

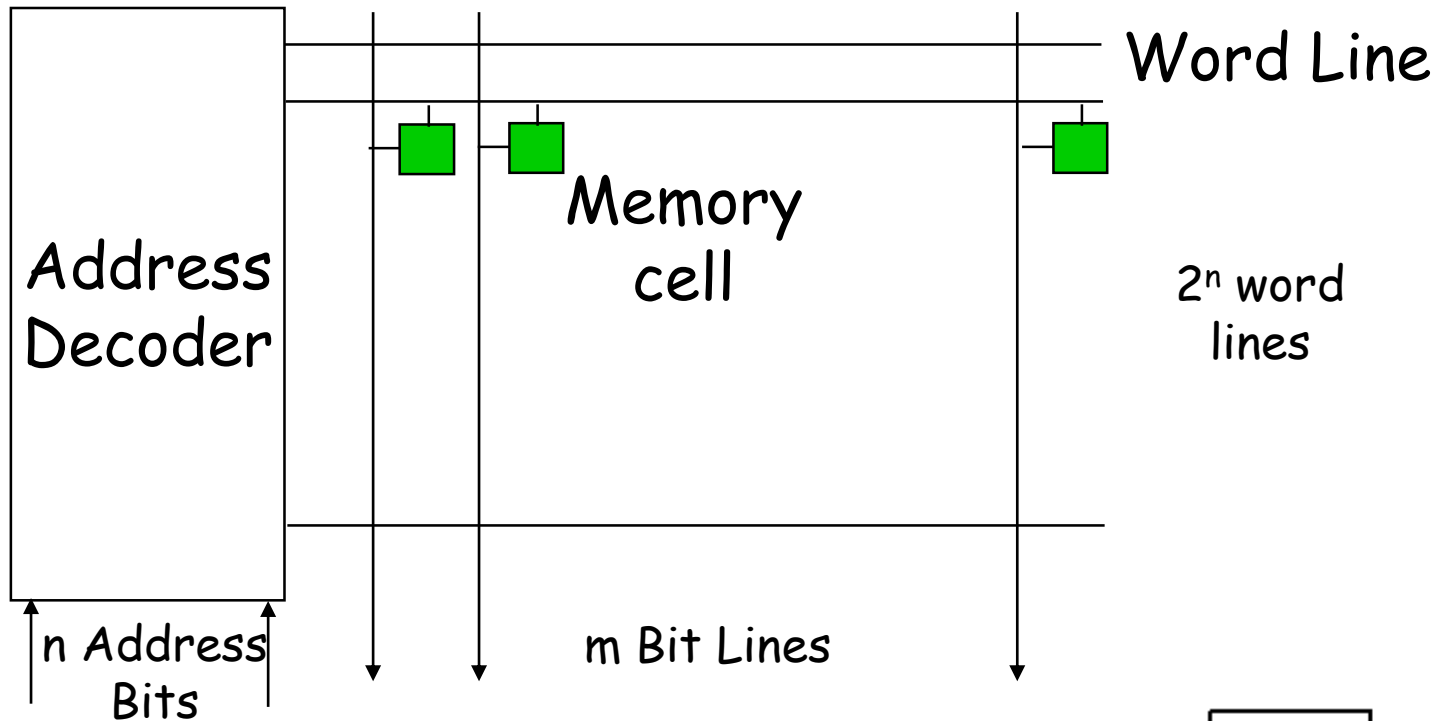
# HY220

## Εργαστήριο Ψηφιακών Κυκλωμάτων

Εαρινό Εξάμηνο  
2022

Δυναμικές Μνήμες - DRAM

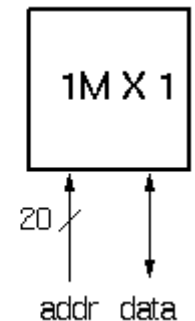
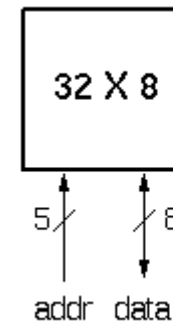
# Βασικό Block Diagram Υποσυστημάτων Μνήμης



RAM/ROM ονοματολογία:

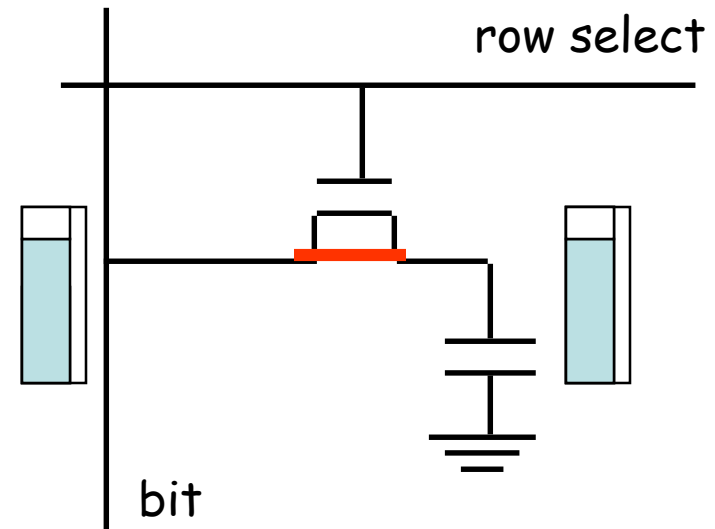
32 X 8, "32 by 8" => 32 8-bit words

1M X 1, "1 meg by 1" => 1M 1-bit words

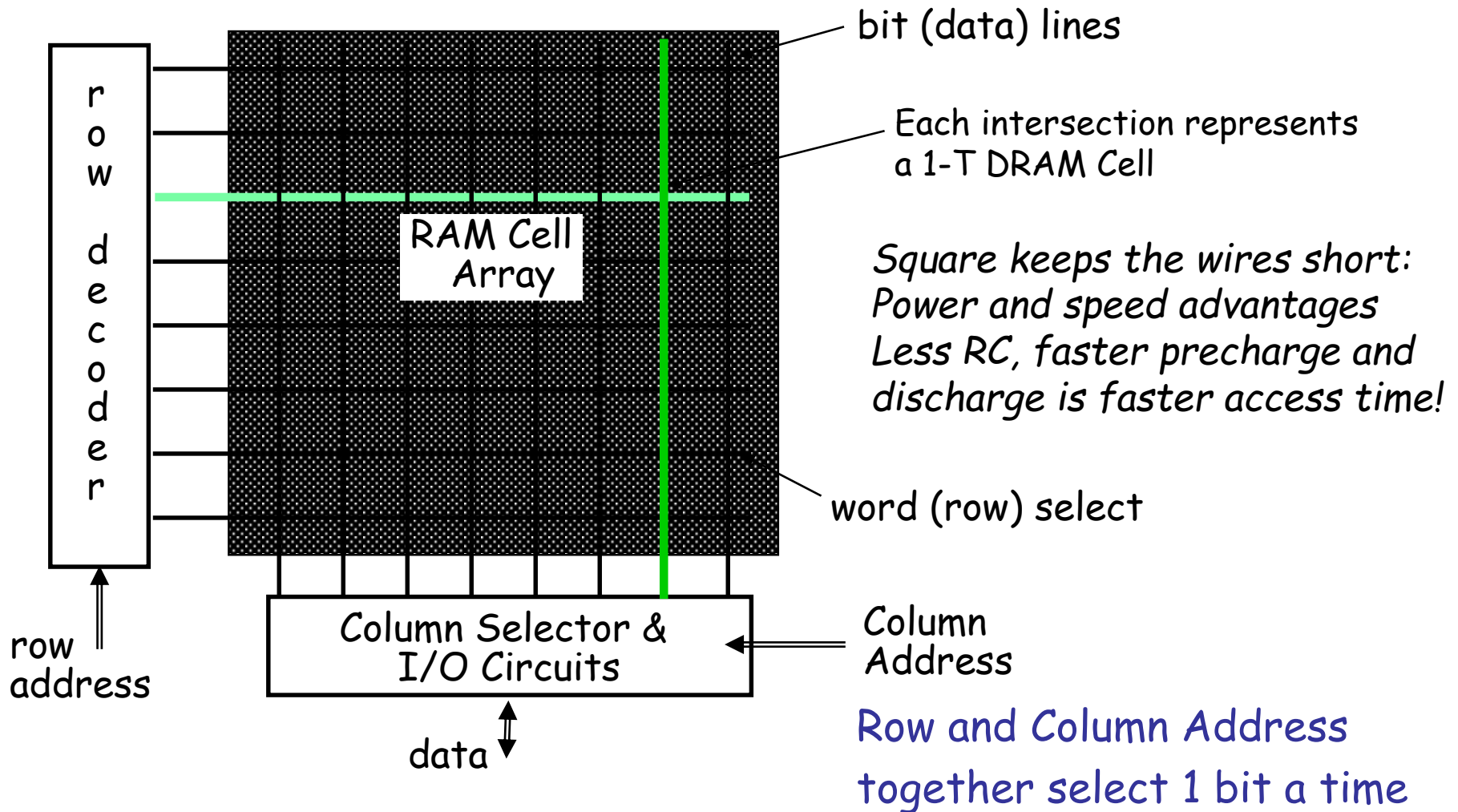


# Κελί μνήμης με 1 transistor (DRAM)

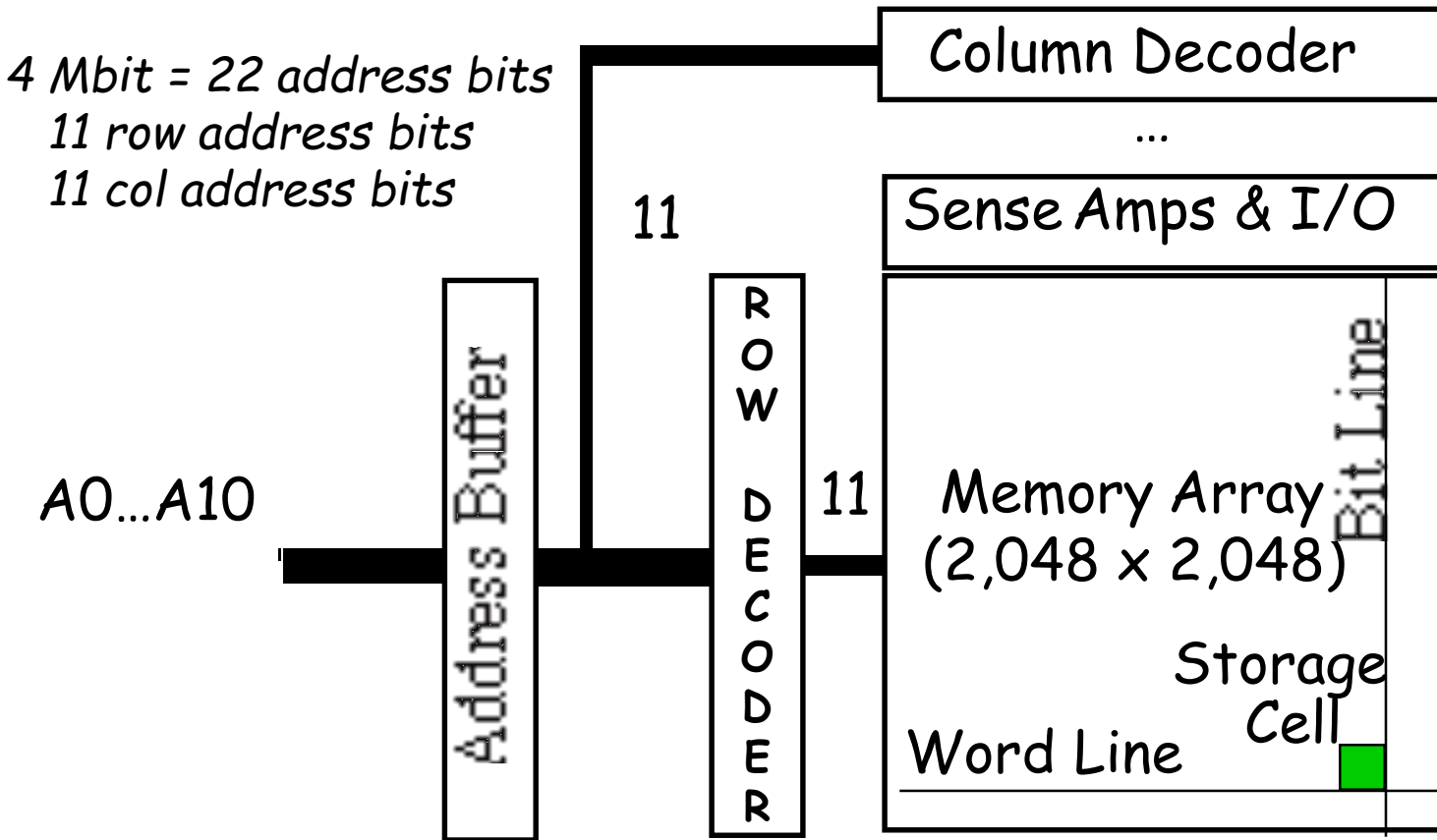
- **Εγγραφή - Write:**
  1. Οδηγούμε την bit line
  2. Επιλέγουμε γραμμή (row select)
- **Ανάγνωση - Read:**
  1. Προφορτίζουμε (precharge) την bit line σε  $V_{dd}/2$
  2. Επιλέγουμε γραμμή (row select)
  3. Το κελί και η bit line μοιράζονται τα φορτία (charge sharing)
  4. Sense στην bit line (sense amplifier)  
Μπορεί να ανιχνεύει πολύ μικρές αλλαγές
  5. Write: επαναφέρουμε την τιμή
- **Ανανέωση - Refresh :**
  - Αρκεί ένα απλό read σε κάθε κελί



# Κλασική Οργάνωση DRAM (τετραγωνική)

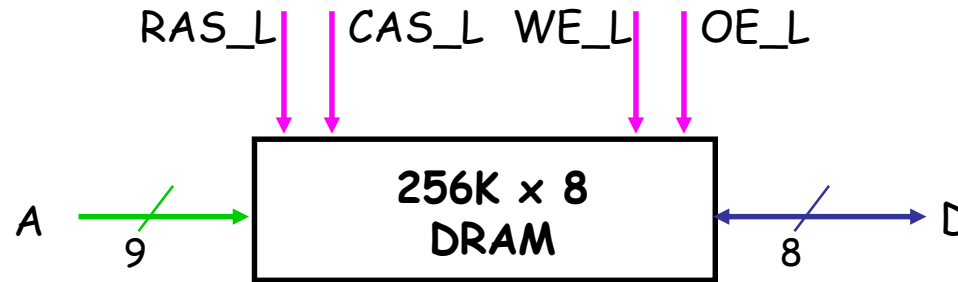


# Λογική Οργάνωση DRAM (4 Mbit)



- Square root of bits per RAS/CAS
  - Row selects 1 row of 2048 bits from 2048 rows
  - Col selects 1 bit out of 2048 bits in such a row

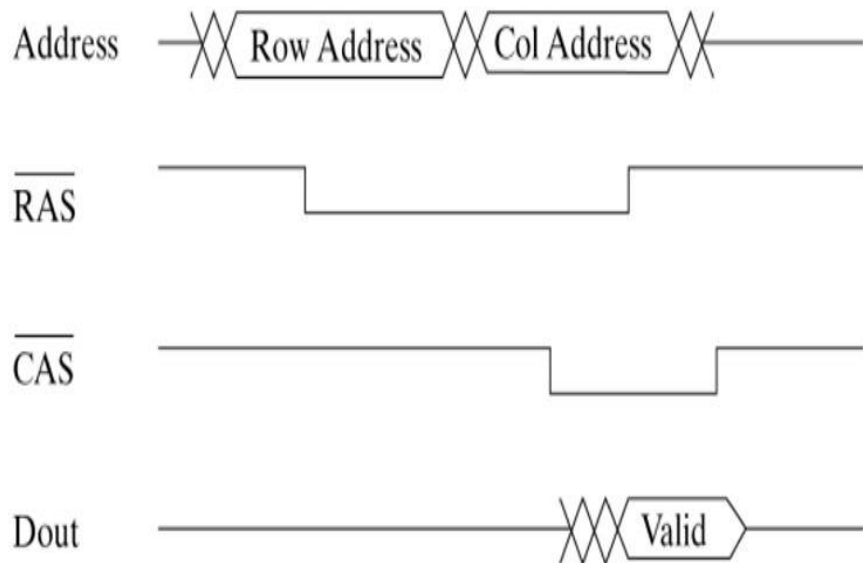
# Τα σήματα της DRAM



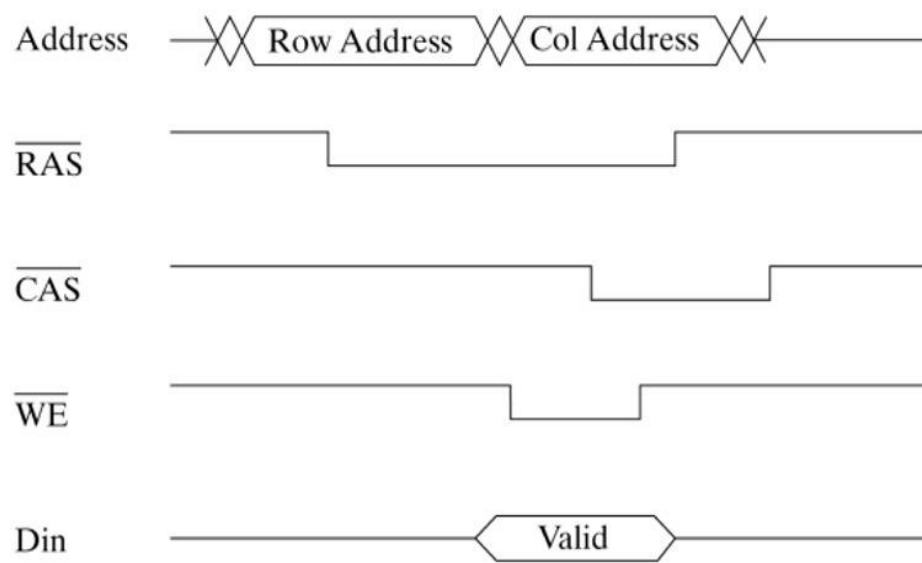
- Σήματα ελέγχου (RAS\_L, CAS\_L, WE\_L, OE\_L) όλα active low
- Κοινό bus δεδομένων D - εισόδου και εξόδου (Din & Dout):
  - WE\_L ενεργοποιείται (Low), OE\_L απενεργοποιείται (High)
    - D χρησιμοποιείται σαν είσοδος στην DRAM
  - WE\_L απενεργοποιείται (High), OE\_L ενεργοποιείται (Low)
    - D χρησιμοποιείται σαν έξοδος από την DRAM
- Οι διευθύνσεις γραμμής και στήλης μοιράζονται τα ίδια pins (A)
  - RAS\_L ενεργοποιείται (low) -> A αποθηκεύεται σαν row address
  - CAS\_L ενεργοποιείται (low) -> A αποθηκεύεται σαν column address
  - RAS/CAS edge-sensitive

