

# EE 457 Unit 7b

## Main Memory Organization

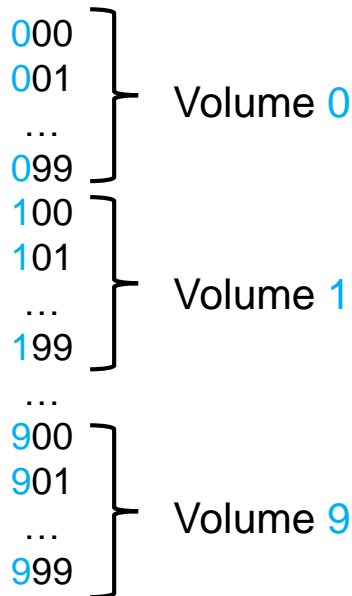
# Motivation

- Organize main memory to
  - Facilitate byte-addressability while maintaining...
  - Efficient fetching of the words in a cache block
- Low order interleaving (L.O.I) helps us achieve this

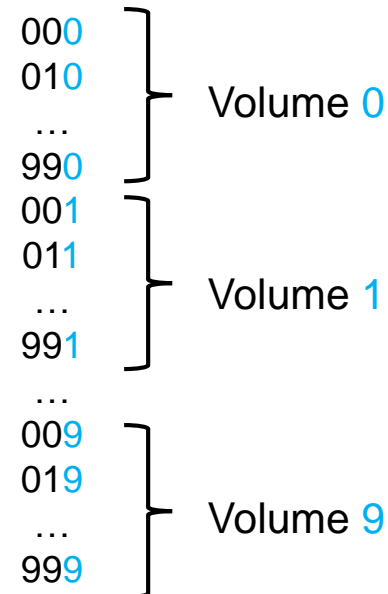
# Interleaving Analogy

- Consider an encyclopedia set consisting of 1000 pages (000-999) bound in
  - 10 volumes (0-9) of
  - 100 pages each (00-99)

Method I  
 (Consecutive pages  
 in a volume)



Method II  
 (Consecutive pages in  
 consecutive volumes)



# Interleaving Analogy

- Example: Say chapter 73 runs from page 730-739
  - In Method I: Chapter 73 is completely in volume 7
  - In Method II: The 73<sup>rd</sup> page of each volume form chapter 73 as shown below
- Which do you prefer?
  - If reading the chapter you may say method I
  - If you have to make a copy of the chapter and you have 10 photocopy machines with 10 friends to help you might say method II
    - Back to the scenario of reading the chapter, given those same 10 friends they could open each volume to page 73 for you so that you can read in a continuous manner

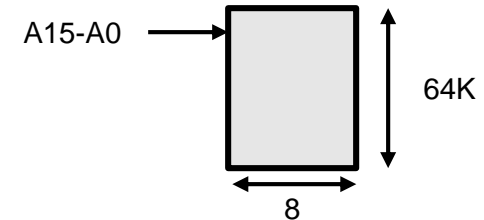
Page 730 is page 73 of volume 0  
Page 731 is page 73 of volume 1  
...  
Page 739 is page 73 of volume 9

} Low Order Interleaving

# Byte Addressability

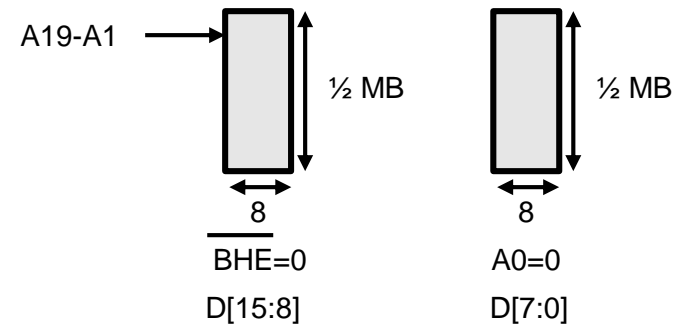
1. Intel 8085: 16-bit addr., 8-bit data, byte addressable processor.

