

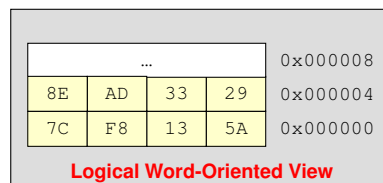
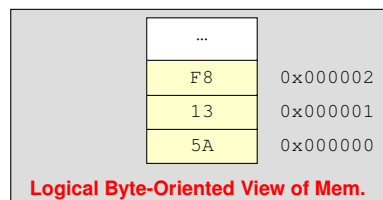
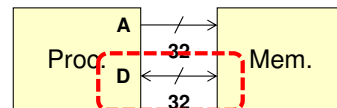
EE 457 Unit 7b

Main Memory Organization

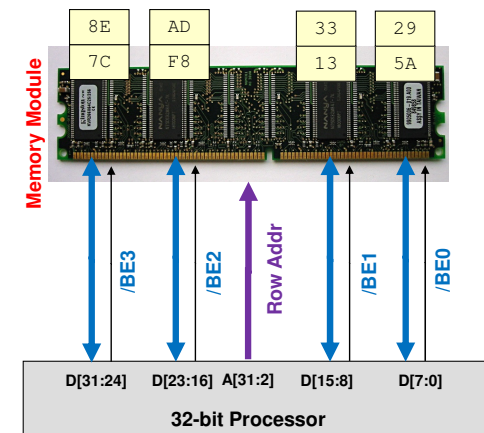
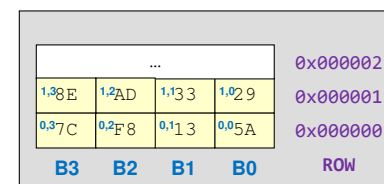
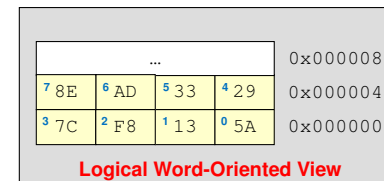
PROC/MEM PHYSICAL INTERFACE

Recall: MIPS Memory Data Organization

- We can logically picture memory in the units (sizes) that we actually access them
- We can access 1-byte at a time but the data bus allows for wider access (32-bits)
- Logical view of memory arranged in rows of largest access size (word)
 - Still with separate addresses for each byte
 - Can get word, halfwords, or bytes

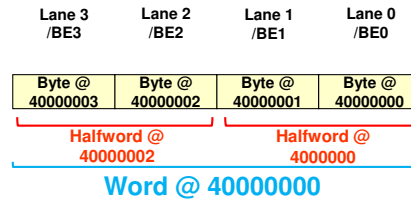


Byte Enables and the Data Bus



Byte Enables and the Data Bus

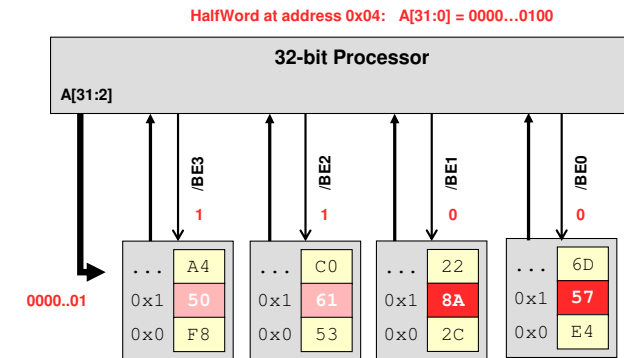
- What are the control signals that indicate access size?
- Though we may have a 32-bit address bus A[31:0], physically the processor will convert the lower 2 address bits A[1:0] and the size information into 4 separate **[/BE3../BE0]**



| Desired Memory | Internal A[31:0] | Physical Addr. Bus A[31:2] | /BE3 | /BE2 | /BE1 | /BE0 |
|-------------------|------------------|----------------------------|------|------|------|------|
| Word @ 0x40000000 | 0100...00 | 0100...00 | | | | |
| Half @ 0x40000002 | 0100...00 | 0100...00 | | | | |
| Byte @ 0x40000002 | 0100...00 | 0100...00 | | | | |
| Byte @ 0x40000001 | 0100...00 | 0100...00 | | | | |
| Half @ 0x40000004 | 0100...01 | 0100...01 | | | | |