# Απλοποιημένο διάγραμμα χρονισμού DRAM

Read



Write

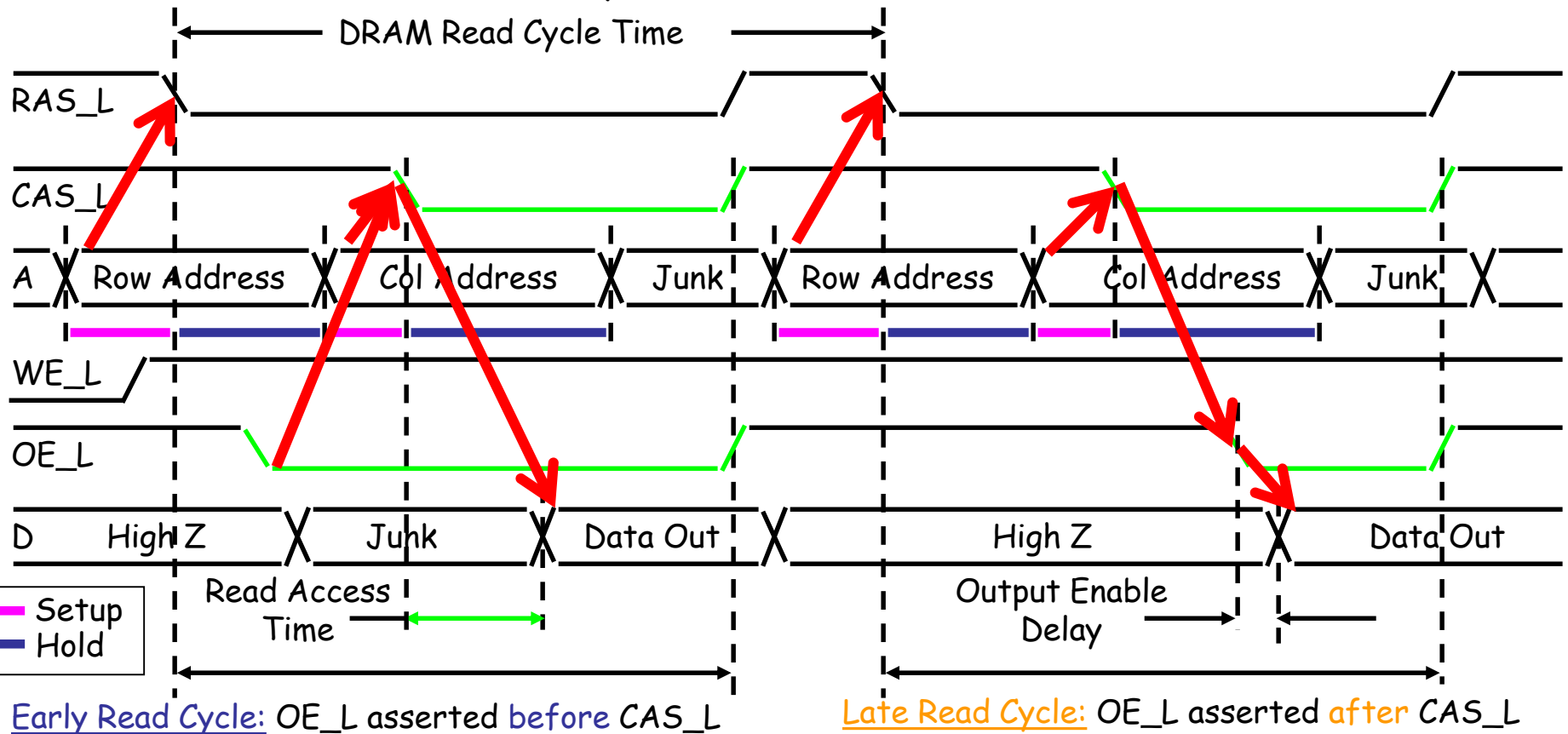
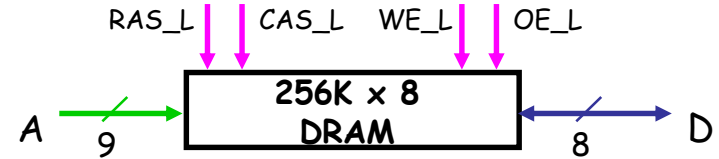


- Η διεύθυνση δίνεται σε 2 βήματα

# Τυπικός χρονοισμός Ανάγνωσης DRAM

• Κάθε προσπέλαση DRAM ξεκινάει με:

- Ενεργοποίηση του RAS\_L
- 2 τρόποι ανάγνωση: early or late





# Τα βήματα για Early Read

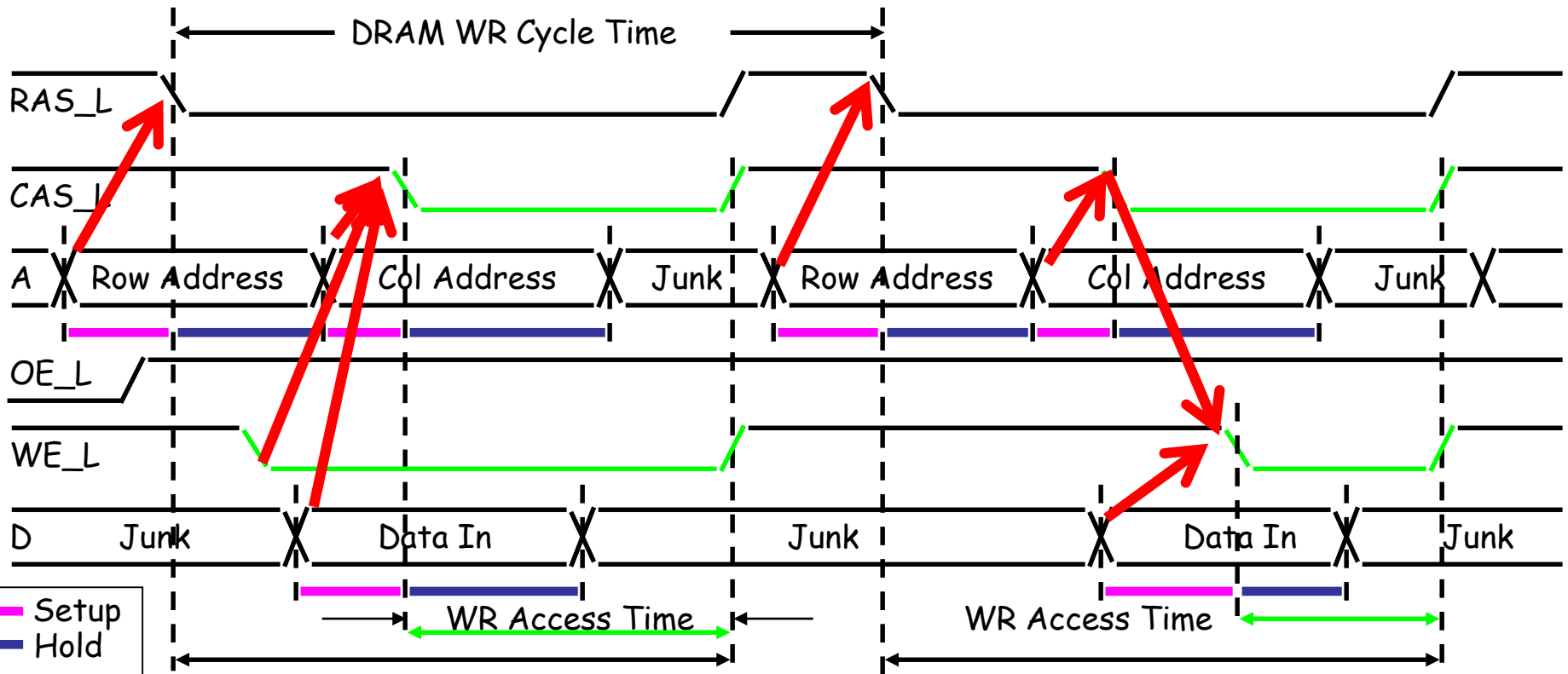
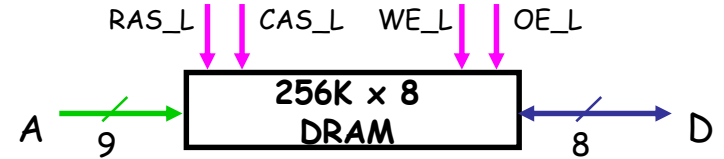
- Assert Row Address
- Assert RAS\_L
  - Start read cycle
  - Meet Row Addr setup time before RAS/hold time after RAS
- Assert OE\_L
- Assert Col Address
- Assert CAS\_L
  - Meet Col Addr setup time before CAS/hold time after CAS
- Valid Data Out after access time
- Disassert OE\_L, CAS\_L, RAS\_L to end cycle

