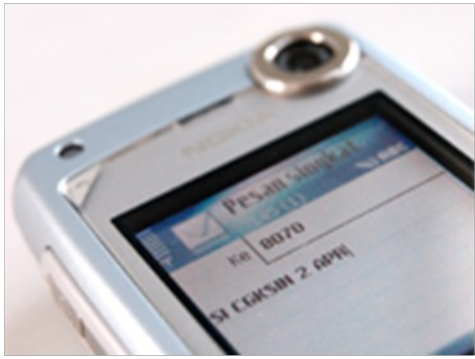




Computer Architecture

Random Access Memory Technologies

Gábor Horváth
associate professor
BUTE Dept. Of Networked Systems and Services
ghorvath@hit.bme.hu

2017. március 17.
Budapest

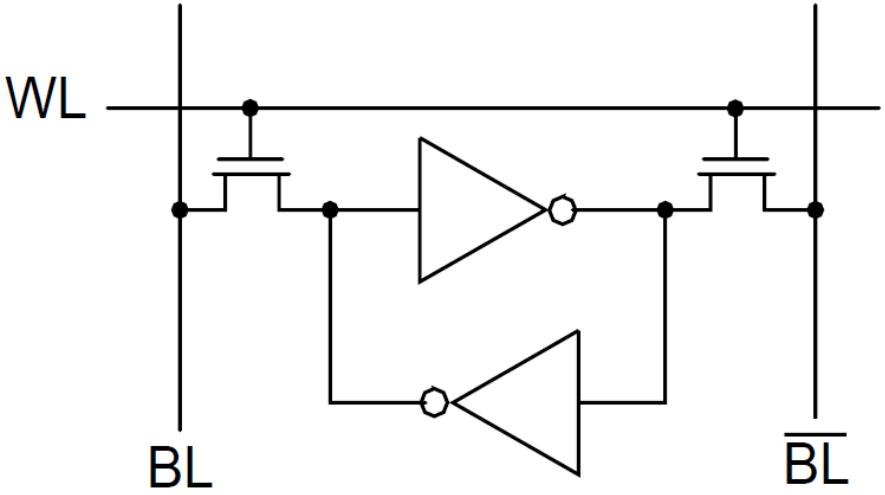
Storing data

- Possible types of memories:
  - ROM: read-only
    - Classical ROM: the content is stored during the manufacturing process
    - PROM: one-time programmable
    - EPROM: can be erased using ultraviolet light
    - Etc.
  - SRAM: Static Random Access Memory
    - Can be read and modified any time
  - DRAM: Dynamic Random Access Memory
    - Can be read and modified any time
    - It forgets its content! Needs to be refreshed periodically.

Storing data in SRAM

Storing a single bit in SRAM

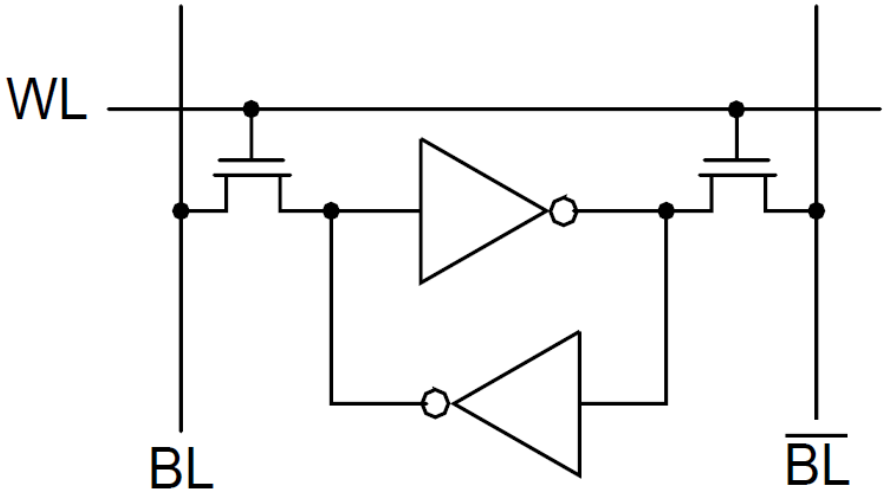
- An SRAM cell consists of a bi-stable flip-flop
  - The bit and its inverse is also available



  - WL = word line: selects the memory cell for operation
  - BL = bit line: reflects the stored bit once the cell is selected
  - $\overline{BL}$ = the inverse of the stored bit
- It does not forget the data till power supply is present

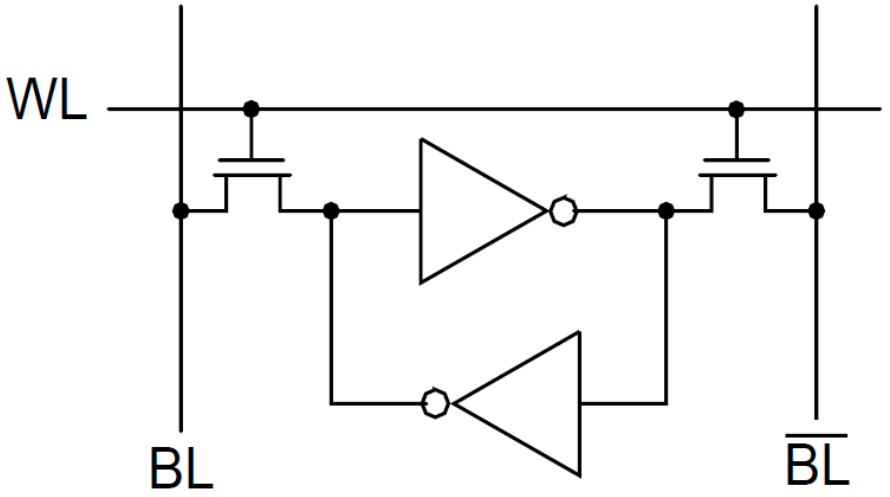
Storing a single bit in SRAM

- **Reading** the value of the bit:
  - The BL and $\overline{BL}$ are precharged
  - A logical „high” value is given to the WL
    - The switches represented by the transistors get closed
    - Selects the cell
    - The BL will be equal to the bit stored
    - The $\overline{BL}$ will be equal to the inverse of the bit stored
  - The sense amplifiers detect the difference of BL and $\overline{BL}$
  - ...providing the bit stored in the cell



Storing a single bit in SRAM

- **Writing** the value of the bit:
  - The BL and $\overline{BL}$ lines are set according to the bit to store
    (BL=1, $\overline{BL}$=0 if a bit 1 needs to be stored and vice versa)
  - A logical „high” value is given to the WL
    - The switches represented by the transistors get closed
    - Selects the cell
    - Since the driver transistors of the BL and BL are stronger than the transistors of the cell, the value if the bit is forced to the flip-flop

