

قبل التصحيح للحلول الجاهزة  
الحصول على الامتحان في الامتحان ، أنزل بجملة في حل الامتحان

## Chapter 6 Review Questions Key

"Locality of reference" refers to:

- A) data always being in cache
- B) programs always referencing data in RAM
- C) clustering of memory references
- D) the requirement that forces us to use a large amount of expensive memory

Which of the following types of memory has the shortest (fastest) access time?

- A) cache memory
- B) main memory
- C) secondary memory
- D) registers

A major advantage of direct mapped cache is its simplicity and ease of implementation. The main disadvantage of direct mapped cache is:

- A) it is more expensive than fully associative and set associative mapping
- B) it has a greater access time than any other method
- C) its performance is degraded if two or more blocks that map to the same location are used alternately
- D) it does not allow the cache to store the tag that corresponds to the block currently residing in that cache location

The average time required to reach a memory storage location and retrieve its contents is called:

- A) latency
- B) response time
- C) hit time
- D) effective access time

Memory that is accessed by searching for content is called:

- A) read only memory
- B) erasable memory
- C) associative memory
- D) virtual memory

Cache memory is effective because:

- A) it is very inexpensive
- B) it is very large
- C) it is very small
- D) of the principle of locality

Cache memory is typically positioned between:

- A) the CPU and RAM
- B) the CPU and the hard drive
- C) ROM and RAM
- D) none of the above

Cache mapping is necessary because:

- A) the address generated by the CPU must be converted to a cache location
- B) cache is so small that its use requires a map
- C) cache is larger than main memory and mapping allows us to store multiple copies of each piece of data from main memory
- D) none of the above

The tag field of a main memory address is used to determine:

- A) if the cache entry is valid
- B) if the cache entry is the desired block
- C) if the memory address is valid
- D) none of the above

The offset field of a main memory address is used to determine:

- A) if the cache entry is valid
- B) if the cache entry is the desired block
- C) the location of the desired data in the cache block
- D) none of these

Cache replacement policies are necessary:

- A) to determine which cache mapping policy to use
- B) to determine which block in cache should be the victim block
- C) to decide where to put blocks when cache is empty
- D) all of the above

All of the following are cache replacement algorithms except:

- A) LRU
- B) FIFO
- C) random
- D) thrashing

Assuming an 8-bit virtual address with pages of 32 bytes, the virtual address format is:

- A) 5 bits for the page and 3 bits for the offset
- B) 3 bits for the page and 5 bits for the offset
- C) 8 bits for the page and 32 bits for the offset
- D) 32 bits for the page and 8 bits for the offset

Cache memory improves performance by improving memory \_\_\_\_\_ while virtual memory improves performance by increasing memory \_\_\_\_\_.

- A) execution time/access time
- B) locality/access time
- C) access time/address space
- D) organization/paging

The purpose of a TLB is:

- A) to cache page table entries
- B) to cache frequently used data from memory
- C) to hold the starting address of the page table
- D) to hold the length of the page table

# Key

## Short Answer

Suppose we have a byte-addressable computer using direct mapping with 16-bit main memory addresses and 32 blocks of cache. If each block contains 8 bytes, determine the size of the offset field.

offset field = 3 bits

Suppose we have a byte-addressable computer using direct mapping with 16-bit main memory addresses and 32 blocks of cache. If each block contains 8 bytes, determine the size of the tag field.

5 bits ← ~~size~~

main memory → 3 bits

Size of tag Field =  $16 - 5 - 3 = 8$  bits

Index → 5 bits

offset → 3 bits

Suppose we have a byte-addressable computer using 2-way set associative mapping with 16-bit main memory addresses and 32 blocks of cache. If each block contains 8 bytes, determine the size of the offset field.

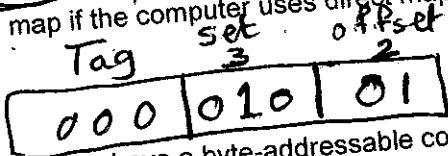
offset = 3 bits

Suppose we have a byte-addressable computer using 2-way set associative mapping with 16-bit main memory addresses and 32 blocks of cache. If each block contains 8 bytes, determine the size of the set field.

# of sets =  $\frac{32 \text{ blocks}}{2} = 16 \text{ set} = 2^4$

Size of set Field = 4 bits

Suppose we have a byte-addressable computer with a cache that holds 8 blocks of 4 bytes each. Assuming that each memory address has 8 bits, to which cache block would the hexadecimal address 0x09 map if the computer uses direct mapping?



