Samiksha Modi 2019331

Cache Assignment (End Sem)

The cache is of one level only. The word length of the machine is 32 bits by default but you can change it while giving input. The assumption is that the size of each memory word is 1 byte. All the inputs have to be in the power of 2. The write policy used is Write Through (WT) and the replacement policy used is Least Recently Used (LRU). The address is input in decimal (base 10).

Usage

- Open the terminal and navigate to the directory where the files are saved.
- Run samiksha_2019331_endsem.py (for endsem assignment)
- Enter the input as instructed
- The program is written in Python 3.6.9

Input

- Cache Lines - in the power of 2
- Block Size - in the power of 2 bytes
- Address Bits - 32 bits by default but can be 16,32,64 bits based on user discretion
- Set Size - is n in the power of 2 for n-way set associative mapping
- Mapping
    - 0 - Direct
    - 1 - Associative
    - 2 - N-way Set Associative

```
samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$ python3 samiksha_2019331_endsem.py
Input cache lines in power of 2: 4
Input block size in power of 2 bytes: 4
Input no of bits in memory address in power of 2: 32
cache lines:  4
address bits:  32
offset bits:  2
 Enter the desired number
 0 for Direct Mapping
 1 for Fully Associative Mapping
 2 for Set Associative Mapping
 0
index bits:  2
tag bits: 28

------------------------------------------------------------------------
                        CACHE TABLE
------------------------------------------------------------------------
Index   Valid   Tag             Data(Decimal)           Dirty Bit
  0       0      -                    0                    0
  1       0      -                    0                    0
  2       0      -                    0                    0
  3       0      -                    0                    0
```

Commands

- read address
- write address data
- quit - to exit the program

After every valid command, the entire cache will be printed. In case of a MISS, it will print address not found and will update the cache table with the new dataset. In case of a replacement, it will also print the address by which the block will be replaced.

```
read 0
Address bits: 32

Instruction Breakdown
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000000     00              00
Index in decimal:  0

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 0 with offset 0 to 3 is transferred to cache at index 0

-------------------------------------------------------------------------
                          CACHE TABLE
-------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000    BLOCK 0         0
1        0       -       0       0
2        0       -       0       0
3        0       -       0       0
write 18 43



write  18 43
Address bits: 32

Instruction Breakdown
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000001     00              10
Index in decimal:  0

Cache MISS!! - Address not found
Cache table is updated accordingly
Address 18 will replace the block at index 0
BLOCK 4 with offset 0 to 3 is transferred to cache at index 0
Content is updated based on Write Policy

-------------------------------------------------------------------------
                          CACHE TABLE
-------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000001    BLOCK 4         0
1        0       -       0       0
2        0       -       0       0
3        0       -       0       0
quit
```



```
samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$
```

Logic

Cache HIT - When requested data is found in the cache
Cache MISS - When requested data is not found in the cache

Caches are of 3 types

- Direct Mapped Cache
- Fully Associative Cache
- Set Associative Cache

Write Policy - Write Through (Hence Dirty Bit is always 0)

Replacement Policy - LRU (Least Recently Used)

Direct Mapped Cache

Definitions
Tag - distinguishes one cache memory block from another
Index - identifies the cache block
Offset - points to desired data in the cache block

Instruction Breakdown
Each of the address of read or write instruction is broken into three parts: Tag, Index and Offset

Tag bits = Address bits - Index bits - Offset bits
Index bits = $\log_2$(No of Cache Lines)
Offset bits = $\log_2$(Block Size in bytes)

```
write  18 43
Address bits: 32

Instruction Breakdown
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000001     00              10
Index in decimal:  0
```

The requested address is converted to binary and is broken down into tag, index and offset. Then the cache table with the corresponding index is examined. If valid bit of the index is 0, cache miss is obtained. Else tag bit of the requested address is compared with tag bit in cache

table. If tag is a match then cache hit is obtained else cache miss is obtained. When cache miss occurs the cache table will be updated with the new dataset.

