Compare result

Paper simulation and code simulation have same result.

2）

Paper simulation

block size = 8

cache size = 256

block size = 8

way = 4

set = 8

tag = 18

set = 3

offset = 3

A[0] to A[1] & B[0] to B[1] -> Set0

A[2] to A[3] & B[2] to B[3] -> Set1

A[4] to A[5] & B[4] to B[5] -> Set2

A[6] to A[7] & B[6] to B[7] -> Set3

Hit 12 miss 4

Hit rate：0.75

Set 0 way0	Set 0 way1	Set 1 way0	Set 1 way1	Set 2 way0	Set 2 way1	Set 3 way0	Set 3 way1	Address	Outcome
0-7	0-7	8-f	8-f	10-17	10-17	18-1f	18-1f	0x0	Hit
0-7	0-7	8-f	8-f	10-17	10-17	18-1f	18-1f	0x100	Miss
100-107	0-7	8-f	8-f	10-17	10-17	18-1f	18-1f	0x4	Hit
100-107	0-7	8-f	8-f	10-17	10-17	18-1f	18-1f	0x104	Hit
100-107	0-7	8-f	8-f	10-17	10-17	18-1f	18-1f	0x8	Hit
100-107	0-7	8-f	8-f	10-17	10-17	18-1f	18-1f	0x108	Miss
100-107	0-7	108-10f	8-f	10-17	10-17	18-1f	18-1f	0xc	Hit
100-107	0-7	108-10f	8-f	10-17	10-17	18-1f	18-1f	0x10c	Hit
100-107	0-7	108-10f	8-f	10-17	10-17	18-1f	18-1f	0x10	Hit
100-107	0-7	108-10f	8-f	110-117	10-17	18-1f	18-1f	0x110	Miss
100-107	0-7	108-10f	8-f	110-117	10-17	18-1f	18-1f	0x14	Hit
100-107	0-7	108-10f	8-f	110-117	10-17	18-1f	18-1f	0x114	Hit
100-107	0-7	108-10f	8-f	110-117	10-17	18-1f	18-1f	0x18	Hit
100-107	0-7	108-10f	8-f	110-117	10-17	18-1f	18-1f	0x118	Miss
100-107	0-7	108-10f	8-f	110-117	10-17	118-11f	18-1f	0x1c	Hit
100-107	0-7	108-10f	8-f	110-117	10-17	118-11f	18-1f	0x11c	Hit

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 8 -c 256 -w t -a 8
0KB 4-way associative cache:
Block size = 8 bytes
Number of [sets, blocks] = [8,32]
Extra space for tag storage = 72 bytes(21.95%)
Bits for [tag,index,offset]=[18,3,3]= 24
write plicy = write-through
Hex address      Binary address          Tag     Set     Blk     Way     UWay    Read    Write
==========================================================================================

     0R         000000000000            0       0       0       0      -1       0       0
   100R         000100000000            4       0       0      -1       1       1       0
     4R         000000000100            0       0       4       0      -1       0       0
   104R         000100000100            4       0       4       1      -1       0       0
     8R         000000001000            0       1       0       0      -1       0       0
   108R         000100001000            4       1       0      -1       1       1       0
     cR         000000001100            0       1       4       0      -1       0       0
   10cR         000100001100            4       1       4       1      -1       0       0
    10R         000000010000            0       2       0       0      -1       0       0
   110R         000100010000            4       2       0      -1       1       1       0
    14R         000000010100            0       2       4       0      -1       0       0
   114R         000100010100            4       2       4       1      -1       0       0
    18R         000000011000            0       3       0       0      -1       0       0
   118R         000100011000            4       3       0      -1       1       1       0
    1cR         000000011100            0       3       4       0      -1       0       0
   11cR         000100011100            4       3       4       1      -1       0       0

  5573                 341        6     3       -1       0       0       0
nref= 16, nread = 16, nwrite = 0
hit: 12, hit_rate:0.75
miss: 4, miss_rate:0.25
```

Compare result

Paper simulation and code simulation have same result.

3）

Paper simulation

block size = 4

cache size = 256

block size = 4

way = 4

set = 16

tag = 18

set = 4

offset = 2

A[0] & B[0] -> Set0

A[1] & B[1] -> Set1

A[2] & B[2] -> Set2

A[3] & B[3] -> Set3

A[4] & B[4] -> Set4

A[5] & B[5] -> Set5

A[6] & B[6] -> Set6

A[7] & B[7] -> Set7

Hit 8 miss 8

Hit rate：0.5

Set 0	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7	Address	Outcome
0-3	4-7	8-b	c-f	10-13	14-17	18-1b	1c-1f	0x0	Hit
0-3	4-7	8-b	c-f	10-13	14-17	18-1b	1c-1f	0x100	miss
100-103	4-7	8-b	c-f	10-13	14-17	18-1b	1c-1f	0x4	Hit
100-103	4-7	8-b	c-f	10-13	14-17	18-1b	1c-1f	0x104	Miss
100-103	104-107	8-b	c-f	10-13	14-17	18-1b	1c-1f	0x8	Hit
100-103	104-107	8-b	c-f	10-13	14-17	18-1b	1c-1f	0x108	Miss
100-103	104-107	108-10b	c-f	10-13	14-17	18-1b	1c-1f	0xc	Hit
100-103	104-107	108-10b	c-f	10-13	14-17	18-1b	1c-1f	0x10c	Miss
100-103	104-107	108-10b	10c-10f	10-13	14-17	18-1b	1c-1f	0x10	Hit
100-103	104-107	108-10b	10c-10f	10-13	14-17	18-1b	1c-1f	0x110	Miss
100-103	104-107	108-10b	10c-10f	110-113	14-17	18-1b	1c-1f	0x14	Hit
100-103	104-107	108-10b	10c-10f	110-113	14-17	18-1b	1c-1f	0x114	Miss
100-103	104-107	108-10b	10c-10f	110-113	114-117	18-1b	1c-1f	0x18	Hit
100-103	104-107	108-10b	10c-10f	110-113	114-117	18-1b	1c-1f	0x118	Miss
100-103	104-107	108-10b	10c-10f	110-113	114-117	118-11b	1c-1f	0x1c	Hit
100-103	104-107	108-10b	10c-10f	110-113	114-117	118-11b	1c-1f	0x11c	Miss

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 4 -c 256 -w t -a 16
0KB 4-way associative cache:
Block size = 4 bytes
Number of [sets, blocks] = [16,64]
Extra space for tag storage = 144 bytes(36.00%)
Bits for [tag,index,offset]=[18,4,2]= 24
write plicy = write-through
Hex address      Binary address          Tag     Set     Blk     Way     UWay    Read    Write
=============================================================================================

     0R         000000000000            0       0       0       0       -1      0       0

   100R         000100000000            4       0       0       -1      1       1       0

     4R         000000000100            0       1       0       0       -1      0       0

   104R         000100000100            4       1       0       -1      1       1       0

     8R         000000001000            0       2       0       0       -1      0       0

   108R         000100001000            4       2       0       -1      1       1       0

     cR         000000001100            0       3       0       0       -1      0       0

   10cR         000100001100            4       3       0       -1      1       1       0

    10R         000000010000            0       4       0       0       -1      0       0

   110R         000100010000            4       4       0       -1      1       1       0

    14R         000000010100            0       5       0       0       -1      0       0

   114R         000100010100            4       5       0       -1      1       1       0

    18R         000000011000            0       6       0       0       -1      0       0

   118R         000100011000            4       6       0       -1      1       1       0

    1cR         000000011100            0       7       0       0       -1      0       0

   11cR         000100011100            4       7       0       -1      1       1       0

  561f                  344         7     3       -1       2       1       0
nref= 16, nread = 16, nwrite = 0
hit: 8, hit_rate:0.50
miss: 8, miss_rate:0.50
```