Address & Data Bus Connections

- Organize memory into several byte-size memories running in parallel (sometimes known as “banks”)
- Convert lower address bits into bank enables to selectively enable each bank
- A[31:2] is provided to all memory banks specifying the same internal location



Byte Addressable Processors

| Proc. | External Data Bus | Address Pin-Out | Min. # of Banks | Shift in Address |
|----------------------------|-------------------|-----------------|-----------------|------------------|
| 8088 (8-bit proc.) | D[7:0] | A[19:__] | | |
| 8086 (16-bit proc.) | D[15:0] | A[19:__] | | |
| 80386 (32-bit proc.) | D[31:0] | A[23:__] | | |
| Core Series (64-bit proc.) | D[63:0] | A[35:__] | | |

MEMORY INTERLEAVING

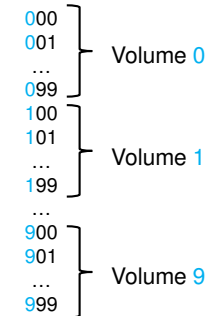
Motivation

- Organize main memory to
 - Facilitate byte-addressability while maintaining...
 - Efficient fetching of the words in a cache block
- _____ helps us achieve this

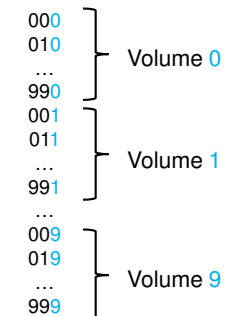
Interleaving Analogy

- Consider a journal 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 article 73 runs from page 730-739
 - In Method I: Article 73 is _____
 - In Method II: The _____ page of _____ volume form article 73 as shown below
- Which do you prefer?
 - If reading the article you may say method I
 - If you have to make a copy of the article and you have 10 photocopy machines with 10 friends to help you might say _____
 - Back to the scenario of reading the article, given those same 10 friends they could _____ for you so that you can still read in a continuous manner

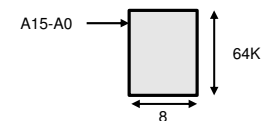
Page 730 is page 73 of volume 0
 Page 731 is page 73 of volume 1
 ...
 Page 739 is page 73 of volume 9

Low Order Interleaving

Byte Addressability

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

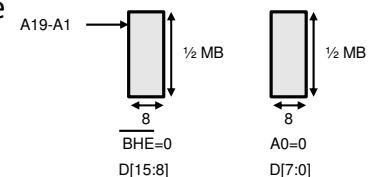
Memory space: $2^{16} = 64\text{KB}$, A15-A0, D7-D0



- 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

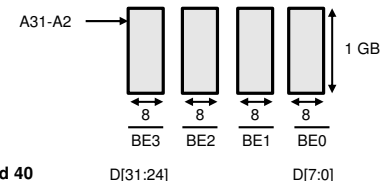
Byte 41 Byte 40 = Word 40



- 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

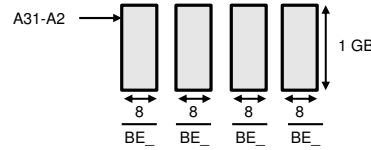
Byte 43 Byte 42 Byte 41 Byte 40 = Word 40



Byte Addressability

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

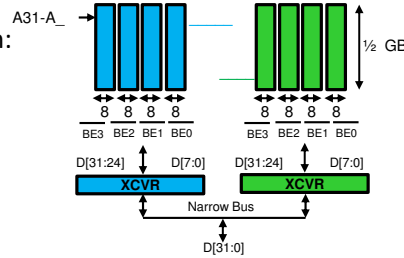
Memory space: $2^{32} = 4\text{GB}$, A31-A0
[A31-A2, BE3, BE2, BE1, BE0], D31-D0



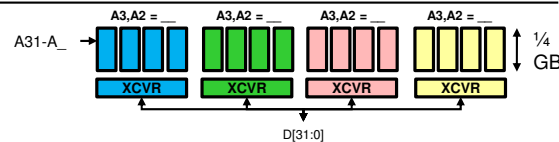
Byte 40 Byte 41 Byte 42 Byte 43 = Word 40

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

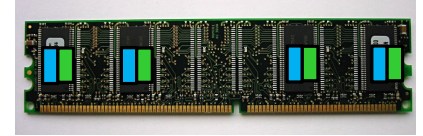
Memory space: $2^{32} = 4\text{GB}$, A31-A0
[A31-A2, BE3, BE2, BE1, BE0], D31-D0



6. Same as 5 above,
but _____



2-Way L.O.I.