Fully Associative Cache

Definitions
Tag - distinguishes one cache memory block from another
Offset - points to desired data in the cache block

Instruction Breakdown
Each of the address of read or write instruction is broken into two parts: Tag and Offset

Tag bits = Address bits - Offset bits
Offset bits = $\log_2$(Block Size in bytes)

```
read 4
Address bits: 32

Instruction Breakdown
Tag( 30 bits )          Offset( 2 bits )
00000000000000000000000000000001           00

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 1 with offset 0 to 3 is transferred to cache at index 0

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000001          BLOCK 1         0
1        0       -       0       0
2        0       -       0       0
3        0       -       0       0
lru:  [0]
```

The requested address is converted to binary and is broken down into tag and offset. The requested tag is then searched in the cache. If the tag matches with one of the entries in the cache table, cache hit is obtained. Else, cache miss obtained. When cache miss occurs the cache table will be updated with the new dataset. If the cache is full then it will place the new dataset at the least recently used block in the cache. The lru array has the least recently used cache block at 0th

index and it's most recently used cache block at the last index. Everytime a particular cache block is accessed it is removed from the lru array and placed at the last index in the array.

N-way Set Associative Cache

N-way set associative cache combines the idea of direct mapped cache and fully associative cache. It has direct mapped principle to an index, yet fully associative concept within the index. An index contains N blocks of way.

Definitions
Tag - distinguishes one cache memory block from another
Index - identifies the cache block
Offset - points to desired data in the cache block

Instruction Breakdown
Each of the address of read or write instruction is broken into three parts: Tag, Index and Offset

Total sets = No of Cache Lines/Set Size
Tag bits = Address bits - Index bits - Offset bits
Index bits = $\log_2$(Total sets)
Offset bits = $\log_2$(Block Size in bytes)

```
read 4
Address bits: 32

Instruction Breakdown
Tag( 29 bits )          Index( 1 bits )         Offset( 2 bits )
00000000000000000000000000000    1               00
Index in decimal:  1

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 1 with offset 0 to 3 is transferred to cache at index 1

------------------------------------------------------------------------
                          CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
  0       0       -                       0                       0
  0       0       -                       0                       0
  1       1       00000000000000000000000000000                    BLOCK 1
  1       0       -                       0                       0
lru: [[], [0]]
```

The requested address is converted to binary and is broken down into tag, index and offset. Then the cache table with the corresponding index is examined. N-ways of cache blocks is searched and if the tag matches with one of the entries, cache hit is obtained. Else, cache miss obtained. When cache miss occurs the cache table will be updated with the new dataset. If the cache is full then it will place the new dataset at the least recently used block in the corresponding index. The lru array is a 2d array. It's 0^{th} array corresponds to set 0 (Index 0 in the above picture) , it's 1^{st} array corresponds to set 1 (Index 1 in the above picture) and so on for all the sets in the cache. Each 1d array in the lru 2d array works the same way as it does in fully associative mapping.

