|  |  |  |
| --- | --- | --- |
| **Course No: CS F342** | **TUTORIAL 12** | **Course Title: Computer Architecture** |

1. Consider a direct mapped cache with eight bytes, a main memory of 32 bytes and block size of 1 byte.

Following byte addresses are reference by a program.

13,17, 26,13,17,25

Write the state of the cache after each address is referenced for the following cache mappings.

1. Direct mapped cache

b. Fully associative cache (LRU replacement)

Assume that for each mapping the cache is started with valid/invalid bit set to ‘N’ indicating none of the memory content present initially are valid. Also compute the total number of misses in each case

ANS:

1. Direct mapped cache (Tag, Line no, Block/Line/Cache offset)

Given: MM size = 32 bytes, Cache size = 8 bytes, Block size = 1byte

Block offset => 1 byte => 20 => 0 bits (Block offset required bits)

Line no => No of lines = 8 => 23 => 3 bits (Line no required bits)

Tag = 5 – (3+0) = 2 bits

|  |  |  |
| --- | --- | --- |
| Tag = 2 bits | Line no = 3 bits | Block offset = 0 bits |

Initial state of cache

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | V | Hit/Miss | Tag | Data |
| 000 | N |  |  |  |
| 001 | N |  |  |  |
| 010 | N |  |  |  |
| 011 | N |  |  |  |
| 100 | N |  |  |  |
| 101 | N |  |  |  |
| 110 | N |  |  |  |
| 111 | N |  |  |  |

An address X maps to the direct-mapped cache block X modulo 8

After handling a miss of address 13 (01**101**)2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | V | Hit/Miss | Tag | Data |
| 000 | N |  |  |  |
| 001 | N |  |  |  |
| 010 | N |  |  |  |
| 011 | N |  |  |  |
| 100 | N |  |  |  |
| 101 | Y | Miss | (01)2 | Memory(01101)2 |
| 110 | N |  |  |  |
| 111 | N |  |  |  |

After handling a miss of address 17 (10**001**)2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | V | Hit/Miss | Tag | Data |
| 000 | N |  |  |  |
| 001 | Y | Miss | (10)2 | Memory(10001)2 |
| 010 | N |  |  |  |
| 011 | N |  |  |  |
| 100 | N |  |  |  |
| 101 | Y |  | (01)2 | Memory(01101)2 |
| 110 | N |  |  |  |
| 111 | N |  |  |  |

After handling a miss of address 26 (11**010**)2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | V | Hit/Miss | Tag | Data |
| 000 | N |  |  |  |
| 001 | Y |  | (10)2 | Memory(10001)2 |
| 010 | Y | Miss | (11)2 | Memory(11010)2 |
| 011 | N |  |  |  |
| 100 | N |  |  |  |
| 101 | Y |  | (01)2 | Memory(01101)2 |
| 110 | N |  |  |  |
| 111 | N |  |  |  |

After handling a hit of address 13 (01**101**)2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | V | Hit/Miss | Tag | Data |
| 000 | N |  |  |  |
| 001 | Y |  | (10)2 | Memory(10001)2 |
| 010 | N |  | (11)2 | Memory(11010)2 |
| 011 | N |  |  |  |
| 100 | N |  |  |  |
| 101 | Y | Hit | (01)2 | Memory(01101)2 |
| 110 | N |  |  |  |
| 111 | N |  |  |  |

After handling a hit of address 17 (10**001**)2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | V | Hit/Miss | Tag | Data |
| 000 | N |  |  |  |
| 001 | Y | Hit | (10)2 | Memory(10001)2 |
| 010 | N |  | (11)2 | Memory(11010)2 |
| 011 | N |  |  |  |
| 100 | N |  |  |  |
| 101 | Y |  | (01)2 | Memory(01101)2 |
| 110 | N |  |  |  |
| 111 | N |  |  |  |

After handling a miss of address 25 (11**001**)2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | V | Hit/Miss | Tag | Data |
| 000 | N |  |  |  |
| 001 | Y | miss | (11)2 | Memory(11001)2 |
| 010 | N |  | (11)2 | Memory(11010)2 |
| 011 | N |  |  |  |
| 100 | N |  |  |  |
| 101 | Y |  | (01)2 | Memory(01101)2 |
| 110 | N |  |  |  |
| 111 | N |  |  |  |