- System address bus uses

- A1:A0 and size info to generate /BE3../BE0 (Byte Enables)

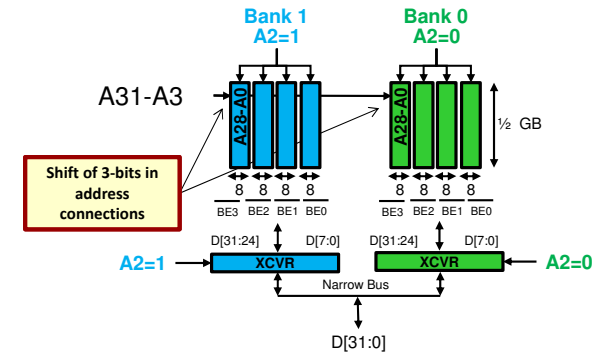
- In a 32-bit data bus, we need 2 address bits to produce the 4 BE's
 - In a 64-bit data bus, we would need _____ address bits to produce _____ BE's

- 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 1/2 GB requiring 29 address bits...we can connect appropriate 29 bits



This Photo by Unknown Author is licensed under CC BY-SA-NC

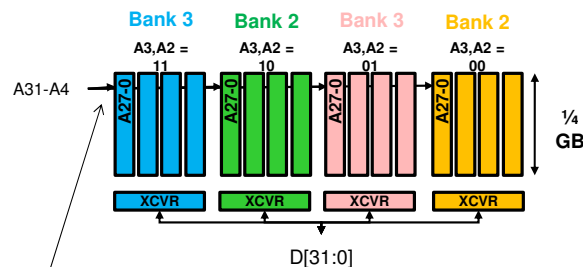
4-Way L.O.I.

- System address bus uses

- A1:A0 and size info to generate /BEi (Byte Enables)

- Lower order bits to select a "bank"

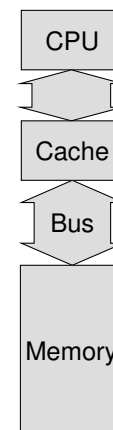
- Upper bits connect to each memory chip



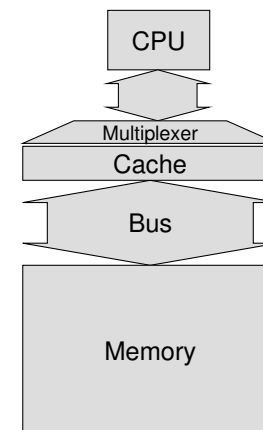
Shift of 4-bits in address connections

Organization Options

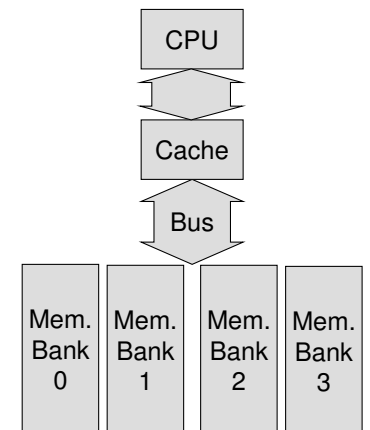
a.) One-word-wide memory Organization



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 |

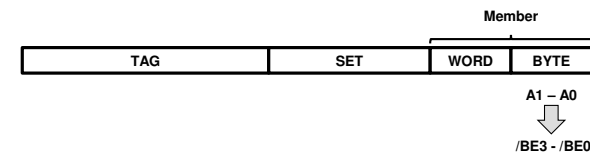
- Find time to access a cache line of 4-words

| | |
|-----------------------|--|
| a. Narrow Memory | (assume mem. controller will auto-increment address) |
| b. Wide Memory | |
| c. Interleaved Memory | |