# Τα βήματα για Late Read

- Assert Row Address
- Assert RAS\_L
  - Start read cycle
  - Meet Row Addr setup time before RAS/hold time after RAS
- Assert Col Address
- Assert CAS\_L
  - Meet Col Addr setup time before CAS/hold time after CAS
- Assert OE\_L
- Valid Data Out after access time
- Disassert OE\_L, CAS\_L, RAS\_L to end cycle

# Τυπικός χρονοσμός Εγγραφής DRAM

- Κάθε προσπέλαση DRAM ξεκινάει με:
  - Ενεργοποίηση του RAS\_L
  - 2 τρόποι εγγραφής: early or late



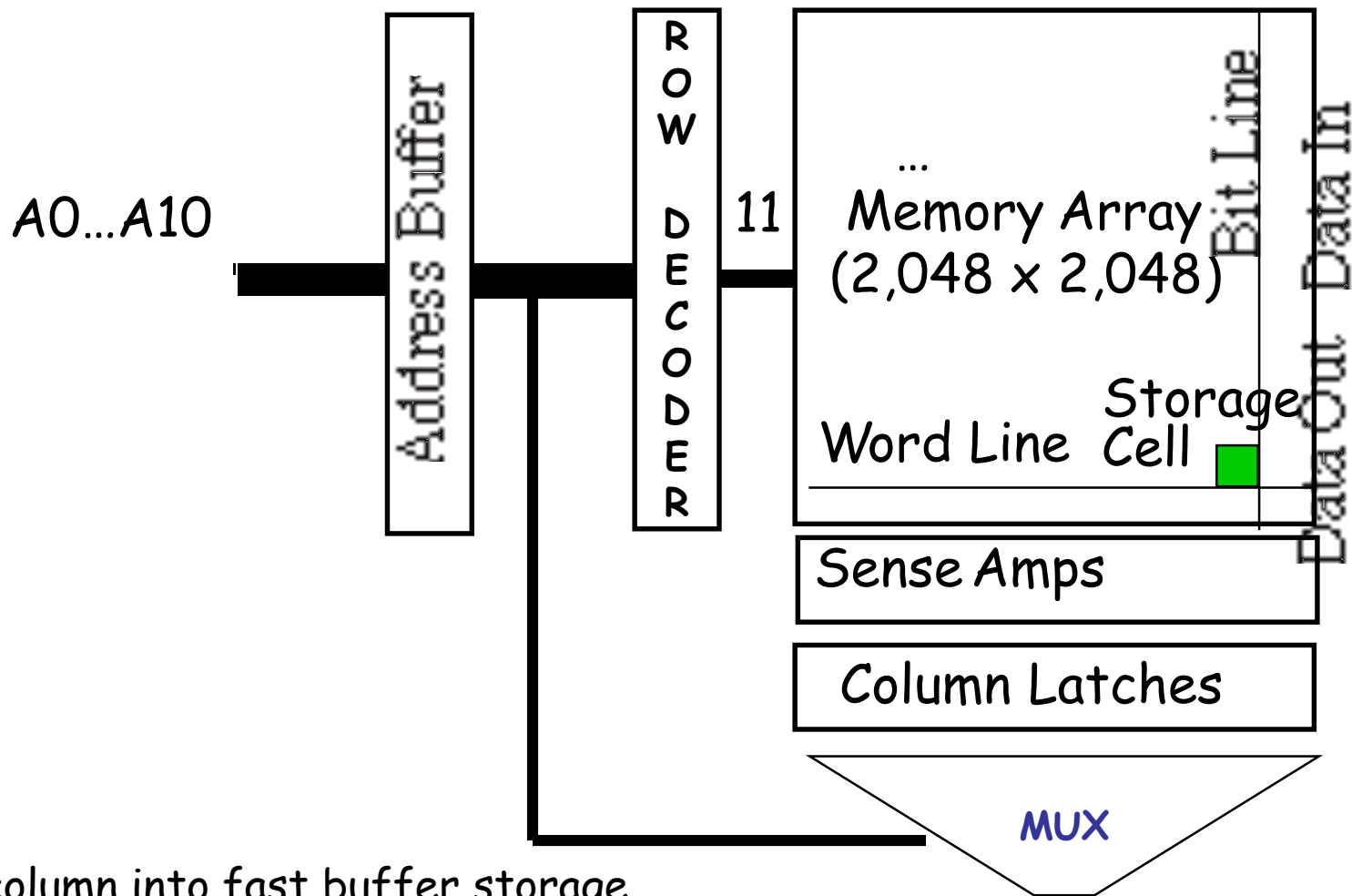
Early Wr Cycle: WE\_L asserted before CAS\_L