Examples

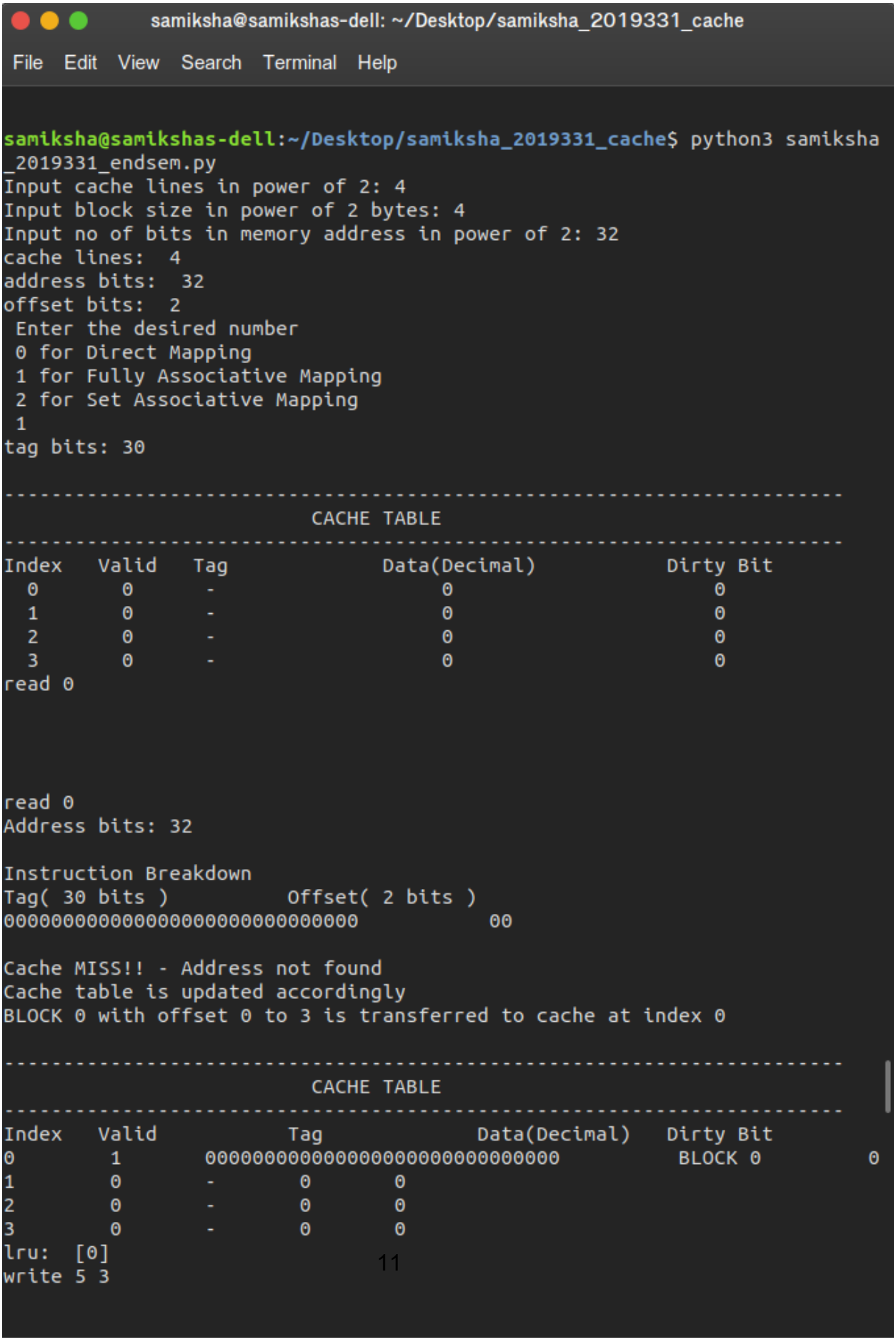
1. Direct Mapping

```
samiksha@samikshas-dell: ~/Desktop/samiksha_2019331_cache
File  Edit  View  Search  Terminal  Help

samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$ python3 samiksha
_2019331_endsem.py
Input cache lines in power of 2: 4
Input block size in power of 2 bytes: 4
Input no of bits in memory address in power of 2: 32
cache lines:  4
address bits:  32
offset bits:  2
 Enter the desired number
 0 for Direct Mapping
 1 for Fully Associative Mapping
 2 for Set Associative Mapping
 0
index bits:  2
tag bits: 28

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid   Tag             Data(Decimal)          Dirty Bit
  0       0      -                    0                    0
  1       0      -                    0                    0
  2       0      -                    0                    0
  3       0      -                    0                    0
read 0




read 0
Address bits: 32

Instruction Breakdown
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000     00              00
Index in decimal:  0

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 0 with offset 0 to 3 is transferred to cache at index 0

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000    BLOCK 0         0
1        0       -       0       0
2        0       -       0       0
3        0       -       0       0
write 1 3
```

```
write  1 3
Address bits: 32

Instruction Breakdown
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000000     00              01
Index in decimal:  0

Cache HIT!
Content is updated based on Write Policy

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000    BLOCK 0         0
1        0       -       0       0
2        0       -       0       0
3        0       -       0       0
read 5




read 5
Address bits: 32

Instruction Breakdown
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000000     01              01
Index in decimal:  1

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 1 with offset 0 to 3 is transferred to cache at index 1

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000    BLOCK 0         0
1        1       00000000000000000000000000    BLOCK 1         0
2        0       -       0       0
3        0       -       0       0
read 18
```

```
read 18
Address bits: 32

Instruction Breakdown
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000001     00              10
Index in decimal:  0

Cache MISS!! - Address not found
Cache table is updated accordingly
Address 18 will replace the block at index 0
BLOCK 4 with offset 0 to 3 is transferred to cache at index 0

------------------------------------------------------------------------
                          CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000001    BLOCK 4         0
1        1       00000000000000000000000000000    BLOCK 1         0
2        0       -       0       0
3        0       -       0       0
quit



samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$
```