Compare result

Paper simulation and code simulation have same result.

Part 4

1)

Paper simulation

When iteration = 1

cache size = 256

block size = 16

way = 4

set = 4

tag = 18

set = 2

offset = 4

A[0] to A[3] & B[0] to B[3] -> Set0

A[4] to A[7] & B[4] to B[7] -> Set1

14 hits 2 miss

Hit rate: 0.875

Set0 Way0	Set0 Way1	Set0 Way2	Set0 Way3	Set1 Way0	Set1 Way1	Set1 Way2	Set1 Way3	Address	Outcome
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x0	Hit
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x100	Miss
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x4	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x104	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x8	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x108	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0xc	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x10c	hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x10	Hit
100-10e	0-e	0-e	0-e	10-1e	10-1e	10-1e	0-e	0x110	miss
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x14	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x114	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x18	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x118	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x1c	hit
100-10e	0-e	0-e	0-e	110-11e	10-1e	10-1e	0-e	0x11c	hit

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 4
0KB 4-way associative cache:
Block size = 16 bytes
Number of [sets, blocks] = [4,16]
Extra space for tag storage = 36 bytes(12.33%)
Bits for [tag,index,offset]=[18,2,4]= 24
write plicy = write-through
Hex address      Binary address          Tag     Set     Blk     Way     UWay    Read    Write
=============================================================================================

     0R          000000000000            0       0       0       0      -1       0       0

   100R          000100000000            4       0       0      -1       1       1       0

     4R          000000000100            0       0       4       0      -1       0       0

   104R          000100000100            4       0       4       1      -1       0       0

     8R          000000001000            0       0       8       0      -1       0       0

   108R          000100001000            4       0       8       1      -1       0       0

     cR          000000001100            0       0      12       0      -1       0       0

   10cR          000100001100            4       0      12       1      -1       0       0

    10R          000000010000            0       1       0       0      -1       0       0

   110R          000100010000            4       1       0      -1       1       1       0

    14R          000000010100            0       1       4       0      -1       0       0

   114R          000100010100            4       1       4       1      -1       0       0

    18R          000000011000            0       1       8       0      -1       0       0

   118R          000100011000            4       1       8       1      -1       0       0

    1cR          000000011100            0       1      12       0      -1       0       0

   11cR          000100011100            4       1      12       1      -1       0       0

  5601                  344         0      1       -1       2       0       0
nref= 16, nread = 16, nwrite = 0
hit: 14, hit_rate:0.88
miss: 2, miss_rate:0.12
```

Compare result

Paper simulation and code simulation have same result.

2)

Paper simulation

When iteration =5,10,100

The exact address of A[0]-A[7],B[0]-B[7] will come several times.

The data in the cache is not changed and does not have any eviction.

Because the temporal locality, all the outcomes will hit after iteration 1.

Iteration	Hit	Miss	Hit rate
1	14	2	0.875
5	78	2	0.975
10	158	2	0.9875
100	1598	2	0.99875

Iteration: 5

Code simulation

```
  100R        000100000000          4          0     0       1      -1      0      0
  104R        000100000100          4          0     4       1      -1      0      0
  108R        000100001000          4          0     8       1      -1      0      0
  10cR        000100001100          4          0    12       1      -1      0      0
  110R        000100010000          4          1     0       1      -1      0      0
  114R        000100010100          4          1     4       1      -1      0      0
  118R        000100011000          4          1     8       1      -1      0      0
  11cR        000100011100          4          1    12       1      -1      0      0
    0R        000000000000          0          0     0       0      -1      0      0
    4R        000000000100          0          0     4       0      -1      0      0
    8R        000000001000          0          0     8       0      -1      0      0
    cR        000000001100          0          0    12       0      -1      0      0
   10R        000000010000          0          1     0       0      -1      0      0
   14R        000000010100          0          1     4       0      -1      0      0
   18R        000000011000          0          1     8       0      -1      0      0
   1cR        000000011100          0          1    12       0      -1      0      0
  100R        000100000000          4          0     0       1      -1      0      0
  104R        000100000100          4          0     4       1      -1      0      0
  108R        000100001000          4          0     8       1      -1      0      0
  10cR        000100001100          4          0    12       1      -1      0      0
  110R        000100010000          4          1     0       1      -1      0      0
  114R        000100010100          4          1     4       1      -1      0      0
  118R        000100011000          4          1     8       1      -1      0      0
  11cR        000100011100          4          1    12       1      -1      0      0
 5611                  344          1     1       -1       2       0       0
nref= 80, nread = 80, nwrite = 0
hit: 78, hit_rate:0.98
miss: 2, miss_rate:0.03
```

Iteration: 10

Code simulation