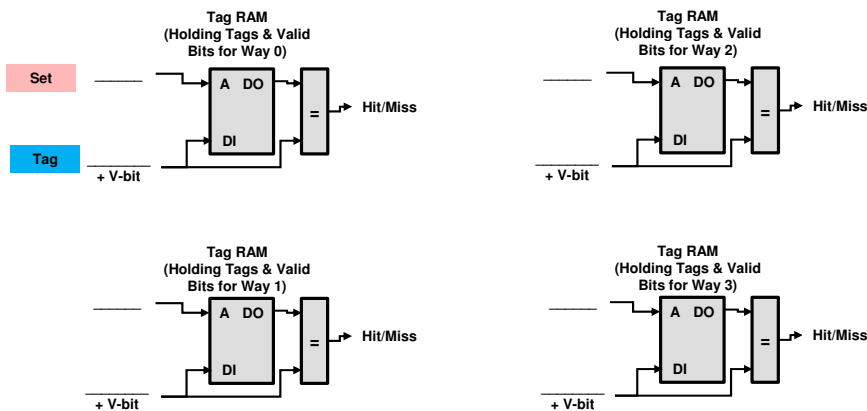
Example

- 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)

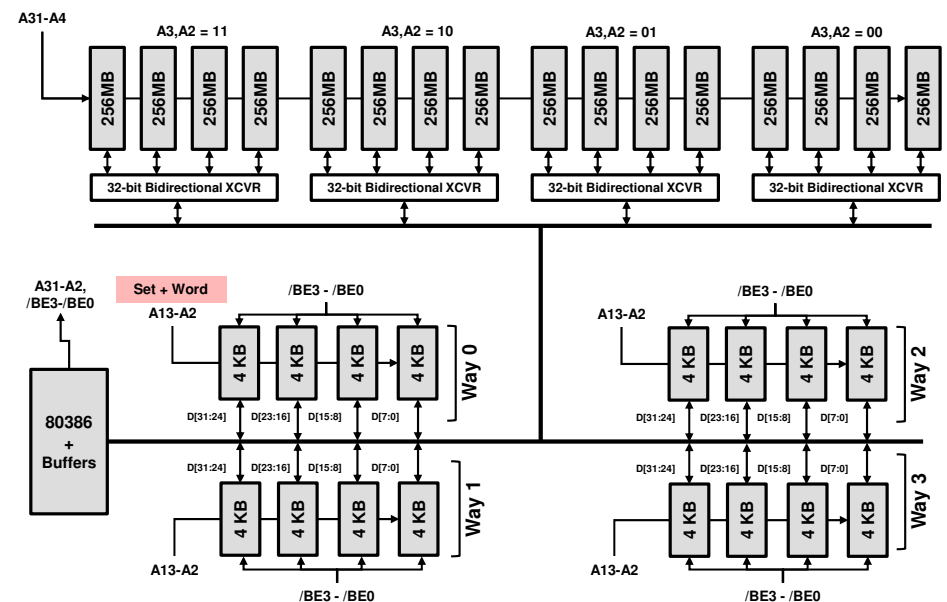
of MM Blocks = _____
 # of Cache Blocks = _____
 # of Sets = _____
 # of Groups = _____