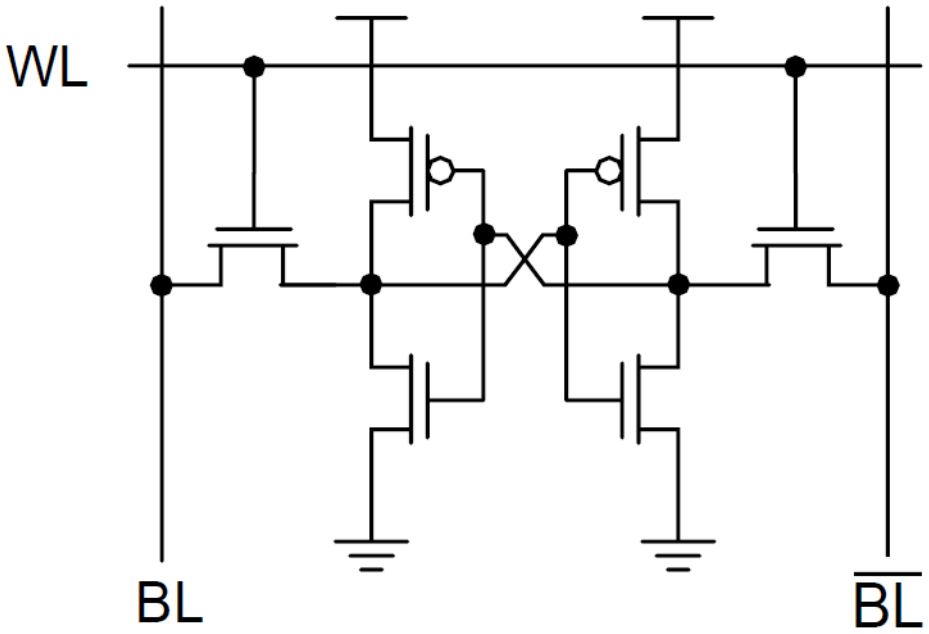


Internals of an SRAM cell

- Each inverter is implemented by two transistors

  → **6 transistors are needed to store a single bit!**

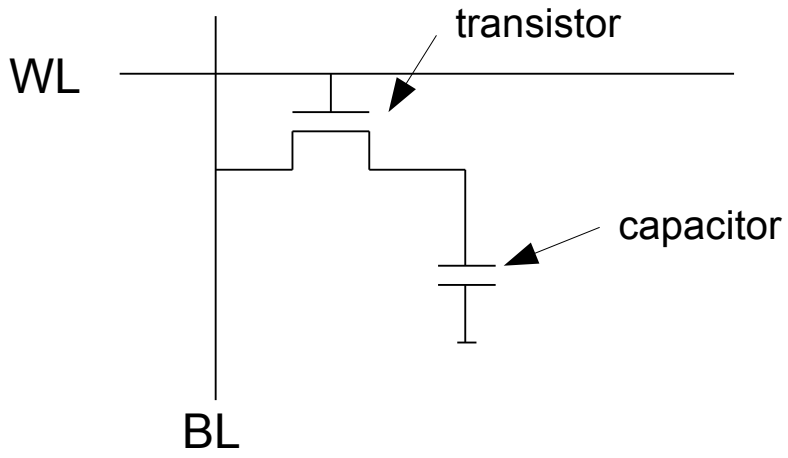
- This makes SRAM very expensive



Storing data in DRAM

Storing a single bit in DRAM

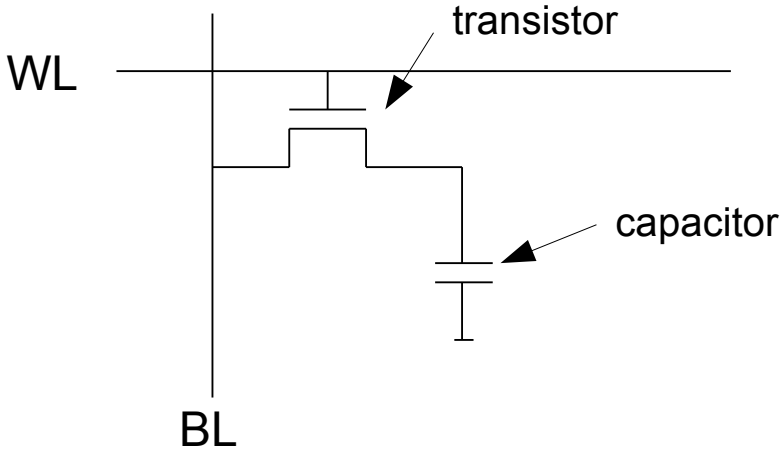
- A DRAM cell consists of a capacitor and a transistor
  - The capacitor stores the data:
    - There is charge: bit 1
    - No charge: bit 0



  - WL = word line: selects the memory cell for operation
  - BL = bit line: reflects the stored bit once the cell is selected
- **It does forget the data with time!** (The charge escapes)
  - Periodical refresh is necessary

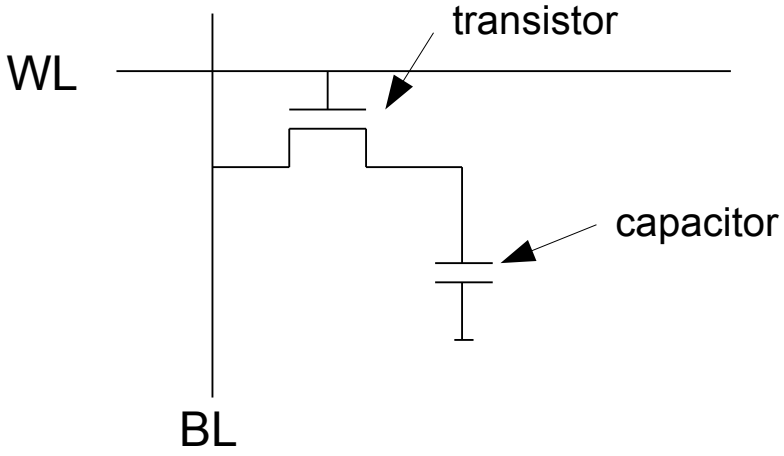
Storing a single bit in DRAM

- **Reading** the value of the bit:
  - The BL is precharged exactly to the middle between logical high and low
  - A logical „high” value is given to the WL
    - The switch represented by the transistor get closed
    - Selects the cell
    - The charge of the capacitor (if any) leaves towards the BL
  - The sense amplifiers detects the level of BL
  - ...providing the bit stored in the cell
- **Reading the bit is destructive!**
  - The charge representing the bit leaves



Storing a single bit in DRAM

- **Writing** the value of the bit:
  - The BL is set according to the bit to store
  - A logical „high” value is given to the WL
    - The switch represented by the transistor gets closed
    - Selects the cell
    - The charge of BL charges the capacitor