```
   100R          000100000000            4            0     0        1      -1       0       0
   104R          000100000100            4            0     4        1      -1       0       0
   108R          000100001000            4            0     8        1      -1       0       0
   10cR          000100001100            4            0    12        1      -1       0       0
   110R          000100010000            4            1     0        1      -1       0       0
   114R          000100010100            4            1     4        1      -1       0       0
   118R          000100011000            4            1     8        1      -1       0       0
   11cR          000100011100            4            1    12        1      -1       0       0
     0R          000000000000            0            0     0        0      -1       0       0
     4R          000000000100            0            0     4        0      -1       0       0
     8R          000000001000            0            0     8        0      -1       0       0
     cR          000000001100            0            0    12        0      -1       0       0
    10R          000000010000            0            1     0        0      -1       0       0
    14R          000000010100            0            1     4        0      -1       0       0
    18R          000000011000            0            1     8        0      -1       0       0
    1cR          000000011100            0            1    12        0      -1       0       0
   100R          000100000000            4            0     0        1      -1       0       0
   104R          000100000100            4            0     4        1      -1       0       0
   108R          000100001000            4            0     8        1      -1       0       0
   10cR          000100001100            4            0    12        1      -1       0       0
   110R          000100010000            4            1     0        1      -1       0       0
   114R          000100010100            4            1     4        1      -1       0       0
   118R          000100011000            4            1     8        1      -1       0       0
   11cR          000100011100            4            1    12        1      -1       0       0

  560c                  344         0    12       -1       2       0       0
nref= 160, nread = 160, nwrite = 0
hit: 158, hit_rate:0.99
miss: 2, miss_rate:0.01
```

Iteration: 100

Code simulation

```
  100R        000100000000          4          0     0       1      -1      0      0
  104R        000100000100          4          0     4       1      -1      0      0
  108R        000100001000          4          0     8       1      -1      0      0
  10cR        000100001100          4          0    12       1      -1      0      0
  110R        000100010000          4          1     0       1      -1      0      0
  114R        000100010100          4          1     4       1      -1      0      0
  118R        000100011000          4          1     8       1      -1      0      0
  11cR        000100011100          4          1    12       1      -1      0      0
    0R        000000000000          0          0     0       0      -1      0      0
    4R        000000000100          0          0     4       0      -1      0      0
    8R        000000001000          0          0     8       0      -1      0      0
    cR        000000001100          0          0    12       0      -1      0      0
   10R        000000010000          0          1     0       0      -1      0      0
   14R        000000010100          0          1     4       0      -1      0      0
   18R        000000011000          0          1     8       0      -1      0      0
   1cR        000000011100          0          1    12       0      -1      0      0
  100R        000100000000          4          0     0       1      -1      0      0
  104R        000100000100          4          0     4       1      -1      0      0
  108R        000100001000          4          0     8       1      -1      0      0
  10cR        000100001100          4          0    12       1      -1      0      0
  110R        000100010000          4          1     0       1      -1      0      0
  114R        000100010100          4          1     4       1      -1      0      0
  118R        000100011000          4          1     8       1      -1      0      0
  11cR        000100011100          4          1    12       1      -1      0      0

 5611                344         1     1       -1       2       0       0
nref= 1600, nread = 1600, nwrite = 0
hit: 1598, hit_rate:1.00
miss: 2, miss_rate:0.00
```

Compare result

Paper simulation and code simulation have same result.

5) What you expect to change if sum and both arrays are uint_8 instead of int?

If the type of the array change to uint_8, it means the size of the array is not 4 byte anymore, and it becomes 1 byte.

That will enhance the hit rate because of the spatial locality. The data saved sequent in a 16-byte block can hit more elements.

Part 5

cache size = 256

block size = 16

way = 4

set = 4

tag = 18

set = 2

offset = 4

stride = 1,2,4,8,16,21,64

num_elements = 128

iteration = 100

1)

Paper simulation

stride = 64

set 0: 0x0, 1x100

If iteration = 1 then:

Hit 1, Miss 1

Hit rate: 0.5

Set 0 way0	Set 0 way 1	Address	Outcome
0-e	0-e	0x0	hit
0-e	0-e	0x100	miss

Analyze the miss

There are 4 ways so the address will not conflict, and both of the data remain in the cache. no miss after the first cache line update.

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 4
0KB 4-way associative cache:
Block size = 16 bytes
Number of [sets, blocks] = [4,16]
Extra space for tag storage = 36 bytes(12.33%)
Bits for [tag,index,offset]=[18,2,4]= 24
write plicy = write-through
Hex address      Binary address          Tag     Set     Blk     Way     UWay    Read    Write
=============================================================================================

     0R         000000000000            0           0     0       0      -1       0       0

   100R         000100000000            4           0     0      -1       1       1       0

  55e8                 343        2     8       -1       0       1       0
nref= 2, nread = 2, nwrite = 0
hit: 1, hit_rate:0.50
miss: 1, miss_rate:0.50
```

If iteration = 100 then:

Hit 199, Miss 1

Hit rate: 0.995

Code simulation

```
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0

  5578              341        3     8       -1      0       0       0
nref= 200, nread = 200, nwrite = 0
hit: 199, hit_rate:1.00
miss: 1, miss_rate:0.00
```

If iteration = 10 then:

Hit 19, Miss 1

Hit rate: 0.95

Code simulation

```
0KB 4-way associative cache:
Block size = 16 bytes
Number of [sets, blocks] = [4,16]
Extra space for tag storage = 36 bytes(12.33%)
Bits for [tag,index,offset]=[18,2,4]= 24
write plicy = write-through
Hex address       Binary address          Tag      Set      Blk      Way      UWay     Read     Write
==========================================================================================================

     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0     -1      1      1      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0
     0R        000000000000          0          0     0      0     -1      0      0
   100R        000100000000          4          0     0      1     -1      0      0

  55e8              343        2     8       -1      0       0       0
nref= 20, nread = 20, nwrite = 0
hit: 19, hit_rate:0.95
miss: 1, miss_rate:0.05
```

If iteration = 50 then:

Hit 99, Miss 1

Hit rate: 0.99

Code simulation

```
  100R        000100000000          4          0     0        1     -1       0        0

    0R        000000000000          0          0     0        0     -1       0        0

  100R        000100000000          4          0     0        1     -1       0        0

    0R        000000000000          0          0     0        0     -1       0        0

  100R        000100000000          4          0     0        1     -1       0        0

    0R        000000000000          0          0     0        0     -1       0        0

  100R        000100000000          4          0     0        1     -1       0        0

    0R        000000000000          0          0     0        0     -1       0        0

  100R        000100000000          4          0     0        1     -1       0        0

    0R        000000000000          0          0     0        0     -1       0        0

  100R        000100000000          4          0     0        1     -1       0        0

    0R        000000000000          0          0     0        0     -1       0        0

  100R        000100000000          4          0     0        1     -1       0        0

    0R        000000000000          0          0     0        0     -1       0        0

  100R        000100000000          4          0     0        1     -1       0        0

 5625                 344       2     5       -1       0       0       0
nref= 100, nread = 100, nwrite = 0
hit: 99, hit_rate:0.99
miss: 1, miss rate:0.01
```