Late Wr Cycle: WE\_L asserted after CAS\_L

# 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 with Column buffer



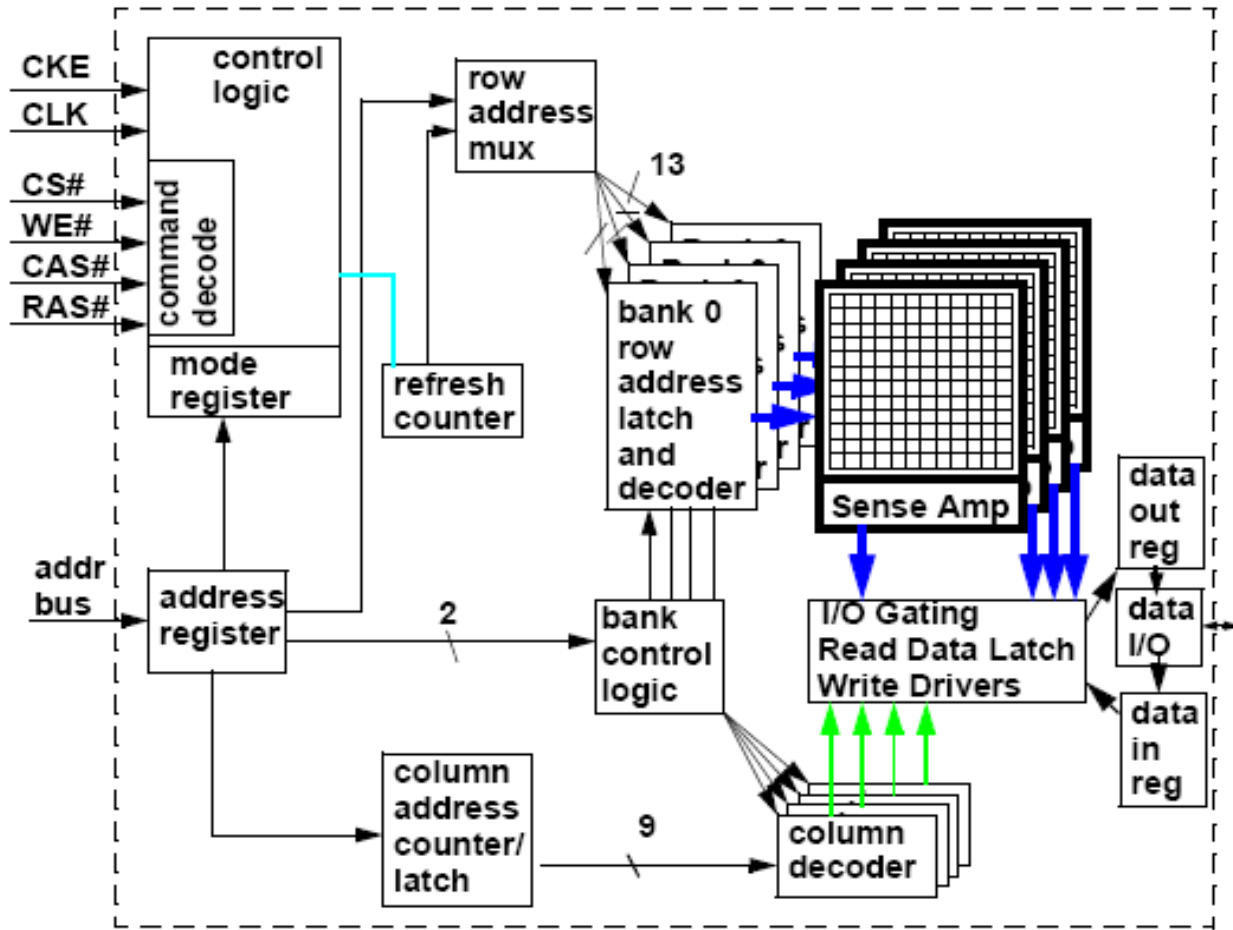
Pull column into fast buffer storage

Access sequence of bit from there

# Optimized Access to Cols in Row

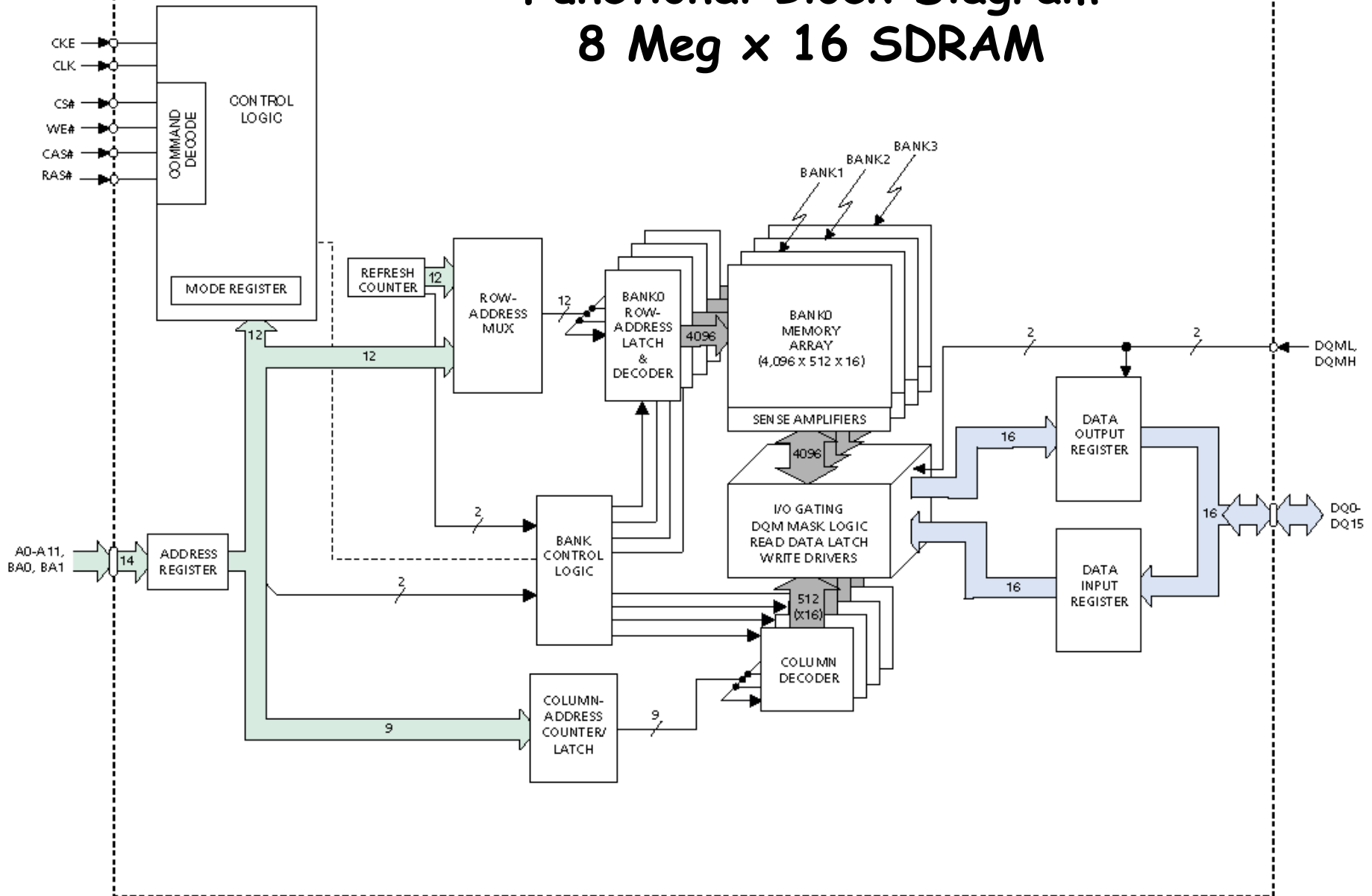
- Often want to access a sequence of bits
- Page mode
  - After RAS / CAS, can access additional bits in the row by changing column address and strobing CAS
- Static Column mode
  - Change column address (without repeated CAS) to get different bit
- Nibble mode
  - Pulsing CAS gives next bit mod 4
- Video ram
  - Serial access

# 256 Mbit SDRAM Addressing



256 Mbit chip: 8192 rows, 512 columns, x16 data, 4 banks

# Functional Block Diagram 8 Meg x 16 SDRAM

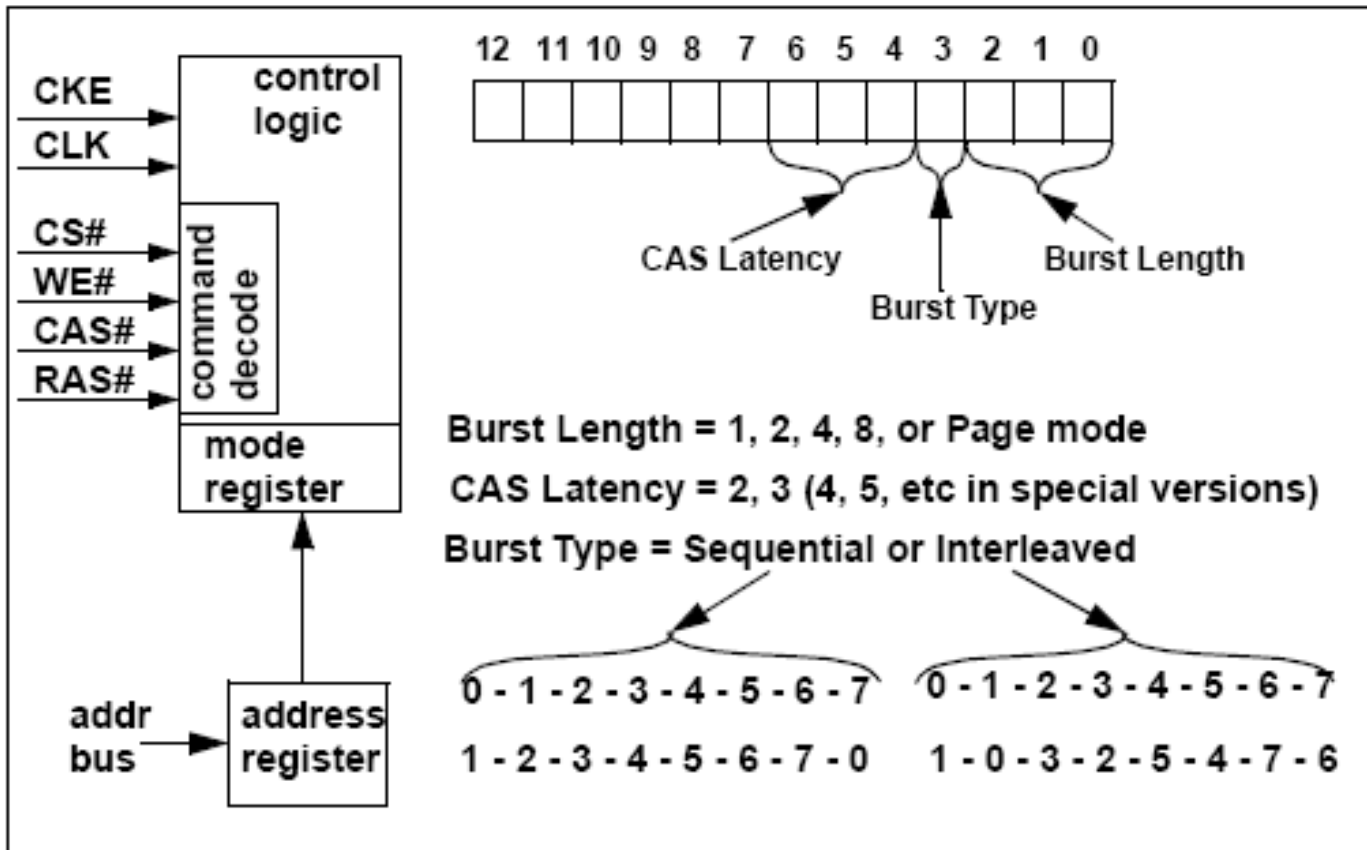




# SDRAM Details

- Multiple "banks" of cell arrays are used to reduce access time:
  - Each bank is 4K rows by 512 "columns" by 16 bits (for our part)
- Read and Write operations are split into RAS (row access) followed by CAS (column access)
- These operations are controlled by sending commands
  - Commands are sent using the RAS, CAS, CS, & WE pins.
- Address pins are "time multiplexed"
  - During RAS operation, address lines select the bank and row
  - During CAS operation, address lines select the column.
- "ACTIVE" command "opens" a row for operation
  - transfers the contents of the entire row to a row buffer
- Subsequent "READ" or "WRITE" commands modify the contents of the row buffer.
- For burst reads and writes during "READ" or "WRITE" the starting address of the block is supplied.
  - Burst length is programmable as 1, 2, 4, 8 or a "full page" (entire row) with a burst terminate option.
- Special commands are used for initialization (burst options etc.)
- A burst operation takes  $\approx 4 + n$  cycles (for  $n$  words)