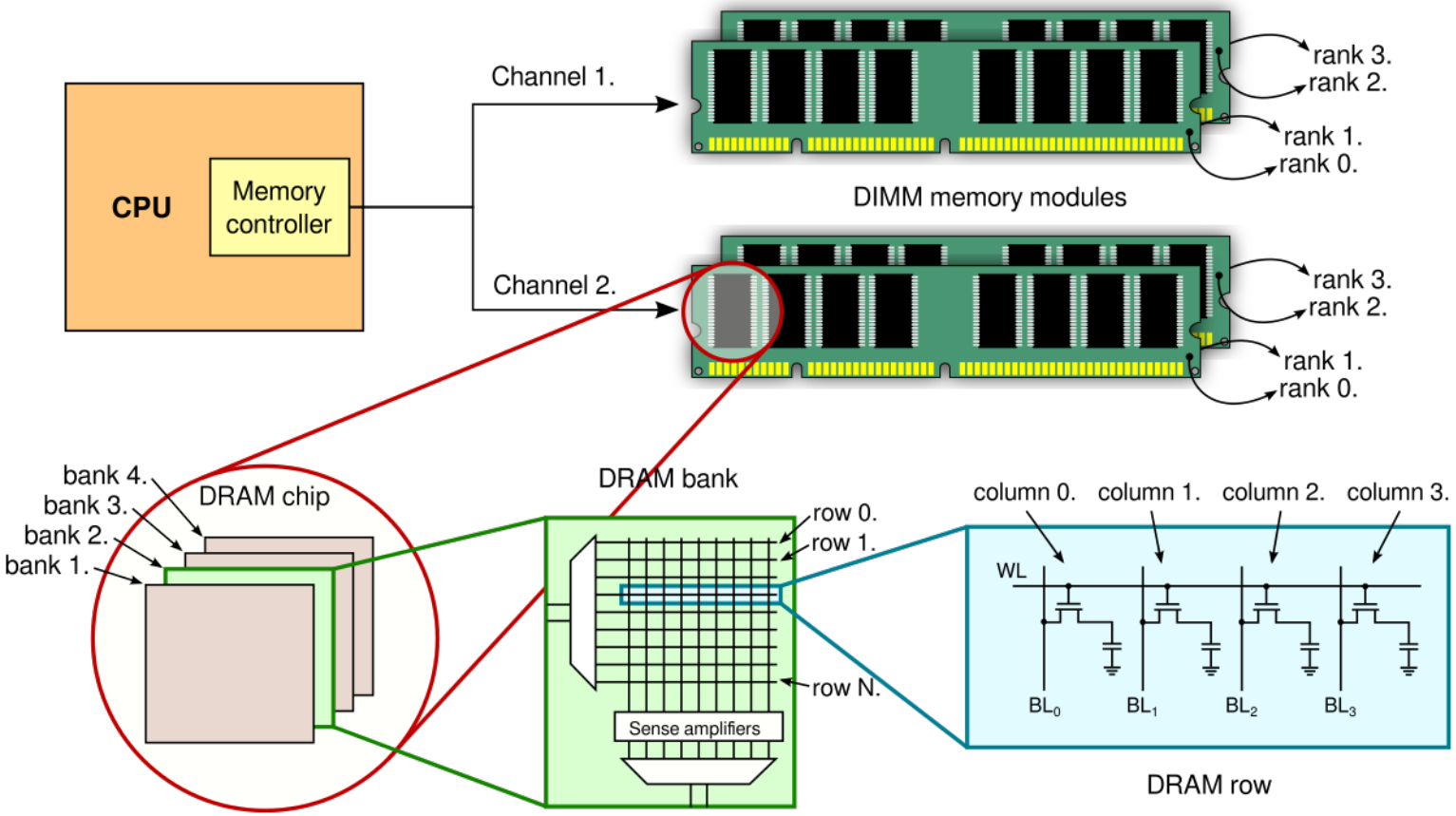
SRAM vs. DRAM

- Cost:
  - SRAM: 6 transistors vs DRAM: 1 transistor + 1 capacitor
  - The **DRAM is much cheaper**, and higher data density can be achieved
- Speed:
  - SRAM: Reading out the bit means detecting the state of a flip-flop (1-2 ns)
  - DRAM: Reading out the bit means detecting the extremely small charge stored in the capacitor (10-20 ns)
  - Reading out a bit from an **SRAM is much faster**
- Integration:
  - SRAM can be integrated with the CPU as they share the same manufacturing process
  - DRAM is produced by a different manufacturing process
- Applications:
  - SRAM: cache memory
  - DRAM: system memory

DRAM based memory systems

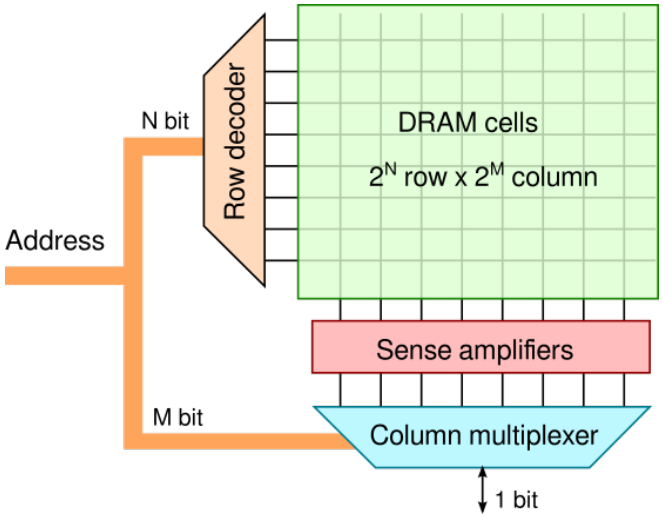
Overview

- How to create a memory system for the computer
  - It must be: cheap, large, low latency, high throughput



DRAM bank

- Structure:



- DRAM cells in a 2D grid
  - Each row shares the same word line
  - Each column shares the same bit line
- Reading:
  - The **row decoder** selects (activates) a row
  - The **sense amplifiers** detect and store the bits of the row
  - The **column multiplexer** selects the desired column from the row
- Two-phase operations:
  - To reduce the width of the address bus
  - Address bus: row address → wait → address bus: column address → data bus: the desired data

DRAM commands

- The 5 most important commands:
  - **ACTIVATE**
    - Opens a row (data → moved to the sense amplifiers)
  - **READ**
    - Reads a column from the open row
    - It reads from the sense amplifiers
  - **WRITE**
    - Writes a data to the open row
    - It writes to the sense amplifiers
  - **PRECHARGE**
    - Closes the open row
    - Precharges the bit lines to make the next row activation fast
  - **REFRESH**
    - Refreshes the content of a row
    - Almost an activate + precharge

DRAM commands

- Example read requests:

```
(row 3, column 8)
(row 3, column 14)
(row 1, column 3)
(row 1, column 4)
```

- Commands (assume the DRAM is precharged initially):

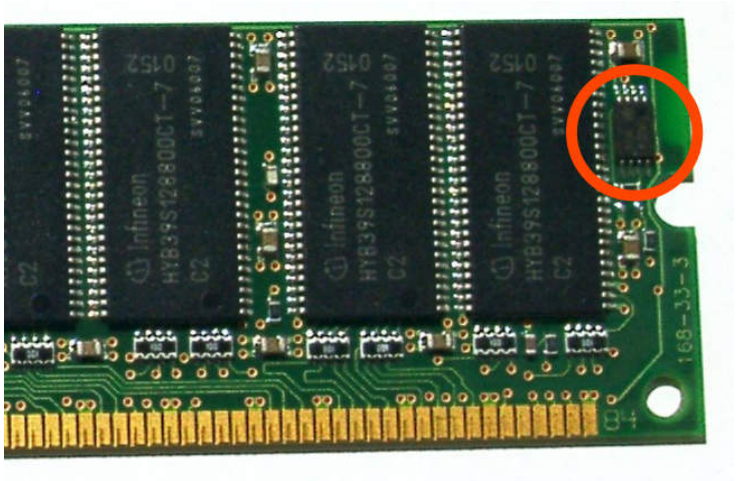
```
ACTIVATE 3
READ 8
READ 14
PRECHARGE
ACTIVATE 1
READ 3
READ 4
```

DRAM timing

- The execution time of the commands
- The 4 most important ones:
  - T_{RCD}: The time needed to open a row
  - T_{CAS} (CL): The delay between receiving the column address and the appearance of the data
  - T_{RP}: The delay of the PRECHARGE command
  - T_{RAS}: The minimal time a row must stay open
- There are many of these
- Unit: clocks (synchronous DRAM), ns (asynchronous)
- A DRAM module parameterized by 8-9-10-11 means:
  - T_{CAS}=8, T_{RCD}=9, T_{RP}=10, T_{RAS}=11
- If only „CL7” is provided: T_{CAS}=7