2. Fully Associative Mapping


samiksha@samikshas-dell: ~/Desktop/samiksha_2019331_cache
File Edit View Search Terminal Help
samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$ python3 samiksha_2019331_endsem.py
Input cache lines in power of 2: 4
Input block size in power of 2 bytes: 4
Input no of bits in memory address in power of 2: 32
cache lines: 4
address bits: 32
offset bits: 2
Enter the desired number
0 for Direct Mapping
1 for Fully Associative Mapping
2 for Set Associative Mapping
1
tag bits: 30
CACHE TABLE
Index Valid Tag Data(Decimal) Dirty Bit
0 0 - 0 0
1 0 - 0 0
2 0 - 0 0
3 0 - 0 0
read 0
read 0
Address bits: 32
Instruction Breakdown
Tag(30 bits) Offset(2 bits)
00000000000000000000000000000000 00
Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 0 with offset 0 to 3 is transferred to cache at index 0
CACHE TABLE
Index Valid Tag Data(Decimal) Dirty Bit
0 1 00000000000000000000000000000000 BLOCK 0 0
1 0 - 0 0
2 0 - 0 0
3 0 - 0 0
lru: [0]
write 5 3

```
write 5 3
Address bits: 32

Instruction Breakdown
Tag( 30 bits )          Offset( 2 bits )
00000000000000000000000000000001          01

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 1 with offset 0 to 3 is transferred to cache at index 1
Content is updated based on Write Policy

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000000          BLOCK 0         0
1        1       00000000000000000000000000000001          BLOCK 1         0
2        0       -       0       0
3        0       -       0       0
lru:  [0, 1]
write 8




Invalid input
write 8 4




write 8 4
Address bits: 32

Instruction Breakdown
Tag( 30 bits )          Offset( 2 bits )
00000000000000000000000000000010          00

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 2 with offset 0 to 3 is transferred to cache at index 2
Content is updated based on Write Policy

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000000          BLOCK 0         0
1        1       00000000000000000000000000000001          BLOCK 1         0
2        1       00000000000000000000000000000010          BLOCK 2         0
3        0       -       0       0
lru:  [0, 1, 2]
read 12
```

```
read 12
Address bits: 32

Instruction Breakdown
Tag( 30 bits )          Offset( 2 bits )
00000000000000000000000000000011          00

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 3 with offset 0 to 3 is transferred to cache at index 3

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000000          BLOCK 0         0
1        1       00000000000000000000000000000001          BLOCK 1         0
2        1       00000000000000000000000000000010          BLOCK 2         0
3        1       00000000000000000000000000000011          BLOCK 3         0
lru:  [0, 1, 2, 3]
read 1




read 1
Address bits: 32

Instruction Breakdown
Tag( 30 bits )          Offset( 2 bits )
00000000000000000000000000000000          01

Cache HIT! at index 0

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000000          BLOCK 0         0
1        1       00000000000000000000000000000001          BLOCK 1         0
2        1       00000000000000000000000000000010          BLOCK 2         0
3        1       00000000000000000000000000000011          BLOCK 3         0
lru:  [1, 2, 3, 0]
read 30
```

```
read 30
Address bits: 32

Instruction Breakdown
Tag( 30 bits )          Offset( 2 bits )
00000000000000000000000000000111           10

Cache MISS!! - Address not found
Cache table is updated accordingly
Address 30 will replace the block at index 1
BLOCK 7 with offset 0 to 3 is transferred to cache at index 1

------------------------------------------------------------------------
                         CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0         1       00000000000000000000000000000000          BLOCK 0         0
1         1       00000000000000000000000000000111          BLOCK 7         0
2         1       00000000000000000000000000000010          BLOCK 2         0
3         1       00000000000000000000000000000011          BLOCK 3         0
lru:  [2, 3, 0, 1]
quit



samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$
```