Tag RAM Example



MM & Data RAM Example

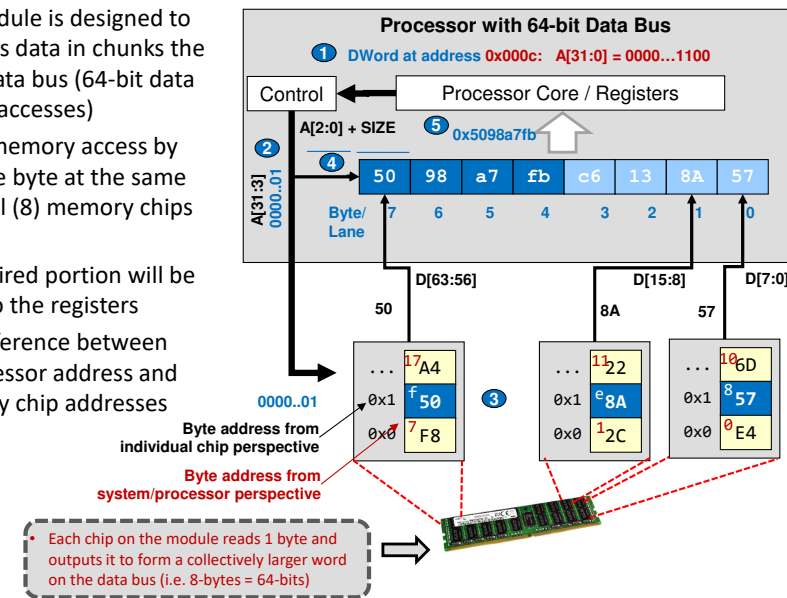


Main memory organization

DRAM TECHNOLOGIES

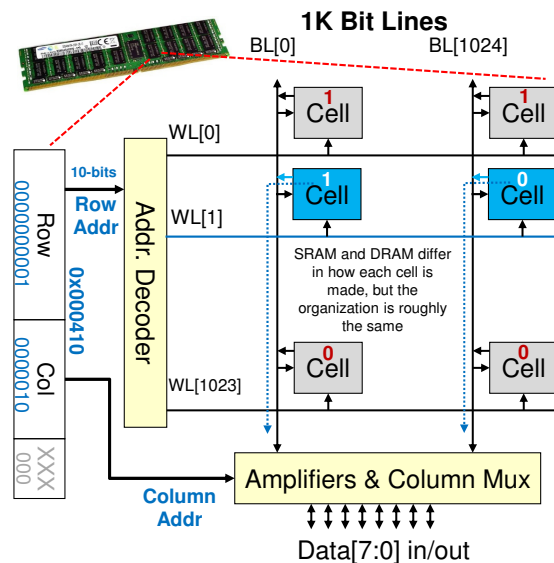
Memory Module Organization

- Memory module is designed to always access data in chunks the size of the data bus (64-bit data bus = 64-bit accesses)
- Parallelizes memory access by accessing the byte at the same location in all (8) memory chips at once
- Only the desired portion will be forwarded to the registers
- Note the difference between system processor address and local memory chip addresses



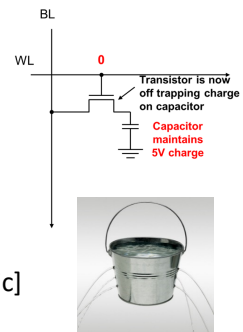
Memory Chip Organization

- Memory technologies share the same layout but differ in their cell implementation
- Memories require the row bits be sent first and are used to select one row (aka "line")
 - Uses a hardware component known as a decoder
- All cells in the selected row access their data bits and output them on their respective "line"
- The column address is sent next and used to select the desired 8 bit lines (i.e. 1 byte)
 - Uses a hardware component known as a mux



SRAM vs. DRAM

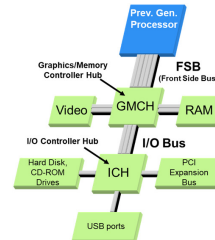
- Dynamic RAM (DRAM) Cells (store 1 bit)
 - Will _____ if not refreshed periodically every few _____ [i.e. dynamic]
 - Extremely small (_____ & a capacitor)
 - Means we can have very high density (GB of RAM)
 - Small circuits require more time to access the bit
 - Used for _____
- Static RAM (SRAM) Cells (store 1 bit)
 - Will retain values as long as _____ [i.e. static]
 - Larger (____ transistors)
 - Larger circuitry can access bit faster
 - FASTER
 - Used for _____ memory



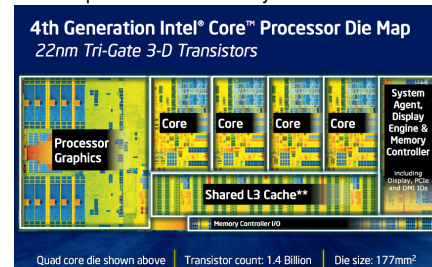
This Photo by Unknown Author is licensed under CC BY-NC

Memory Controller

- DRAMs require non-trivial hardware controller (aka memory controller)
 - To split up the address and send the row and column address as the right time
 - To periodically refresh the DRAM cells
 - Plus more...
- Used to require a separate chip from the processor
- But due to scaling (i.e. Moore's Law) most processors integrate the controller on-chip
 - Helps reduce access time since fewer hops



