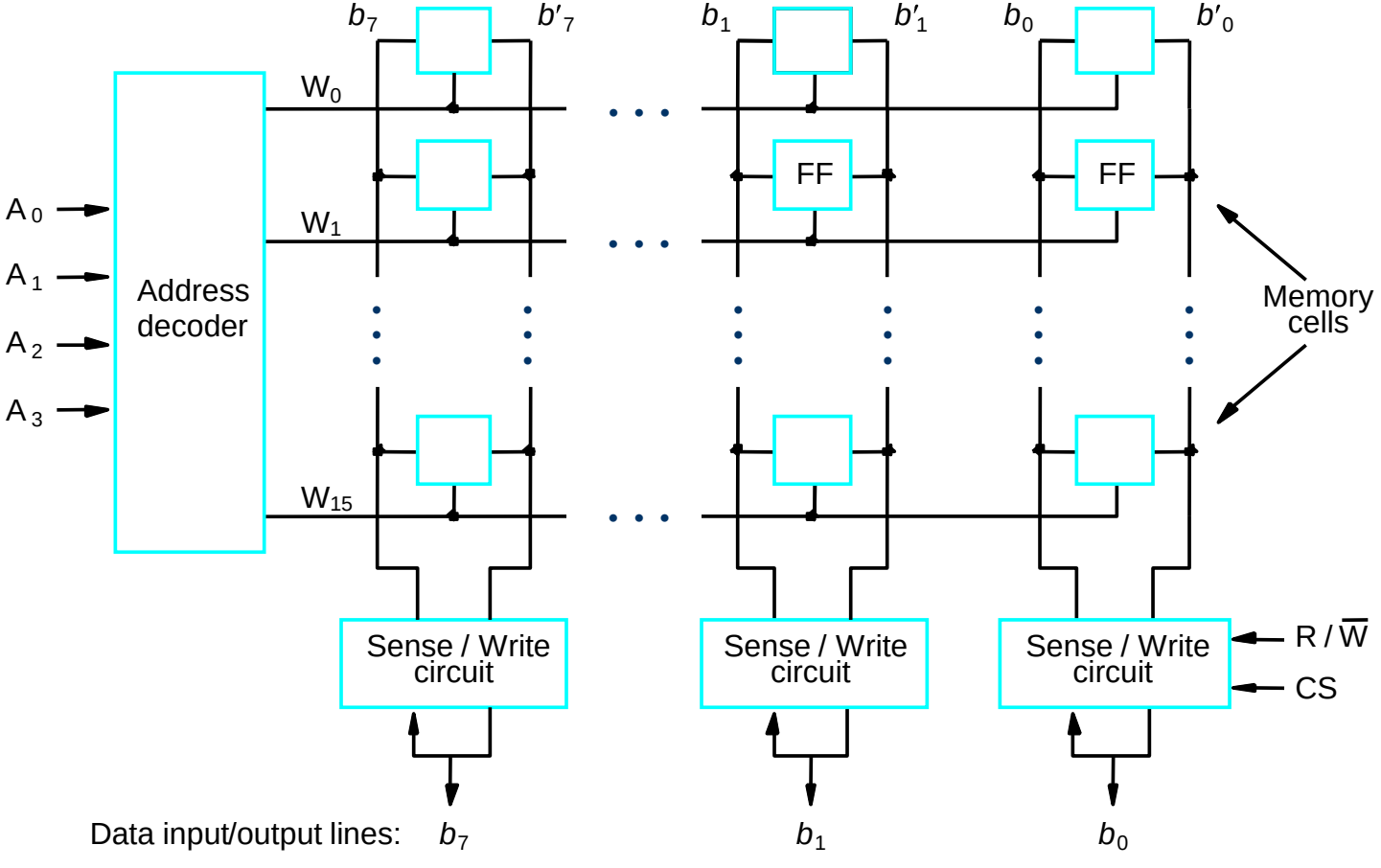
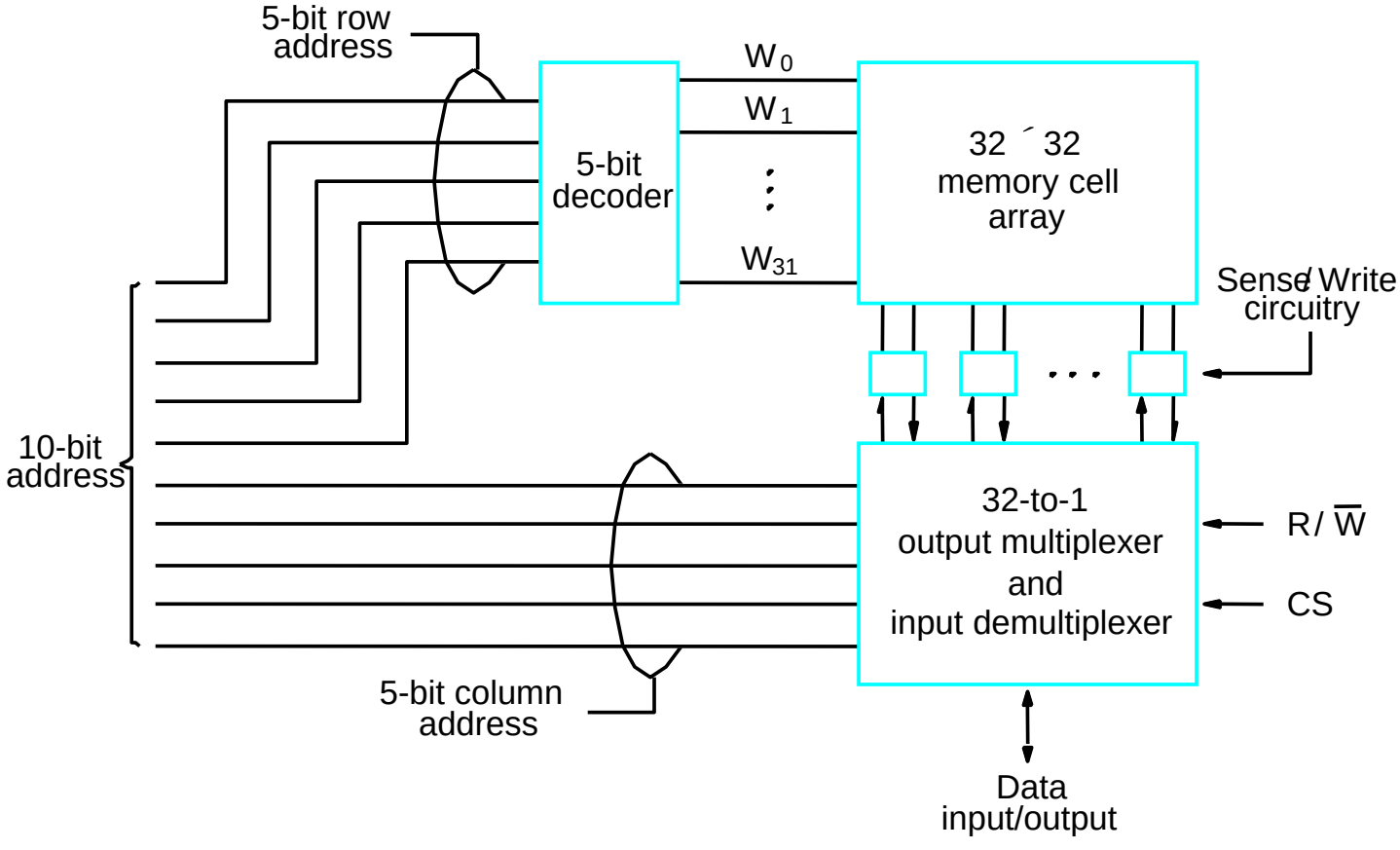
# Considerazioni Tecnologiche