Memory space:  $2^{16} = 64\text{KB}$ , A15-A0, D7-D0



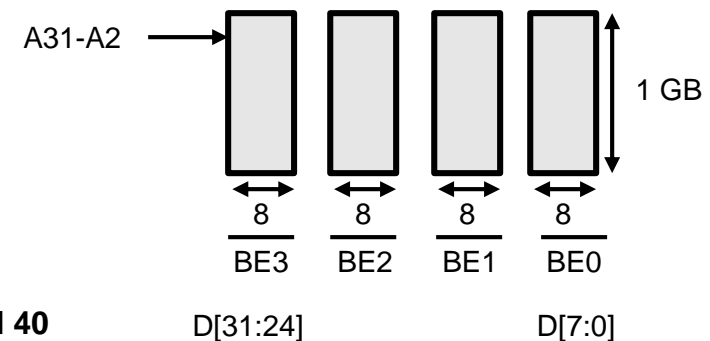
2. Intel 8086: 20-bit addr., 16-bit data, byte addressable, little-endian proc.

Memory space:  $2^{20} = 1\text{MB}$ , A19-A0  
 [A19-A1, BHE (BE1), A0 (BE0)], D15-D0



3. Intel 80386: 32-bit addr., 32-bit data, byte addressable, little-endian proc.

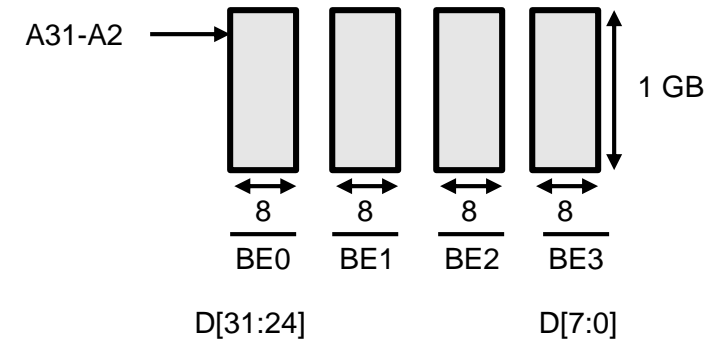
Memory space:  $2^{32} = 4\text{GB}$ , A31-A0  
 [A31-A2, BE3, BE2, BE1, BE0], D31-D0



# Byte Addressability

4. Intel 80386: 32-bit addr., 32-bit data, byte addressable, big-endian proc.

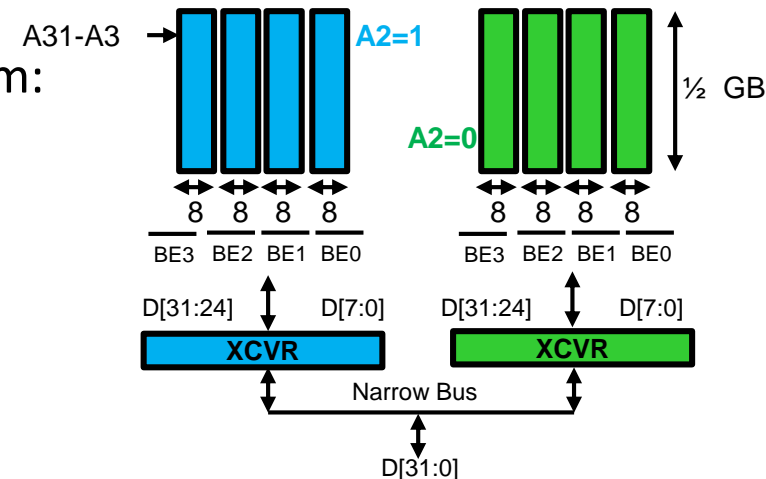
Memory space:  $2^{32} = 4\text{GB}$ ,  $A_{31}-A_0$   
 $[A_{31}-A_2, BE_3, BE_2, BE_1, BE_0], D_{31}-D_0$