2)

Paper simulation

stride = 21

set 0: 0x0

set 1: 0x54, 0x150

set 2: 0xa8, 0x1a4

set 3: 0xfc, 0x1f8

If iteration = 1 then:

hit 1,miss 6

hit rate: 0.143

Set0 way0	Set0 way1	Set1 way0	Set1 way1	Set2 way0	Set2 way1	Set3 way0	Set3 way1	Address	Outcome
0-e	0-e	10-1e	10-1e	20-2e	20-2e	30-3e	30-3e	0x0	hit
0-e	0-e	10-1e	10-1e	20-2e	20-2e	30-3e	30-3e	0x54	miss
0-e	0-e	50-5e	10-1e	20-2e	20-2e	30-3e	30-3e	0xa8	miss
0-e	0-e	50-5e	10-1e	a0-ae	20-2e	30-3e	30-3e	0xfc	miss
0-e	0-e	50-5e	10-1e	a0-ae	20-2e	30-3e	f0-fe	0x150	miss
0-e	0-e	50-5e	150-15e	a0-ae	20-2e	30-3e	f0-fe	0x1a4	miss
0-e	0-e	50-5e	150-15e	a0-ae	1a0-1ae	30-3e	f0-fe	0x1f8	miss

Analyze the miss

There are 4 ways so the address will not conflict, and both of the data remain in the cache. no miss after the first cache line update.

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 4
0KB 4-way associative cache:
Block size = 16 bytes
Number of [sets, blocks] = [4,16]
Extra space for tag storage = 36 bytes(12.33%)
Bits for [tag,index,offset]=[18,2,4]= 24
write plicy = write-through
Hex address      Binary address          Tag     Set     Blk     Way     UWay    Read    Write
=============================================================================================

     0R         000000000000            0       0       0       0       -1      0       0

    54R         000001010100            1       1       4       -1      0       1       0

    a8R         000010101000            2       2       8       -1      0       1       0

    fcR         000011111100            3       3       12      -1      0       1       0

   150R         000101010000            5       1       0       -1      1       1       0

   1a4R         000110100100            6       2       4       -1      1       1       0

   1f8R         000111111000            7       3       8       -1      1       1       0

  5581                  342         0     1       -1       1       1       0
nref= 7, nread = 7, nwrite = 0
hit: 1, hit_rate:0.14
miss: 6, miss_rate:0.86
```

If iteration = 100 then:

hit 694,miss 6

hit rate: 0.991

Code simulation

```
     0R         000000000000            0       0       0       0       -1      0       0

    54R         000001010100            1       1       4       0       -1      0       0

    a8R         000010101000            2       2       8       0       -1      0       0

    fcR         000011111100            3       3       12      0       -1      0       0

   150R         000101010000            5       1       0       1       -1      0       0

   1a4R         000110100100            6       2       4       1       -1      0       0

   1f8R         000111111000            7       3       8       1       -1      0       0

  55c9                  343         0     9       -1       1       0       0
nref= 700, nread = 700, nwrite = 0
hit: 694, hit_rate:0.99
miss: 6, miss_rate:0.01
```

If iteration = 10 then:

hit 64,miss 6

hit rate: 0.914

Code simulation

```
     0R         000000000000            0       0       0       0       -1      0       0

    54R         000001010100            1       1       4       0       -1      0       0

    a8R         000010101000            2       2       8       0       -1      0       0

    fcR         000011111100            3       3       12      0       -1      0       0

   150R         000101010000            5       1       0       1       -1      0       0

   1a4R         000110100100            6       2       4       1       -1      0       0

   1f8R         000111111000            7       3       8       1       -1      0       0

  563d                  344         3     13       -1       2       0       0
nref= 70, nread = 70, nwrite = 0
hit: 64, hit_rate:0.91
miss: 6, miss_rate:0.09
```

If iteration = 50 then:

hit 344,miss 6

hit rate: 0.983

Code simulation

```
     0R          000000000000            0           0     0        0     -1     0     0

    54R          000001010100            1           1     4        0     -1     0     0

    a8R          000010101000            2           2     8        0     -1     0     0

    fcR          000011111100            3           3    12        0     -1     0     0

   150R          000101010000            5           1     0        1     -1     0     0

   1a4R          000110100100            6           2     4        1     -1     0     0

   1f8R          000111111000            7           3     8        1     -1     0     0

  55b5                  342         3     5       -1       2        0      0
nref= 350, nread = 350, nwrite = 0
hit: 344, hit_rate:0.98
miss: 6, miss_rate:0.02
```

3)

Paper simulation

stride = 16

set 0: all address is in set 0

If iteration = 1 then:

hit 1, miss 7

hit rate: 0. 125

set0 way0	set0 way1	set0 way2	set0 way3	Address	Outcome
0-e	0-e	0-e	0-e	0x0	hit
0-e	0-e	0-e	0-e	0x40	miss
40-4e	0-e	0-e	0-e	0x80	miss
40-4e	80-8e	0-e	0-e	0xc0	miss
40-4e	80-8e	c0-ce	0-e	0x100	miss
40-4e	80-8e	c0-ce	100-10e	0x140	miss
140-14e	80-8e	c0-ce	100-10e	0x180	miss
140-14e	180-18e	c0-ce	100-10e	0x1c0	miss

Analyze the miss

There are 8 addresses in set0, but set0 only has 4 ways that cause conflict, and the data in the cache have to evict and reload all the time. Every iteration will cause miss.

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 4
0KB 4-way associative cache:
Block size = 16 bytes
Number of [sets, blocks] = [4,16]
Extra space for tag storage = 36 bytes(12.33%)
Bits for [tag,index,offset]=[18,2,4]= 24
write plicy = write-through
Hex address      Binary address          Tag     Set     Blk     Way     UWay    Read    Write
=============================================================================================

     0R         000000000000            0       0       0       0      -1       0       0

    40R         000001000000            1       0       0      -1       1       1       0

    80R         000010000000            2       0       0      -1       2       1       0

    c0R         000011000000            3       0       0      -1       3       1       0

   100R         000100000000            4       0       0      -1       0       1       0

   140R         000101000000            5       0       0      -1       1       1       0

   180R         000110000000            6       0       0      -1       2       1       0

   1c0R         000111000000            7       0       0      -1       3       1       0

  55ce                  343         0    14       -1       0       1       0
nref= 8, nread = 8, nwrite = 0
hit: 1, hit_rate:0.12
miss: 7, miss_rate:0.88
```