# Mode Register

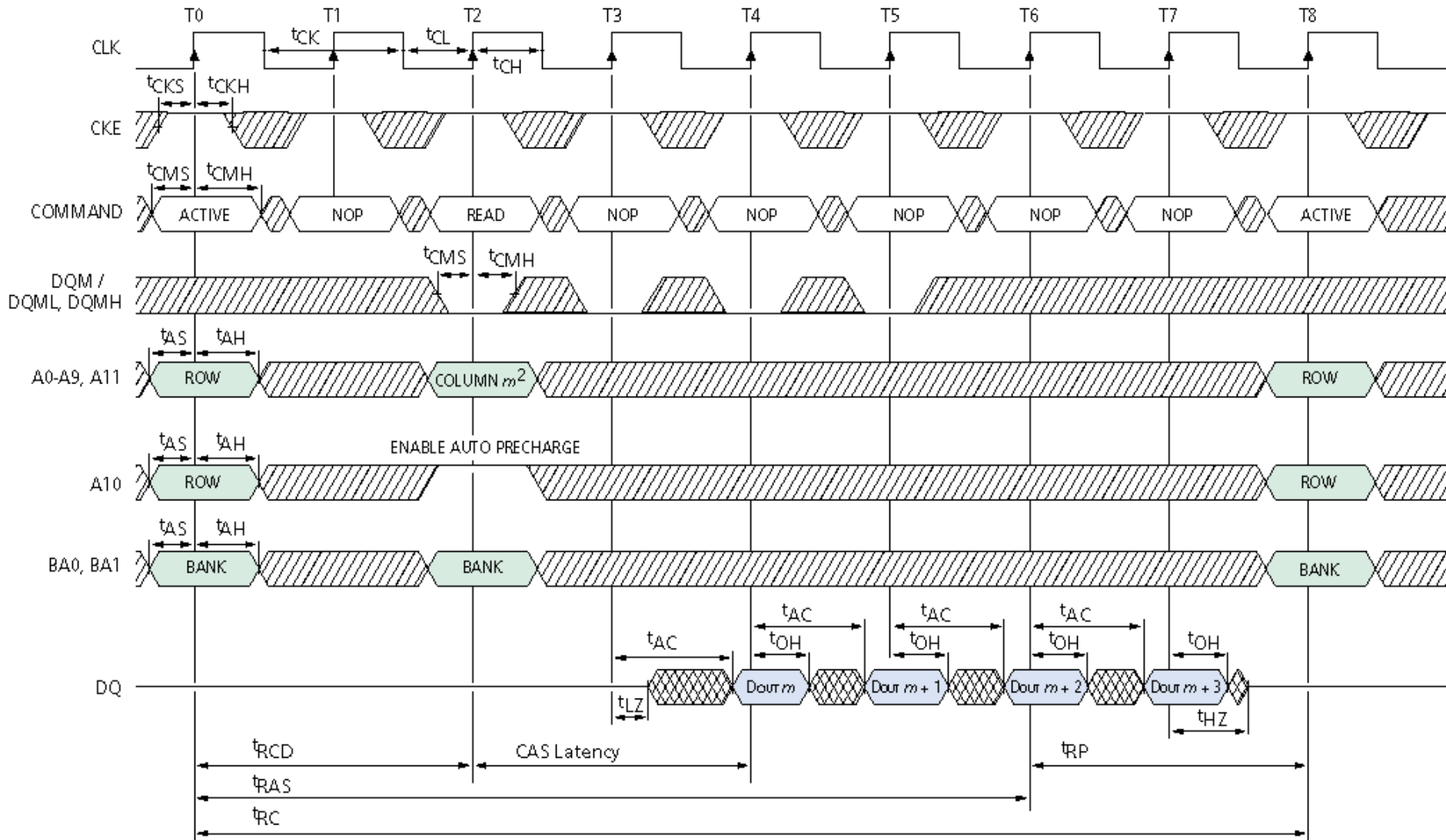


**SDRAM Device can be programmed to respond in slightly different manners**

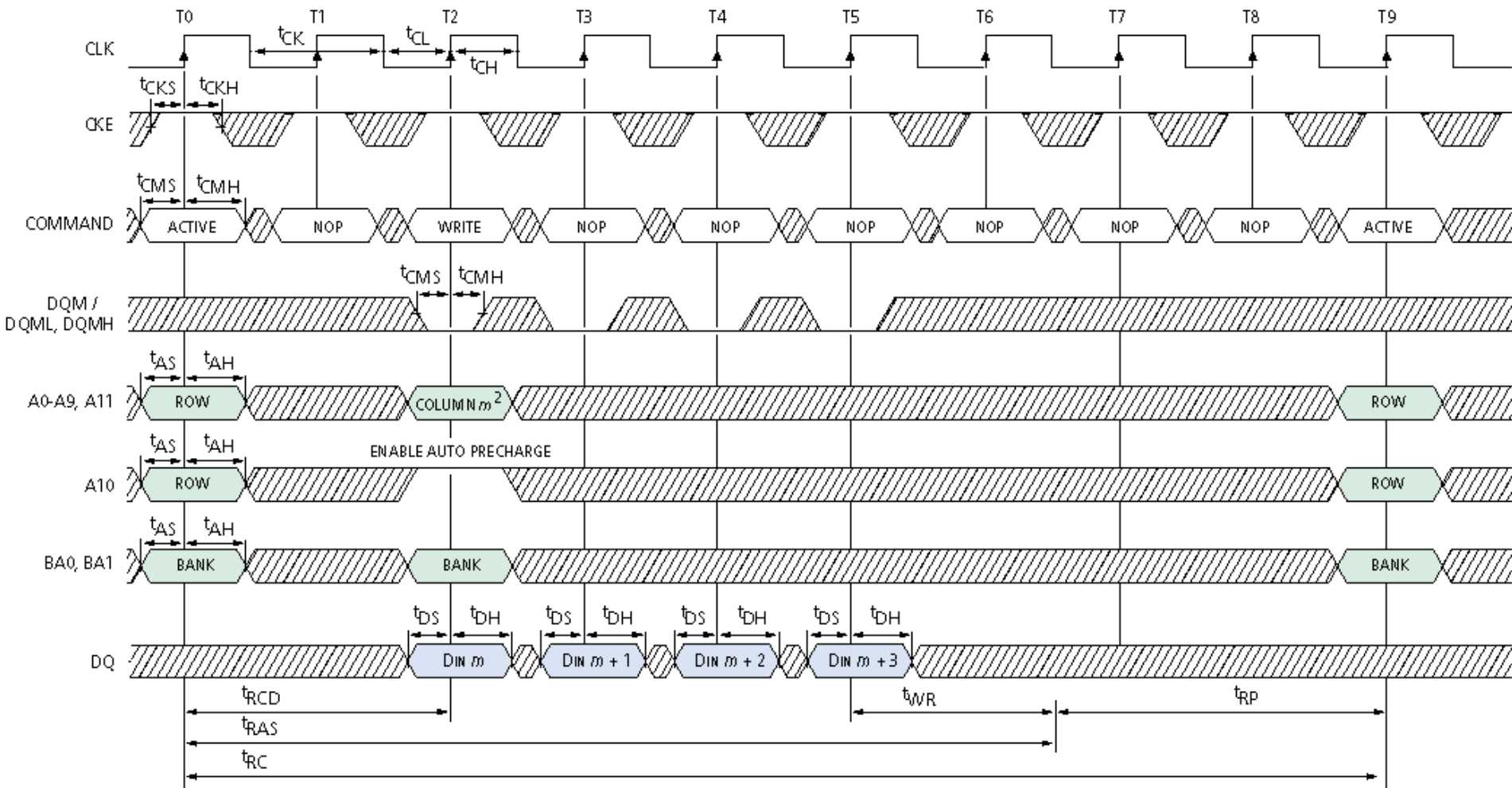
# Βασικές Χρονικές Παράμετροι SDRAM

- $t_{RCD}$  : ACTIVE to READ or WRITE delay
- $CL$  : CAS Latency
- $t_{RAS}$  : ACTIVE to PRECHARGE time
- $t_{RP}$  : PRECHARGE Period
- $t_{WR}$  : WRITE recovery time
- $t_{RC}$  : ACTIVE to ACTIVE time

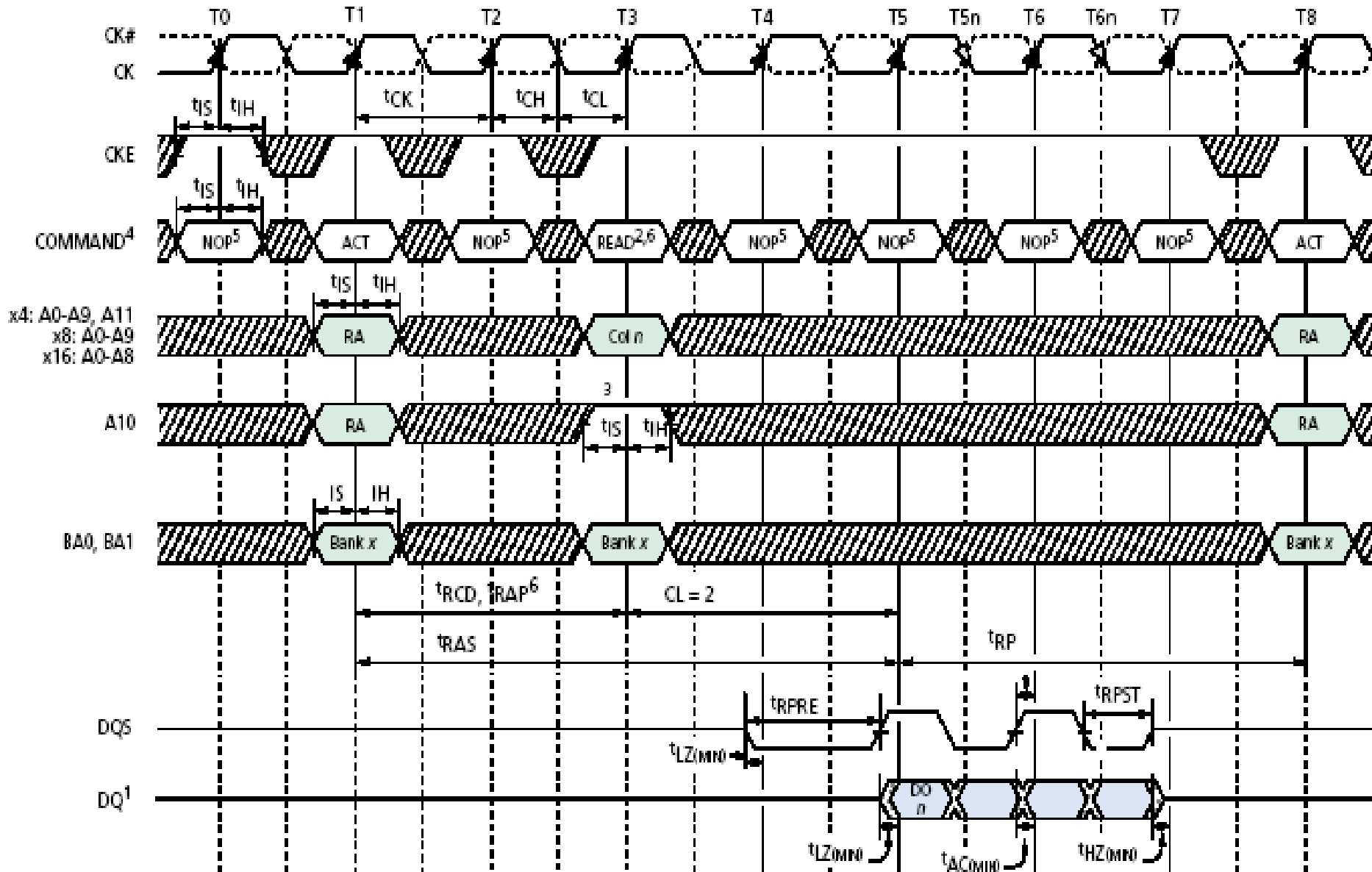
# READ burst (with auto precharge)



