Αρχιτεκτονική Υπολογιστών

Κεφάλαιο #5 (b)

Συστήματα Μνήμης

Διονύσης Πνευματικάτος

pnevmati@cslab.ece.ntua.gr

5ο εξάμηνο ΣΗΜΜΥ – Ακαδημαϊκό Έτος: 2019-20

Τμήμα 3 (ΠΑΠΑΔ-Ω)

http://www.cslab.ece.ntua.gr/courses/comparch/

Memory Hierarchy Technology

- Random Access:
  - "Random" is good: access time is the same for all locations
  - DRAM: Dynamic Random Access Memory
    - High density, low power, cheap, slow
    - Dynamic: need to be "refreshed" regularly
  - SRAM: Static Random Access Memory
    - Low density, high power, expensive, fast
    - Static: content will last "forever"(until lose power)
- "Non-so-random" Access Technology:
  - Access time varies from location to location and from time to time
  - Examples: Disk, CDROM, DRAM page-mode access
- Sequential Access Technology: access time linear in location (e.g.,Tape)
- The next two lectures will concentrate on random access technology
  - The Main Memory: DRAMs + Caches: SRAMs

National Technical University of Athens CSLab

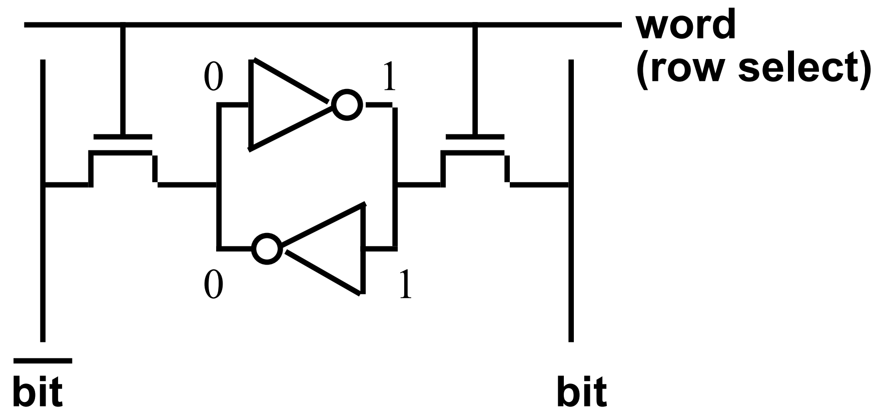
Main Memory Background

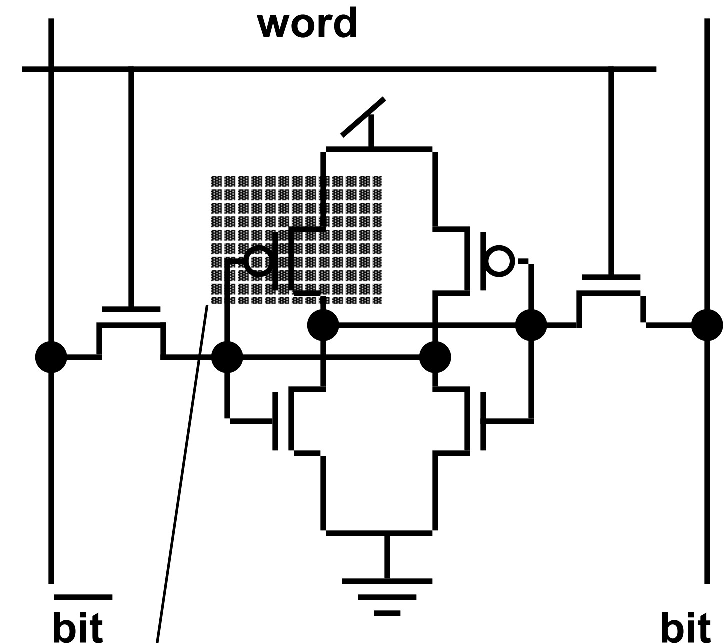
- Performance of Main Memory:
  - Latency: Cache Miss Penalty
    - *Access Time*: time between request and word arrives
    - *Cycle Time*: time between requests
  - Bandwidth: I/O & Large Block Miss Penalty (L2)
- Main Memory is *DRAM* : Dynamic Random Access Memory
  - Dynamic since needs to be refreshed periodically (8 ms)
  - Addresses divided into 2 halves (Memory as a 2D matrix):
    - *RAS* or *Row Access Strobe*
    - *CAS* or *Column Access Strobe*
- Cache uses *SRAM* : Static Random Access Memory
  - No refresh (6 transistors/bit vs. 1 transistor)
    Size: DRAM/SRAM *4-8*
    Cost/Cycle time: SRAM/DRAM *8-16*

Random Access Memory (RAM) Technology

- Why do computer designers need to know about RAM technology?
  - Processor performance is usually limited by memory bandwidth
  - As IC densities increase, lots of memory will fit on processor chip
    - Tailor on-chip memory to specific needs
      - **Instruction cache**
      - **Data cache**
      - **Write buffer**
- What makes RAM different from a bunch of flip-flops?
  - Density: RAM is much denser