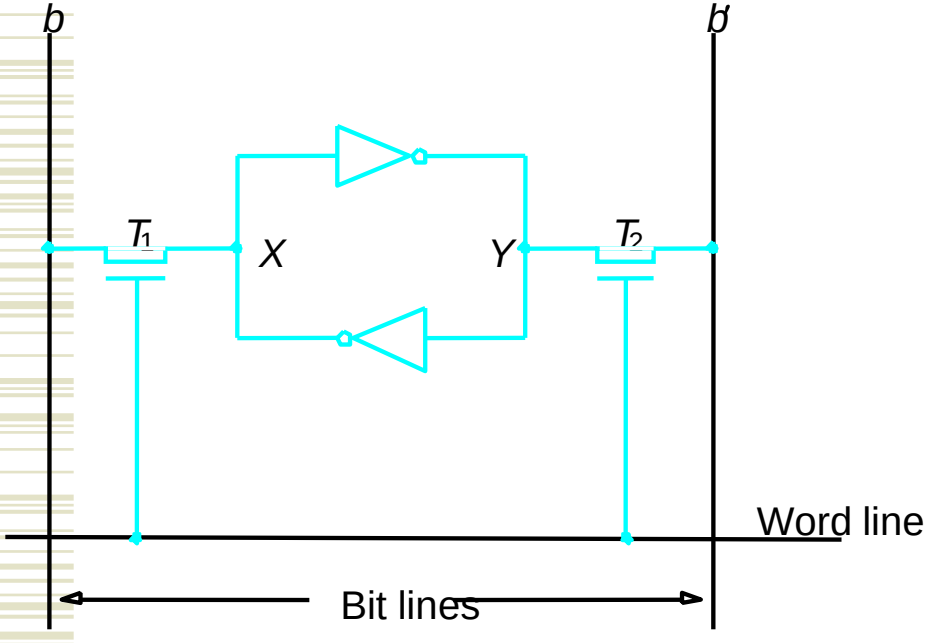
# Organizzazione di una RAM statica (16parole x 8bit)



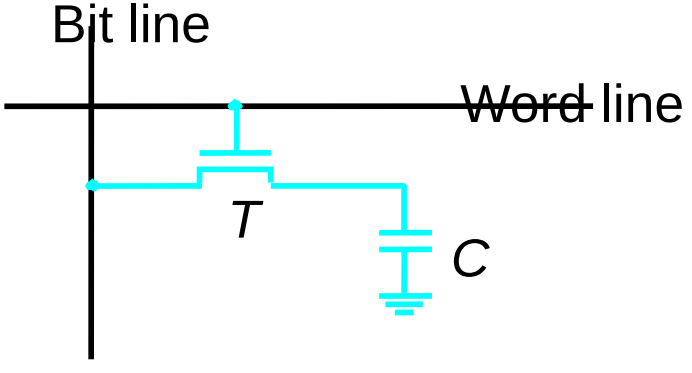
# Organizzazione di una memoria di 1K × 1



# Bit cell per memorie statiche e dinamiche (memorie volatili)

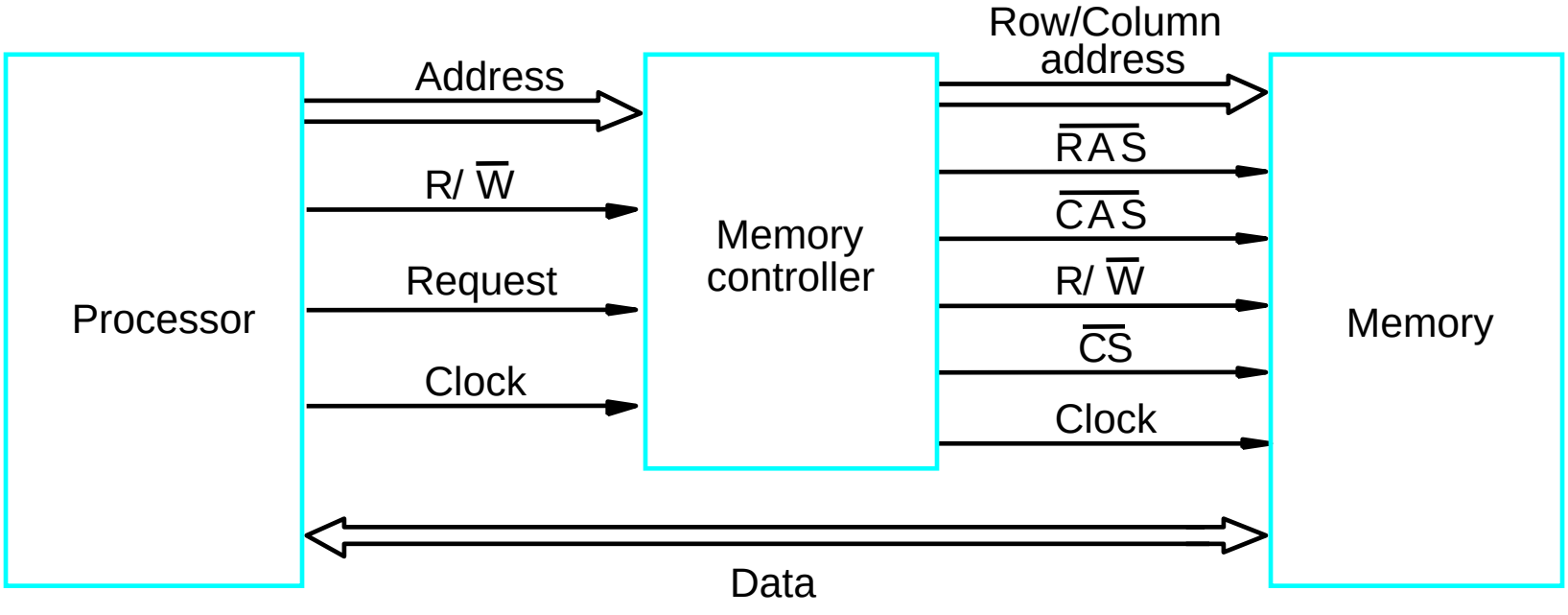



Cella di RAM statica (SRAM)