3. 2-Way Set Associative Mapping

```
samiksha@samikshas-dell: ~/Desktop/samiksha_2019331_cache
File  Edit  View  Search  Terminal  Help

samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$ python3 samiksha_2019331_e
ndsem.py
Input cache lines in power of 2: 4
Input block size in power of 2 bytes: 4
Input no of bits in memory address in power of 2: 32
cache lines:  4
address bits:  32
offset bits:  2
 Enter the desired number
 0 for Direct Mapping
 1 for Fully Associative Mapping
 2 for Set Associative Mapping
 2
Input set size: 2
total sets:  2
set bits:  1
tag bits: 29

--------------------------------------------------------------------------
                         CACHE TABLE
--------------------------------------------------------------------------
Index   Valid   Tag             Data(Decimal)          Dirty Bit
  0       0      -                    0                    0
  0       0      -                    0                    0
  1       0      -                    0                    0
  1       0      -                    0                    0
read 0



read 0
Address bits: 32

Instruction Breakdown
Tag( 29 bits )          Index( 1 bits )         Offset( 2 bits )
00000000000000000000000000000    0               00
Index in decimal:  0

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 0 with offset 0 to 3 is transferred to cache at index 0

--------------------------------------------------------------------------
                         CACHE TABLE
--------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
  0       1      00000000000000000000000000000               BLOCK 0         0
  0       0      -                    0                    0
  1       0      -                    0                    0
  1       0      -                    0                    0
lru: [[0], []]
write 5 3
```

```
write 5 3
Address bits: 32

Instruction Breakdown
Tag( 29 bits )          Index( 1 bits )         Offset( 2 bits )
00000000000000000000000000000    1               01
Index in decimal:  1

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 1 with offset 0 to 3 is transferred to cache at index 1
Content is updated based on Write Policy

------------------------------------------------------------------------
                        CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
  0       1      00000000000000000000000000000               BLOCK 0         0
  0       0      -                   0                   0
  1       1      00000000000000000000000000000               BLOCK 1         0
  1       0      -                   0                   0
lru: [[0], [0]]
write 8 4



write 8 4
Address bits: 32

Instruction Breakdown
Tag( 29 bits )          Index( 1 bits )         Offset( 2 bits )
00000000000000000000000000001    0               00
Index in decimal:  0

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 2 with offset 0 to 3 is transferred to cache at index 0
Content is updated based on Write Policy

------------------------------------------------------------------------
                        CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
  0       1      00000000000000000000000000000               BLOCK 0         0
  0       1      00000000000000000000000000001               BLOCK 2         0
  1       1      00000000000000000000000000000               BLOCK 1         0
  1       0      -                   0                   0
lru: [[0, 1], [0]]
read 12
```

```
read 12
Address bits: 32

Instruction Breakdown
Tag( 29 bits )          Index( 1 bits )         Offset( 2 bits )
00000000000000000000000000001    1               00
Index in decimal:  1

Cache MISS!! - Address not found
Cache table is updated accordingly
BLOCK 3 with offset 0 to 3 is transferred to cache at index 1

------------------------------------------------------------------------
                        CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
  0       1      00000000000000000000000000000               BLOCK 0         0
  0       1      00000000000000000000000000001               BLOCK 2         0
  1       1      00000000000000000000000000000               BLOCK 1         0
  1       1      00000000000000000000000000001               BLOCK 3         0
lru: [[0, 1], [0, 1]]
read 1




read 1
Address bits: 32

Instruction Breakdown
Tag( 29 bits )          Index( 1 bits )         Offset( 2 bits )
00000000000000000000000000000    0               01
Index in decimal:  0

Cache HIT! at index 0

------------------------------------------------------------------------
                        CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
  0       1      00000000000000000000000000000               BLOCK 0             0
  0       1      00000000000000000000000000001               BLOCK 2             0
  1       1      00000000000000000000000000000               BLOCK 1             0
  1       1      00000000000000000000000000001               BLOCK 3             0
lru: [[1, 0], [0, 1]]
read 30
```

```
read 30
Address bits: 32

Instruction Breakdown
Tag( 29 bits )          Index( 1 bits )         Offset( 2 bits )
00000000000000000000000000011    1               10
Index in decimal:  1

Cache MISS!! - Address not found
Cache table is updated accordingly
Address 30 will replace the block at index 1
BLOCK 7 with offset 0 to 3 is transferred to cache at index 1

-------------------------------------------------------------------------
                        CACHE TABLE
-------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
  0       1      00000000000000000000000000000              BLOCK 0              0
  0       1      00000000000000000000000000001              BLOCK 2              0
  1       1      00000000000000000000000000011              BLOCK 7              0
  1       1      00000000000000000000000000001              BLOCK 3              0
lru: [[1, 0], [1, 0]]
quit



samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache$
```