Static RAM Cell

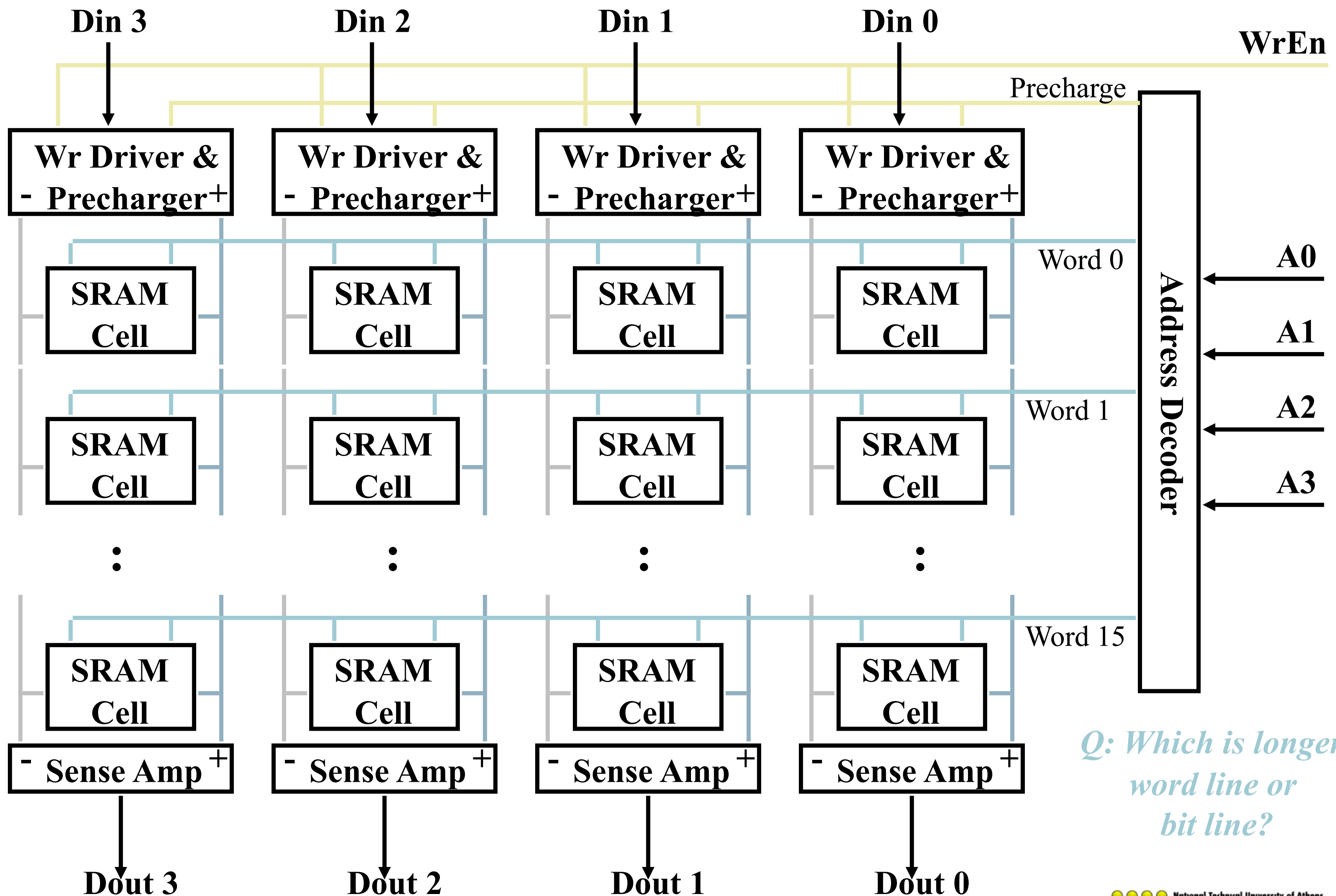




- Write:
  - 1. Drive bit lines (bit=1, bit=0)
  - 2.. Select row
- Read:
  - 1. Precharge bit and $\overline{\text{bit}}$ to Vdd or Vdd/2 => make sure equal!
  - 2.. Select row
  - 3. Cell pulls one line low
  - 4. Sense amp on column detects difference between bit and $\overline{\text{bit}}$

National Technical University of Athens
CSLab

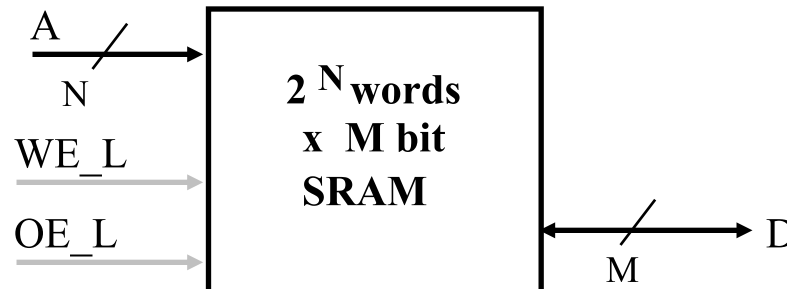
Typical SRAM Organization: 16-word x 4-bit



National Technical University of Athens
CSLab

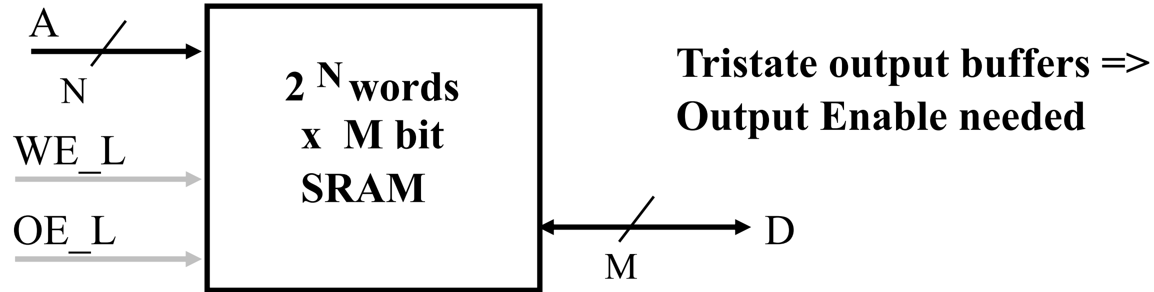
Logic Diagram of a Typical SRAM

- Write Enable is usually active low (WE_L)
- Din and Dout are combined to save pins:
  - A new control signal, output enable (OE_L) is needed
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
  - Both WE_L and OE_L are asserted:
    - Result is unknown. Don't do that!!!
- Although could change VHDL to do what desire, must do the best with what you've got (vs. what you need)



National Technical University of Athens
CSLab

Typical SRAM Timing (Asynchronous Handshake)



Write Timing:

Read Timing:

D
Data In
High Z
Junk
Data Out
Data Out

A
Write Address
Read Address
Read Address

OE_L

WE_L

Write Setup Time

Write Hold Time

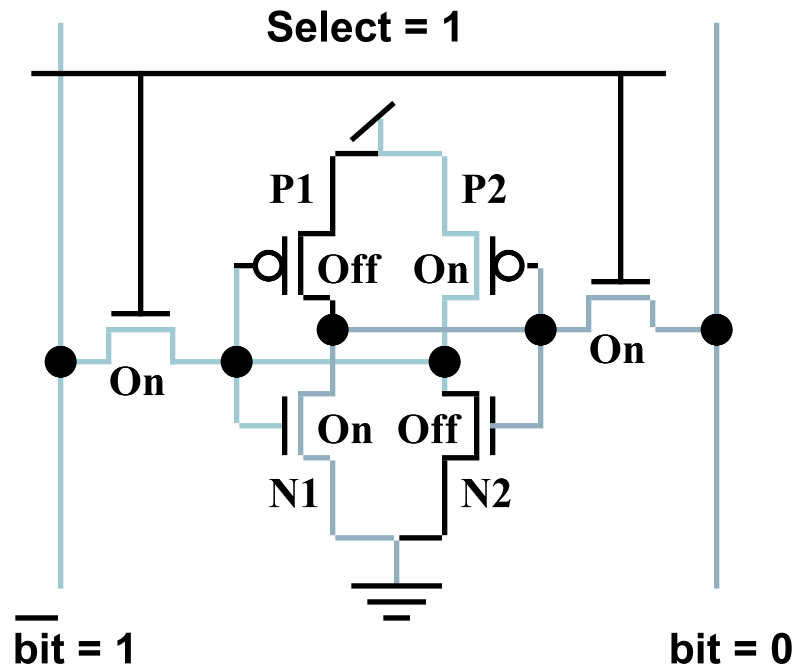
Read Access Time

Read Access Time

National Technical University of Athens
CSLab

Problems with SRAM

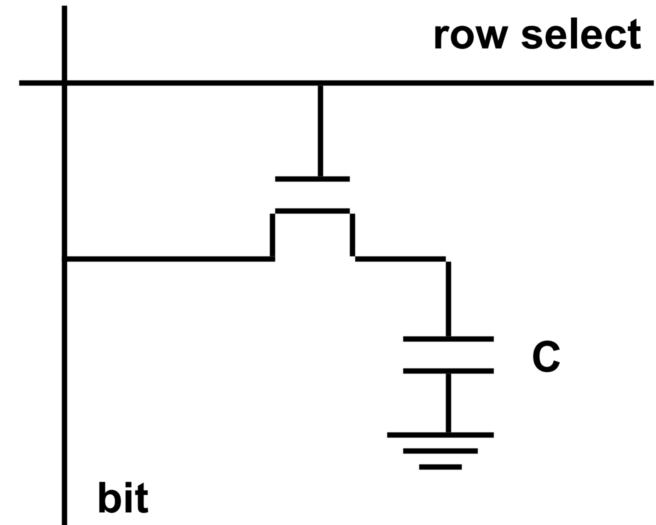
- Six transistors use up a lot of area
- Consider a "Zero" is stored in the cell:
  - Transistor N1 will try to pull "bit" to 0
  - Transistor P2 will try to pull "bit bar" to 1
- But bit lines are precharged to high: Are P1 and P2 necessary?



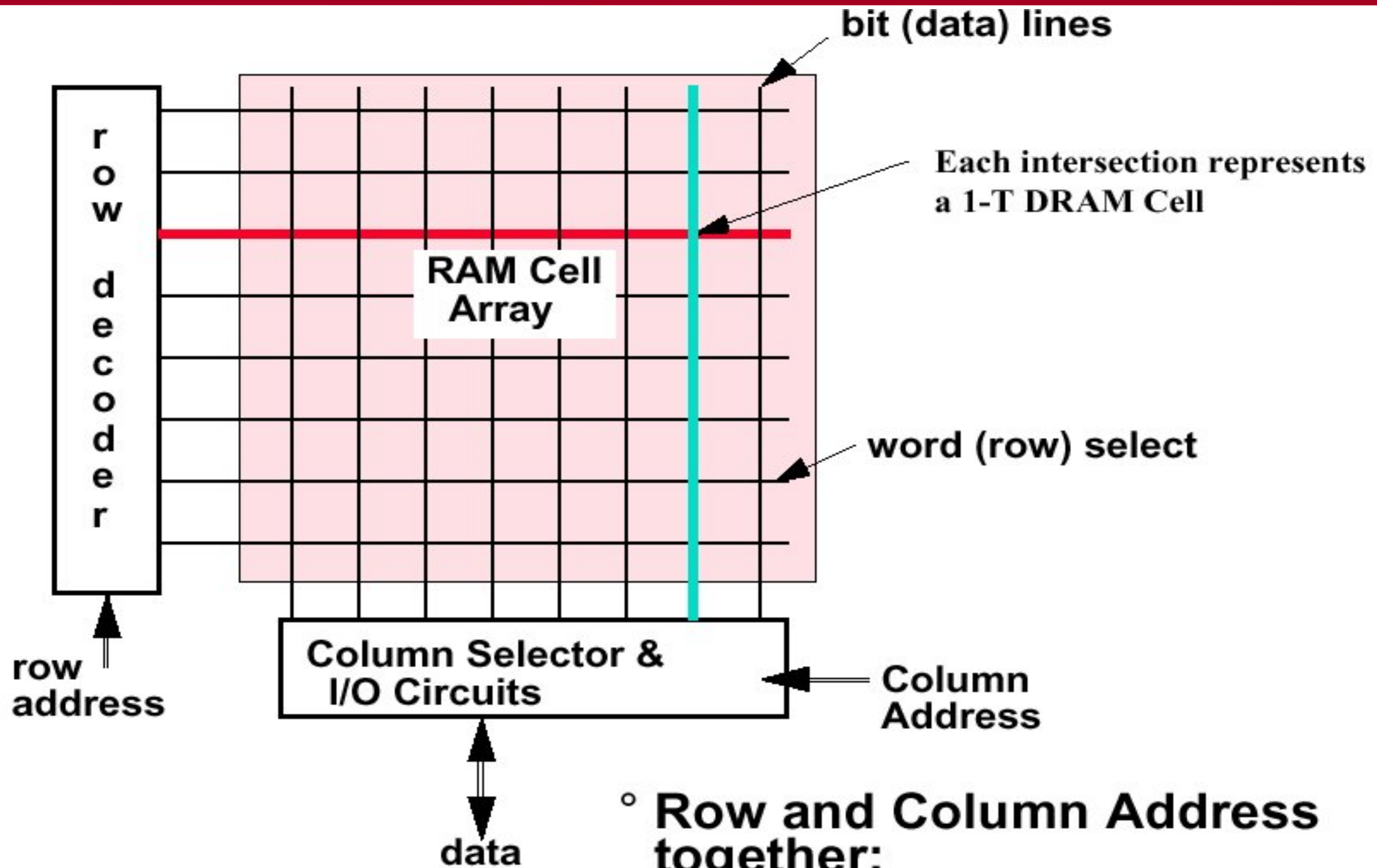
National Technical University of Athens
CSLab

1-Transistor Memory Cell (DRAM)

- Write:
  - 1. Drive bit line
  - 2.. Select row
- Read (*destructive*):
  - 1. Precharge bit line to Vdd/2
  - 2.. Select row
  - 3. Cell and bit line share charges
    - Very small voltage changes on the bit line
  - 4. Sense (fancy sense amp)
    - Can detect changes of ~1 million electrons
  - 5. Write: restore the value
- Refresh
  - 1. Just do a dummy read to every cell.