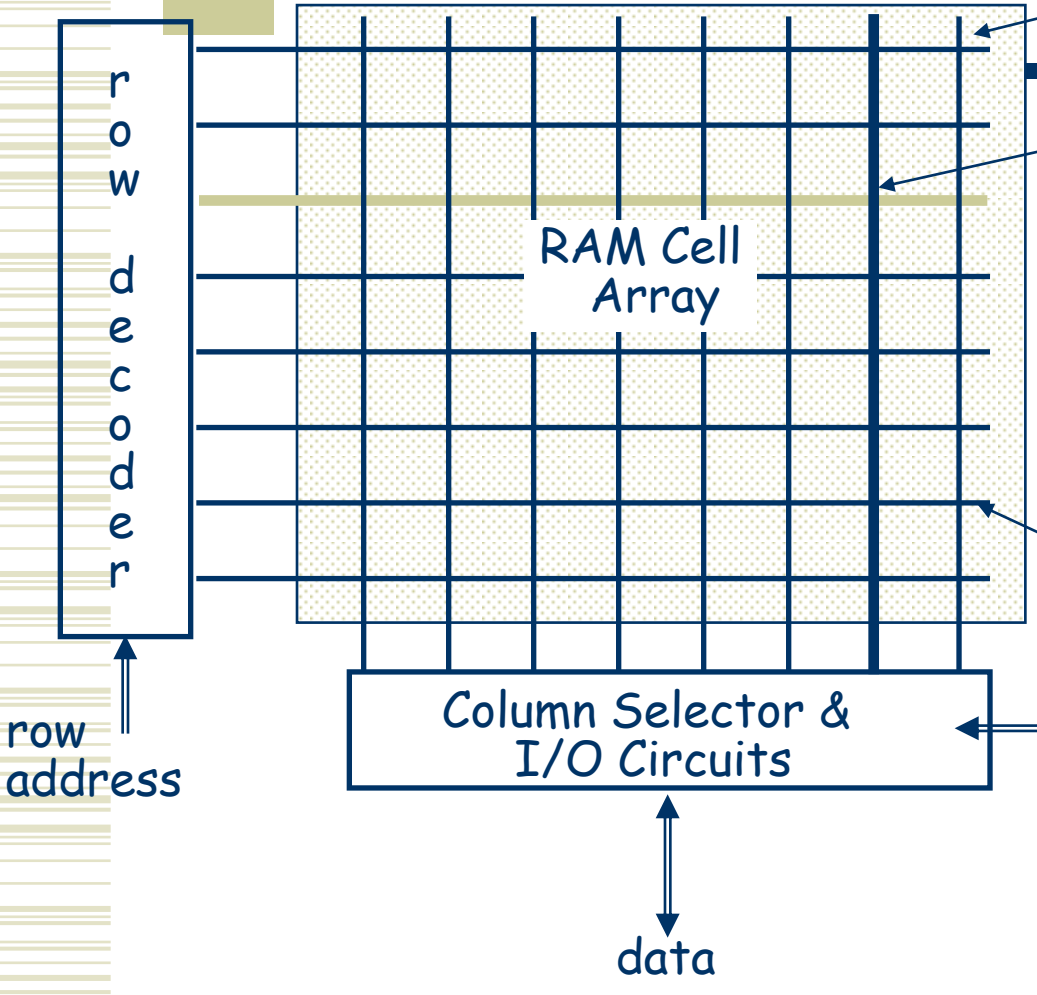



Cella di ram dinamica

# Utilizzo del memory controller



# Classical DRAM Organization

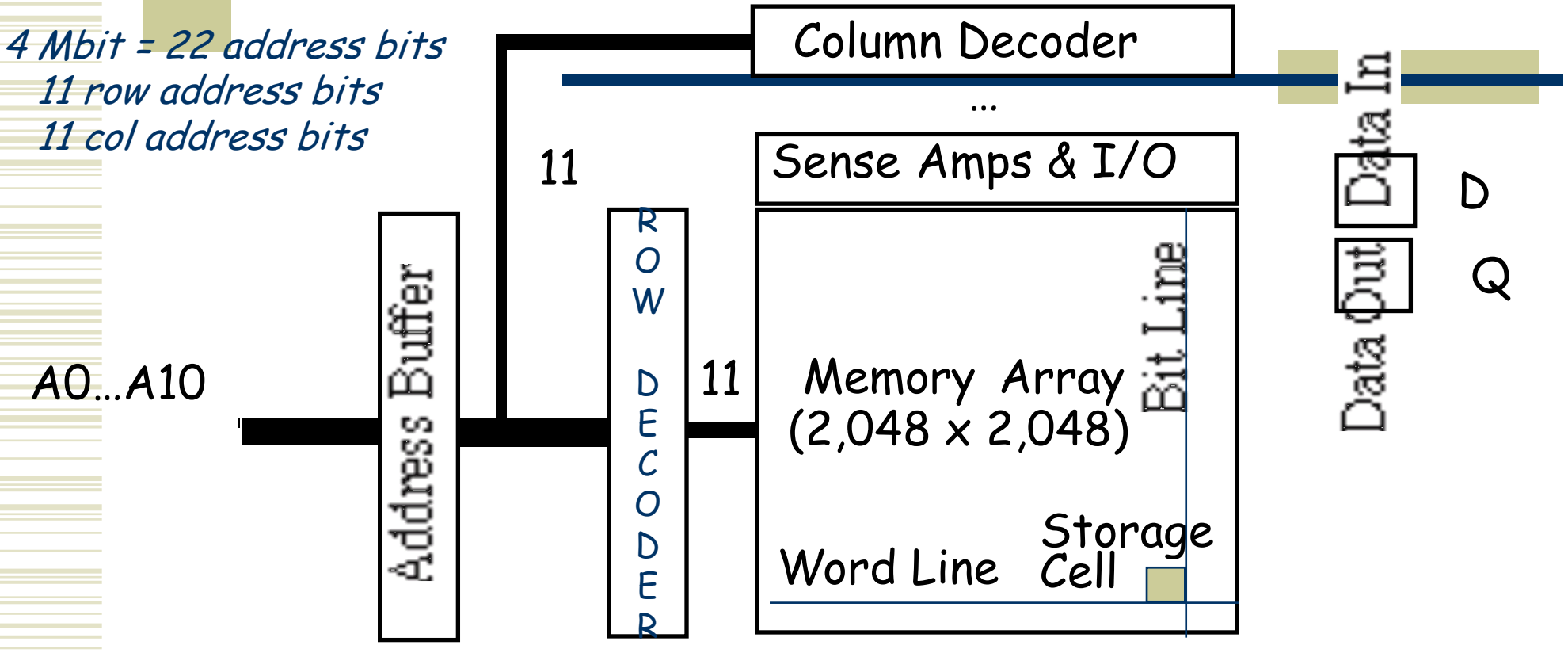



Each intersection represents a 1-T DRAM Cell

*Square keeps the wires short: Power and speed advantages Less RC, faster precharge and discharge is faster access time!*

