Data Cache Operations

Consider a direct-mapped, write-back data cache. Each line of the cache contains the following:

V bit
Tag bits
M bit
data block

When the cache controller receives a read or write request for a particular address, it views the address as being composed of three fields: the leftmost bits are viewed as the tag bits, the middle bits are viewed as the index bits, and the rightmost bits are viewed as the offset bits.

The size of the offset field determines the size of one data block: if the offset field is K bits wide, then the data block contains 2^K bytes. The size of the index field determines the number of lines in the cache: if the index field is L bits wide, then the cache contains 2^L lines.

For example, consider a system with 32 bit addresses, where the index field is 9 bits wide and the offset field is 6 bits wide: the tag field is 17 bits wide, there are 512 lines in the cache (2^9 lines), and there are 64 bytes in one data block (2^6 bytes).

The following outline describes the steps used to process a read or write request to the data cache:

```
// Process one memory reference (read or write)
decompose address into fields:  tag, index, offset
if (not (cache[index].V = 1 and cache[index].Tag = tag)) then
    // Perform miss processing
    if (cache[index].V = 1 and cache[index].M = 1) then
        // Perform write back
        copy old block from cache[index] to lower level
    endif
    copy new block from lower level to cache[index]
    cache[index].M ← 0
    cache[index].Tag ←tag
    cache[index].V ← 1
endif
if (read operation) then
    // Perform read operation
    copy item from cache[index] data block to CPU register
else if (write operation) then
    // Perform write operation
    copy item from CPU register to cache[index] data block
    cache[index].M ← 1
endif
```