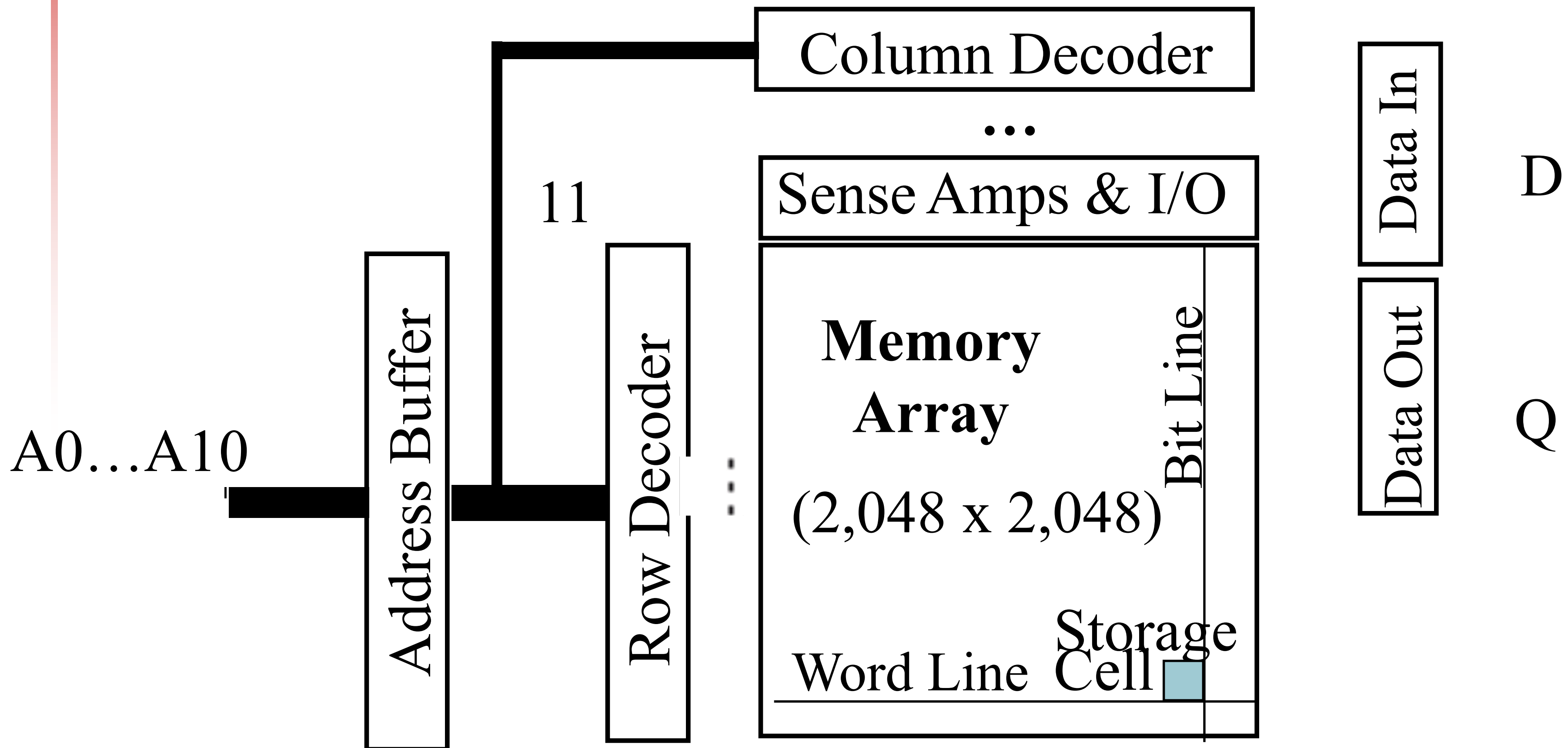
Οργάνωση της DRAM



- Row and Column Address together:
  - Select 1 bit a time

National Technical University of Athens CSLab

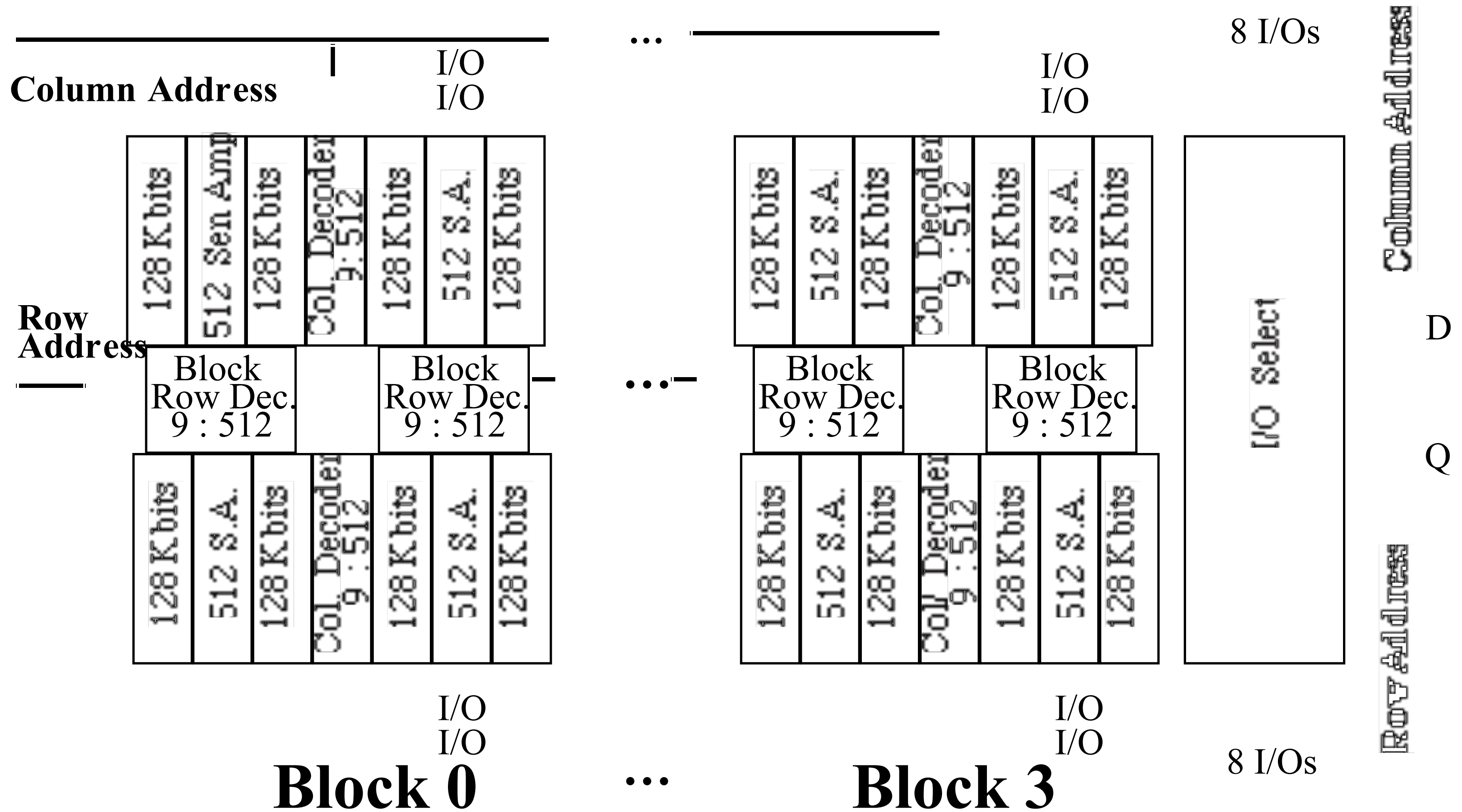
DRAM logical organization (4 Mbit)