If iteration = 100 then:

hit 1, miss 799

hit rate: 0.00125

Code simulation

```
     0R         000000000000            0       0       0      -1       0       1       0

    40R         000001000000            1       0       0      -1       1       1       0

    80R         000010000000            2       0       0      -1       2       1       0

    c0R         000011000000            3       0       0      -1       3       1       0

   100R         000100000000            4       0       0      -1       0       1       0

   140R         000101000000            5       0       0      -1       1       1       0

   180R         000110000000            6       0       0      -1       2       1       0

   1c0R         000111000000            7       0       0      -1       3       1       0

  5580                  342         0     0       -1       0       1       0
nref= 800, nread = 800, nwrite = 0
hit: 1, hit_rate:0.00
miss: 799, miss_rate:1.00
```

If iteration = 10 then:

hit 1, miss 79

hit rate: 0. 0125

Code simulation

```
     0R         000000000000            0          0     0       -1       0       1       0
    40R         000001000000            1          0     0       -1       1       1       0
    80R         000010000000            2          0     0       -1       2       1       0
    c0R         000011000000            3          0     0       -1       3       1       0
   100R         000100000000            4          0     0       -1       0       1       0
   140R         000101000000            5          0     0       -1       1       1       0
   180R         000110000000            6          0     0       -1       2       1       0
   1c0R         000111000000            7          0     0       -1       3       1       0

  5634                 344         3     4        -1       0       1       0
nref= 80, nread = 80, nwrite = 0
hit: 1, hit_rate:0.01
miss: 79, miss_rate:0.99
```

If iteration = 50 then:

hit 1, miss 399

hit rate: 0. 0025

Code simulation

```
     0R         000000000000            0          0     0       -1       0       1       0
    40R         000001000000            1          0     0       -1       1       1       0
    80R         000010000000            2          0     0       -1       2       1       0
    c0R         000011000000            3          0     0       -1       3       1       0
   100R         000100000000            4          0     0       -1       0       1       0
   140R         000101000000            5          0     0       -1       1       1       0
   180R         000110000000            6          0     0       -1       2       1       0
   1c0R         000111000000            7          0     0       -1       3       1       0

  564e                 345         0    14        -1       0       1       0
nref= 400, nread = 400, nwrite = 0
hit: 1, hit_rate:0.00
miss: 399, miss_rate:1.00
```

4)

Paper simulation

stride = 8

set 0: 0x0, 0x40, 0x80, 0xc0, 0x100, 0x140, 0x180, 0x1c0

set 2: 0x20, 0x60, 0xa0, 0xe0, 0x120, 0x160, 0x1a0, 0x1e0,

If iteration = 1 then:

hit 2, miss 14

hit rate: 0. 125

Set0 way0	Set0 way1	Set0 way2	Set0 way3	Set2 way0	Set2 way1	Set2 way2	Set2 way3	Address	Outcome
0-e	0-e	0-e	0-e	20-2e	20-2e	20-2e	20-2e	0x0	hit
0-e	0-e	0-e	0-e	20-2e	20-2e	20-2e	20-2e	0x20	hit
0-e	0-e	0-e	0-e	20-2e	20-2e	20-2e	20-2e	0x40	miss
40-4e	0-e	0-e	0-e	20-2e	20-2e	20-2e	20-2e	0x60	miss
40-4e	0-e	0-e	0-e	60-6e	20-2e	20-2e	20-2e	0x80	miss
40-4e	80-8e	0-e	0-e	60-6e	20-2e	20-2e	20-2e	0xa0	miss
40-4e	80-8e	0-e	0-e	60-6e	a0-ae	20-2e	20-2e	0xc0	miss

40-4e	80-8e	c0-ce	0-e	60-6e	a0-ae	20-2e	20-2e	0xe0	miss
40-4e	80-8e	c0-ce	0-e	60-6e	a0-ae	e0-ee	20-2e	0x100	miss
40-4e	80-8e	c0-ce	100-10e	60-6e	a0-ae	e0-ee	20-2e	0x120	miss
40-4e	80-8e	c0-ce	100-10e	60-6e	a0-ae	e0-ee	120-12e	0x140	miss
140-14e	80-8e	c0-ce	100-10e	60-6e	a0-ae	e0-ee	120-12e	0x160	miss
140-14e	80-8e	c0-ce	100-10e	160-16e	a0-ae	e0-ee	120-12e	0x180	miss
140-14e	180-18e	c0-ce	100-10e	160-16e	a0-ae	e0-ee	120-12e	0x1a0	miss
140-14e	180-18e	c0-ce	100-10e	160-16e	1a0-1ae	e0-ee	120-12e	0x1c0	miss
140-14e	180-18e	1c0-1ce	100-10e	160-16e	1a0-1ae	e0-ee	120-12e	0x1e0	miss

Analyze the miss

There are 8 addresses in both set0 & set2, but set0 & set2 only has 4 ways that cause conflict, and the data in the cache have to evict and reload all the time. Every iteration will cause miss.

Code simulation

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 4
0KB 4-way associative cache:
Block size = 16 bytes
Number of [sets, blocks] = [4,16]
Extra space for tag storage = 36 bytes(12.33%)
Bits for [tag,index,offset]=[18,2,4]= 24
write plicy = write-through
Hex address     Binary address          Tag     Set     Blk     Way     UWay    Read    Write
=============================================================================================

     0R         000000000000            0       0       0       0       -1      0       0
    20R         000000100000            0       2       0       0       -1      0       0
    40R         000001000000            1       0       0       -1      1       1       0
    60R         000001100000            1       2       0       -1      1       1       0
    80R         000010000000            2       0       0       -1      2       1       0
    a0R         000010100000            2       2       0       -1      2       1       0
    c0R         000011000000            3       0       0       -1      3       1       0
    e0R         000011100000            3       2       0       -1      3       1       0
   100R         000100000000            4       0       0       -1      0       1       0
   120R         000100100000            4       2       0       -1      0       1       0
   140R         000101000000            5       0       0       -1      1       1       0
   160R         000101100000            5       2       0       -1      1       1       0
   180R         000110000000            6       0       0       -1      2       1       0
   1a0R         000110100000            6       2       0       -1      2       1       0
   1c0R         000111000000            7       0       0       -1      3       1       0
   1e0R         000111100000            7       2       0       -1      3       1       0
  55a9                  342         2       9       -1      0       1       0
nref= 16, nread = 16, nwrite = 0
hit: 2, hit_rate:0.12
miss: 14, miss_rate:0.88
```

If iteration = 100 then:

hit 2, miss 1598

hit rate: 0.00125

Code simulation