References

- https://www.ntu.edu.sg/home/smitha/ParaCache/Paracache/kb.pdf
- https://www.ntu.edu.sg/home/smitha/ParaCache/Paracache/dmc.html
- http://web.cecs.pdx.edu/~zeshan/cache_solved_examples.pdf
- https://www.gatevidyalay.com/set-associative-mapping-practice-problems/
- Computer Organisation and Architecture - Smruti Sarangi

Cache Assignment (Bonus)

The cache is a two level cache. The word length of the machine is 32 bits by default but you can change it while giving input. The assumption is that the size of each memory word is 1 byte. All the inputs have to be in the power of 2. The write policy used is Write Through (WT) and the replacement policy used is Least Recently Used (LRU). The address is input in decimal (base 10). The size of the level 1 cache is S/2 and the size of level 2 cache is S.

Usage

- Open the terminal and navigate to the directory where the files are saved.
- Run samiksha_2019331_bonus.py (for bonus assignment)
  Make sure all the 4 files samiksha_2019331_bonus.py,direct.py,associative.py, kassociative.py are in the same folder
- Enter the input as instructed
- The program is written in Python 3.6.9

Input

- Cache Lines in Level 1 - in the power of 2
- Block Size - in the power of 2 bytes
- Address Bits - 32 bits by default but can be 16,32,64 bits based on user discretion
- Set Size - is n in the power of 2 for n-way set associative mapping
- Mapping
  - 0 - Direct
  - 1 - Associative
  - 2 - N-way Set Associative

Commands

- read address
- write address data
- quit - to exit the program

After every valid command, the entire cache will be printed. In case of a MISS, it will print address not found and will update the cache table with the new dataset. In case of a replacement, it will also print the address by which the block will be replaced.

Logic

Cache HIT - When requested data is found in the cache
Cache MISS - When requested data is not found in the cache

Caches are of 3 types

- Direct Mapped Cache
- Fully Associative Cache
- Set Associative Cache

Write Policy - Write Through (Hence Dirty Bit is always 0)

Replacement Policy - LRU (Least Recently Used)

Cache Inclusion Policy - NINE Policy [Link here]

Direct Mapped Cache

The requested address is converted to binary and is broken down into tag, index and offset. Then the level 1 cache with the corresponding index is examined. If valid bit of the index is 0, cache miss is obtained. Else tag bit of the requested address is compared with tag bit in level 1 cache. If tag is a match then cache hit is obtained else cache miss is obtained.

In case there is a cache miss at level 1, then we search level 2 cache in the same way. If there is a cache hit then we load the data from level 2 cache to level 1 cache. Else if there is a cache miss then both the cache levels are updated with the new dataset.

Fully Associative Cache

The requested address is converted to binary and is broken down into tag and offset. The requested tag is then searched in the cache. If the tag matches with one of the entries in the cache table, cache hit is obtained. Else, cache miss obtained. When cache miss occurs the cache table will be updated with the new dataset. If the cache is full then it will place the new dataset at the least recently used block in the cache. The lru array has the least recently used cache block at 0th

index and it's most recently used cache block at the last index. Everytime a particular cache block is accessed it is removed from the lru array and placed at the last index in the array.