- Square root of bits per RAS/CAS

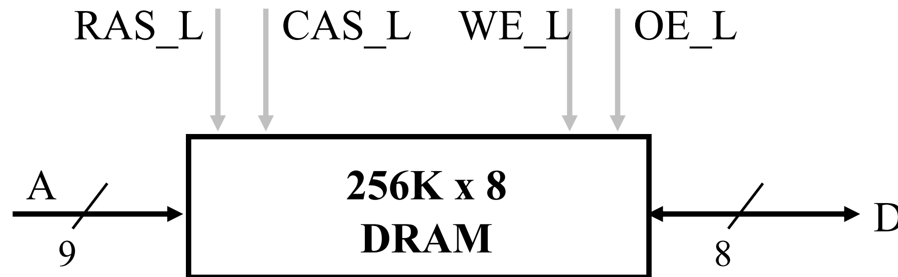
National Technical University of Athens CSLab

DRAM physical organization (4 Mbit)



National Technical University of Athens
CSLab

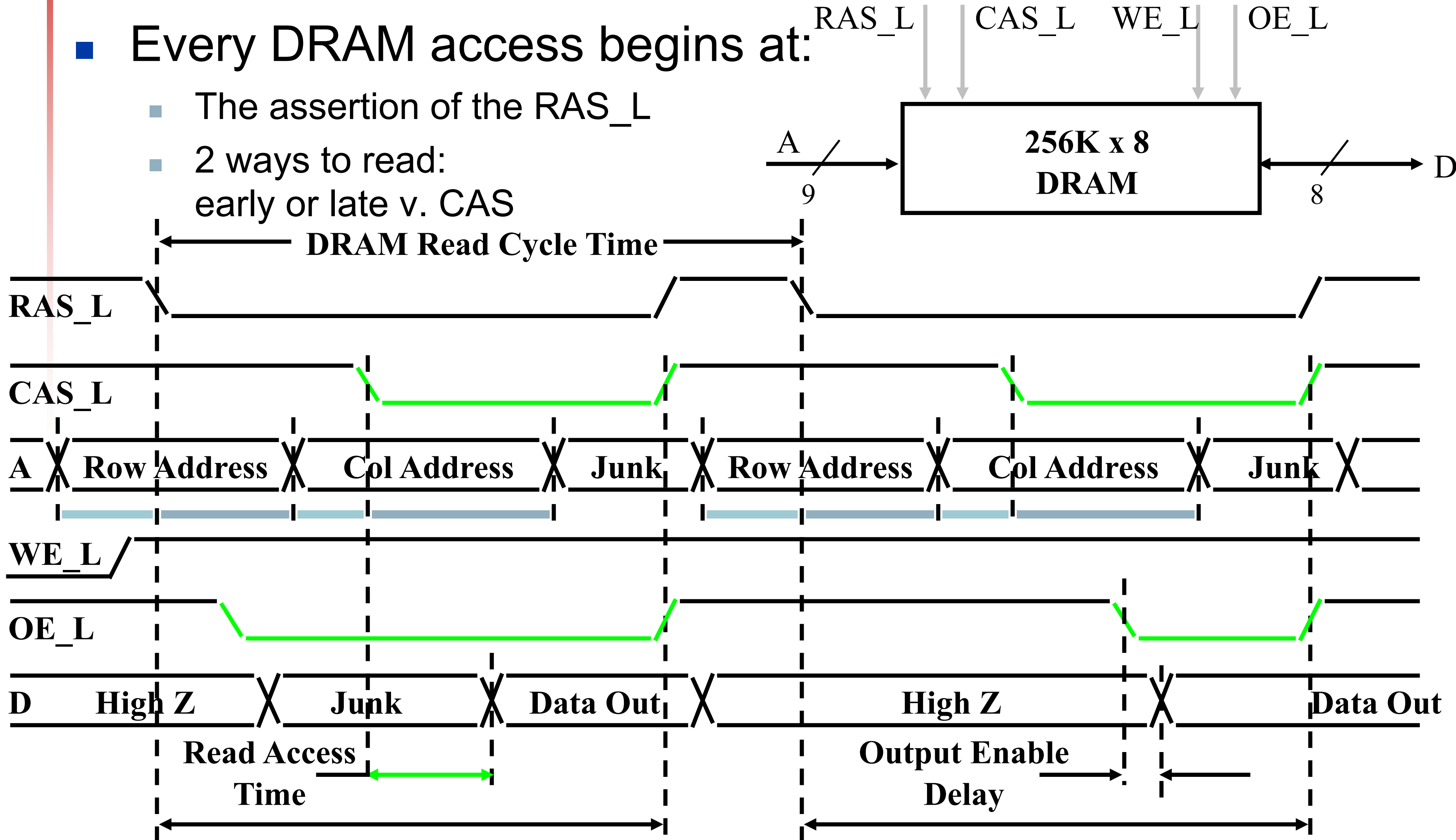
Logic Diagram of a Typical DRAM



- Control Signals (RAS_L, CAS_L, WE_L, OE_L) are all active low
- Din and Dout are combined (D):
  - WE_L is asserted (Low), OE_L is disasserted (High)
    - D serves as the data input pin
  - WE_L is disasserted (High), OE_L is asserted (Low)
    - D is the data output pin
- Row and column addresses share the same pins (A)
  - RAS_L goes low: Pins A are latched in as row address
  - CAS_L goes low: Pins A are latched in as column address
  - RAS/CAS edge-sensitive