```
   100R         000100000000           4          0     0       -1       0       1       0

   120R         000100100000           4          2     0       -1       0       1       0

   140R         000101000000           5          0     0       -1       1       1       0

   160R         000101100000           5          2     0       -1       1       1       0

   180R         000110000000           6          0     0       -1       2       1       0

   1a0R         000110100000           6          2     0       -1       2       1       0

   1c0R         000111000000           7          0     0       -1       3       1       0

   1e0R         000111100000           7          2     0       -1       3       1       0

  5625                 344         2     5       -1       0       1       0
nref= 1600, nread = 1600, nwrite = 0
hit: 2, hit_rate:0.00
miss: 1598, miss_rate:1.00
```

If iteration = 10 then:

hit 2, miss 158

hit rate: 0. 0125

Code simulation

```
   100R         000100000000           4          0     0       -1       0       1       0

   120R         000100100000           4          2     0       -1       0       1       0

   140R         000101000000           5          0     0       -1       1       1       0

   160R         000101100000           5          2     0       -1       1       1       0

   180R         000110000000           6          0     0       -1       2       1       0

   1a0R         000110100000           6          2     0       -1       2       1       0

   1c0R         000111000000           7          0     0       -1       3       1       0

   1e0R         000111100000           7          2     0       -1       3       1       0

  563f                 344         3    15       -1       0       1       0
nref= 160, nread = 160, nwrite = 0
hit: 2, hit_rate:0.01
miss: 158, miss_rate:0.99
```

If iteration = 50 then:

hit 2, miss 798

hit rate: 0. 0025

Code simulation

```
   100R         000100000000           4          0     0       -1       0       1       0

   120R         000100100000           4          2     0       -1       0       1       0

   140R         000101000000           5          0     0       -1       1       1       0

   160R         000101100000           5          2     0       -1       1       1       0

   180R         000110000000           6          0     0       -1       2       1       0

   1a0R         000110100000           6          2     0       -1       2       1       0

   1c0R         000111000000           7          0     0       -1       3       1       0

   1e0R         000111100000           7          2     0       -1       3       1       0

  556b                 341         2    11       -1       0       1       0
nref= 800, nread = 800, nwrite = 0
hit: 2, hit_rate:0.00
miss: 798, miss_rate:1.00
```

5)

Paper simulation

stride = 4

If iteration = 1 then:

hit 4, miss 28

hit rate: 0.125

set0 way0	set0 way1	set0 way2	set0 way3	set1 way0	set1 way1	set1 way2	set1 way3	set2 way0	set2 way1
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	10-1e	20-2e	20-2e
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	10-1e	20-2e	20-2e
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	10-1e	20-2e	20-2e
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	10-1e	20-2e	20-2e
0-e	0-e	0-e	0-e	10-1e	10-1e	10-1e	10-1e	20-2e	20-2e
40-4e	0-e	0-e	0-e	10-1e	10-1e	10-1e	10-1e	20-2e	20-2e
40-4e	0-e	0-e	0-e	50-5e	10-1e	10-1e	10-1e	20-2e	20-2e
40-4e	0-e	0-e	0-e	50-5e	10-1e	10-1e	10-1e	60-6e	20-2e
40-4e	0-e	0-e	0-e	50-5e	10-1e	10-1e	10-1e	60-6e	20-2e
40-4e	80-8e	0-e	0-e	50-5e	10-1e	10-1e	10-1e	60-6e	20-2e
40-4e	80-8e	0-e	0-e	50-5e	90-9e	10-1e	10-1e	60-6e	20-2e
40-4e	80-8e	0-e	0-e	50-5e	90-9e	10-1e	10-1e	60-6e	a0-ae
40-4e	80-8e	0-e	0-e	50-5e	90-9e	10-1e	10-1e	60-6e	a0-ae
...	...	...	...	...	...	...	...	...	...

Analyze the miss

Because each set only has 4 ways and each block only has 16 bytes, if stride = 4, the address needs to move 16 bytes to the next address, and the cache never stops to renew the data in blocks. All the address will miss.

Code simulation

```
  150R        000101010000          5          1     0      -1      1      1      0
  160R        000101100000          5          2     0      -1      1      1      0
  170R        000101110000          5          3     0      -1      1      1      0
  180R        000110000000          6          0     0      -1      2      1      0
  190R        000110010000          6          1     0      -1      2      1      0
  1a0R        000110100000          6          2     0      -1      2      1      0
  1b0R        000110110000          6          3     0      -1      2      1      0
  1c0R        000111000000          7          0     0      -1      3      1      0
  1d0R        000111010000          7          1     0      -1      3      1      0
  1e0R        000111100000          7          2     0      -1      3      1      0
  1f0R        000111110000          7          3     0      -1      3      1      0

  55b1                342          3     1       -1       0       1       0
nref= 32, nread = 32, nwrite = 0
hit: 4, hit_rate:0.12
miss: 28, miss_rate:0.88
```

If iteration = 100 then:

hit 4, miss 3196

hit rate: 0.001

Code simulation

```
   150R         000101010000            5           1     0       -1       1       1       0

   160R         000101100000            5           2     0       -1       1       1       0

   170R         000101110000            5           3     0       -1       1       1       0

   180R         000110000000            6           0     0       -1       2       1       0

   190R         000110010000            6           1     0       -1       2       1       0

   1a0R         000110100000            6           2     0       -1       2       1       0

   1b0R         000110110000            6           3     0       -1       2       1       0

   1c0R         000111000000            7           0     0       -1       3       1       0

   1d0R         000111010000            7           1     0       -1       3       1       0

   1e0R         000111100000            7           2     0       -1       3       1       0

   1f0R         000111110000            7           3     0       -1       3       1       0