Row and Column Address together select 1 bit a time

# DRAM Logical Organization (4 Mbit)

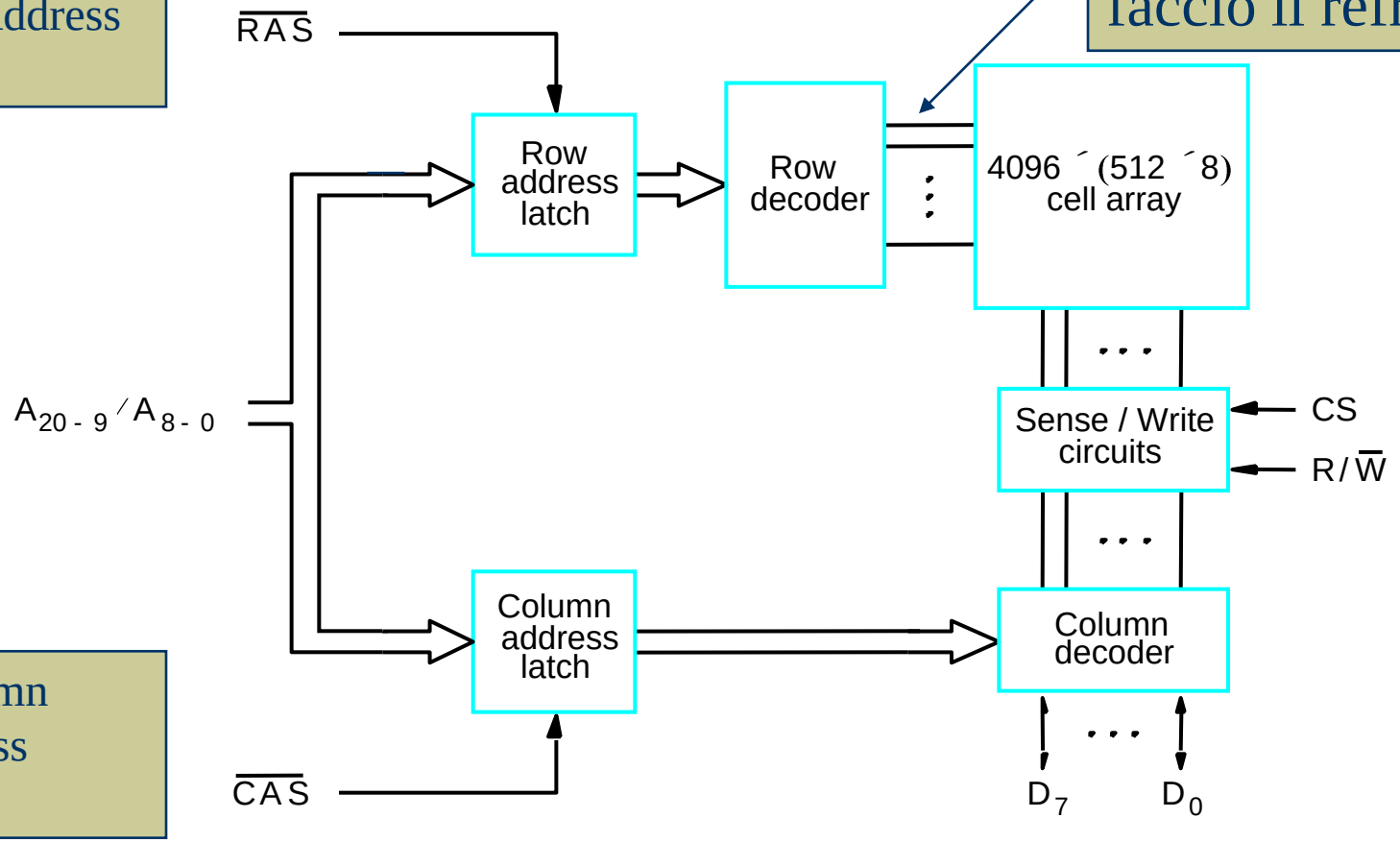



- Square root of bits per RAS/CAS
  - Row selects 1 row of 2048 bits from 2048 rows
  - Col selects 1 bit out of 2048 bits in such a row