DRAM timing

- How does the memory controller know the timing values?
  - It asks the memory modules
  - Components of the memory modules:
    - DRAM chips
    - ...and an **SPD** chip! It stores the timing parameters (among others)



Making DRAMs more efficient

- For every column read operation targetting the same row:

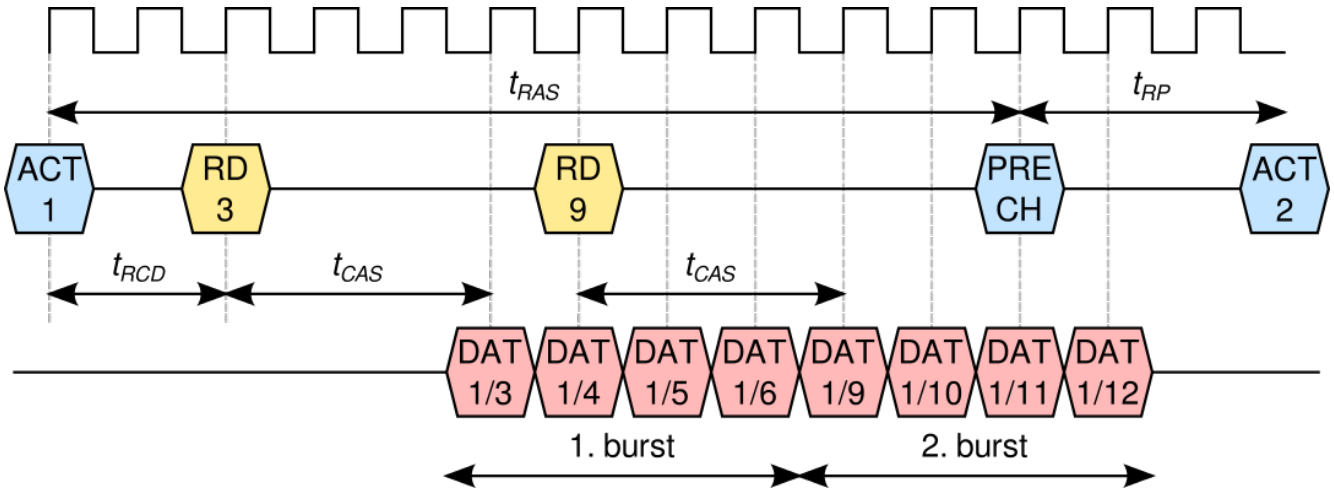
  Column address → T_{CAS} → appearance of the data

  - Waste of time!
- **Burst mode**
  - Column address is given to the DRAM

    … the response is not a single column, but a burst (a series of columns)!
  - Burst length: configuration parameter
  - Minimal burst length:
    - DDR: 2 columns
    - DDR2: 4 columns
    - DDR3: 8 columns
    - DDR4: 8 columns
- **The commands and the data can be overlapped**
  - The next command does not have to wait till the current one finishes

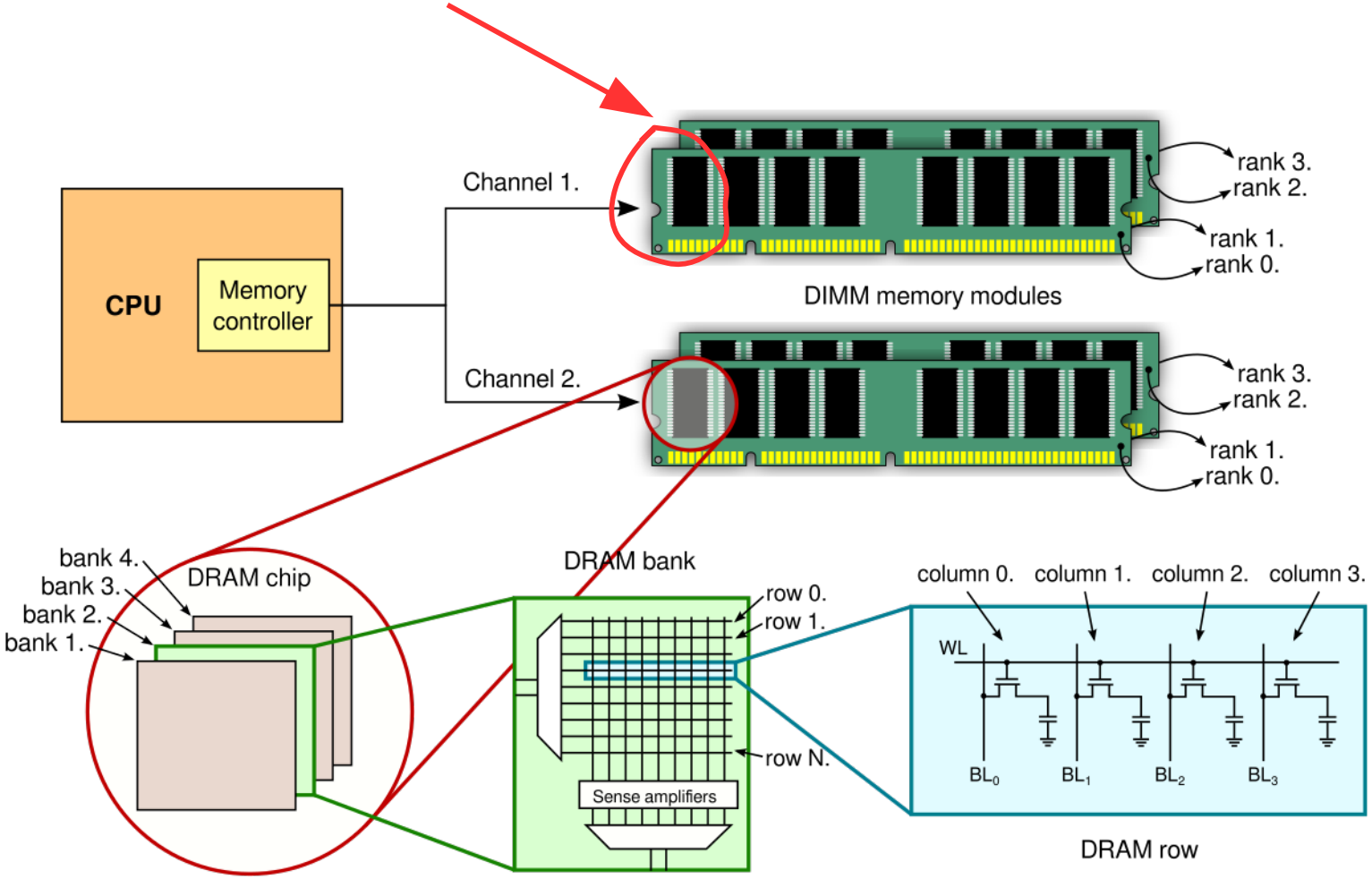
Making DRAMs more efficient

- Example:
  - Reading from row 1, starting at columns 3 and 9



DRAM chips

DRAM chip



DRAM chips

- A DRAM chip consists of banks
- Bank:
  - Independent DRAM cell grids
  - Each of them have their own row decoders, column multiplexers, sense amplifiers
  - Each of them has a row open