  5593                  342         1     3       -1       0       1       0
nref= 3200, nread = 3200, nwrite = 0
hit: 4, hit_rate:0.00
miss: 3196, miss_rate:1.00
```

If iteration = 10 then:

hit 4, miss 316

hit rate: 0. 01

Code simulation

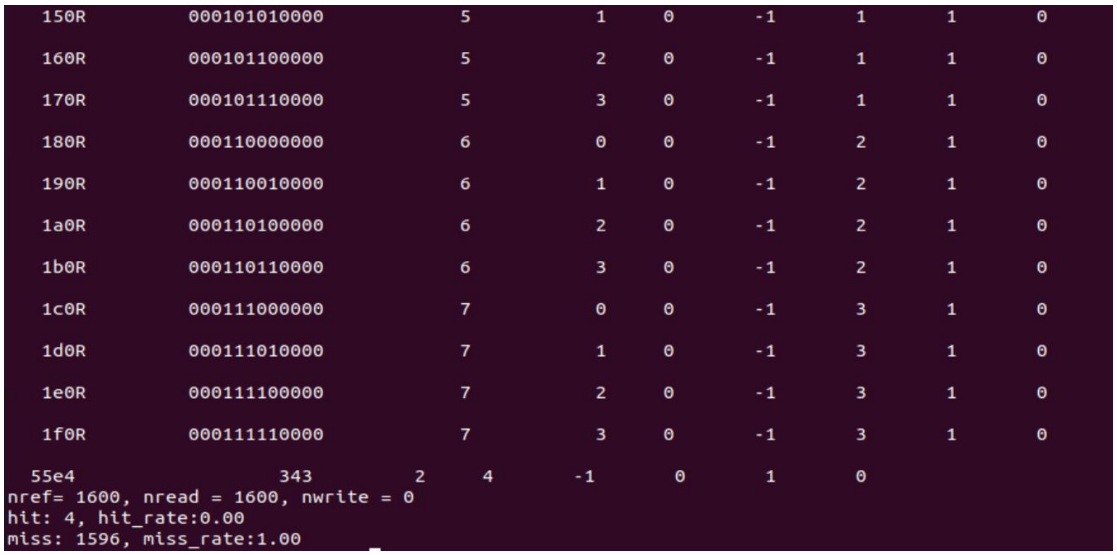
```
   180R         000110000000            6           0     0       -1       2       1       0

   190R         000110010000            6           1     0       -1       2       1       0

   1a0R         000110100000            6           2     0       -1       2       1       0

   1b0R         000110110000            6           3     0       -1       2       1       0

   1c0R         000111000000            7           0     0       -1       3       1       0

   1d0R         000111010000            7           1     0       -1       3       1       0

   1e0R         000111100000            7           2     0       -1       3       1       0

   1f0R         000111110000            7           3     0       -1       3       1       0

  55af                  342         2    15       -1       0       1       0
nref= 320, nread = 320, nwrite = 0
hit: 4, hit_rate:0.01
miss: 316, miss_rate:0.99
```

If iteration = 50 then:

hit 4, miss 1596

hit rate: 0. 0025

Code simulation



6)

Paper simulation

stride = 2

If iteration = 1 then:

hit 33, miss 31

hit rate: 0.515

	Address	Outcome
Set 0 way 0	0x0	hit
Set 0 way 0	0x8	hit
Set 1 way 0	0x10	miss
Set 1 way 0	0x18	hit
Set 2 way 0	0x20	miss
Set 2 way 0	0x28	hit
Set 3 way 0	0x30	miss
Set 3 way 0	0x38	hit
…	…	…

Analyze the miss

When the stride = 2, a 16 bytes block can hold 2 spatial addresses, but the conflict still happens since there are only 4 ways in the set. As the result, the address in same set will hit and miss.

Code simulation

```
   1c0R         000111000000            7           0     0       -1       3       1       0
   1c8R         000111001000            7           0     8        3      -1       0       0
   1d0R         000111010000            7           1     0       -1       3       1       0
   1d8R         000111011000            7           1     8        3      -1       0       0
   1e0R         000111100000            7           2     0       -1       3       1       0
   1e8R         000111101000            7           2     8        3      -1       0       0
   1f0R         000111110000            7           3     0       -1       3       1       0
   1f8R         000111111000            7           3     8        3      -1       0       0

  55d9                 343         1     9       -1       0       0       0
nref= 64, nread = 64, nwrite = 0
hit: 36, hit_rate:0.56
miss: 28, miss_rate:0.44
```

If iteration = 100 then:

hit 3201 miss 3199

hit rate:0.5001

Code simulation

```
   1c0R         000111000000            7           0     0       -1       3       1       0
   1c8R         000111001000            7           0     8        3      -1       0       0
   1d0R         000111010000            7           1     0       -1       3       1       0
   1d8R         000111011000            7           1     8        3      -1       0       0
   1e0R         000111100000            7           2     0       -1       3       1       0
   1e8R         000111101000            7           2     8        3      -1       0       0
   1f0R         000111110000            7           3     0       -1       3       1       0
   1f8R         000111111000            7           3     8        3      -1       0       0

  5560                 341         2     0       -1       0       0       0
nref= 6400, nread = 6400, nwrite = 0
hit: 3204, hit_rate:0.50
miss: 3196, miss_rate:0.50
```

If iteration = 10 then:

hit 321, miss 319

hit rate: 0.501

Code simulation

```
   1b8R         000110111000            6           3     8        2      -1       0       0
   1c0R         000111000000            7           0     0       -1       3       1       0
   1c8R         000111001000            7           0     8        3      -1       0       0
   1d0R         000111010000            7           1     0       -1       3       1       0
   1d8R         000111011000            7           1     8        3      -1       0       0
   1e0R         000111100000            7           2     0       -1       3       1       0
   1e8R         000111101000            7           2     8        3      -1       0       0
   1f0R         000111110000            7           3     0       -1       3       1       0
   1f8R         000111111000            7           3     8        3      -1       0       0

  556e                 341         2    14       -1       0       0       0
nref= 640, nread = 640, nwrite = 0
hit: 324, hit_rate:0.51
miss: 316, miss_rate:0.49
```

If iteration = 50 then:

hit 1601, miss 1599

hit rate: 0.5

Code simulation

```
   1c0R         000111000000            7           0     0       -1       3       1       0
   1c8R         000111001000            7           0     8        3      -1       0       0
   1d0R         000111010000            7           1     0       -1       3       1       0
   1d8R         000111011000            7           1     8        3      -1       0       0
   1e0R         000111100000            7           2     0       -1       3       1       0
   1e8R         000111101000            7           2     8        3      -1       0       0
   1f0R         000111110000            7           3     0       -1       3       1       0
   1f8R         000111111000            7           3     8        3      -1       0       0