# Asynchronous DRAM: Organizzazione di una memoria dinamica da 2M ´ 8

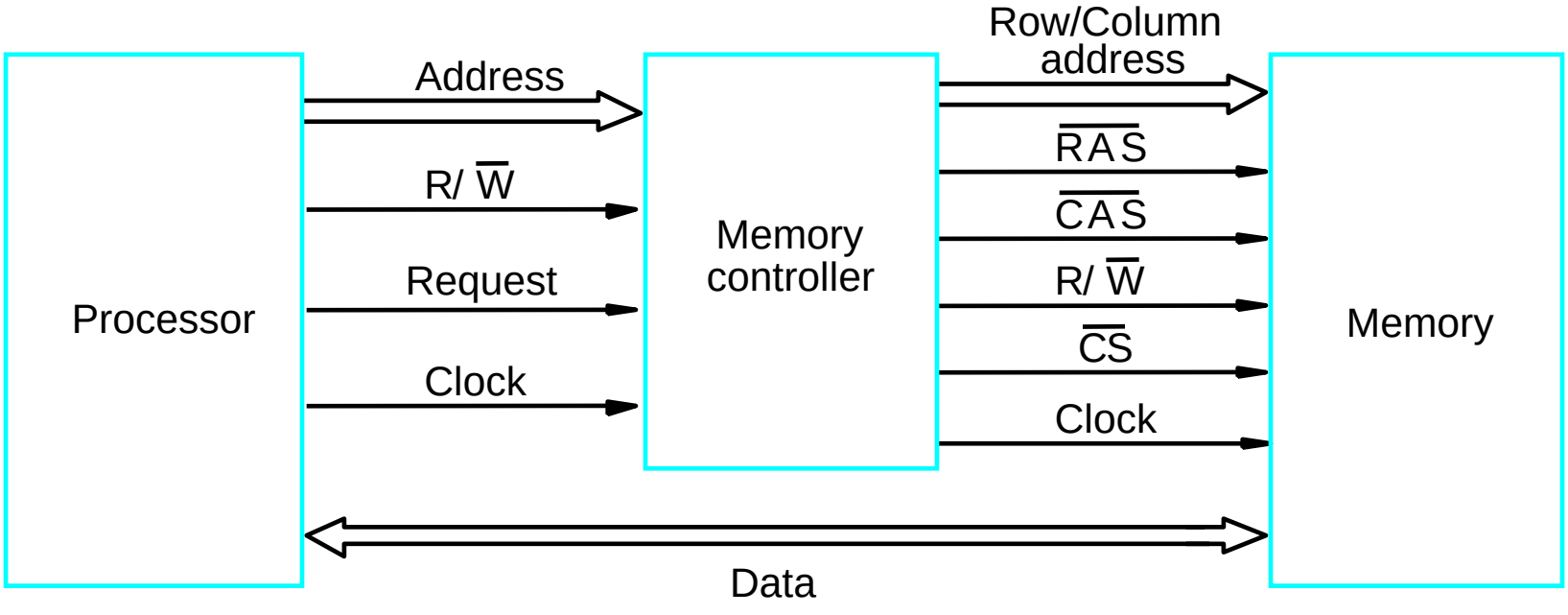
Seleziono una riga e faccio il refresh

Raw Address Strobe

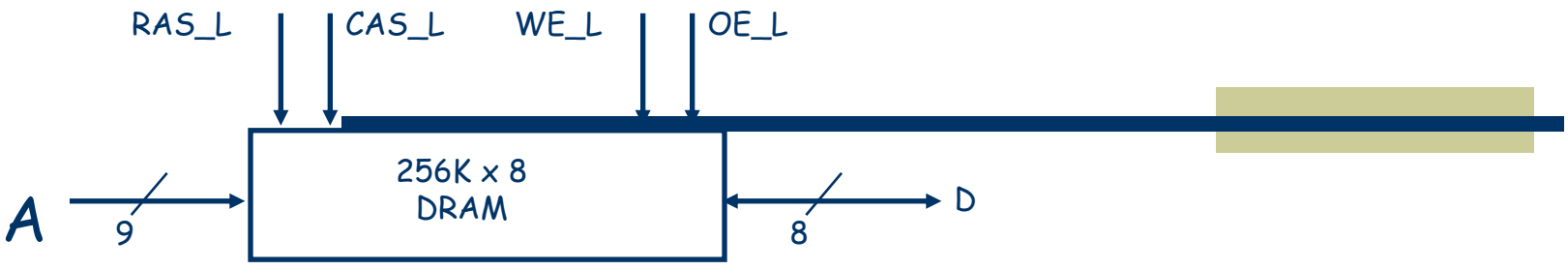



Coloumn Address Strobe

# Utilizzo del memory controller

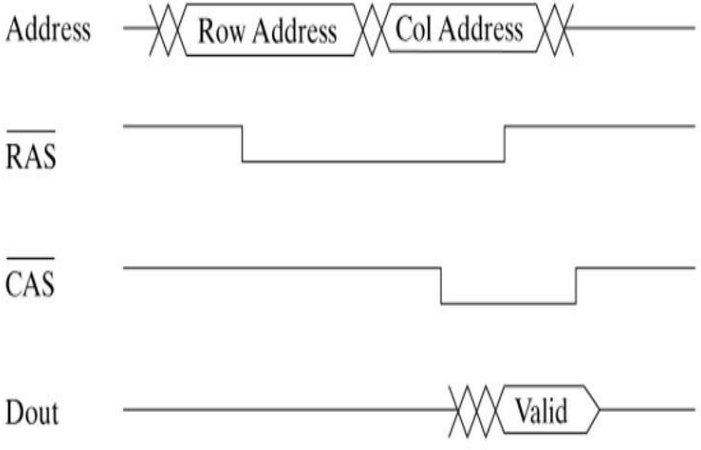


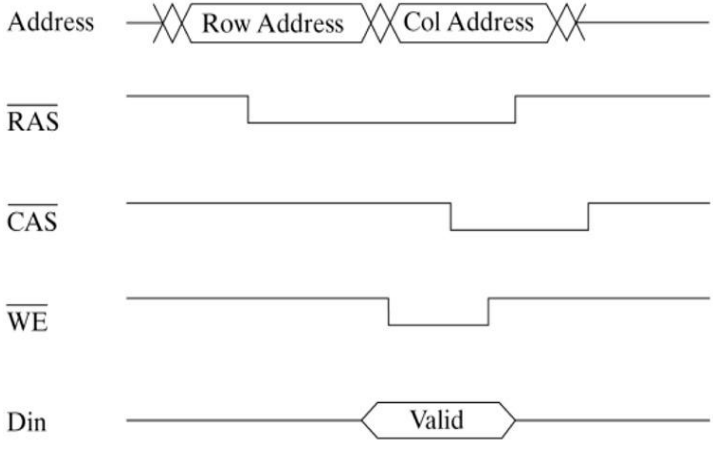
# Logic Diagram of a Typical DRAM




- 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

# Basic DRAM read & write

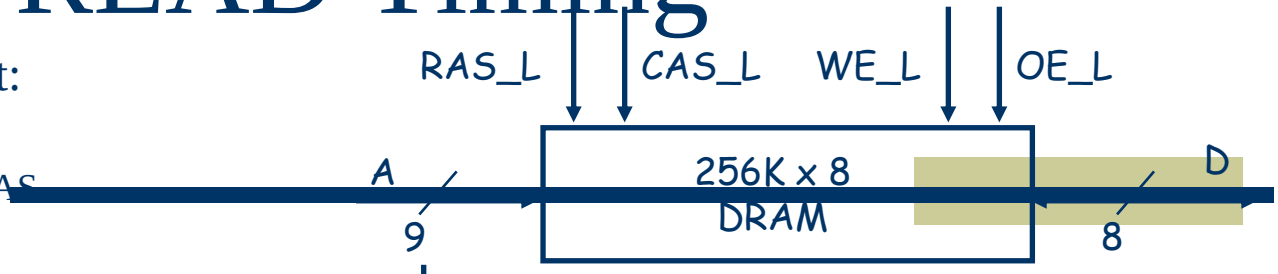


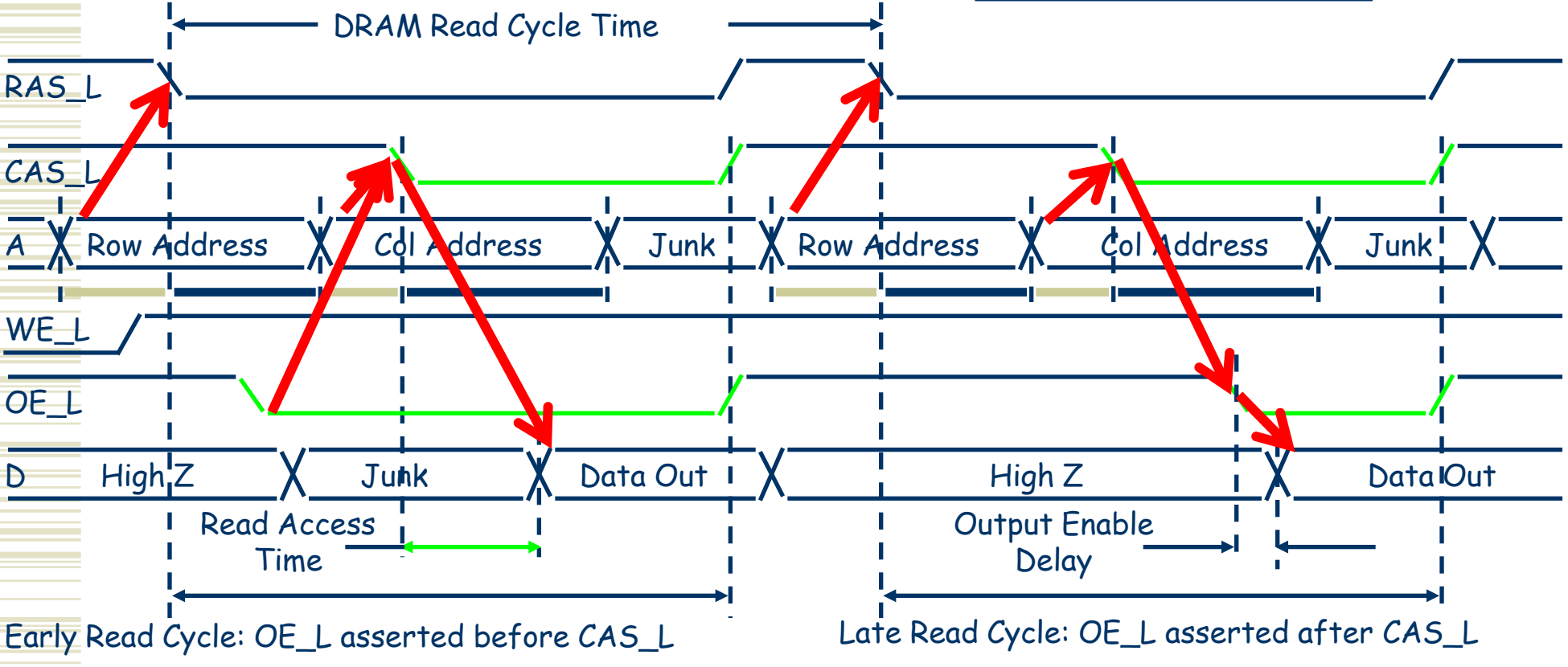




- Strobe address in two steps

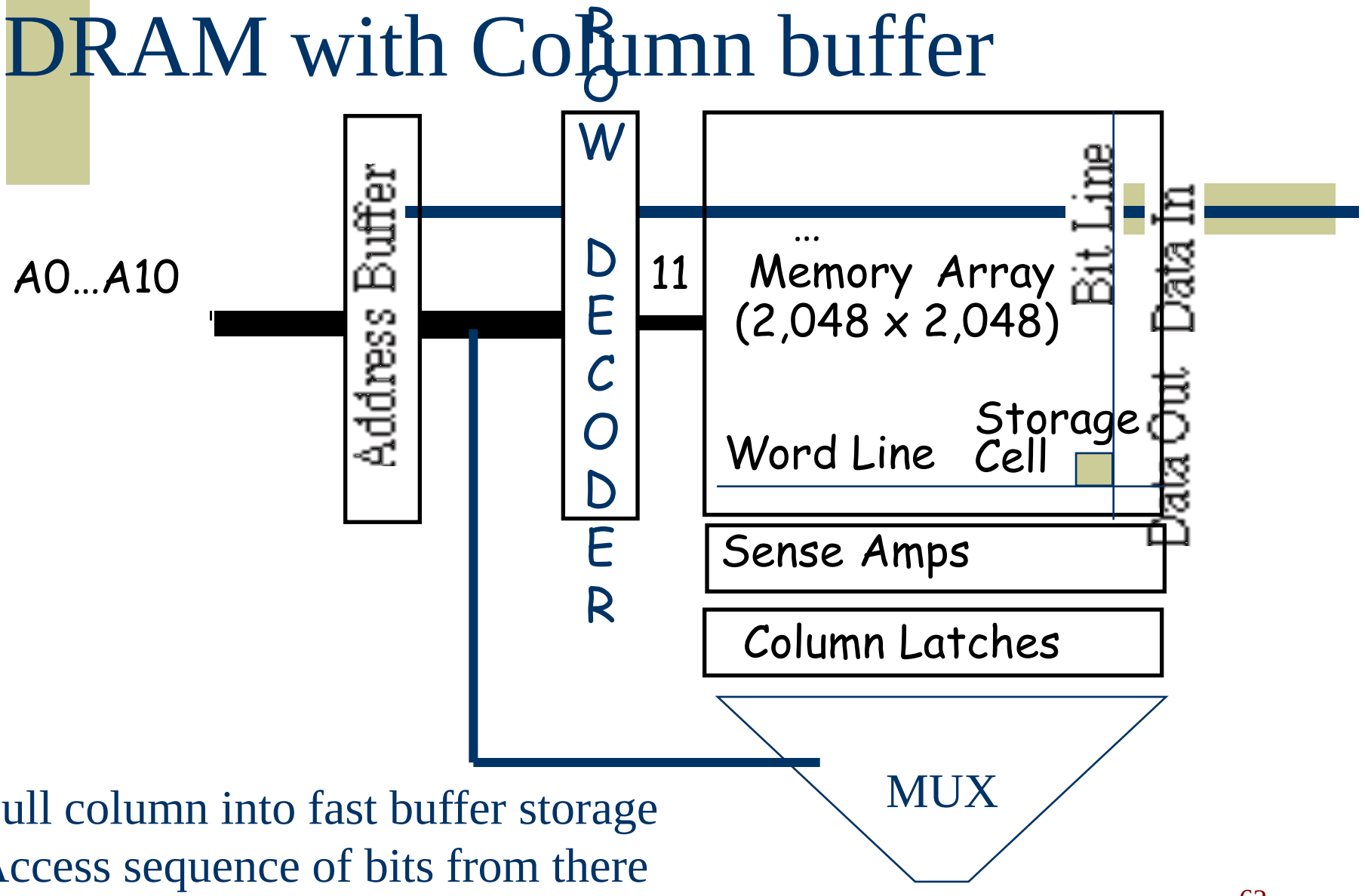
# DRAM READ Timing

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



# DRAM with Column buffer




Pull column into fast buffer storage
Access sequence of bits from there

# Optimized Access to Cols in Row

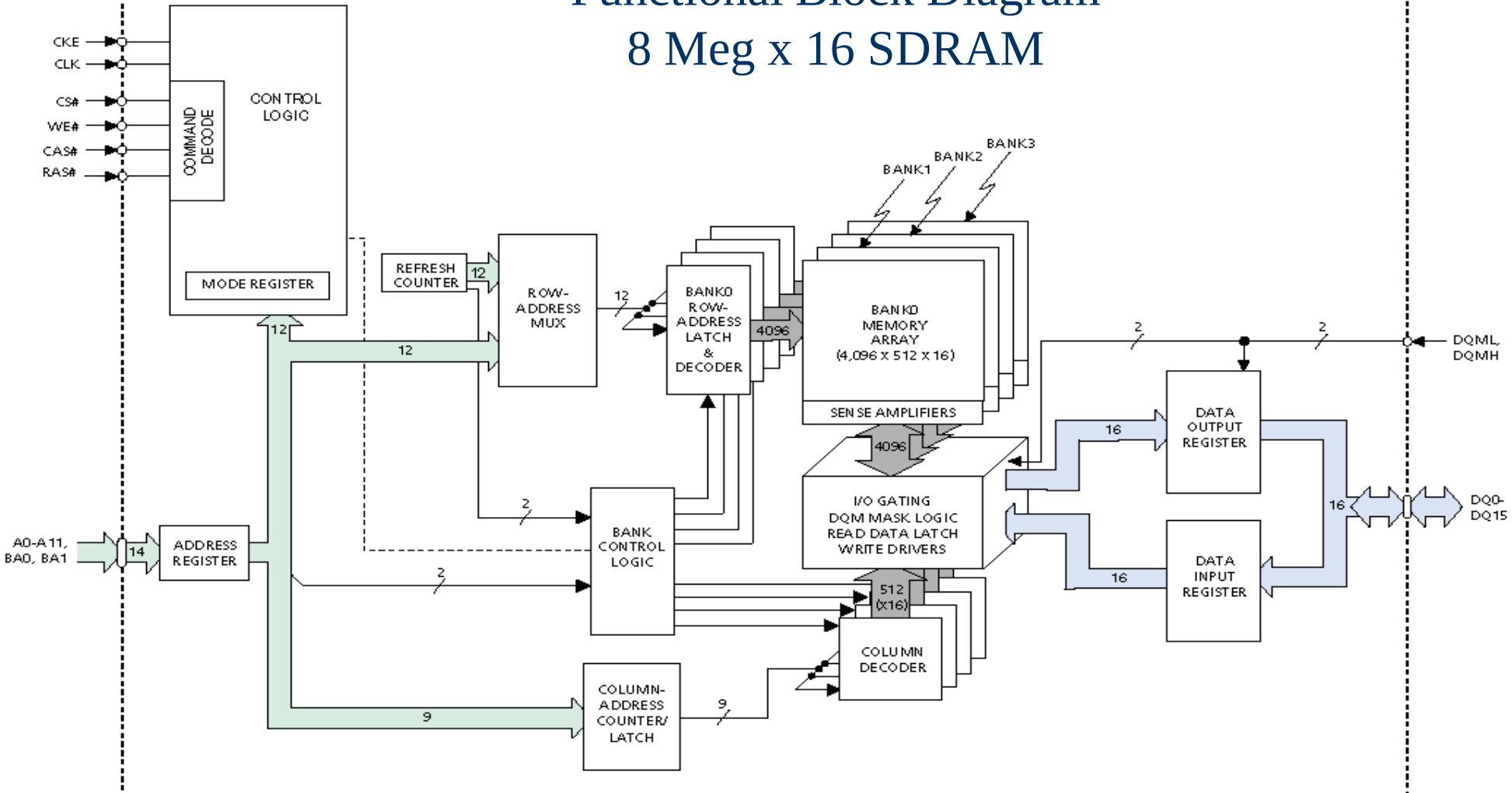
- Often want to access a sequence of bits

- 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

# More recent DRAM enhancements

- EDO - extended data out (similar to fast-page mode)
  - RAS cycle fetched rows of data from cell array blocks (long access time, around 100ns)
  - Subsequent CAS cycles quickly access data from row buffers if within an address page (page is around 256 Bytes)
- SDRAM - synchronous DRAM
  - clocked interface
  - uses dual banks internally. Start access in one bank then next, then receive data from first then second.
- DDR - Double data rate SDRAM
  - Uses both rising (positive edge) and falling (negative) edge of clock for data transfer. (typical 100MHz clock with 200 MHz transfer).
- RDRAM - Rambus DRAM
  - Entire data blocks are access and transferred out on a high-speed bus-like interface (500 MB/s, 1.6 GB/s)
  - Tricky system level design. More expensive memory chips.

