Gasser Ahmed
gasser18@vt.edu
ECE 5484, Homework 5

1. a. Blocks of main memory = $2^{64} / 64 = 2^{64} / 2^6 = 2^{58}$ blocks
   b. Since cache has $2048 = 2^{11}$ blocks, therefore size of the block = 11 bits
   Since each block contains $64 = 2^6$ bytes, therefore size of the offset = 6 bits
   Therefore, size of the tag = 64 – (11 + 6) = 64 – 17 = 47 bits
   c. 0x00000000000163FA
   = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0110 0011 1111 1010
   Tag = 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 000
   Block = 10110001111
   Offset = 111010

   Therefore, 10110001111 = 0x58F = Cache block 1423

---

2. a. Blocks of main memory = $2^{24} / 64 = 2^{24} / 2^6 = 2^{18}$ blocks
   b. Since each block contains $64 = 2^6$ bytes, therefore size of the offset = 6 bits
   Therefore, size of the tag = 24 – 6 = 18 bits
   c. Since it is associative cache, it can map anywhere.

---

3. a. Since number of sets = $64 / 2 = 32 = 2^5$ sets, therefore size of set = 5 bits
   Since each block contains $4 = 2^2$ bytes, therefore size of the offset = 2 bits
   Therefore, size of the tag = 21 – (5+2) = 14 bits
   b. Since number of sets = $64 / 4 = 16 = 2^4$ sets, therefore size of set = 4 bits
   Since each block contains $4 = 2^2$ bytes, therefore size of the offset = 2 bits
   Therefore, size of the tag = 21 – (4+2) = 15 bit

---

4. a. $2^{20} / 2^8 = 2^{12}$ pages
   b. $2^{16} / 2^8 = 2^8$ page frames
   c. $2^{20} / 2^8 = 2^{12}$ entries

---

5. a. 7ns + 15ns = 22ns
   b. EAT for Cache = Ratio for hit * (time for hit) + Ratio for miss * (time for miss)
   = 0.97 * (15ns) + 0.03 * (15ns + 30ns) = 15.9ns
   EAT for TLB= Ratio for hit TLB * (time for hit TLB + EAT Cache)
   = 0.95 * (7ns + 15.9ns) = 21.755ns

---

6. a. Since there are 32MB, or $2^5 x 2^{20} = 2^{25}$ addresses
   Therefore, we need 25 bits for a virtual address.
   b. Since there are 4MB, or $2^2 x 2^{20} = 2^{22}$
   Therefore, we need 22 bits for physical address.
   c. Since there are $2^{25}/2^{11}$ pages in virtual memory, therefore the page table can have 2^{14} entries.
   d. 0x37F = 00000000000000 01101111111,
   Since the first 14 bits are the page, therefore the remaining bits (11 bits) are the offset
   Therefore, replace the first 14 bits (00000000000000) by (00000000000001) (page 0 maps to frame 1), to get the physical address 00000000000000101101111111, or 0xB7F
   e. 0x1203 = 00000000000010 01000000011
   Since the first 14 bits are the frame, therefore the frame is 2.
   Since virtual page 4 maps to frame 2, so we will replace the first 14 bits (00000000000010) with (00000000000100), to get the virtual address 00000000000100001000000011, or 0x2203