Legacy architectures used separate chipsets for the memory and I/O controller



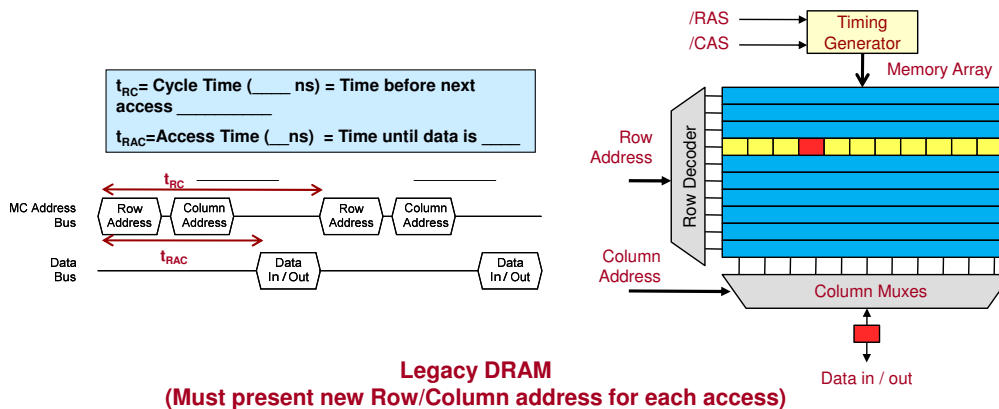
Current general-purpose processors usually integrate the memory controller on chip.

Implications of Memory Technology

- Memory latency of a single access using current DRAM technology will be slow
- We must improve bandwidth
 - Idea 1: Access _____ a single word at a time (to exploit spatial locality)
 - Technology: Fast Page Mode, DDR SDRAM, etc.
 - Idea 2: Increase number of accesses serviced in _____
 - Technology: Banking

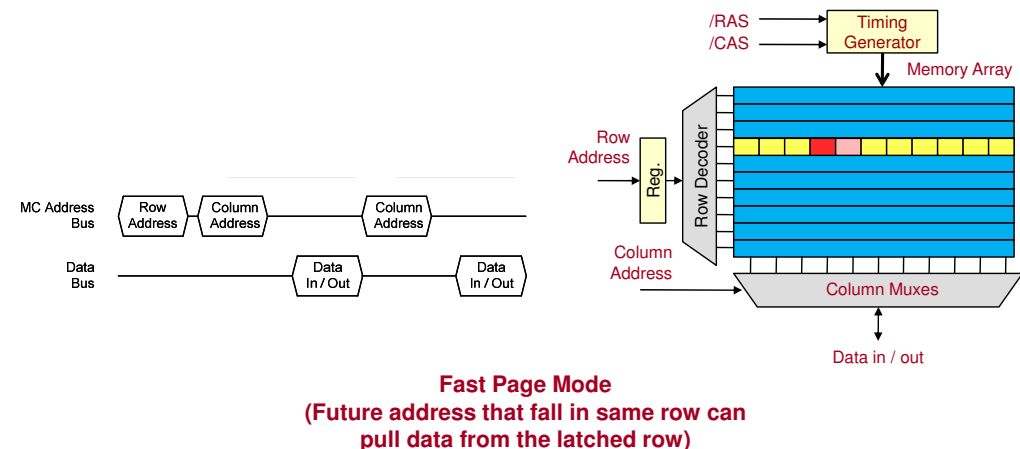
Legacy DRAM Timing

- Can have only a single access “in-flight” at once
- Memory controller must send row and column address portions for each access



