CPSC 313

06W Term 2

Problem Set #9 - Solution

1. A. 00 0010 0111 1100

   B.

VPN	0x9
TLBI	0x1
TLBT	0x2
TLB hit?	N
page fault?	N
PPN	0x17

   C. 0101 1111 1100

   D.

CO	0x0
CI	0xf
CT	0x17
cache hit?	N
cache byte?	-

2. A. 00 0011 1010 1001

   B.

VPN	0xe
TLBI	0x2
TLBT	0x3
TLB hit?	N
page fault?	N
PPN	0x11

   C. 0100 0110 1001

   D.

CO	0x1
CI	0xa
CT	0x11
cache hit?	N
cache byte?	-

3. A. 00 0000 0100 0000

   B.

VPN	0x1
TLBI	0x1
TLBT	0x0
TLB hit?	N
page fault?	Y
PPN	-

   C. n/a

   D. n/a

4. (a) First we must read the PTE, which is at index VA.VPN (0x08063) or at address PTBR + 0x08063 * 4, or PTE[PTBR + 0x2018C].
   (b) Next we check fields P (must be 1) and S (must be 0) of that PTE (no additional memory reads). If either of these are not as required, we are done.
   (c) Finally we compute the physical address, which is (PTE[PTBR + 0x2018C].PFN << 12) + VA.PO (= 0x472), and read from that physical address.

5. (a) First we must read the PDE, which is at index VA[31:22] (0x020) or at address PTBR + 0x020 * 4, or PTE[PTBR + 0x080].
   (b) Next we check fields P (must be 1) and S (must be 0) of that PDE (no additional memory reads). If either of these are not as required, we are done.
   (c) Next we must read the PTE, which is at index VA[21:12] (0x063) or at address PDE[PTBR + 0x080].PFN << 12 + 0x063 * 4, or PTE[PDE[PTBR + 0x080].PFN << 12 + 0x18C].
   (d) Next we check fields P (must be 1) and S (must be 0) of that PTE (no additional memory reads). If either of these are not as required, we are done.
   (e) Finally we compute the physical address, which is (PTE[PDE[PTBR + 0x080].PFN << 12 + 0x18C].PFN << 12) + VA.PO (= 0x472), and read from that physical address.

6. Both set associative caches and virtual memory want to replace the item that will be used longest in the future, but this can't be implemented. The best approximation is LRU (Least Recently Used), which virtual memory can use (or approximate with multiple FIFOs) because the VM system has lots of time and wants to work hard to avoid a miss. A cache must use a cheaper approximation because it can't afford the overhead and so normally uses random.