In case there is a cache miss at level 1, then we search level 2 cache in the same way. If there is a cache hit then we load the data from level 2 cache to level 1 cache. Else if there is a cache miss then both the cache levels are updated with the new dataset.

N-way Set Associative Cache

The requested address is converted to binary and is broken down into tag, index and offset. Then the cache table with the corresponding index is examined. N-ways of cache blocks is searched and if the tag matches with one of the entries, cache hit is obtained. Else, cache miss obtained. When cache miss occurs the cache table will be updated with the new dataset. If the cache is full then it will place the new dataset at the least recently used block in the corresponding index. The lru array is a 2d array. It's 0^{th} array corresponds to set 0 (Index 0 in the above picture) , it's 1^{st} array corresponds to set 1 (Index 1 in the above picture) and so on for all the sets in the cache. Each 1d array in the lru 2d array works the same way as it does in fully associative mapping.

In case there is a cache miss at level 1, then we search level 2 cache in the same way. If there is a cache hit then we load the data from level 2 cache to level 1 cache. Else if there is a cache miss then both the cache levels are updated with the new dataset.

Example

Direct Mapping

```
samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache/Bonus$ python3 samiksha
_2019331_bonus.py
 Enter the desired number
 0 for Direct Mapping
 1 for Fully Associative Mapping
 2 for Set Associative Mapping
 0
Input cache lines in level 1 in power of 2: 4
Input block size in power of 2 bytes: 4
Input no of bits in memory address in power of 2: 32
cache lines level 1:  4
cache lines level 2:  8
address bits:  32
offset bits:  2
index bits level 1:  2
tag bits level 1: 28

---------------------------------------------------------------------
                    LEVEL 1 CACHE TABLE
---------------------------------------------------------------------
Index   Valid   Tag             Data(Decimal)          Dirty Bit
  0       0      -                    0                    0
  1       0      -                    0                    0
  2       0      -                    0                    0
  3       0      -                    0                    0
read 0
```

```
read 0
Address bits: 32

Instruction Breakdown Level 1
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000000     00              00
Index in decimal:  0

Cache MISS at Level 1!! - Address not found
Searching in Level 2

Instruction Breakdown Level 2
Tag( 27 bits )          Index( 3 bits )         Offset( 2 bits )
0000000000000000000000000000      000             00
Index in decimal:  0

Cache MISS at Level 2!! - Address not found
Both caches are updated accordingly
BLOCK 0 with offset 0 to 3 is transferred to cache Level 2 at index 0

------------------------------------------------------------------------
                  LEVEL 2 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000     BLOCK 0         0
1        0       -       0       0
2        0       -       0       0
3        0       -       0       0
4        0       -       0       0
5        0       -       0       0
6        0       -       0       0
7        0       -       0       0

BLOCK 0 with offset 0 to 3 is transferred to cache Level 1 at index 0

------------------------------------------------------------------------
                  LEVEL 1 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000    BLOCK 0         0
1        0       -       0       0
2        0       -       0       0
3        0       -       0       0
read 4
```

```
read 4
Address bits: 32

Instruction Breakdown Level 1
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000000     01              00
Index in decimal:  1

Cache MISS at Level 1!! - Address not found
Searching in Level 2

Instruction Breakdown Level 2
Tag( 27 bits )          Index( 3 bits )         Offset( 2 bits )
0000000000000000000000000000      001             00
Index in decimal:  1

Cache MISS at Level 2!! - Address not found
Both caches are updated accordingly
BLOCK 1 with offset 0 to 3 is transferred to cache Level 2 at index 1

-------------------------------------------------------------------------
                  LEVEL 2 CACHE TABLE
-------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000     BLOCK 0         0
1        1       00000000000000000000000000000     BLOCK 1         0
2        0       -       0       0
3        0       -       0       0
4        0       -       0       0
5        0       -       0       0
6        0       -       0       0
7        0       -       0       0

BLOCK 1 with offset 0 to 3 is transferred to cache Level 1 at index 1

-------------------------------------------------------------------------
                  LEVEL 1 CACHE TABLE
-------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000     BLOCK 0         0