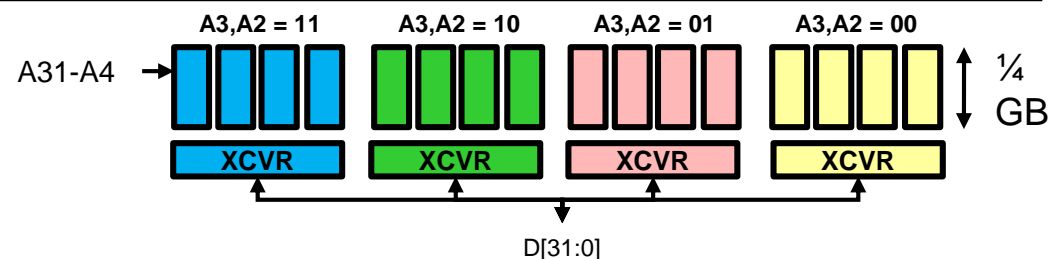
Byte 40    Byte 41    Byte 42    Byte 43    = Word 40

5. Little-Endian system, **2-way interleaved** system:  
 32-bit addr., 32-bit data,  
 byte addressable  
 (Narrow, 32-bit data bus b/w mem. and cache)

Memory space:  $2^{32} = 4\text{GB}$ ,  $A_{31}-A_0$   
 $[A_{31}-A_2, BE_3, BE_2, BE_1, BE_0], D_{31}-D_0$