1. fully associative cache (Tag, Block offset)

Block offset => 1 byte => 20 => 0 bits (Block offset required bits)

Tag = 5 – (0) = 5 bits

|  |  |
| --- | --- |
| Tag = 5 bits | Block offset = 0 bits |

Memory reference: 13,17, 26,13,17, 25

Initial state of cache

|  |  |  |  |
| --- | --- | --- | --- |
| V | Hit/Miss | Tag | Data |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |

After handling a miss of address 13 (01101)2

|  |  |  |  |
| --- | --- | --- | --- |
| V | Hit/Miss | Tag | Data |
| Y | Miss | (01101)2 | Memory(01101)2 |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| Y |  |  |  |
| N |  |  |  |
| N |  |  |  |

After handling a miss of address 17 (10001)2

|  |  |  |  |
| --- | --- | --- | --- |
| V | Hit/Miss | Tag | Data |
| Y |  | (01101)2 | Memory(01101)2 |
| Y | Miss | (10001)2 | Memory(10001)2 |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |

After handling a miss of address 26 (11010)2

|  |  |  |  |
| --- | --- | --- | --- |
| V | Hit/Miss | Tag | Data |
| Y |  | (01101)2 | Memory(01101)2 |
| Y |  | (10001)2 | Memory(10001)2 |
| Y | Miss | (11010)2 | Memory(11010)2 |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |

After handling a hit of address 13 (01101)2

|  |  |  |  |
| --- | --- | --- | --- |
| V | Hit/Miss | Tag | Data |
| Y | Hit | (01101)2 | Memory(01101)2 |
| Y |  | (10001)2 | Memory(10001)2 |
| Y |  | (11010)2 | Memory(11010)2 |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |

After handling a hit of address 17 (10001)2

|  |  |  |  |
| --- | --- | --- | --- |
| V | Hit/Miss | Tag | Data |
| Y |  | (01101)2 | Memory(01101)2 |
| Y | Hit | (10001)2 | Memory(10001)2 |
| Y |  | (11010)2 | Memory(11010)2 |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |

After handling a miss of address 25 (11001)2

|  |  |  |  |
| --- | --- | --- | --- |
| V | Hit/Miss | Tag | Data |
| Y |  | (01101)2 | Memory(01101)2 |
| Y |  | (10001)2 | Memory(10001)2 |
| Y |  | (11010)2 | Memory(11010)2 |
| Y | miss | (11001)2 | Memory(11001)2 |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |
| N |  |  |  |

2. A computer system has a main memory size of 16MB, and direct mapped cache size of 64KB. Assume that block size is 4 words and 4bytes/word. For the following byte memory references indicate in decimal which block number in the cache the reference maps to, and whether is a hit or not. Assume initially all cache entries are invalid.

0x010035, 0x020048, 0x010038, 0x020145

**Ans:**

Given,

MM size = 16 MB

Mapping function = Direct mapping (Tag, Line no, Block/Cache offset)

Cache Size = 64 KB

Block Size = 4 words

1 word = 4 bytes

Address subdivision:

MM address size => 16 MB => 224 => 24 bits MM address

Block/Cache offset => 4 words =>

Size of block in bytes = Block size \* word size = 4\*4 = 16 bytes => 24

No of bits required for Block/Cache offset = 4 bits

Line no =>

Number of lines (blocks) in a Cache = Cache size (bytes) / block size (bytes)

= (64 \* 1024) / 16

= 4096 => 212

No of bits require for line no = 12 bits

Tag =MM address size – (Line no + block/cache offset) = 24 – (12+4) = 8 bits

Address subdivision

|  |  |  |
| --- | --- | --- |
| Tag = **8** bits | Index or cache line no = **12** bits | Cache offset = 4 bits |

|  |  |  |  |
| --- | --- | --- | --- |
| Byte Address | Cache line no | Cache line number in decimal | Hit/Miss |
| 0x01**003**5 | 0x**003** | 3 | Miss |
| 0x02**004**8 | 0x**004** | 4 | Miss |
| 0x01**003**8 | 0x**003** | 3 | Hit |
| 0x02**014**5 | 0x**014** | 20 | Miss |