    → There can be more open rows in a DRAM chip (one in each bank)

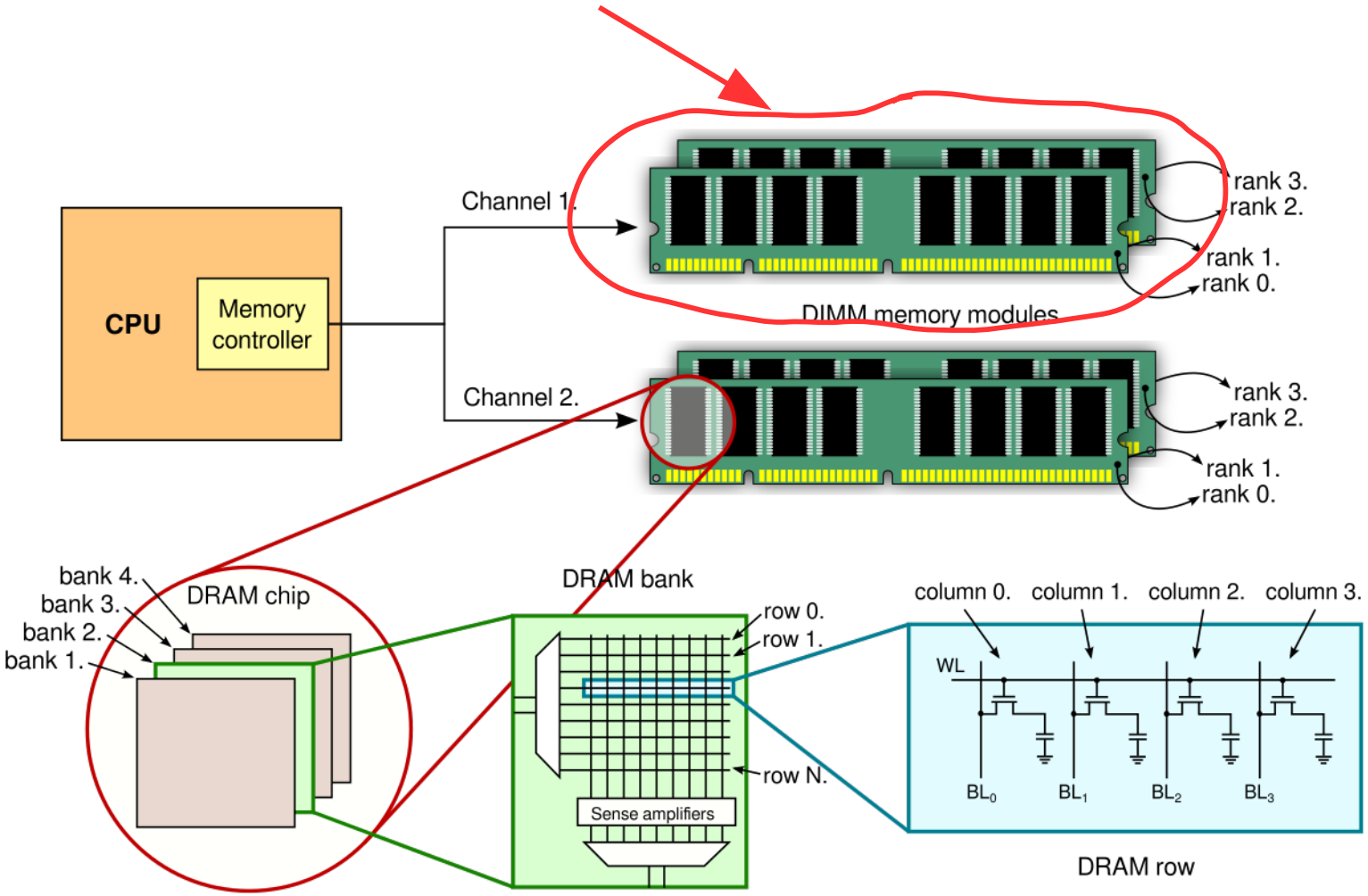
    → **The latency is reduced (less row activations are needed)!**

DRAM chips

- Banks do not store individual bits
  - One column: 4 bit, 8 bit, 16 bit (x4, x8, x16)
- Interface:
  - **Command lines**
    - What the chip has to do (ACTIVATE, READ, WRITE, etc.)
  - **Bank selection lines**
    - The bank the command is given to
  - **Address lines**
    - ACTIVATE: row address
    - READ/WRITE: column address
  - **Data lines**
    - 4, 8, or 16 bit