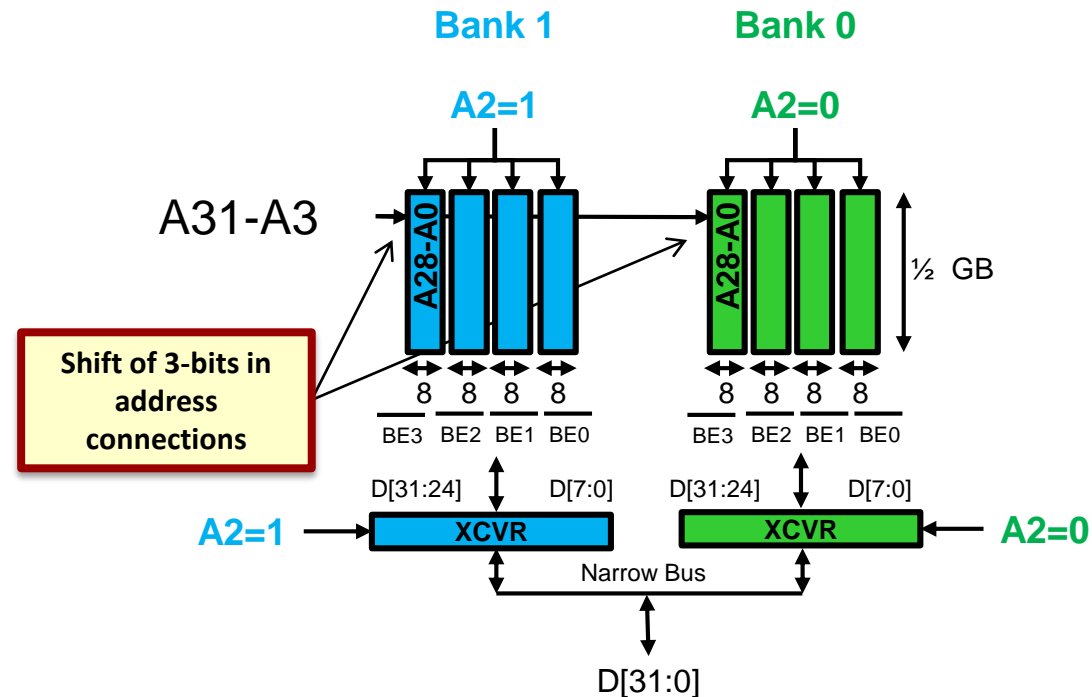


6. Same as 5 above,  
 but **4-way interleaved**



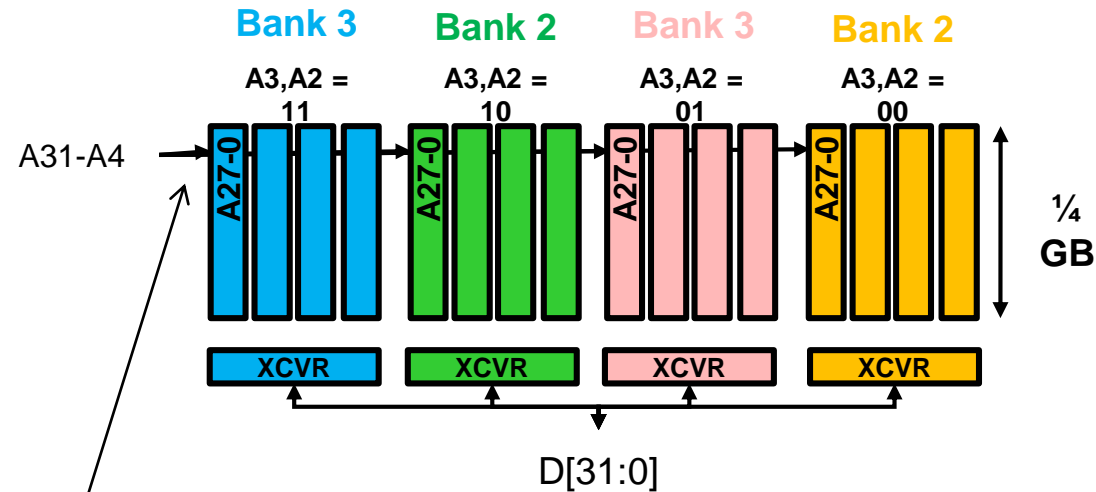
# 2-Way L.O.I.

- System address bus uses
  - A1:A0 and size info to generate /BE3../BE0 (Byte Enables)
    - In a 32-bit data bus, we need 2 address bits to produce the 4 BE's
    - In a 64-bit data bus, we would need 3 address bits to produce 8 BE's
  - Lower order bits to select a “bank”
    - Only 1 address bit, A2, to select one of 2 banks
  - Upper bits connect to each memory chip
    - Each memory chip is just a collection of ½ GB requiring 29 address bits...we can connect appropriate 29 bits



# 4-Way L.O.I.

- System address bus uses
  - A1:A0 and size info to generate /BE<sub>i</sub> (Byte Enables)
  - Lower order bits to select a “bank”
  - Upper bits connect to each memory chip

