* If the valid bit is false
  + Copy the block into cache
  + Write data to block element
  + Set tag
  + Set dirty = true
  + “It is a miss”
* Else, the valid bit is true
  + The tag matches
    - Write data to the block
    - “It is a hit”
  + The tag DOES NOT match
    - Write the block that’s there to MM
    - Copy correct block to cache
    - Write data to block element
    - Set tag
    - Set dirty
    - “It is a miss”

Great question - I think I can explain it.  :)

What you've written is not what I put into the pseudocode.

You write above:

In a valid=1, and tag doesn't match... I get that I need to write the block in cache back to MM. At this point dirty is set to false. Then I copy the correct tag/block into the slot in cache and write the new data. Now dirty = true because I now have data that is not in MM.

That's not quite right. It should be: (I get this… just forget to write it)

In a valid=1, and tag doesn't match... Only if the dirty bit is true, I need to write the block in cache back to MM. At this point dirty is set to false. Then I copy the correct tag/block into the slot in cache.

Separately - then you process the write.  (write the new data. set dirty to true because I now have data that is not in MM.)  **BUT THIS WOULDN”T BE DONE IN A WRITE**

If this had only been a READ, you would still have to do those steps above.  Think of the process as a READ process, and if the instruction is a WRITE instruction, you'll process the WRITE.

So, the dirty bit is set to false when the cache and the main memory are back in sync because there's been a cache miss and the dirty bit had been true.

The dirty bit is set to true when something is written to the cache because now what's in cache doesn't match what's in memory.

Now, on a write, you can set the dirty bit to true and**then process some reads that are cache hits**. In that case, the dirty bit will remain true until it hits a cache miss (when you will have to write the block from the cache to the memory).  Likewise, if you set the dirty bit to false after a cache miss when the dirty bit had been true, **you can process a whole bunch of reads** before you ever set it to true again!  In this case the dirty bit will be false until the next time there's a write.

So, think of the following instructions:

                     B  E  F  O  R  E         A   F   T   E   R

Inst   Slot  Tag   Valid  Slot  Tag  Dirty | Valid  Slot  Tag  Dirty

 R     2     1  |   0     0     0     0   |   1     2     1     0 “That byte is not in memory and initialized to 0 (Cache Miss)”

 R     2     5  |    1     2     1     0   |   1     2     5     0

So does this write the block to memory?

What I think ::: Go out to main memory, write the block to the cache and set the dirty bit to false

What would print out? ::: “Value \_\_\_\_\_ has been written to address \_\_\_\_\_\_ (Cache Miss)

Miss because the tag is different

 W     2     4   |    1     2     5     0   |   1     2     4      1   (because a write) Miss

 R     2     4  |    1     2     4     1   |   1     2     4     1

 R     2     5  |    1     2     4     1   |   1     2     5     0  (because the cache was written back to memory and they match now)

 R     2   1  |     1     2     5     0   |   1     2     1     0

 W     2     3  |    1     2     1     0   |   1     2     3     1  (because a write)

 W     2     6  |    1     2     3     1   |   1     2     6     1   (although the read set the dirty bit to true, it was immediately reset again because it's a write)

 R     2     5  |    1     2     6     1   |   1     2     5     0   (because the cache was written back to memory)

Notice what happens to the dirty bit during these instructions.

Write

* Is valid
  + Tag matches
    - is dirty
      * (Write Only) add data to the block in cache
      * “data x is in address y (hit)
    - is **NOT** dirty
      * (Write Only) add data to the block in cache
      * (Write Only) set dirty to true
      * “data x is in address y (hit)
  + Tag does NOT match
    - is dirty
      * (Read & Write) copy block in cache to MM
      * (Read & Write) copy block w correct tag into cache
      * (Read & Write) set dirty bit to false
      * (Write only) add data to the block in cache
      * (Write only) set dirty bit to true (or can we skip setting it false above?)
      * “data x is in address y (miss)”
    - is NOT dirty (No need to copy block to MM because dirty=false)
      * (Read & Write) copy block w correct tag into cache
      * (Write only) add data to the block in cache
      * (Write only) set dirty bit to true
      * “data x is in address y (miss)”
* Is not valid
  + “Not in cache (miss)”

system.array(copy)