

# CSC 3501 Computer Organization and Design

## Homework # 8 & 9 Solutions

---

(2) a.  $2^{32}/2^5 = 2^{27}$

b. 32 bit addresses with 17 bits in the tag field, 10 in the block field, and 5 in the word field

c. 000063FA = 0000000000000000 1100011111 11010, which implies Block 799

(4) a.  $2^{24}/2^6 = 2^{18}$

b. 24 bit addresses with 18 bits in the tag field and 6 in the word field

c. Since it's associative cache, it can map anywhere

(7) a. A total of  $2^{16}$  words of main memory implies we have 16 bits in an address. Cache contains  $2^5$  blocks, but each set must have 2 blocks, so we have  $2^5/2=2^4$  sets.

Therefore our 16-bit address is divided into 9 bits for the tag field, 4 bits for the set field, and 3 bits for the word field.

b. The 32 blocks in cache must now be divided into sets with 4 blocks each, implying we have only 8 sets. The tag field would now have 10 bits, the set field 3 bits, and the word field 3 bits.

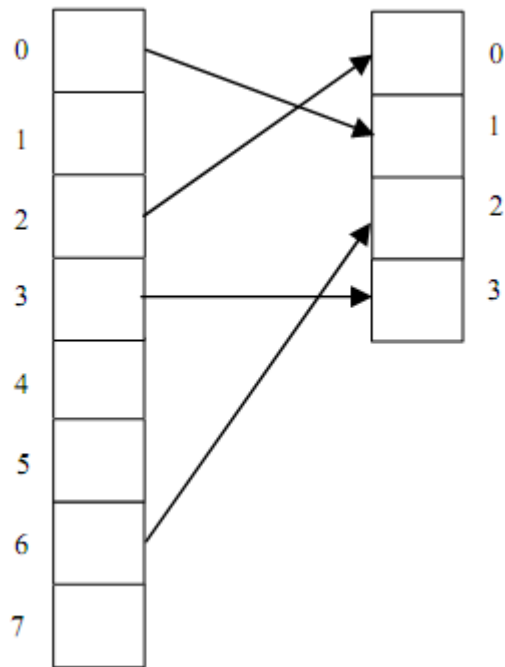
(11) a.  $64K = 2^6 \cdot 2^{10} = 2^{16}$ ;  $2^{16}/2^5 = 2^{11}$  blocks in cache, so 11 bits are needed for the block field.

5 bits are needed for the word field, leaving 8 for the tag.

b. Again, 5 bits are needed for the word field, leaving 19 for the tag.

c. There are  $2^{11}/2^2 = 2^9$  sets in cache, so 9 bits are needed for the set field. We still need 5 bits for the word field, leaving 10 for the tag field.

(12)



(14) a.  $2^3 * 2^4 = 2^7$ , so there are 7 bits in a virtual address

b.  $2^2 * 2^4 = 2^6$ , so there are 6 bits in a physical address

c.  $18 = 001\ 0010$  (where 001 is the page field and 0010 is the offset). Using the page table, and going to entry 1, we see that page 1 maps to frame 0, so the actual physical address would be  $00\ 0010$ , or 2.

d.  $54 = 11\ 0\ 110$ , where 11 is the tag, 0 is the set, and 110 is the offset. Therefore, this would map to Set 0 in cache. Once there, if the tag is found, the block is in cache. If not, it is a miss.

e.  $25 = 001\ 1001$ , where 001 is the virtual page, and 1001 is the offset. (This maps to physical page 00 with offset 1001.) The TLB would first be checked to see if the pair (1,0) was present (virtual page 1, physical frame 0). If so, 00 can be substituted for 001, giving the actual physical address  $00\ 1001$ . If the entry is not found in the TLB,

the page table must be accessed, at which time the physical address 00 1001 is determined. At this point (regardless of whether we found the physical address using the TLB or the page table), the cache is checked to see if the block containing this physical address is currently cached. The memory address 001001 is divided up into 00 1 001, where 00 is the tag, 1 is the set, and 001 is the offset. Set 1 in cache is then checked for the tag 00 (it would be found as address 25 is in block D). The value in cache is used. (If it had not been found in cache, main memory would be accessed.)

- (15) a. Possible, 5ns (TLB access) + 12ns (cache access)
- b. Possible, 5ns (TLB access) + 25ns (page table reference) + 12ns (cache access)
- c. Possible, 5ns (TLB access) + 25ns (page table reference) + 12ns (cache access) + 25ns (main memory reference)
- d. Not possible (the cache value would be stale if the page no longer resides in page table)
- e. Possible, 5ns (TLB access) + 25ns (page table reference) + 200ms (disk reference) + 5ns (TLB, since access is restarted) + 12ns (cache hit)

$$\begin{aligned}
 \text{EAT} = & .90[5\text{ns} + \underbrace{.98(12\text{ns})}_{\text{Cache hit}} + \underbrace{.02(25\text{ns})}_{\text{Cache miss}}] \\
 & \underbrace{\hspace{10em}}_{\text{TLB hit}} \\
 + & .10 [5\text{ns} + \underbrace{.999[25\text{ns} + \underbrace{.98(12\text{ns})}_{\text{Cache hit or miss}} + \underbrace{.02(12\text{ns} + 25\text{ns})}_{\text{Page table miss}}]}_{\text{Page table hit}} + \underbrace{.001(25\text{ns} + 200\text{ms} + 5\text{ns} + 12\text{ns})}_{\text{Find in cache}}] \\
 & \underbrace{\hspace{15em}}_{\text{Restart access, find in TLB}} \\
 & \underbrace{\hspace{20em}}_{\text{TLB miss}}
 \end{aligned}$$