**Homework 1**

**ECE 486/586 Spring 2020**

Name: Griffin Mack Student Number: 11713813

**DUE: April 14, 2020 at the beginning of class (2:00 p.m.). This is an individual assignment.**

Consider the characteristics of a memory system given below.

Cache size = 512 bytes

Main memory size = 8192 bytes

Cache line/block size = 32 bytes

Replacement policy = least recently used

Write policy = write-back

Assume the cache is initially “empty” meaning that no cache blocks are considered valid at the start of the program.

Assume a program to be executed on this system generates the following sequence of main memory byte addresses: 0 (write), 1011 (read), 96 (read), 8191 (read), 992 (read), 736 (write), 767 (read).

**1)** What is the best possible hit rate for this sequence of addresses? (5 points)

W 0

R 1011

R 96

R 8191

R 992

W 736

R 767

2/7 = **28.57%**

**2)** For a direct-mapped cache configuration:

**2A)** Complete a table showing all address fields, mm block numbers, cm set numbers, cm block numbers, and hit/miss status. (10 points)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| mm address | tag  (binary) | index  (binary) | offset  (binary) | mm blk # (/ 32) | cm set # (%16) | cm blk # | hit/miss |
| 0 | 0000 | 0000 | 00000 | 0 | 0 | 0 | Miss |
| 1011 | 0001 | 1111 | 10011 | 31 | 15 | 15 | Miss |
| 96 | 0000 | 0011 | 00000 | 3 | 3 | 3 | Miss |
| 8191 | 1111 | 1111 | 11111 | 255 | 15 | 15 | Miss |
| 992 | 0001 | 1111 | 00000 | 31 | 15 | 15 | Miss |
| 736 | 0001 | 0111 | 00000 | 23 | 7 | 7 | Miss |
| 767 | 0001 | 0111 | 11111 | 23 | 7 | 7 | Hit |

**2B)** What is the actual hit rate? (5 points)

1/7 = **14.2857%**

**2C)** Calculate the total size of the cache. (5 points)

Cache Data Size + Valid Byte + Dirty Bit + Tag Bytes

= 512 + 2 + 2 + (4\*16)/8

**= 524 Bytes**

**2D)** Draw the cache indicating all its contents as it would appear AFTER the sequence of memory references is completed. (10 points)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Valid Bit | Dirty Bit | Tag | Data | CM Set # | CM Block # |
| 1 | 1 | 0000 | MM0 | 0 | 0 |
| 0 | 0 | xxxx | X | 1 | 1 |
| 0 | 0 | xxxx | X | 2 | 2 |
| 1 | 0 | 0000 | MM3 | 3 | 3 |
| 0 | 0 | xxxx | X | 4 | 4 |
| 0 | 0 | xxxx | X | 5 | 5 |
| 0 | 0 | xxxx | X | 6 | 6 |
| 1 | 1 | 0001 | MM23 | 7 | 7 |
| 0 | 0 | xxxx | X | 8 | 8 |
| 0 | 0 | xxxx | X | 9 | 9 |
| 0 | 0 | xxxx | X | 10 | 10 |
| 0 | 0 | xxxx | X | 11 | 11 |
| 0 | 0 | xxxx | X | 12 | 12 |
| 0 | 0 | xxxx | X | 13 | 13 |
| 0 | 0 | xxxx | X | 14 | 14 |
| 1 | 0 | 0001 | MM31 | 15 | 15 |

**3)** For a 2-way set-associative cache configuration:

**3A)** Complete a table showing all address fields, mm block numbers, cm set numbers, cm block numbers, and hit/miss status. (10 points)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| mm address | tag  (binary) | index  (binary) | offset  (binary) | mm blk # (/ 32) | cm set # (%8) | cm blk # | hit/miss |
| 0 | 00000 | 000 | 00000 | 0 | 0 | 0,1 | Miss |
| 1011 | 00011 | 111 | 10011 | 31 | 7 | 14,15 | Miss |
| 96 | 00000 | 011 | 00000 | 3 | 3 | 6,7 | Miss |
| 8191 | 11111 | 111 | 11111 | 255 | 7 | 14,15 | Miss |
| 992 | 00011 | 111 | 00000 | 31 | 7 | 14,15 | Hit |
| 736 | 00010 | 111 | 00000 | 23 | 7 | 14,15 | Miss |
| 767 | 00010 | 111 | 11111 | 23 | 7 | 14,15 | Hit |