1        1       00000000000000000000000000000     BLOCK 1         0
2        0       -       0       0
3        0       -       0       0
read 8
```

```
read 8
Address bits: 32

Instruction Breakdown Level 1
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000000     10              00
Index in decimal:  2

Cache MISS at Level 1!! - Address not found
Searching in Level 2

Instruction Breakdown Level 2
Tag( 27 bits )          Index( 3 bits )         Offset( 2 bits )
0000000000000000000000000000      010             00
Index in decimal:  2

Cache MISS at Level 2!! - Address not found
Both caches are updated accordingly
BLOCK 2 with offset 0 to 3 is transferred to cache Level 2 at index 2

------------------------------------------------------------------------
                  LEVEL 2 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000     BLOCK 0         0
1        1       00000000000000000000000000000     BLOCK 1         0
2        1       00000000000000000000000000000     BLOCK 2         0
3        0       -       0       0
4        0       -       0       0
5        0       -       0       0
6        0       -       0       0
7        0       -       0       0

BLOCK 2 with offset 0 to 3 is transferred to cache Level 1 at index 2

------------------------------------------------------------------------
                  LEVEL 1 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000    BLOCK 0         0
1        1       00000000000000000000000000000    BLOCK 1         0
2        1       00000000000000000000000000000    BLOCK 2         0
3        0       -       0       0
read 20
```

```
read 20
Address bits: 32

Instruction Breakdown Level 1
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000001     01              00
Index in decimal:  1

Cache MISS at Level 1!! - Address not found
Searching in Level 2

Instruction Breakdown Level 2
Tag( 27 bits )          Index( 3 bits )         Offset( 2 bits )
000000000000000000000000000       101             00
Index in decimal:  5

Cache MISS at Level 2!! - Address not found
Both caches are updated accordingly
BLOCK 5 with offset 0 to 3 is transferred to cache Level 2 at index 5

------------------------------------------------------------------------
                  LEVEL 2 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000     BLOCK 0         0
1        1       00000000000000000000000000     BLOCK 1         0
2        1       00000000000000000000000000     BLOCK 2         0
3        0       -       0       0
4        0       -       0       0
5        1       00000000000000000000000000     BLOCK 5         0
6        0       -       0       0
7        0       -       0       0

Address 20 will replace the block at index 1 in Level 1

BLOCK 5 with offset 0 to 3 is transferred to cache Level 1 at index 1

------------------------------------------------------------------------
                  LEVEL 1 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000    BLOCK 0         0
1        1       00000000000000000000000001    BLOCK 5         0
2        1       00000000000000000000000000    BLOCK 2         0
3        0       -       0       0
read 4
```

```
read 4
Address bits: 32

Instruction Breakdown Level 1
Tag( 28 bits )          Index( 2 bits )         Offset( 2 bits )
00000000000000000000000000000     01              00
Index in decimal:  1

Cache MISS at Level 1!! - Address not found
Searching in Level 2

Instruction Breakdown Level 2
Tag( 27 bits )          Index( 3 bits )         Offset( 2 bits )
0000000000000000000000000000      001             00
Index in decimal:  1

Cache HIT at Level 2! Data is loaded to Cache Level 1

------------------------------------------------------------------------
                  LEVEL 2 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000     BLOCK 0         0
1        1       00000000000000000000000000000     BLOCK 1         0
2        1       00000000000000000000000000000     BLOCK 2         0
3        0       -       0       0
4        0       -       0       0
5        1       00000000000000000000000000000     BLOCK 5         0
6        0       -       0       0
7        0       -       0       0

Address 4 will replace the block at index 1 in Level 1

BLOCK 1 with offset 0 to 3 is transferred to cache Level 1 at index 1

------------------------------------------------------------------------
                  LEVEL 1 CACHE TABLE
------------------------------------------------------------------------
Index   Valid           Tag             Data(Decimal)   Dirty Bit
0        1       00000000000000000000000000000    BLOCK 0         0
1        1       00000000000000000000000000000    BLOCK 1         0
2        1       00000000000000000000000000000    BLOCK 2         0
3        0       -       0       0
quit



samiksha@samikshas-dell:~/Desktop/samiksha_2019331_cache/Bonus$
```

Fully Associative and N-way Set Associative Mapping can be similarly run on my program.