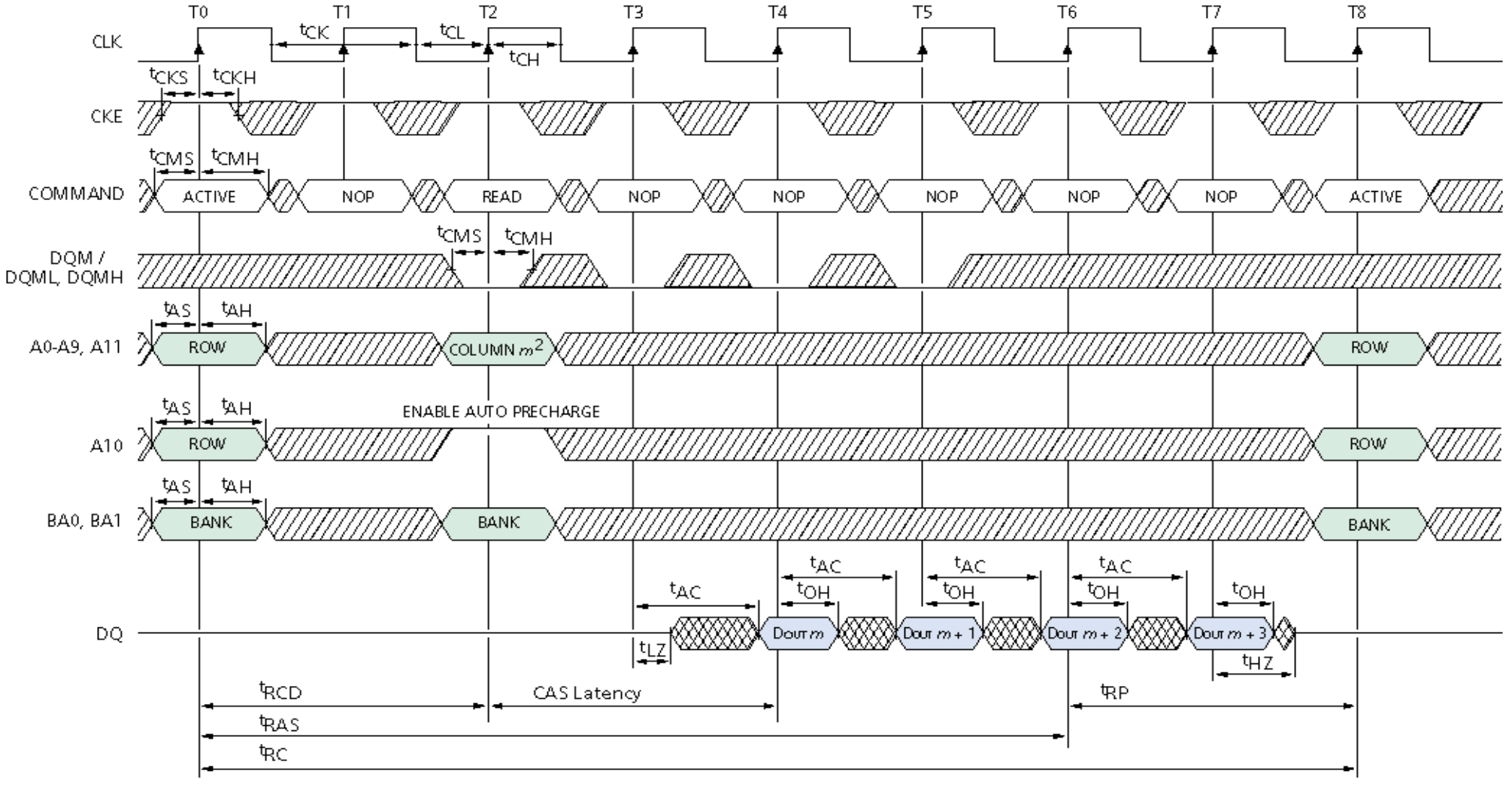
# Functional Block Diagram
# 8 Meg x 16 SDRAM



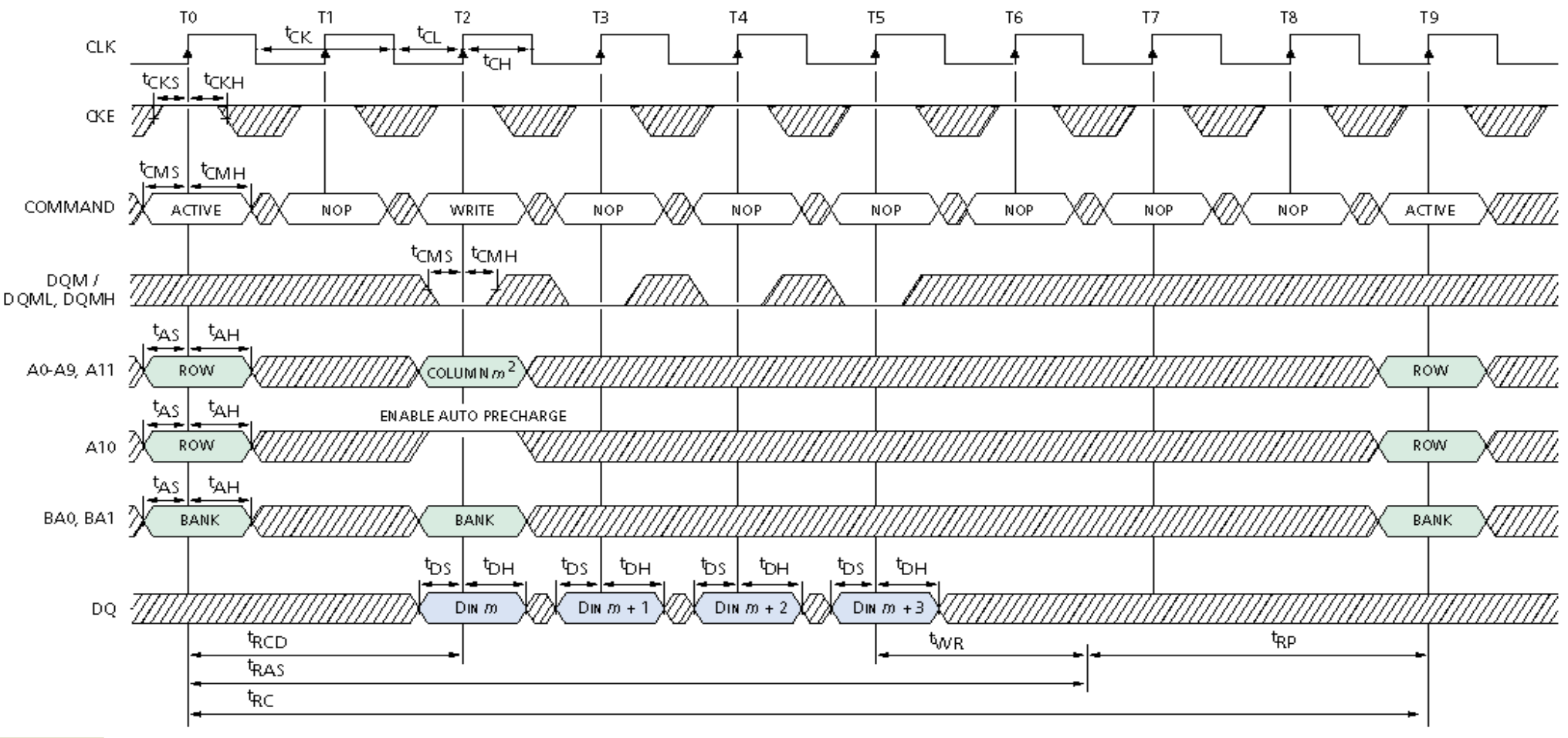
# SDRAM Details

- 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 as 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 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 ≈ 4 + n cycles (for n words)