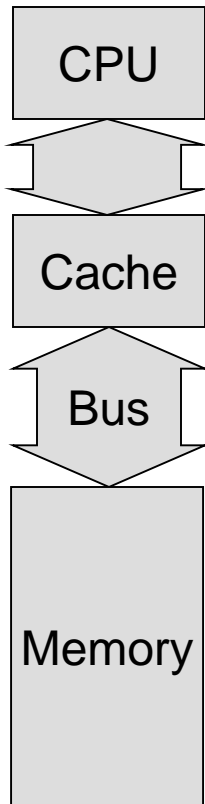


Shift of 4-bits in address connections

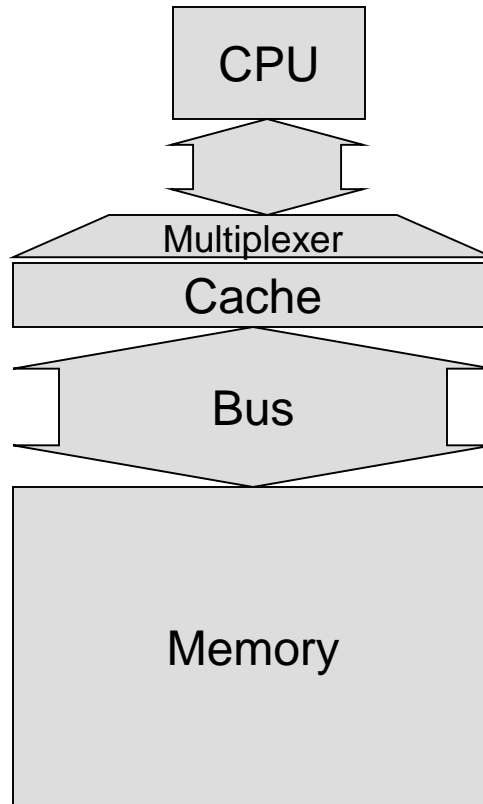


# Organization Options

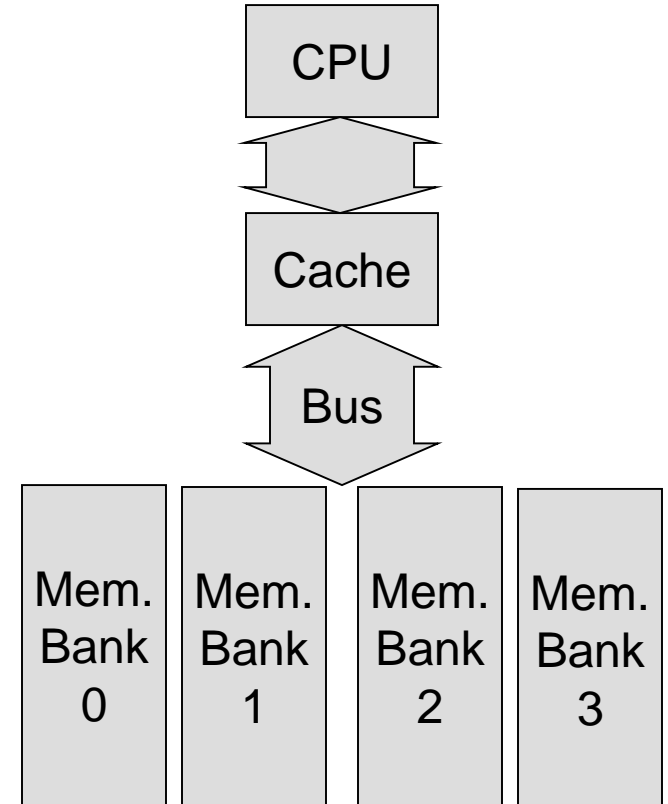
a.) One-word-wide memory Organization



b.) Wide Memory Organization



c.) EE 457 Interleaved



# Organization Comparison

- Assume following latencies

Send address to MM	1 clock
MM (DRAM) Access Time	15 clocks
Transfer time for one word	1 clock

- Find time to access a cache line of 4-words

a. Narrow Memory	$1 + 4 * 15 + 4 * 1 = 65$ clocks (assume mem. controller will auto-increment address)
b. Wide Memory	$1 + 15 + 1 = 17$ clocks
c. Interleaved Memory	$1 + 15 + 4 * 1 = 20$ clocks

# Example

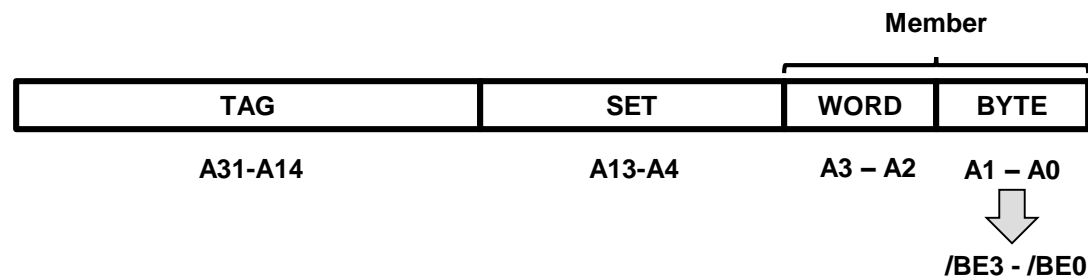
- Consider a set-associative mapping and physical organization of main memory, cache data RAMs, and cache tag RAMs.
- Specs:
  - 32-bit physical address, byte-addressable system
  - Cache Size = 64KB
  - Block Size = 4 words (16 bytes)
  - Set Size = 4 blocks (64 bytes)