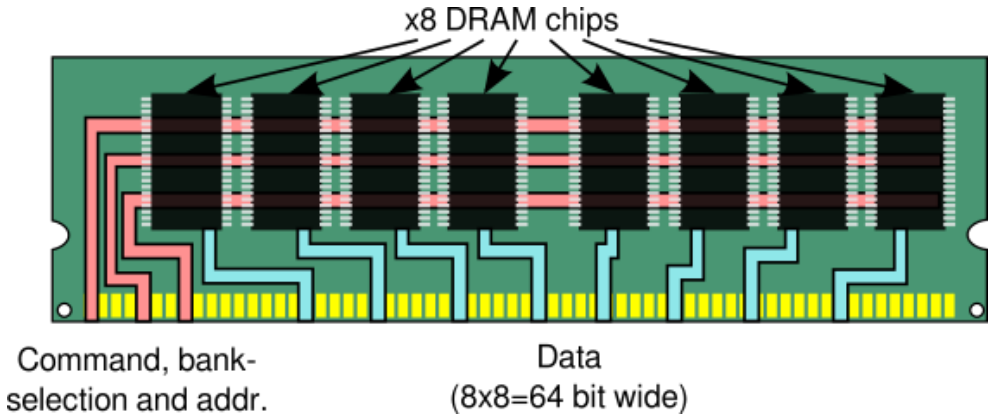
DRAM memory modules

Memory module



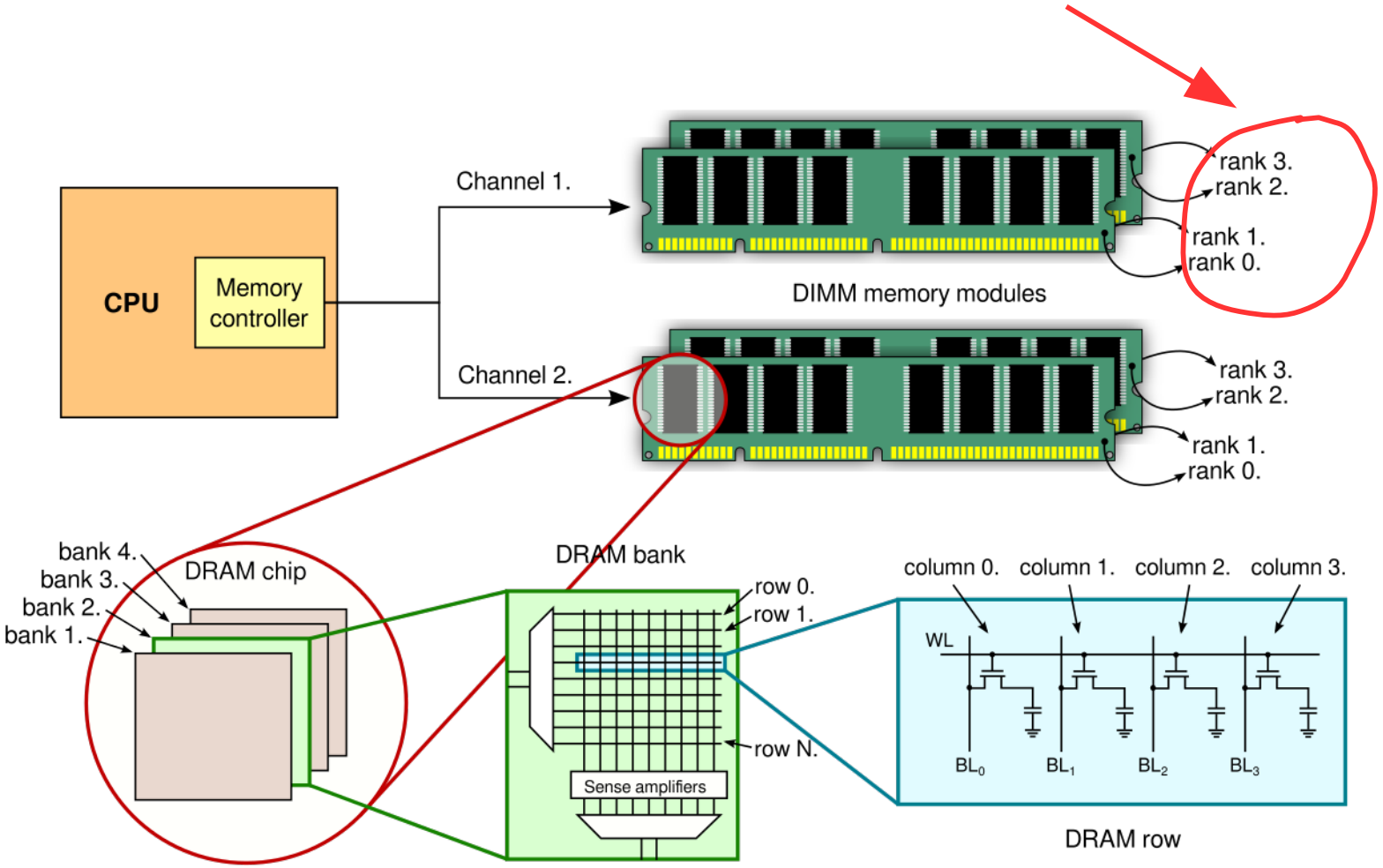
DRAM memory module

- A memory module consists of DRAM chips
- Command lines, bank selection lines, address lines: shared
- Data lines: concatenated



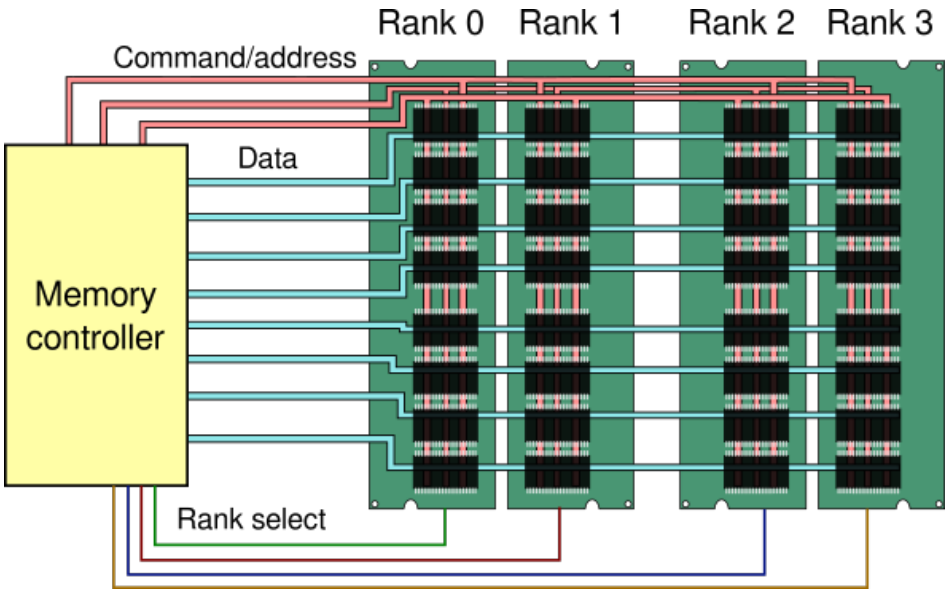
- Each chip receives all commands
- Effect:
  - Throughput increases 8x
  - Delay: the same

DRAM ranks


DRAM ranks
CPU
Memory controller
Channel 1.
Channel 2.
rank 3.
rank 2.
rank 1.
rank 0.
DIMM memory modules
bank 4.
bank 3.
bank 2.
bank 1.
DRAM chip
DRAM bank
row 0.
row 1.
row N.
Sense amplifiers
column 0.
column 1.
column 2.
column 3.
WL
BL0
BL1
BL2
BL3
DRAM row

DRAM ranks

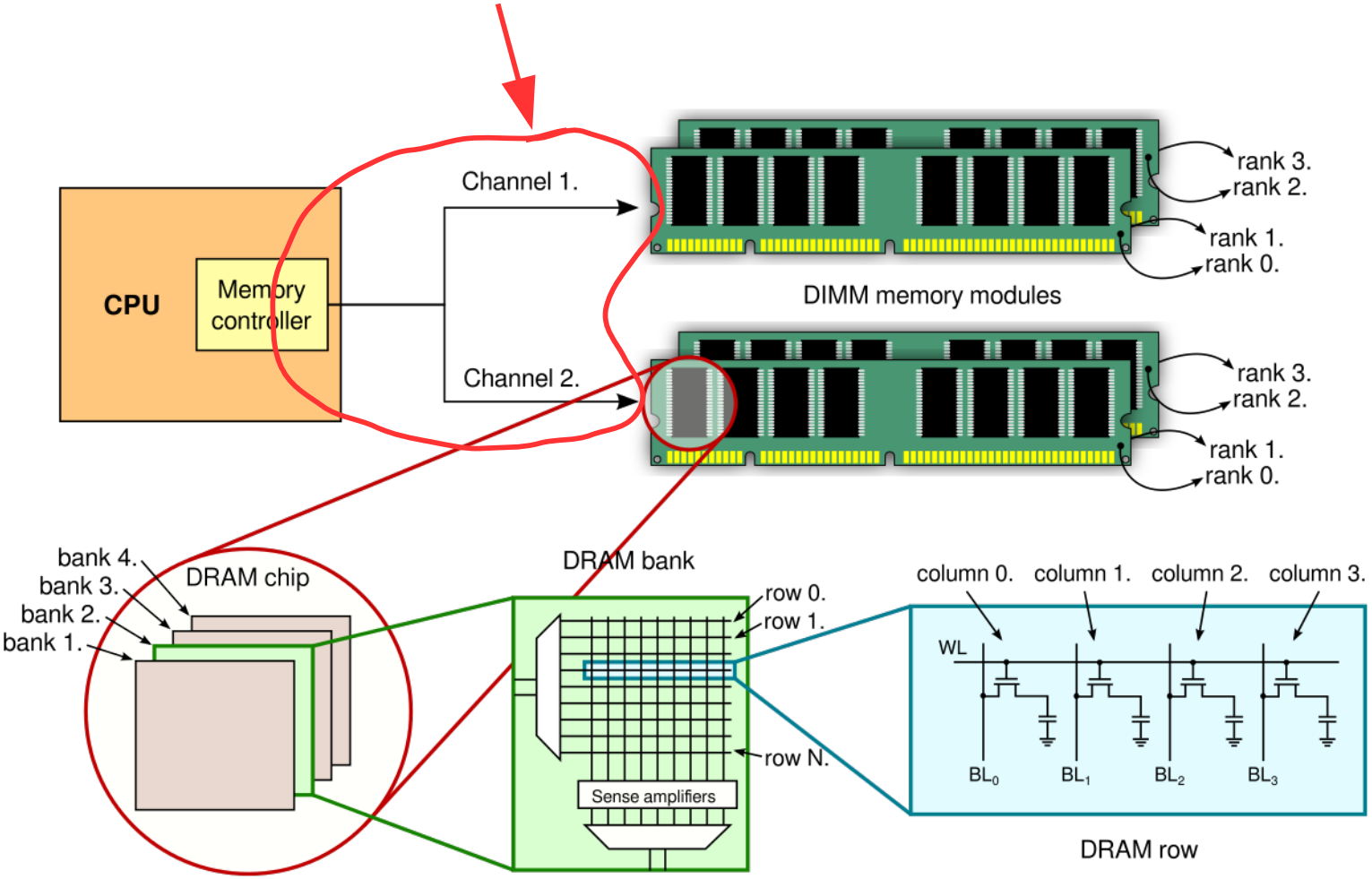
- To increase storage capacity
- Independent memory devices
- All lines are shared
  - ...but only a single rank can be enabled at a time → rank select lines