0x09 maps to block 2

Suppose we have a byte-addressable computer with a cache that holds 8 blocks of 4 bytes each. Assuming that each memory address has 8 bits, to which cache set would the hexadecimal address 0x1F map if the computer uses direct mapping?

Block  $(111)_2 \Rightarrow \text{Block } (7)_{10}$

Suppose the cache access time is 10ns, main memory access time is 200ns, and the cache hit rate is 90%. Assuming parallel (overlapped) access, what is the average access time for the processor to access an item?

Avg Access time =  $H \times \text{Access}_{\text{cache}} + (1-H) \times \text{Access}_{\text{mem}}$

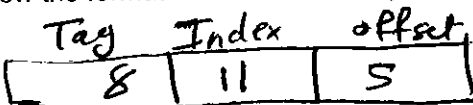
=  $0.9 (10 \times 10^{-9}) + (0.1) (200 \times 10^{-9})$

=  $9 \times 10^{-9} + 20 \times 10^{-9}$

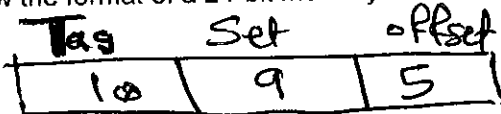
=  $29 \times 10^{-9}$

= 29 n sec

Consider a byte-addressable computer with 24-bit addresses, a cache capable of storing a total of 64K bytes of data, and blocks of 32 bytes. Show the format of a 24-bit memory address if the computer uses direct mapping.



Consider a byte-addressable computer with 24-bit addresses, a cache capable of storing a total of 64K bytes of data, and blocks of 32 bytes. Show the format of a 24-bit memory address if the computer uses 4-way set associative mapping.



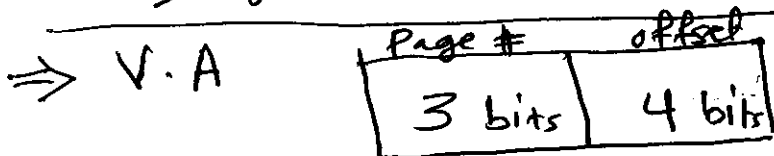
Given a computer using a byte-addressable virtual memory system with a two-entry TLB, a 2-way set associative cache, and a page table for a process P. Assume cache blocks of size 8 bytes. Assume pages of size 16 bytes and a main memory of 4 frames. Assume the following TLB and page table for Process P:

| TLB |   |
|-----|---|
| 0   | 3 |
| 4   | 1 |

- ① Page size = Frame size = 16 bytes =  $2^4$  الحلقة  
 $\Rightarrow$  offset = 4 bits
- ② # of Frames = 4 frames =  $2^2$  frame  
 $\Rightarrow$  Frame field = 2 bits
- ③ From the table  $\exists$  8 pages =  $2^3$   
 $\Rightarrow$  Page field = 3 bits

frame

| Page Table |       |   |
|------------|-------|---|
| f          | Valid |   |
| 0          | 3     | 1 |
| 1          | 0     | 1 |
| 2          | -     | 0 |
| 3          | 2     | 1 |
| 4          | 1     | 1 |
| 5          | -     | 0 |
| 6          | -     | 0 |
| 7          | -     | 0 |



How many bits are in a virtual address for process P?

# of bits = 3 + 4 = 7 bits

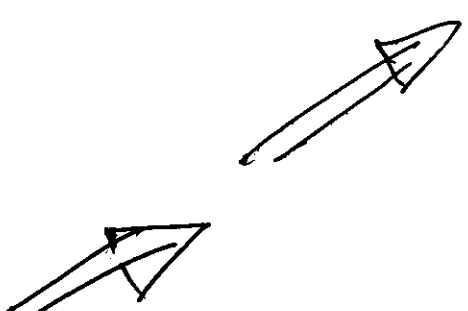
50. Given a computer using a byte-addressable virtual memory system with a two-entry TLB, a 2-way set associative cache, and a page table for a process P. Assume cache blocks of size 8 bytes. Assume pages of size 16 bytes and a main memory of 4 frames. Assume the following TLB and page table for Process P:

| TLB |   |
|-----|---|
| 0   | 3 |
| 4   | 1 |

# of bits in physical address = 2 + 4 = 6 bits

| Page Table |       |   |
|------------|-------|---|
| f          | Valid |   |
| 0          | 3     | 1 |
| 1          | 0     | 1 |
| 2          | -     | 0 |
| 3          | 2     | 1 |
| 4          | 1     | 1 |
| 5          | -     | 0 |
| 6          | -     | 0 |
| 7          | -     | 0 |

How many bits are in a physical address?



متمنياً لكم التوفيق

محبتي