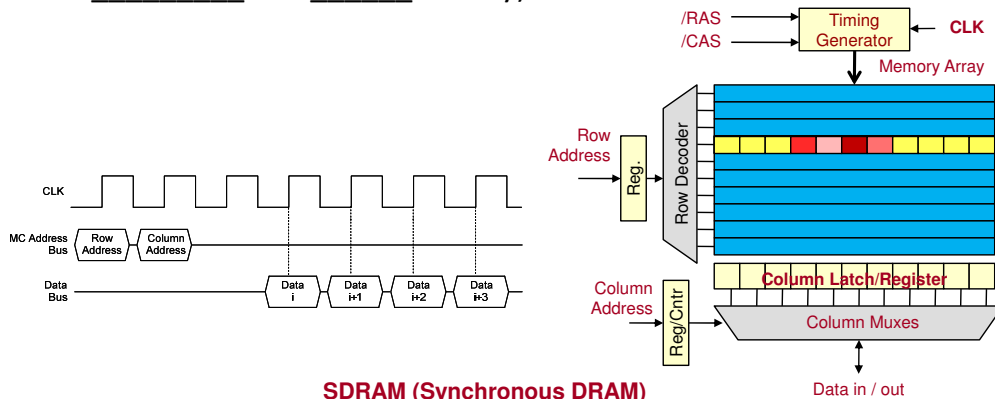
Fast Page Mode DRAM Timing

- Can provide _____ addresses with only one row address



Synchronous DRAM Timing

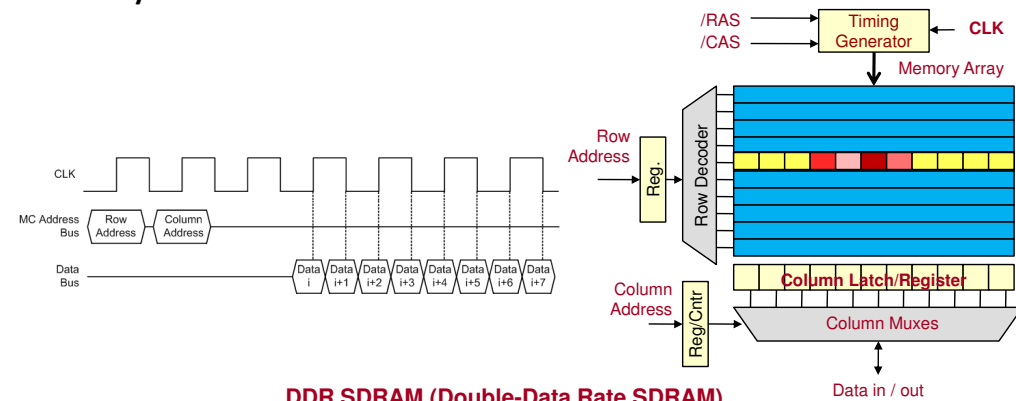
- Registers the column address and automatically increments it, accessing n sequential data words in n successive clocks called _____ ... $n =$ _____ usually)



SDRAM (Synchronous DRAM)
Addition of clock signal. Will get up to 'n' consecutive words in the next 'n' clocks after column address is sent

DDR SDRAM Timing

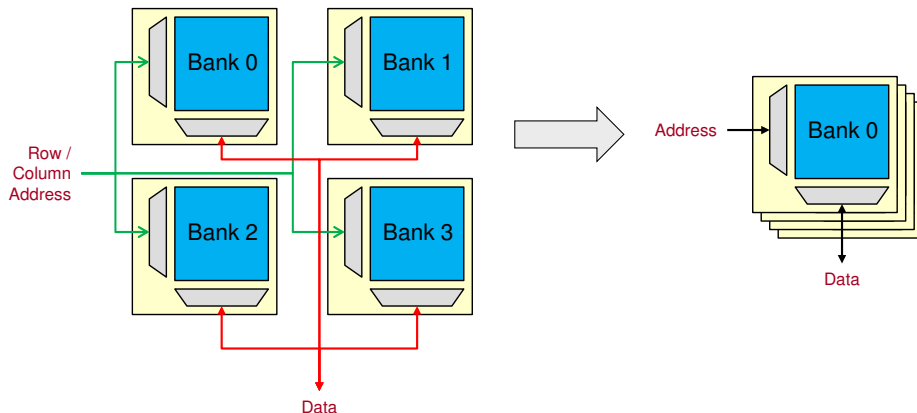
- Double data rate access data every _____ clock cycle



DDR SDRAM (Double-Data Rate SDRAM)
Addition of clock signal. Will get up to '2n' consecutive words in the next 'n' clocks after column address is sent

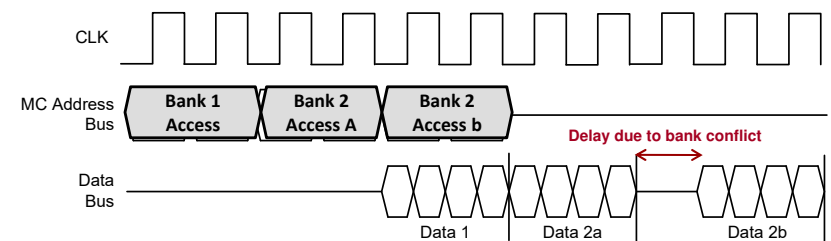
Banking

- Divide memory into "banks" duplicating row/column decoder and other peripheral logic to create _____ memory arrays that can access data in _____
— uses a _____ of the address to determine which bank to access



Bank Access Timing

- Consecutive accesses to different banks can be _____ and hide the time to access the row and select the column
- Consecutive accesses within a bank (to different rows) _____ the access latency



Access 1 maps to bank 1 while access 2a maps to bank 2 allowing parallel access. However, access 2b immediately follows and maps to bank 2 causing a delay.