- Effect:
  - Throughput is the same as without multiple ranks
  - Delay: better (more banks, more open rows)

Multi-channel memory access

Multi-channel memory access

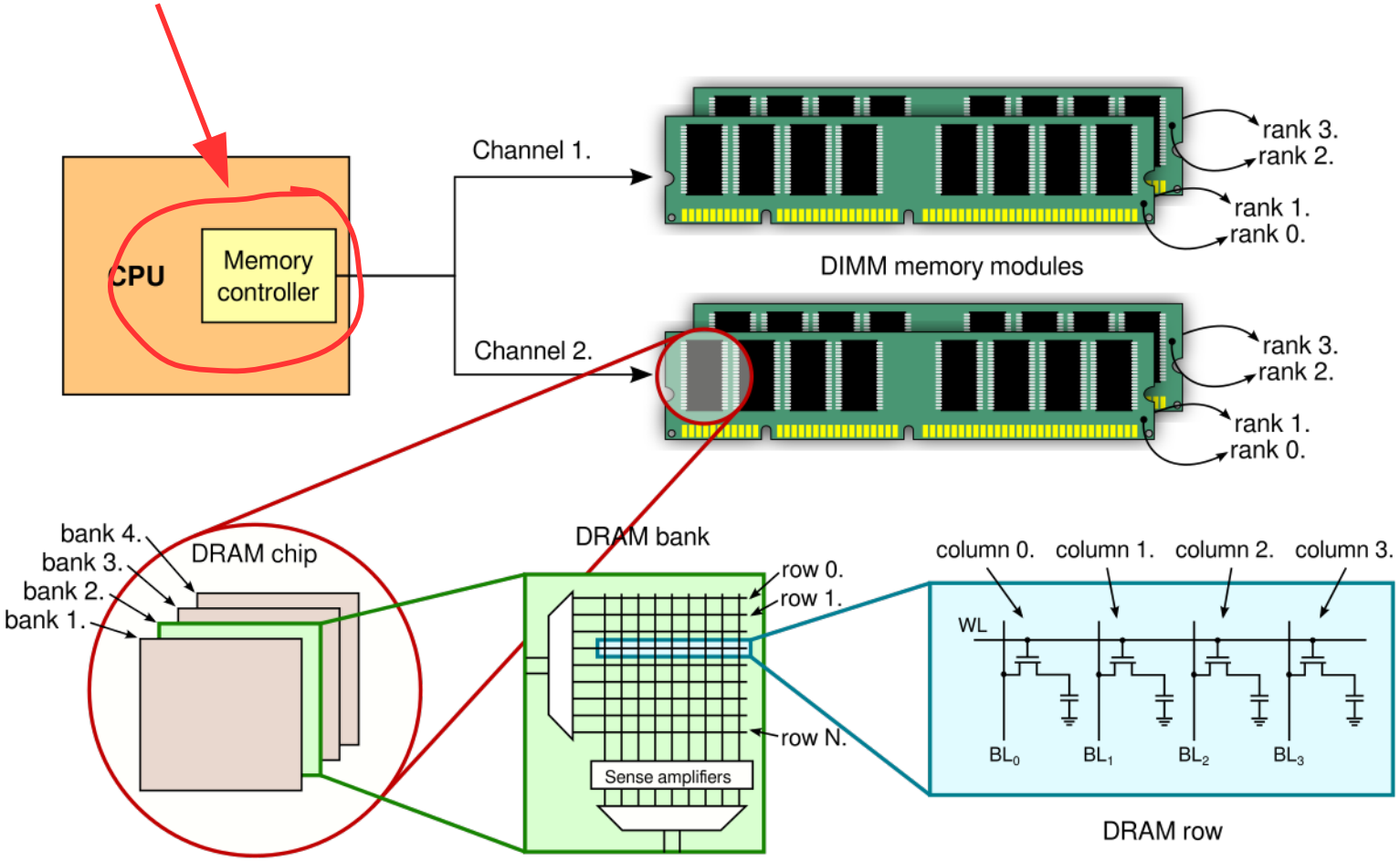


Multi-channel memory access

- Synchronized case:
  - If all modules are identical (size, timing, etc. are the same)
  - Operate in lock step
  - 2x 64 bit wide channels → 1x 128 bit wide channel
- Indipendent channels
  - Modules dont have to be identical in different channels
  - Every channel has its own memory controller

The memory controller

Memory controller



The memory controller

- Purpose:
  - It serves memory read/write requests (coming from the CPU and the I/O devices)
- Main tasks:
  - It translates the memory addresses to channel/rank/bank/row/column coordinates
  - Re-orders memory read/write requests
  - Open row management
  - Scheduling DRAM refresh commands

The memory controller

- Optimizing the order of read/write requests:
  - To minimize the row activation and precharge operations (that have a significant delay)
    - First-Come-First-Serve scheduling (no optimization)
    - First-Response-First-Come-First-Serve scheduling (optimized)