  55f6                 343         3     6       -1       0       0       0
nref= 3200, nread = 3200, nwrite = 0
hit: 1604, hit_rate:0.50
miss: 1596, miss_rate:0.50
```

7)

Paper simulation

stride = 1

If iteration = 1 then:

hit 97, miss 31

hit rate: 0.757

	Address	Outcome
Set 0 way 0	0x0	hit
Set 0 way 0	0x4	hit
Set 0 way 0	0x8	hit
Set 0 way 0	0xc	hit
Set 1 way 0	0x10	miss
Set 1 way 0	0x14	hit
Set 1 way 0	0x18	hit
Set 1 way 0	0x1c	hit
Set 2 way 0	0x20	miss
Set 2 way 0	0x24	hit
Set 2 way 0	0x28	hit
Set 2 way 0	0x2c	hit
Set 3 way 0	0x30	miss
…	…	…

Analyze the miss

When the stride = 1, a 16 bytes block can hold 4 spatial addresses, but the conflict still happens since there are only 4 ways in the set. The miss will happened when all ways in the set full.

Code simulation

```
   1d8R         000111011000            7           1     8        3      -1       0       0
   1dcR         000111011100            7           1    12        3      -1       0       0
   1e0R         000111100000            7           2     0       -1       3       1       0
   1e4R         000111100100            7           2     4        3      -1       0       0
   1e8R         000111101000            7           2     8        3      -1       0       0
   1ecR         000111101100            7           2    12        3      -1       0       0
   1f0R         000111110000            7           3     0       -1       3       1       0
   1f4R         000111110100            7           3     4        3      -1       0       0
   1f8R         000111111000            7           3     8        3      -1       0       0
   1fcR         000111111100            7           3    12        3      -1       0       0
  557e                 341         3    14       -1       0       0       0
nref= 128, nread = 128, nwrite = 0
hit: 100, hit_rate:0.78
miss: 28, miss_rate:0.22
```

If iteration = 100 then:

hit 9601 miss 3199

hit rate:0.75

Code simulation

```
   1e4R         000111100100            7           2     4        3      -1       0       0
   1e8R         000111101000            7           2     8        3      -1       0       0
   1ecR         000111101100            7           2    12        3      -1       0       0
   1f0R         000111110000            7           3     0       -1       3       1       0
   1f4R         000111110100            7           3     4        3      -1       0       0
   1f8R         000111111000            7           3     8        3      -1       0       0
   1fcR         000111111100            7           3    12        3      -1       0       0
  55e8                 343         2     8       -1       0       0       0
  55e8                 343         2     8        0      -1       0       0
  55e8                 343         2     8        0      -1       0       0
nref= 12800, nread = 12800, nwrite = 0
hit: 9604, hit_rate:0.75
miss: 3196, miss_rate:0.25
```

If iteration = 10 then:

hit 961, miss 319

hit rate: 0.751

Code simulation

```
   1d8R         000111011000            7           1     8        3      -1       0       0

   1dcR         000111011100            7           1    12        3      -1       0       0

   1e0R         000111100000            7           2     0       -1       3       1       0

   1e4R         000111100100            7           2     4        3      -1       0       0

   1e8R         000111101000            7           2     8        3      -1       0       0

   1ecR         000111101100            7           2    12        3      -1       0       0

   1f0R         000111110000            7           3     0       -1       3       1       0

   1f4R         000111110100            7           3     4        3      -1       0       0

   1f8R         000111111000            7           3     8        3      -1       0       0

   1fcR         000111111100            7           3    12        3      -1       0       0

  55cc                 343         0    12       -1       0       0       0
nref= 1280, nread = 1280, nwrite = 0
hit: 964, hit_rate:0.75
miss: 316, miss_rate:0.25
```

If iteration = 50 then:

hit 4801, miss 1599

hit rate: 0.75

Code simulation

```
   1d4R         000111010100            7           1     4        3      -1       0       0

   1d8R         000111011000            7           1     8        3      -1       0       0

   1dcR         000111011100            7           1    12        3      -1       0       0

   1e0R         000111100000            7           2     0       -1       3       1       0

   1e4R         000111100100            7           2     4        3      -1       0       0

   1e8R         000111101000            7           2     8        3      -1       0       0

   1ecR         000111101100            7           2    12        3      -1       0       0

   1f0R         000111110000            7           3     0       -1       3       1       0

   1f4R         000111110100            7           3     4        3      -1       0       0

   1f8R         000111111000            7           3     8        3      -1       0       0

   1fcR         000111111100            7           3    12        3      -1       0       0

  5556                 341         1     6       -1       0       0       0

  5556                 341         1     6        0      -1       0       0
nref= 6400, nread = 6400, nwrite = 0
hit: 4804, hit_rate:0.75
miss: 1596, miss_rate:0.25
```

4)How does varying number of iterations and stride value affect the hit rate? How does this relate to spatial and temporal locality?

If stride < 2, 16 bytes block can contain more than one address, the spatial locality is increased.

If stride > 16, the number of array addresses after the loop are less than 8, the conflict risk will be reduced, the temporal locality is increased.

If the spatial or temporal locality increases, the hit rate also grows when the number of iterations grow to a stable and reasonable rate. Else when the number of iterations grows, the hit rate will down to almost 0.

Part 6

cache size = 256
block size = 16
way = 4
set = 4
tag = 18
set = 2
offset = 4
stride = 1
num_elements = 128
iteration = 100

Paper simulation

Same as part5, when stride = 1, the number of the array addresses is 128.
4 sequence addresses can be stored in one block, and there are only 16 blocks.
As a result, 128 addresses cannot be stored in 16 blocks. The data in the block must be replaced in 1 iteration. The hit rate is no difference between these mapping methods.
If iteration = 100 then:
hit 9601 miss 3199
hit rate:0.75

Code simulation

a. Direct Mapped

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 1
nref= 12800, nread = 12800, nwrite = 0
hit: 9601, hit_rate:0.75
miss: 3199, miss_rate:0.25
```

b. 2-way Set Associative

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 8
nref= 12800, nread = 12800, nwrite = 0
hit: 9608, hit_rate:0.75
miss: 3192, miss_rate:0.25
```

c. 4-way Set Associative

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 4
nref= 12800, nread = 12800, nwrite = 0
hit: 9604, hit_rate:0.75
miss: 3196, miss_rate:0.25
```

d. Fully-Associative Cache

```
guangmin@guangmin-VirtualBox:~$ ./a.out -b 16 -c 256 -w t -a 16
nref= 12800, nread = 12800, nwrite = 0
hit: 9616, hit_rate:0.75
miss: 3184, miss_rate:0.25
```