DRAM Read Timing (Asynchronous Handshake)

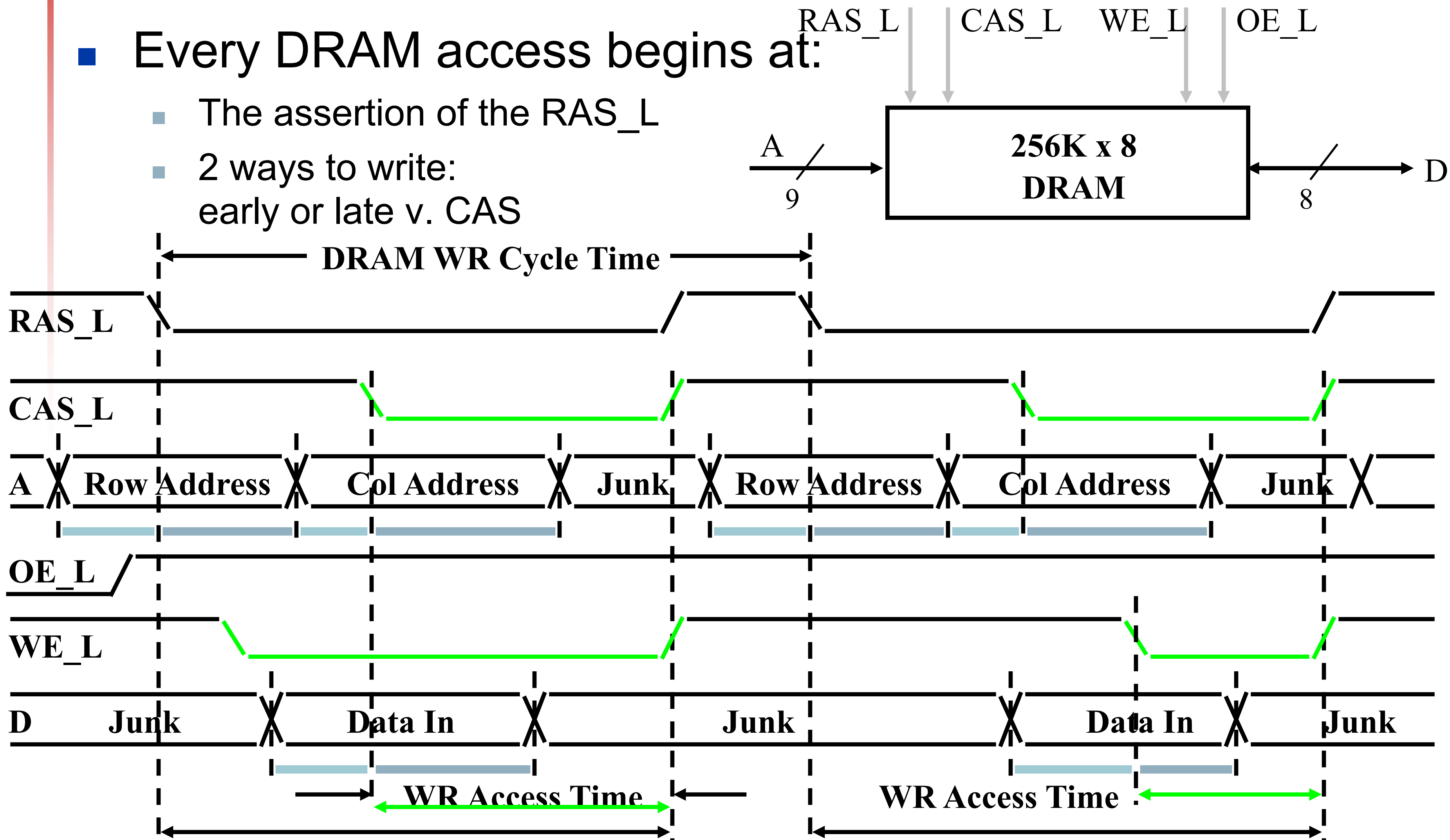
- Every DRAM access begins at:
  - The assertion of the RAS_L
  - 2 ways to read: early or late v. CAS



CSLab National Technical University of Athens

DRAM Write Timing (Asynchronous Handshake)

- Every DRAM access begins at:
  - The assertion of the RAS_L
  - 2 ways to write: early or late v. CAS



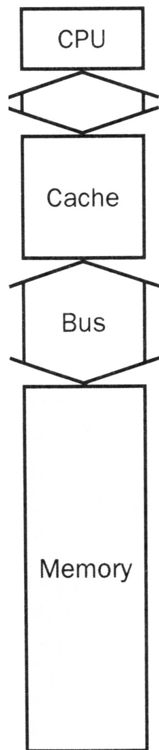
Key DRAM Timing Parameters

- t_{RAC}: minimum time from RAS line falling to the valid data output
  - Quoted as the speed of a DRAM
  - A fast 4Mb DRAM t_{RAC} = 60 ns
- t_{RC}: minimum time from the start of one row access to the start of the next
  - t_{RC} = 110 ns for a 4Mbit DRAM with a t_{RAC} of 60 ns
- t_{CAC}: minimum time from CAS line falling to valid data output
  - 15 ns for a 4Mbit DRAM with a t_{RAC} of 60 ns
- t_{PC}: minimum time from the start of one column access to the start of the next
  - 35 ns for a 4Mbit DRAM with a t_{RAC} of 60 ns

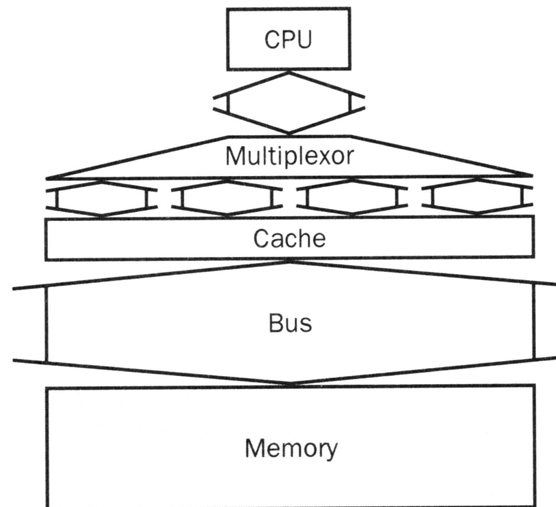
DRAM Performance

- A 60 ns (t_{RAC}) DRAM can
  - perform a row access only every 110 ns (t_{RC})
  - perform column access (t_{CAC}) in 15 ns, but time between column accesses is at least 35 ns (t_{PC}).
    - In practice, external address delays and turning around buses make it 40 to 50 ns
- These times do not include the time to drive the addresses off the microprocessor nor the memory controller overhead.
  - Drive parallel DRAMs, external memory controller, bus to turn around, SIMM module, pins…
  - 180 ns to 250 ns latency from processor to memory is good for a "60 ns" (t_{RAC}) DRAM