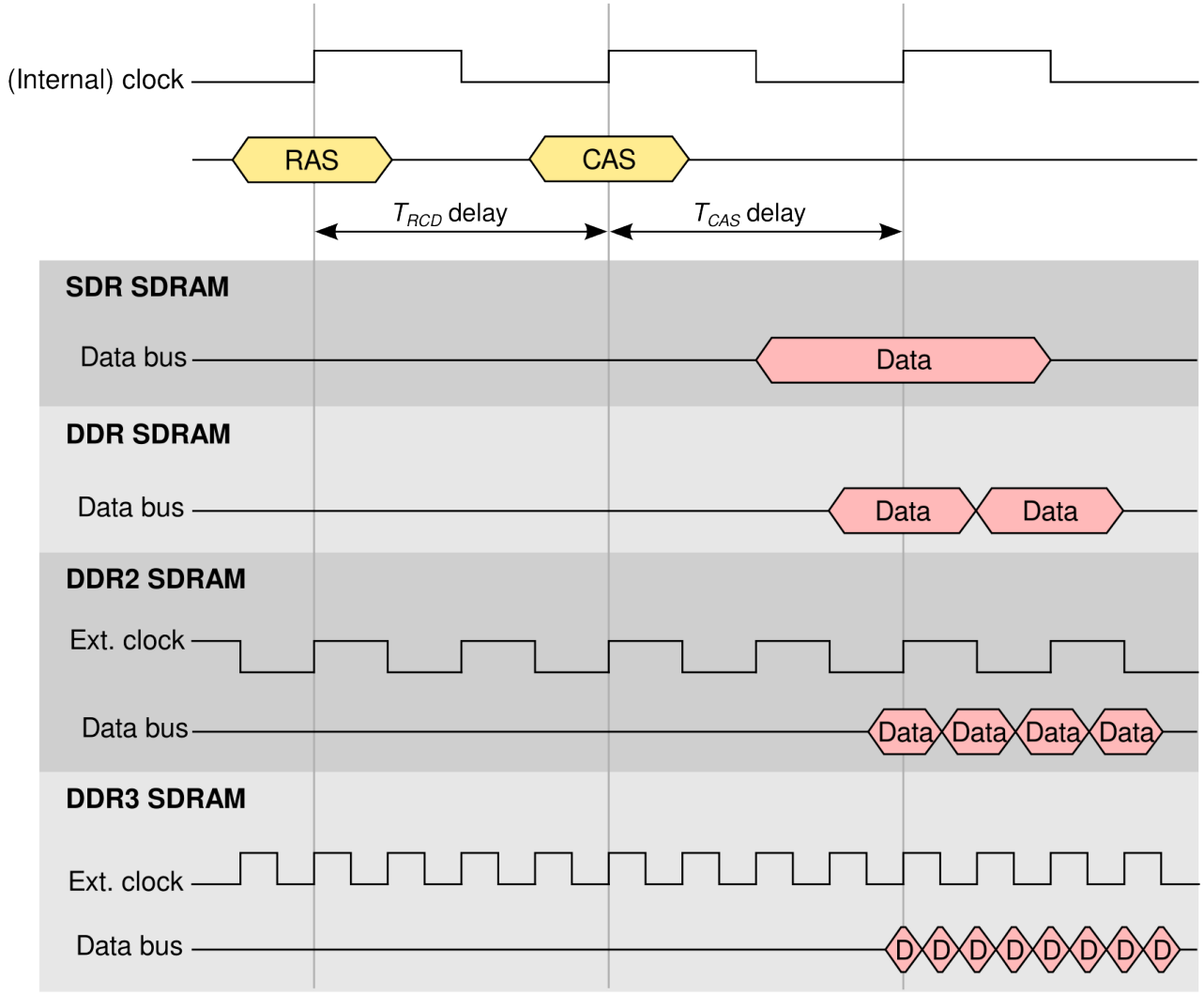
	FCFS	FR-FCFS
Requests:	(row 3, column 8) (row 1, column 3) (row 3, column 14)	(row 3, column 8) (row 3, column 14) (row 1, column 3)
Commands:	ACTIVATE 3 READ 8 PRECHARGE ACTIVATE 1 READ 3 PRECHARGE ACTIVATE 3 READ 14	ACTIVATE 3 READ 8 READ 14 PRECHARGE ACTIVATE 1 READ 3

The memory controller

- Open row management
- All read/requests are served. What to do with the active row?
  - **Dont close it**:
    - If the next requests falls to the same row, we dont have to open it (no extra delay)
    - If the next request falls to a different row, we have to close the current one before opening the next one (extra delay)
  - **Close it**:
    - If the next requests falls to the same row, we have to open it again (extra daly)
    - If the next request falls to a different row, we dont have to close the current one before opening the next one (no extra delay)
  - **Adaptive**:
    - Speculates
    - APM (Active Page Management)

      Core i7: „Adaptive Page Closing” option in the BIOS

Synchronous DRAM systems



Synchronous DRAM systems

- **DDR SDRAM** (Double Data Rate SDRAM)
  - Doubles the transfer speed of the burst
  - It transmits data at both the rising and the falling edge of the clock
  - DDR-200 memory operates at 100 MHz only!
    - The „200” means that the burst is transmitted as fast as a 200 MHz single data rate SDRAM can transmit
  - Notation of DDR SDRAMs:
    - Let us have a DDR SDRAM operating at 200 MHz having 64 bit data units
    - It is sold as:
      - DDR-400
      - PC-3200
        - » As this memory transmits bursts at 400 Mega-data units per second
        - » Data units are 8 bytes → speed is 3200 MB/s

Synchronous DRAM systems

- **DDR2 SDRAM**
  - Classical DDR: 2 data units / clock (raising and falling edge)
  - DDR2: 4 data units / clock
  - DDR2-800 memory operates at 200 MHz only!
    - The „800” means that the burst is transmitted as fast as a 800 MHz single data rate SDRAM can transmit
  - Notation of DDR2 SDRAMs:
    - Let us have a DDR2 SDRAM operating at 200 MHz having 64 bit data units
    - It is sold as:
      - DDR2-800
      - PC2-6400
        - » As this memory transmits bursts at 800 Mega-data units per second
        - » Data units are 8 bytes → speed is 6400 MB/s

Synchronous DRAM systems

- **DDR3 SDRAM**
  - Classical DDR: 2 data units / clock (raising and falling edge)
  - DDR2: 4 data units / clock
  - DDR3: 8 data units / clock
  - DDR3-1600 memory operates at 200 MHz only!
    - The „1600” means that the burst is transmitted as fast as a 1600 MHz single data rate SDRAM can transmit
  - Notation of DDR3 SDRAMs:
    - Let us have a DDR3 SDRAM operating at 200 MHz having 64 bit data units
    - It is sold as:
      - DDR3-1600
      - PC3-12800
        - » As this memory transmits bursts at 1600 Mega-data units per second
        - » Data units are 8 bytes → speed is 12800 MB/s

Synchronous DRAM systems

- DDR4 SDRAM
  - Still transfers 8 data units/clock (like DDR3)
  - Internal clock rate is increased to improve throughput

Comparison

	SDR	DDR	DDR2	DDR3	DDR4
Internal clock	66-133 MHz	133-200 MHz	100-200 MHz	100-200 MHz	266-533 MHz
Data/int. clock	1	2	4	8	8
Throu. MB/s	528-1064	2128-3200	3200-6400	6400-12800	17024-34112
Burst length	1-8	2-8	4-8	8	8
Voltage	3.3V	2.5V	1.8V	1.5V	1.05-1.2V

Synchronous DRAM systems

- Conclusion
  - Internal clock rate is almost the same in the last 10-15 years

    → **latency is the same**

    (latency: delay between the address and the corresponding data)
  - Data units transfered / clock cycle increased significantly

    → **throughput is improving**

    (Throughput: amount of data transmitted / second)