**3B)** What is the actual hit rate? (5 points)

2/7 = **28.5714%**

**3C)** Calculate the total size of the cache. (5 points)

Cache Data Size + Valid Bytes + Dirty Bytes + Tag Bytes

= 512 + 2 + 2 + (16\*5)/8

**= 526 Bytes**

**3D)**  Draw the cache indicating all its contents as it would appear AFTER the sequence of memory references is completed. (10 points)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Valid Bit | Dirty Bit | Tag | Data | CM Set # | CM Block # |
| 1 | 1 | 00000 | MM0 | 0 | 0 |
| 0 | 0 | xxxx | X | 1 |
| 0 | 0 | xxxx | X | 1 | 2 |
| 1 | 0 | xxxx | X | 3 |
| 0 | 0 | xxxx | X | 2 | 4 |
| 0 | 0 | xxxx | X | 5 |
| 1 | 0 | 00000 | MM3 | 3 | 6 |
| 0 | 0 | xxxx | X | 7 |
| 0 | 0 | xxxx | X | 4 | 8 |
| 0 | 0 | xxxx | X | 9 |
| 0 | 0 | xxxx | X | 5 | 10 |
| 0 | 0 | xxxx | X | 11 |
| 0 | 0 | xxxx | X | 6 | 12 |
| 0 | 0 | xxxx | X | 13 |
| 1 | 0 | 00011 | MM31 | 7 | 14 |
| 1 | 1 | 00010 | MM23 | 15 |

**4)** For a 8-way set-associative cache configuration:

**4A)** Complete a table showing all address fields, mm block numbers, cm set numbers, cm block numbers, and hit/miss status. (10 points)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| mm address | tag  (binary) | index  (binary) | offset  (binary) | mm blk # (/ 32) | cm set # (%2) | cm blk # | hit/miss |
| 0 | 0000000 | 0 | 00000 | 0 | 0 | 0,1,2,3,4,5,6,7 | Miss |
| 1011 | 0001111 | 1 | 10011 | 31 | 1 | 8,9,10,11,12,13,14,15 | Miss |
| 96 | 0000001 | 1 | 00000 | 3 | 1 | 8,9,10,11,12,13,14,15 | Miss |
| 8191 | 1111111 | 1 | 11111 | 255 | 1 | 8,9,10,11,12,13,14,15 | Miss |
| 992 | 0001111 | 1 | 00000 | 31 | 1 | 8,9,10,11,12,13,14,15 | Hit |
| 736 | 0001011 | 1 | 00000 | 23 | 1 | 8,9,10,11,12,13,14,15 | Miss |
| 767 | 0001011 | 1 | 11111 | 23 | 1 | 8,9,10,11,12,13,14,15 | Hit |

**4B)** What is the actual hit rate? (5 points)

2/7 = **28.5714%**

**4C)** Calculate the total size of the cache. (5 points)

Cache Data Size + Valid Bytes + Dirty Bytes + Tag Bytes

= 512 + 2 + 2 + (16\*7)/8

**= 530 Bytes**

**4D)** Draw the cache indicating all its contents as it would appear AFTER the sequence of memory references is completed. (10 points)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Valid Bit | Dirty Bit | Tag | Data | CM Set # | CM Block # |
| 1 | 1 | 0000000 | MM0 | 0 | 0 |
| 0 | 0 | xxxxxxx | X | 1 |
| 0 | 0 | xxxxxxx | X | 2 |
| 0 | 0 | xxxxxxx | X | 3 |
| 0 | 0 | xxxxxxx | X | 4 |
| 0 | 0 | xxxxxxx | X | 5 |
| 0 | 0 | xxxxxxx | X | 6 |
| 0 | 0 | xxxxxxx | X | 7 |
| 1 | 0 | 0001111 | MM31 | 1 | 8 |
| 1 | 0 | 0000001 | MM3 | 9 |
| 1 | 0 | 1111111 | MM255 | 10 |
| 1 | 1 | 0001011 | MM23 | 11 |
| 0 | 0 | xxxxxxx | X | 12 |
| 0 | 0 | xxxxxxx | X | 13 |
| 0 | 0 | xxxxxxx | X | 14 |
| 0 | 0 | xxxxxxx | X | 15 |