**CIS 450 – Computer Organization and Architecture**
**Homework #6**
**(25 points)**

**Due: Mon., Apr. 23, 2018 by 11:59 pm.**

1. Consider the Simple Memory System given in Lecture 20 with 14-bit virtual addresses, 12-bit physical addresses, and a page size of $2^6$ = 64 bytes:

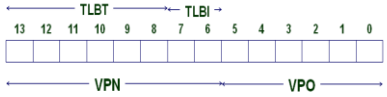
Simple Memory System Page Table

- Only show first 16 entries (out of 256)

| VPN | PPN | Valid | VPN | PPN | Valid |
|---|---|---|---|---|---|
| 00 | 28 | 1 | 08 | 13 | 1 |
| 01 | – | 0 | 09 | 17 | 1 |
| 02 | 33 | 1 | 0A | 09 | 1 |
| 03 | 02 | 1 | 0B | – | 0 |
| 04 | – | 0 | 0C | – | 0 |
| 05 | 16 | 1 | 0D | 2D | 1 |
| 06 | – | 0 | 0E | 11 | 1 |
| 07 | – | 0 | 0F | 0D | 1 |

Simple Memory System TLB

TLB
- 16 entries
- 4-way associative




| Set | Tag | PPN | Valid | Tag | PPN | Valid | Tag | PPN | Valid | Tag | PPN | Valid |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 03 | – | 0 | 09 | 0D | 1 | 00 | – | 0 | 07 | 02 | 1 |
| 1 | 03 | 2D | 1 | 02 | – | 0 | 04 | – | 0 | 0A | – | 0 |
| 2 | 02 | – | 0 | 08 | – | 0 | 06 | – | 0 | 03 | – | 0 |
| 3 | 07 | – | 0 | 03 | 0D | 1 | 0A | 34 | 1 | 02 | – | 0 |

Use the Simple Memory System above to complete the following address translation problems to convert the given Virtual Address to the corresponding Physical Address. Note that all table values are given in hexadecimal format. Complete the virtual address and other fields.

(a) Virtual Address: 0x036A

| 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | | | | | | | | |

VPN _____ TLBI _____ TLBT _____ TLB Hit? ___ Page Fault? ___ PPN: _____

Physical Address: _____

| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | |

(b) Virtual Address: 0x0177

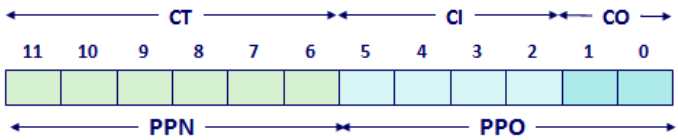
| 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | |

VPN _____ TLBI _____ TLBT _____ TLB Hit? ___ Page Fault? ___ PPN: _____

Physical Address: _____

| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | |

2. Once the physical address is resolved, use the Data Cache shown below to determine if the data can be obtained from the cache, or if access to memory is required (just write MEM for your answer). You may assume that:
   - The memory is byte addressable. Memory accesses are to 1-byte words, not 4-byte words.
   - Physical addresses are 12 bits wide.
   - The cache is a direct mapped cache with a 4-byte block size and 16 lines total as shown below. Note that all numbers are given in **hexadecimal** format.

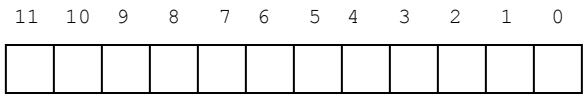



| Idx | Tag | Valid | B0 | B1 | B2 | B3 |
|---|---|---|---|---|---|---|
| 0 | 19 | 1 | 99 | 11 | 23 | 11 |
| 1 | 15 | 0 | – | – | – | – |
| 2 | 1B | 1 | 00 | 02 | 04 | 08 |
| 3 | 36 | 0 | – | – | – | – |
| 4 | 32 | 1 | 43 | 6D | 8F | 09 |
| 5 | 0D | 1 | 36 | 72 | F0 | 1D |
| 6 | 31 | 0 | – | – | – | – |
| 7 | 16 | 1 | 11 | C2 | DF | 03 |

| Idx | Tag | Valid | B0 | B1 | B2 | B3 |
|---|---|---|---|---|---|---|
| 8 | 24 | 1 | 3A | 00 | 51 | 89 |
| 9 | 2D | 0 | – | – | – | – |
| A | 2D | 1 | 93 | 15 | DA | 3B |
| B | 0B | 0 | – | – | – | – |
| C | 12 | 0 | – | – | – | – |
| D | 16 | 1 | 04 | 96 | 34 | 15 |
| E | 13 | 1 | 83 | 77 | 1B | D3 |
| F | 14 | 0 | – | – | – | – |

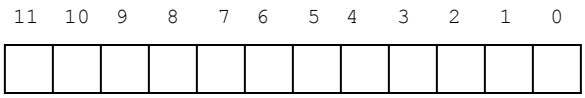
(a) For each physical address given below, if a cache hit occurs, indicate the cache entry accessed and the cache byte value returned in hex. If a cache miss occurs, just write 'N' next to "Cache Hit?" and just write "MEM" for the cache byte returned to indicate that the reference must be resolve by going out to main memory.

**Physical Address: 0xB6A**




| Parameter | Value |
|---|---|
| Cache Byte Offset | 0x |
| Cache Index | 0x |
| Cache Tag | 0x |
| Cache Hit (Y/N)? | |
| If Hit, Cache Byte Returned | 0x |

**Physical Address: 0x05B7**




| Parameter | Value |
|---|---|
| Cache Byte Offset | 0x |
| Cache Index | 0x |
| Cache Tag | 0x |
| Cache Hit (Y/N)? | |
| If Hit, Cache Byte Returned | 0x |