1.

Please choose an appropriate sentence for below situation.

|  |  |  |  |
| --- | --- | --- | --- |
| Cache | TLB | Page table | Possible? Conditions? |
| Miss | Hit | Hit | a |
| Hit | Hit | Miss | e |
| Hit | Miss | Miss | e |
| Miss | Miss | Miss | d |
| Hit | Miss | Hit | b |
| Miss | Hit | Miss | e |
| Hit | Hit | Hit | f |
| Miss | Miss | Hit | c |

1. Yes; but page table never checked if TLB hits
2. TLB miss, but entry found in page table;after retry, data in cache
3. TLB miss, but entry found in page table; after retry, data miss in cache
4. TLB miss and is followed by a page fault; after retry, data miss in cache
5. impossible; not in TLB if page not in memory
6. entry is found in TLB and data in cache

2.

The memory architecture of a machine X is summarized in the table.

|  |  |
| --- | --- |
| Virtual Address | 54 bits |
| Page Size | 16 K bytes |
| PTE Size | 4 bytes |

1. Assume that there are 8 bits reserved for the operating system function (protection, replacement, valid, modified,…) other than required by the hardware translation algorithm. Derive the largest physical memory size (in bytes) allowed by this PTE format. Make sure you consider all the fields required by the translation algorithm.
2. How large (in bytes) is the page table?
3. Assuming 1 application exists in the system and the maximum physical memory is devoted to the process, how much physical space (in bytes) is there for the application’s data and code?
4. Physical page number = 32 – 8 = 24bits

The largest physical memory size = \* 16K bytes = 256 GB

1. The virtual page number has 54 – 14 = 40 bits. The number of page table entries are . Each PTE has 4 bytes. So the total size of the page table is bytes which is 4 terrabytes.
2. The application’s page table has an entry for every physical page that exists on the system, which means the page table size is \* 4 bytes. This leaves the remaining physical space to the process: \* 16K – bytes = bytes.