# READ burst (with auto precharge)



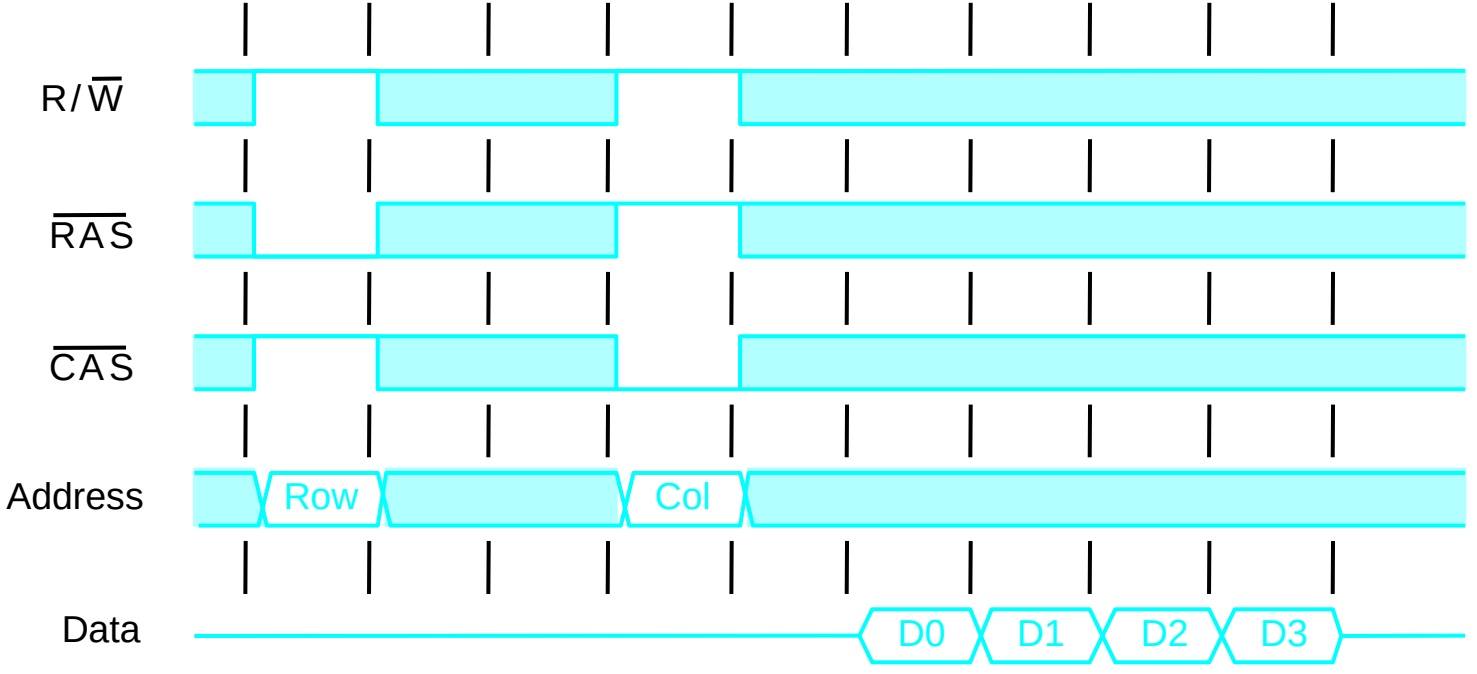
# WRITE burst (with auto precharge)



# Considerazioni sulla organizzazione ad array

- Locazioni di memoria consecutive si trovano sulla stessa linea;
- Se devo prelevare 2 o più byte consecutivi basta decodificare una sola volta l'indirizzo di riga e poi caricare gli indirizzi di colonna nei cicli successivi;
- Conclusione: il trasferimento di byte consecutivi avviene a frequenza doppia rispetto al trasferimento di 2 byte ad indirizzi casuali (aspetto importante nella progettazione delle cache)

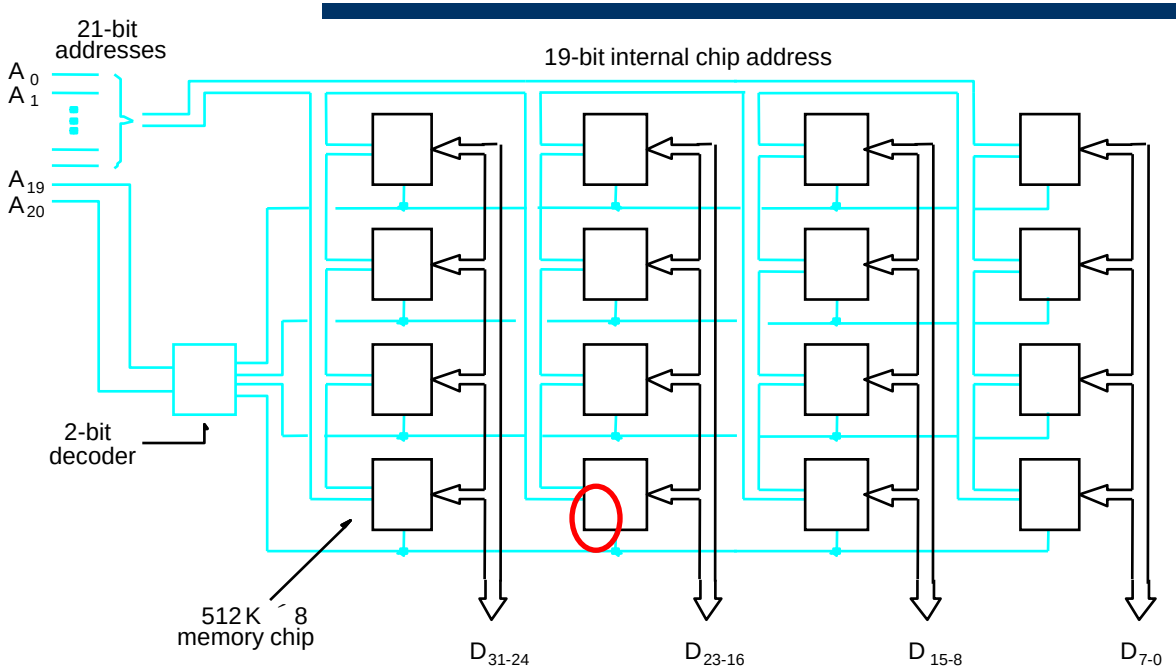
# Burst read of length 4

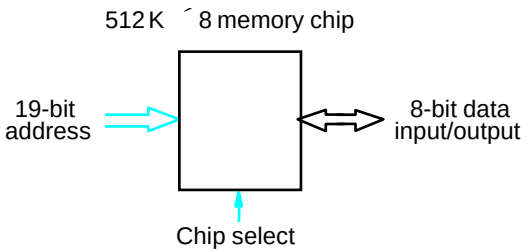


# RAM statiche Vs. RAM dinamiche

- Le RAM statiche sono molto veloci ma hanno costo e dimensioni elevate –> vengono usate per la memoria cache (piccole e veloci)
- Le RAM dinamiche non sono molto veloci ed hanno bisogno di un circuito di refresh però occupano uno spazio molto limitato, sono adatte per memorie più grandi → vengono usate per la memoria principale del calcolatore (più lenta ma molto grande).

# Progetto di sottosistemi di memoria: Organization of a 2M × 32 memory module using 16 modules 512K × 8 static memory chips



# Riferimenti

- K. Hamacher, Computer Organization
  - Cap 5: The memory system
- Hennessy & Patterson: Computer Architecture, a quantitative approach (3rd Edition)
  - Cap 5