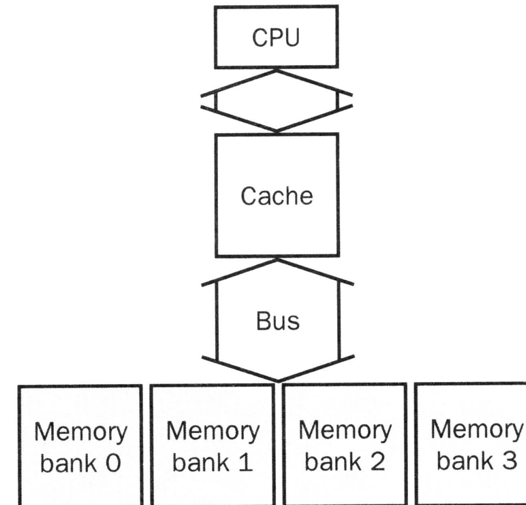
Main Memory Performance



a. One-word-wide memory organization



b. Wide memory organization



c. Interleaved memory organization

- *Simple*:

  CPU, Cache, Bus, Memory same width (32 bits)

- *Wide*:

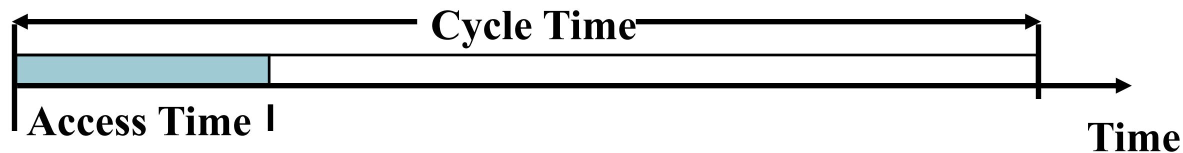
  CPU/Mux 1 word; Mux/Cache, Bus, Memory N words (Alpha: 64 bits & 256 bits)

- *Interleaved*:

  CPU, Cache, Bus 1 word: Memory N Modules (4 Modules); example is *word interleaved*

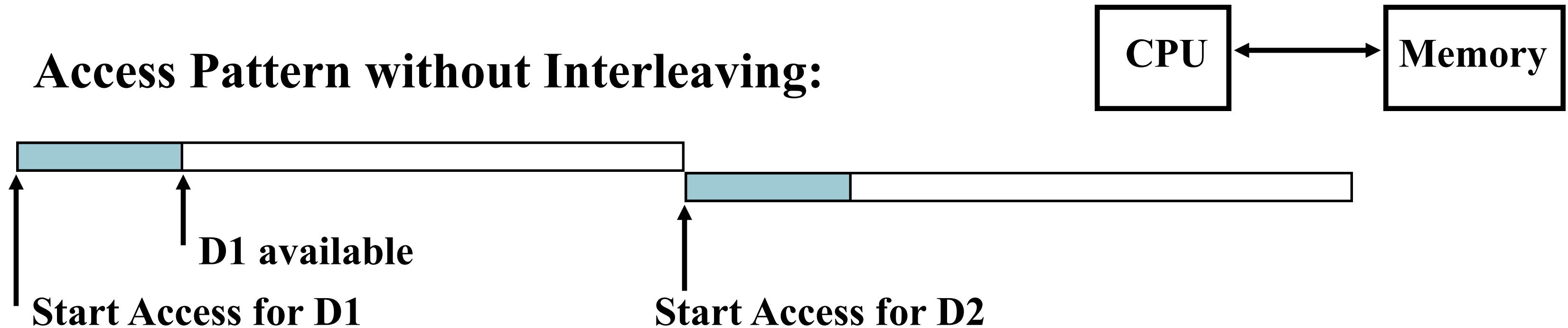
National Technical University of Athens CSLab

Main Memory Performance



- DRAM (Read/Write) Cycle Time >> DRAM (Read/Write) Access Time
  - 2:1; why?
- DRAM (Read/Write) Cycle Time :
  - How frequent can you initiate an access?
- DRAM (Read/Write) Access Time:
  - How quickly will you get what you want once you initiate an access?
- DRAM Bandwidth

Increasing Bandwidth - Interleaving



Memory Bank 0

Memory Bank 1

Memory Bank 2

Memory Bank 3

CPU

Access Pattern with 4-way Interleaving:

Access Bank 0

Access Bank 1

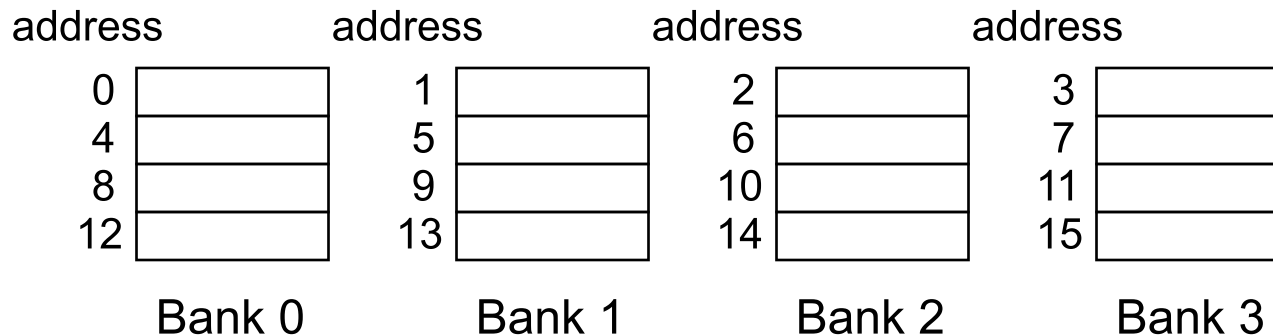
Access Bank 2

Access Bank 3

We can Access Bank 0 again

Main Memory Performance

- Timing model
  - 1 to send address,
  - 4 for access time, 10 cycle time, 1 to send data
  - Cache Block is 4 words
- *Simple M.P.* = 4 x (1+10+1) = 48
- *Wide M.P.* = 1 + 10 + 1 = 12
- *Interleaved M.P.* = 1+10+1 + 3 = 15