Programming Considerations

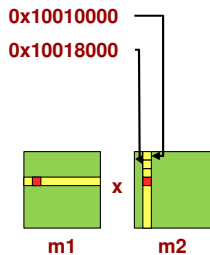
- For memory configuration given earlier, accesses to the same bank but different row occur on an 32KB boundary
- Now consider a matrix multiply of 8K x 8K integer matrices (i.e. 32KB x 32KB)
- In code below...m2[0][0] @ 0x10010000 while m2[1][0] @ 0x10018000

| Unused | Row | Bank | Col. | Unused |
|---------|--------------------|---------|------------|--------|
| A31-A29 | A28...A15 | A14,A13 | A12...A3 | A2..A0 |
| 00 | 1 0000 0000 0001 0 | 00 | 0000000000 | 000 |
| 00 | 1 0000 0000 0001 1 | 00 | 0000000000 | 000 |

```

int m1[8192][8192], m2[8192][8192], result[8192][8192];
int i,j,k;
...
for(i=0; i < 8192; i++){
  for(j=0; j < 8192; j++){
    result[i][j]=0;
    for(k=0; k < 8192; k++){
      result[i][j] += matrix1[i][k] * matrix2[k][j];
    } }
  }

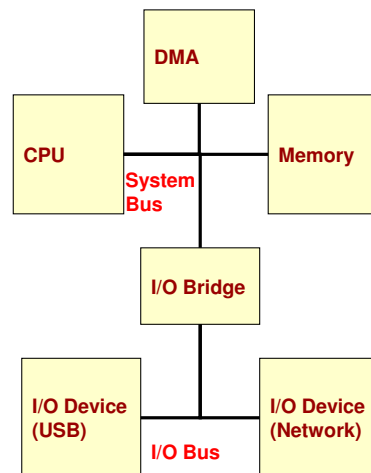
```



DMA

Direct Memory Access (DMA)

- Large buffers of data often need to be copied between:
 - _____ (video data, network traffic, etc.)
 - _____ (OS space to user app. space)
- DMA devices are small hardware devices that copy data from a source to destination freeing the processor to do _____



Data Transfer w/o DMA

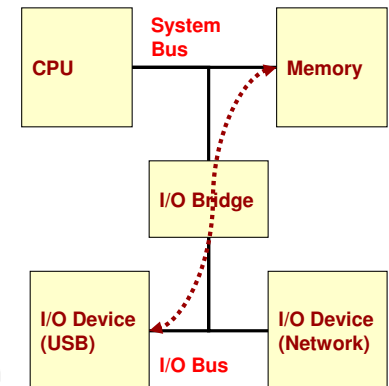
- Without DMA, processor would have to move data using a loop
- Move 16Kwords pointed to by (\$s1) to (\$s2)

```

li      $t0,16384
AGAIN:  lw      $t1,0($s1)
        sw      $t1,0($s2)
        addi    $s1,$s1,4
        addi    $s2,$s2,4
        subi    $t0,$t0,1
        bne     $t0,$zero,AGAIN

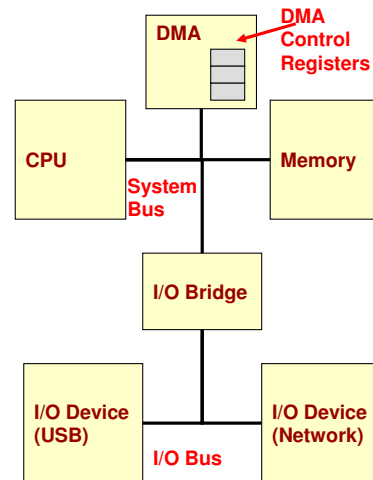
```

- Processor wastes valuable execution time moving data



Data Transfer w/ DMA

- Processor sets values in DMA control registers
 - _____ Address
 - _____ Address
 - _____
 - Control & Status (Start, Stop, Interrupt on Completion, etc.)
- DMA becomes “_____”
(controls system bus to generate reads and writes) while processor is free to execute other code
 - Small problem: _____
 - Hopefully, data & code needed by the CPU will reside in _____



DMA Engines

- Systems usually have multiple DMA engines/channels
- Each can be configured to be started/controlled by the processor or by certain I/O peripherals
 - Network or other peripherals can initiate DMA's on their behalf
- Bus arbiter assigns control of the bus
 - Usually winning requestor has control of the bus until it relinquishes it (turns off its request signal)