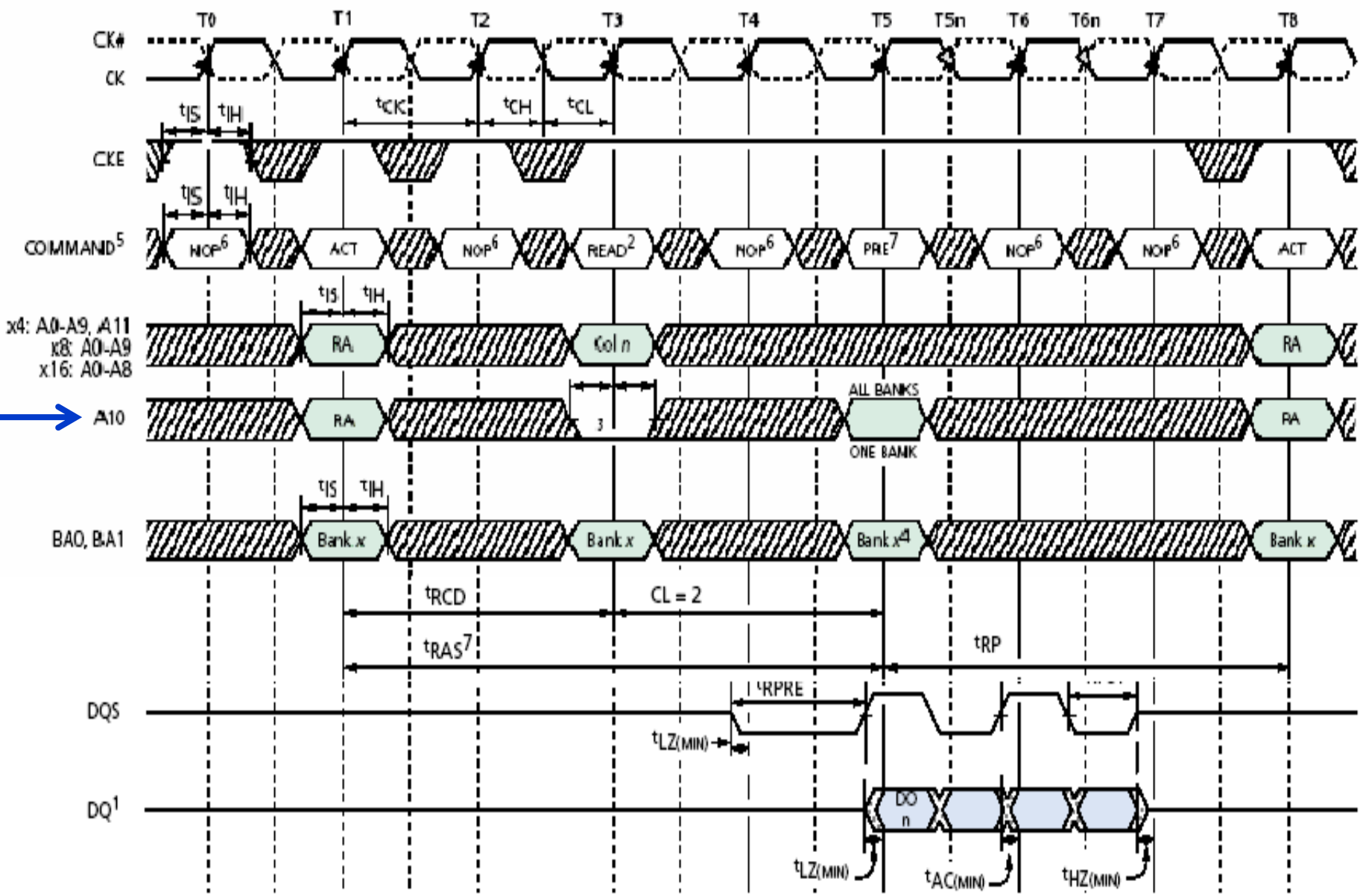
# WRITE burst (with auto precharge)



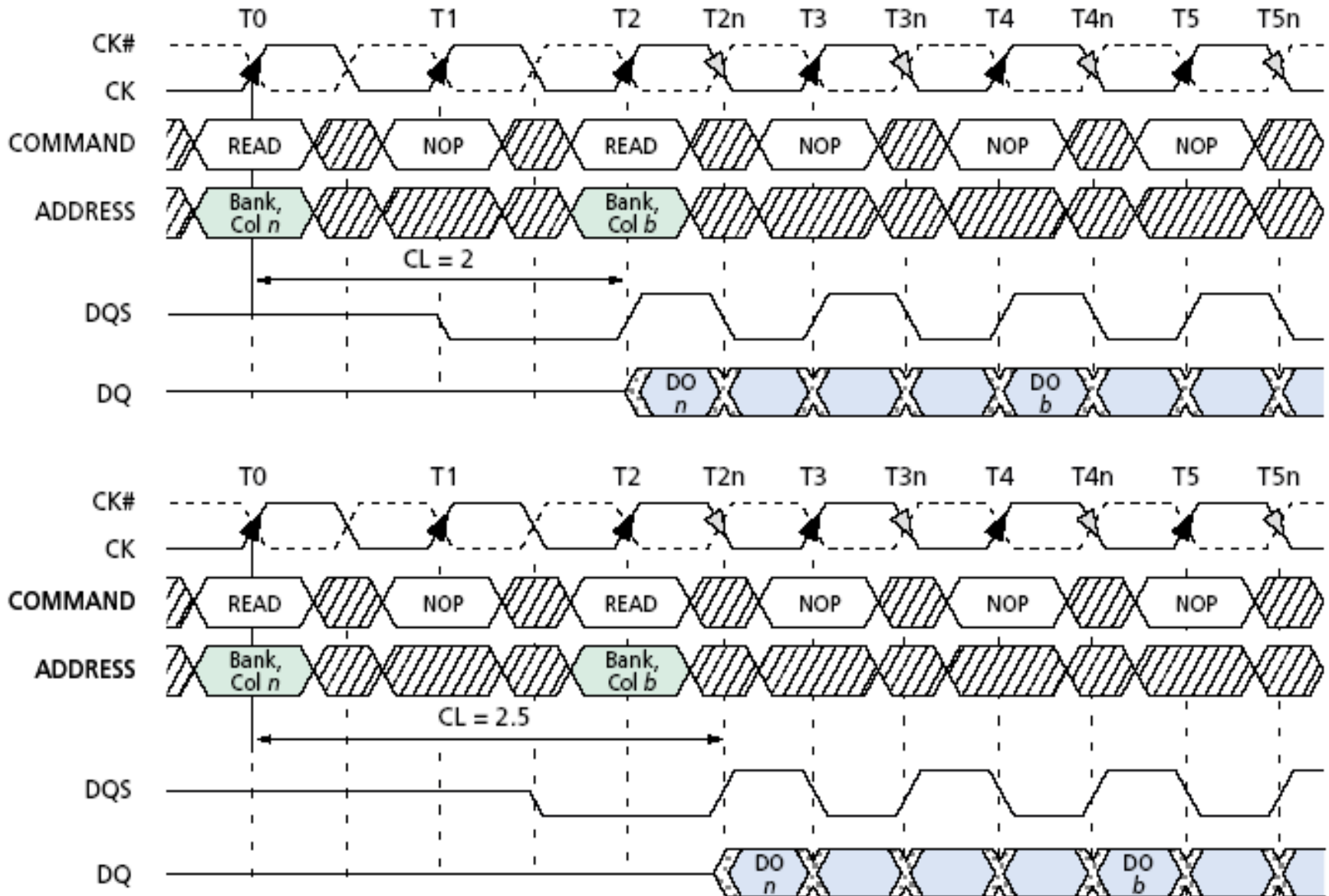
# DDR Read Command (with auto precharge)



