Problem # 1

A computer system uses 16-bit memory addresses. It has a 2K-byte cache organized in a direct-mapped manner with 64 bytes per cache block. Assume that the size of each memory word is 1 byte.

(a) Calculate the number of bits in each of the Tag, Block, and Word fields of the memory address.
(b) When a program is executed, the processor reads data sequentially from the following word addresses:

128, 144, 2176, 2180, 128, 2176

All the above addresses are shown in decimal values. Assume that the cache is initially empty. For each of the above addresses, indicate whether the cache access will result in a hit or a miss.

Solution

(a) Block size = 64 bytes = 2^6 bytes = 2^6 words (since 1 word = 1 byte)
Therefore, **Number of bits in the *Word* field = 6**
Cache size = 2K-byte = 2^{11} bytes
Number of cache blocks = Cache size / Block size = $2^{11}/2^6 = 2^5$
Therefore, **Number of bits in the *Block* field = 5**
Total number of address bits = 16
Therefore, **Number of bits in the *Tag* field = 16 - 6 - 5 = 5**
For a given 16-bit address, the 5 most significant bits, represent the *Tag*, the next 5 bits represent the *Block*, and the 6 least significant bits represent the *Word*.

(b) The cache is initially empty. Therefore, all the cache blocks are invalid.

Access # 1:
Address = $(128)_{10} = (0000000010000000)_2$
(**Note:** Address is shown as a 16-bit number, because the computer uses 16-bit addresses)
For this address, *Tag* = 00000, *Block* = 00010, *Word* = 000000
Since the cache is empty before this access, this will be a cache **miss**
After this access, **Tag field for cache block 00010 is set to 00000**

Access # 2:
Address = $(144)_{10} = (0000000010010000)_2$
For this address, *Tag* = **00000**, *Block* = **00010**, *Word* = 010000
Since tag field for cache block **00010** is **00000** before this access, this will be a cache **hit** (because address tag = block tag)

Access # 3:
Address = $(2176)_{10} = (0000100010000000)_2$
For this address, *Tag* = **00001**, *Block* = **00010**, *Word* = 000000
Since tag field for cache block **00010** is **00000** before this access, this will be a cache **miss** (address tag ≠ block tag)
After this access, **Tag field for cache block 00010 is set to 00001**

Access # 4:
Address = $(2180)_{10}$ = $(0000100010000100)_2$
For this address, *Tag* = **00001**, *Block* = **00010**, *Word* = 000100
Since tag field for cache block **00010** is **00001** before this access, this will be a cache **hit** (address tag = block tag)

Access # 5:
Address = $(128)_{10}$ = $(0000000010000000)_2$
For this address, *Tag* = **00000**, *Block* = **00010**, *Word* = 000000
Since tag field for cache block **00010** is **00001** before this access, this will be a cache **miss** (address tag ≠ block tag)
After this access, **Tag field for cache block 00010 is set to 00000**

Access # 6:
Address = $(2176)_{10}$ = $(0000100010000000)_2$
For this address, *Tag* = **00001**, *Block* = **00010**, *Word* = 000000
Since tag field for cache block **00010** is **00001** before this access, this will be a cache **miss** (address tag ≠ block tag)
After this access, **Tag field for cache block 00010 is set to 00001**

Cache hit rate = Number of hits / Number of accesses = 2/6 = 0.333

Problem # 2

Repeat Problem # 1, if the cache is organized as a 2-way set-associative cache that uses the LRU replacement algorithm.

Solution

(a) Block size = 64 bytes = 2^6 bytes = 2^6 words
Therefore, **Number of bits in the *Word* field = 6**
Cache size = 2K-byte = 2^{11} bytes
Number of cache blocks per set = 2
Number of sets = Cache size / (Block size * Number of blocks per set) = $2^{11}/(2^6 * 2) = 2^4$
Therefore, **Number of bits in the *Set* field = 4**
Total number of address bits = 16
Therefore, **Number of bits in the *Tag* field = 16 - 6 - 4 = 6**

(b) The cache is initially empty. Therefore, all the cache blocks are invalid.

Access # 1:
Address = $(128)_{10}$ = $(0000000010000000)_2$
For this address, *Tag* = 000000, *Set* = 0010, *Word* = 000000
Since the cache is empty before this access, this will be a cache **miss**
After this access, **Tag field for the first block in set 0010 is set to 000000**

Access # 2:

Address = $(144)_{10}$ = $(0000000010010000)_2$

For this address, *Tag* = **000000**, *Set* = **0010**, *Word* = 010000

The tag field for this address matches the tag field for the first block in set 0010. Therefore, this access will be a cache **hit**.

Access # 3:

Address = $(2176)_{10}$ = $(0000100010000000)_2$

For this address, *Tag* = **000010**, *Set*= **0010**, *Word* = 000000

The tag field for this address does not match the tag field for the first block in set 0010. The second block in set 0010 is empty. Therefore, this access will be a cache **miss**.

After this access, **Tag field for the second block in set 0010 is set to 000010**

Access # 4:

Address = $(2180)_{10}$ = $(0000100010000100)_2$

For this address, *Tag* = **000010**, *Set* = **0010**, *Word* = 000100

The tag field for this address matches the tag field for the second block in set 0010. Therefore, this access will be a cache **hit**.

Access # 5:

Address = $(128)_{10}$ = $(0000000010000000)_2$

For this address, *Tag* = **000000**, *Set* = **0010**, *Word* = 000000

The tag field for this address matches the tag field for the first block in set 0010. Therefore, this access will be a cache **hit**.

Access # 6:

Address = $(2176)_{10}$ = $(0000100010000000)_2$

For this address, *Tag* = **000010**, *Set* = **0010**, *Word* = 000000

The tag field for this address matches the tag field for the second block in set 0010. Therefore, this access will be a cache **hit**.

Cache hit rate = Number of hits / Number of accesses = 4/6 = 0.666