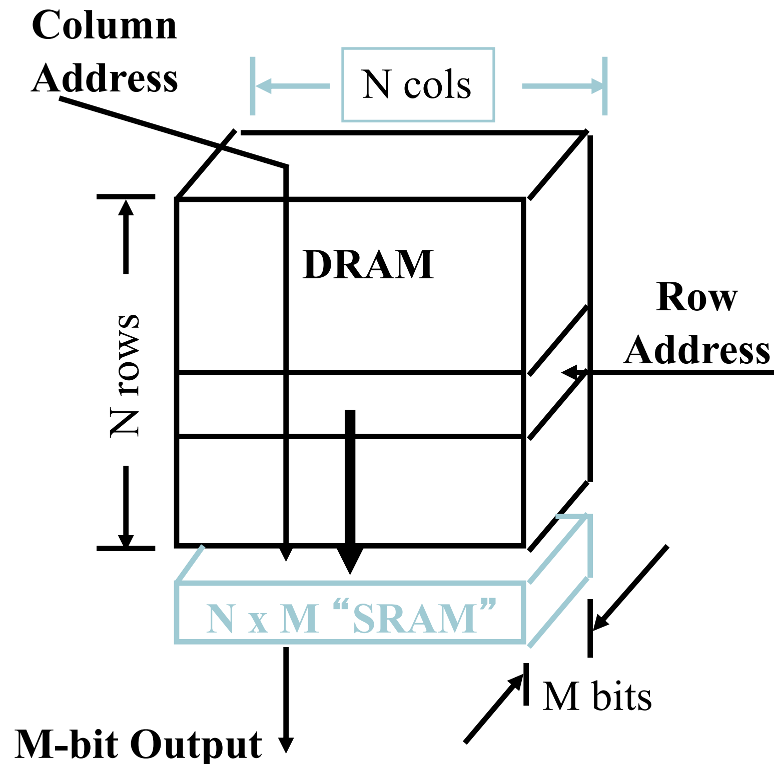
Independent Memory Banks

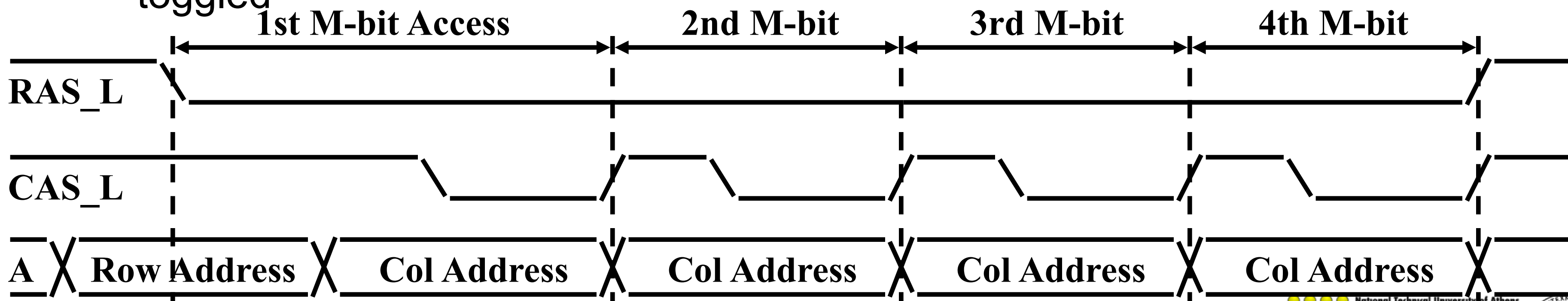
- How many banks?
- number banks ≥ number clocks to access word in bank
  - For sequential accesses, otherwise will return to original bank before it has next word ready
- Increasing DRAM sizes => fewer chips => harder to have many banks
  - Growth bits/chip DRAM : 50%-60%/yr

National Technical University of Athens CSLab

Fast Page Mode Operation

- Regular DRAM Organization:
  - N rows x N column x M-bit
  - Read & Write M-bit at a time
  - Each M-bit access requires a RAS / CAS cycle
- Fast Page Mode DRAM
  - N x M "SRAM" to save a row
- After a row is read into the register
  - Only CAS is needed to access other M-bit blocks on that row
  - RAS_L remains asserted while CAS_L is toggled





National Technical University of Athens
CSLab

Summary:

- Two Different Types of Locality:
  - Temporal Locality (Locality in Time): If an item is referenced, it will tend to be referenced again soon.
  - Spatial Locality (Locality in Space): If an item is referenced, items whose addresses are close by tend to be referenced soon.
- By taking advantage of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.
- DRAM is slow but cheap and dense:
  - Good choice for presenting the user with a BIG memory system
- SRAM is fast but expensive and not very dense:
  - Good choice for providing the user FAST access time.

Memory Width, Interleaving: Παράδειγμα

Δίνεται ένα σύστημα με τις ακόλουθες παραμέτρους:

Μέγεθος Cache Block = 1 word, Memory bus width = 1 word, Miss rate = 3%

Miss penalty = 32 κύκλους :

(4 κύκλοι για αποστολή της διεύθυνσης, 24 κύκλοι access time / λέξη, 4 κύκλοι για αποστολή μιας λέξης)

Memory access / εντολή = 1.2 Ιδανικό execution CPI (αγνοώντας τα cache misses) = 2

Miss rate (μέγεθος block=2 word) = 2% Miss rate (μέγεθος block=4 words) = 1%

- Το CPI του μηχανήματος με blocks της 1 λέξης = 2 + (1.2 x .03 x 32) = 3.15
- Μεγαλώνοντας το μέγεθος του block σε 2 λέξεις δίνει το ακόλουθο CPI:
  - 32-bit bus και memory, καθόλου interleaving = 2 + (1.2 x .02 x 2 x 32) = 3.54
  - 32-bit bus και memory, interleaved = 2 + (1.2 x .02 x (4 + 24 + 8) = 2.86
  - 64-bit bus και memory, καθόλου interleaving = 2 + (1.2 x .02 x 1 x 32) = 2.77
- Μεγαλώνοντας το μέγεθος του block σε 4 λέξεις, δίνει CPI:
  - 32-bit bus και memory, καθόλου interleaving = 2 + (1.2 x 1% x 4 x 32) = 3.54
  - 32-bit bus και memory, interleaved = 2 + (1.2 x 1% x (4 +24 + 16) = 2.53
  - 64-bit bus και memory, καθόλου interleaving = 2 + (1.2 x 2% x 2 x 32) = 2.77