$$\# \text{ of MM Blocks} = 2^{32} / 2^4 = 2^{28}$$

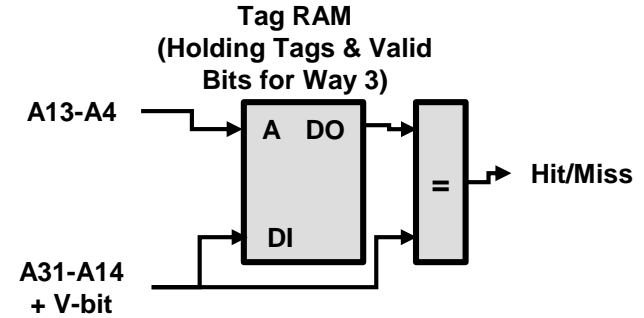
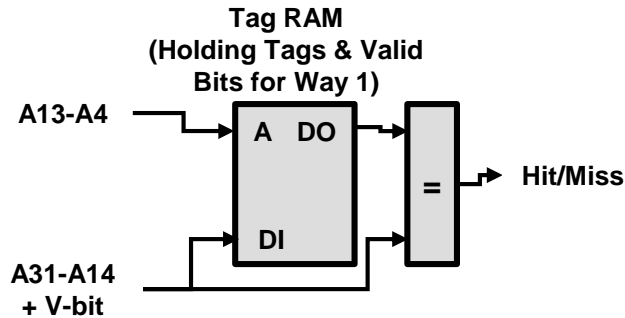
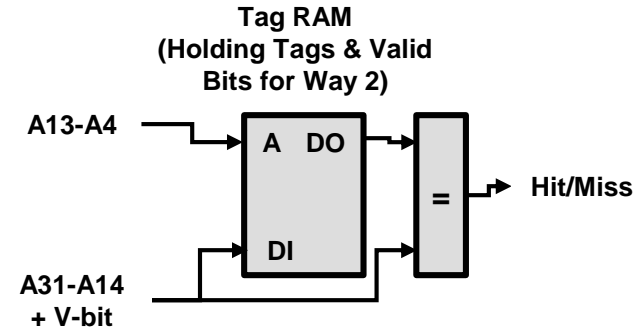
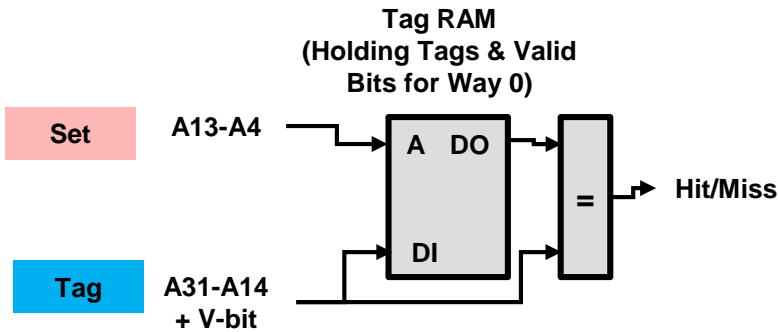
$$\# \text{ of Cache Blocks} = 2^{16} / 2^4 = 2^{12}$$

$$\# \text{ of Sets} = 2^{12} \text{ cache blocks} / 2^2 \text{ blocks/set} = 2^{10}$$

$$\# \text{ of Groups} = 2^{28} \text{ MM blocks} / 2^{10} \text{ sets} = 2^{18}$$



# Tag RAM Example



# MM & Data RAM Example