# DDR Read Command (without auto precharge)

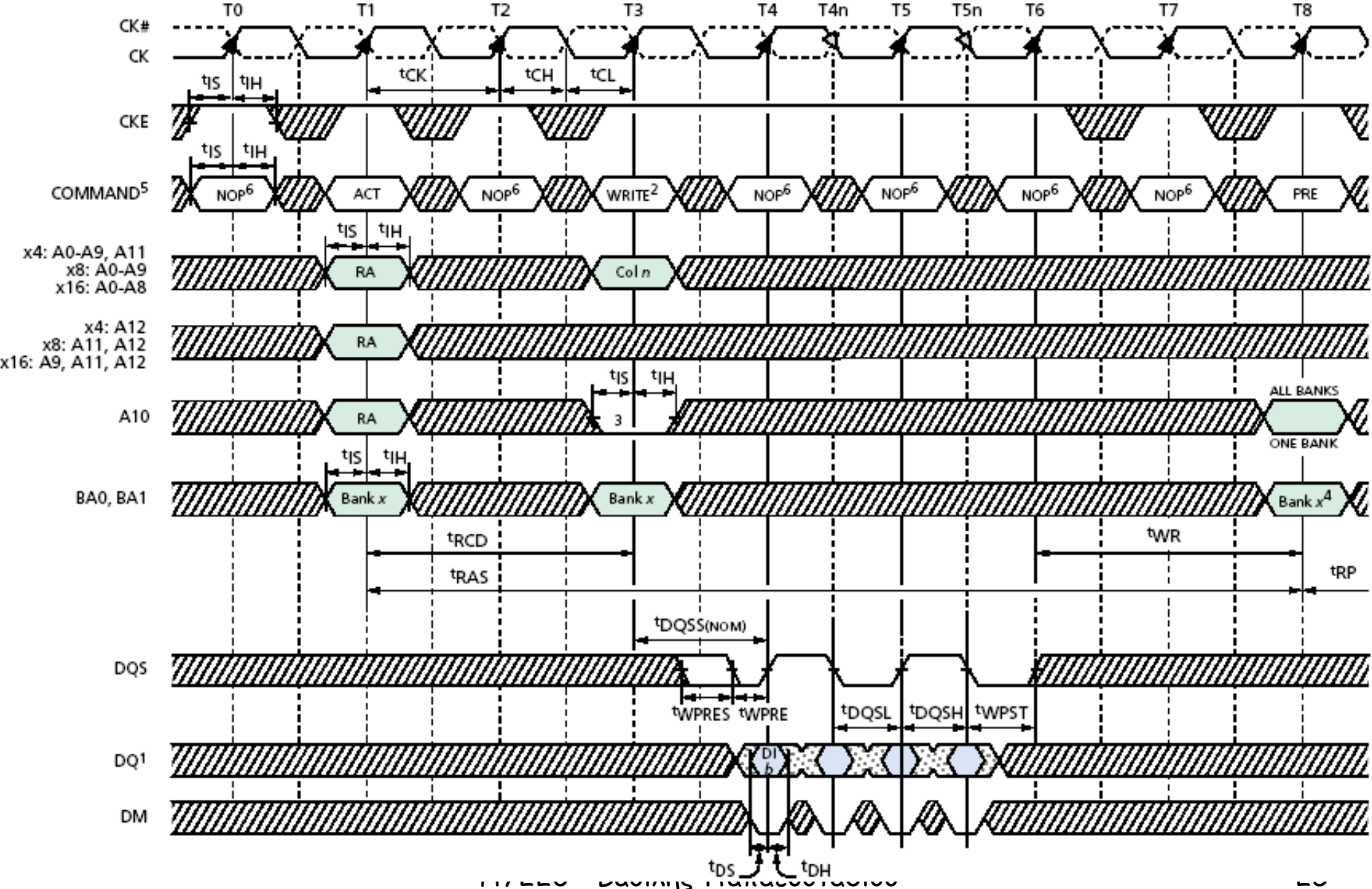


# DDR: Consecutive Read Bursts

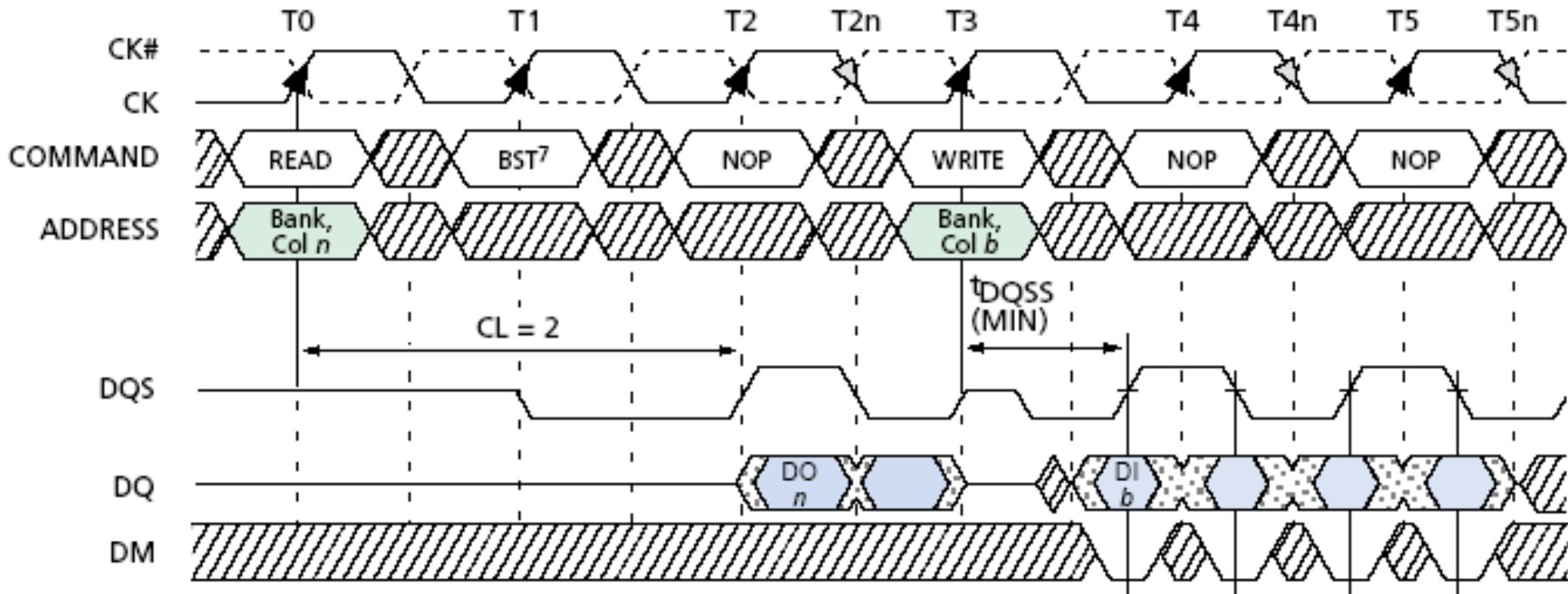




# DDR Write Command (without auto precharge)



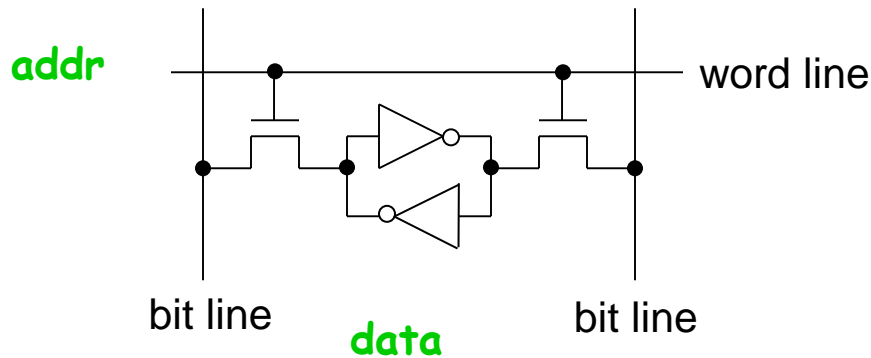
# DDR: Read - Burst Stop - Write



# Volatle Memory Comparison

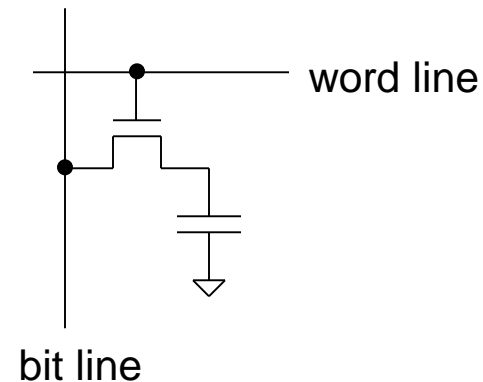
*The primary difference between different memory types is the bit cell.*

## • SRAM Cell



- Larger cell  $\Rightarrow$  lower density, higher cost/bit
- No dissipation
- Read non-destructive
- No refresh required
- Simple read  $\Rightarrow$  faster access
- Standard IC process  $\Rightarrow$  natural for integration with logic

## • DRAM Cell



- Smaller cell  $\Rightarrow$  higher density, lower cost/bit
- Needs periodic refresh, and refresh after read
- Complex read  $\Rightarrow$  longer access time
- Special IC process  $\Rightarrow$  difficult to integrate with logic circuits
- Density impacts addressing