

# Electronica

## Sistemi Digitali

### 10. Le memorie

Roberto Roncella



# Le memorie

## Circuiti di memoria a stato solido

(Millmann 16.1-8, data sheet)

Tipi di memorie

Classificazione

Memorie SRAM

Interfaccia elettrica e logica

Indirizzamento bidimensionale

Realizzazione di moduli SRAM

# Che cos'è una memoria

- Sistema in grado di **conservare informazione**
  - Alcune aree del cervello, un libro, un taccuino, un badge magnetico, un hard-disk, un CD-ROM, un DVD, una flash-pen, una SD card, un elemento bistabile, ... sono tutti esempi di memoria
- Memorie a semiconduttore
  - L'informazione è un livello di tensione
    - Deve essere compatibile con i circuiti logici
  - L'informazione richiede un supporto fisico **elettrico**
    - Presenza o assenza di **connessioni**
    - Circuiti **bistabili** elementari
    - Carica in **condensatori**

# Classificazione delle memorie

- Secondo ... l'attitudine a **mantenere** il contenuto senza uso di alimentazione esterna
  - Non volatili
  - Volatili
- ... il tempo di **permanenza** con alimentazione
  - Statiche
  - Dinamiche, pseudostatiche
- ... la modalità di **accesso**
  - Ad accesso casuale
  - Ad accesso sequenziale
  - Miste

# Tipi di accesso

- Accesso casuale (**random**)
  - Si può accedere a qualsiasi parola interna specificandone l'indirizzo
    - 😊 Tempo di accesso breve, indipendente dalla posizione
    - 😞 Presenza di circuiti complessi di decodifica e selezione
- Accesso sequenziale (**shift**)
  - I dati sono collocati nella memoria sequenzialmente. Per accedere alla locazione n occorrono n accessi successivi
- Accesso misto
  - Si accede casualmente a blocchi che sono trasferiti sequenzialmente
    - Hard disk, flash pen

# Memorie non volatili e volatili

- Memorie **non volatili**
  - Sistemi che conservano l'informazione anche in assenza di alimentazione
  - L'informazione può essere scritta durante la realizzazione o dall'utente con procedure particolari
    - La scrittura è significativamente più lenta della lettura
- Memorie **volatili**
  - In assenza di alimentazione perdono l'informazione e tornano in uno stato "vergine"
  - Le memorie possono essere "scritte e riscritte" dall'utente durante il normale funzionamento
    - Scrittura e lettura sono ragionevolmente simmetriche

# Memorie non volatili

- **ROM** (Read Only Memory)
  - Programmata in fabbrica
- **PROM** (Programmable ROM)
  - Programmabile dall'utente una volta per sempre
- **EPROM** (Erasable PROM)
  - Programmabile più volte dall'utente
    - Si programma con opportuni segnali elettrici ai terminali
    - Si cancellazione esponendo il chip agli UV
- **E<sup>2</sup>PROM** (Electrically EPROM)
  - Programmabile più volte dall'utente
    - Si programma e si cancella con opportuni segnali elettrici

# Tipi di memorie volatili

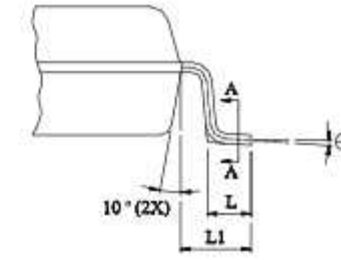
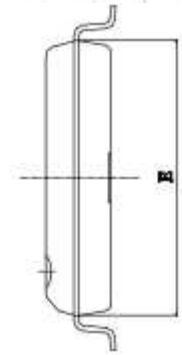
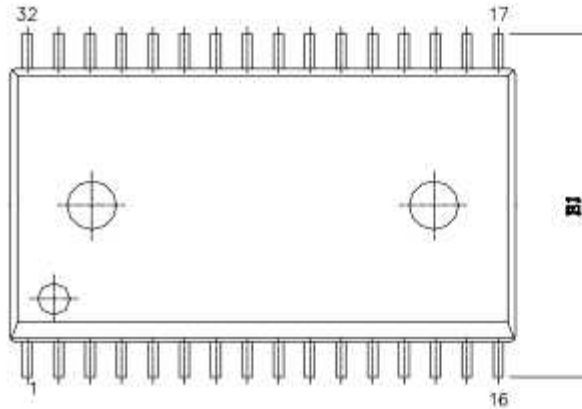
- RAM (Random Access Memory)
  - Memoria nella quale è possibile
    - Scrivere (WRITE, W) e leggere (READ, R)
- Esistono diverse tipologie di RAM
  - RAM Statica (**SRAM**)
    - Conserva l'informazione fino a quando è alimentata
      - 😊 Più veloce
      - 😞 Dissipa di più, ha una area per bit maggiore
  - RAM Dinamica (DRAM, SDRAM, ...)
    - L'informazione deve essere periodicamente “rinfrescata”
    - Usa un condensatore come elemento di memoria
      - 😊 Dissipazione e area per bit minore, più dense
      - 😞 Più lenta e complessa



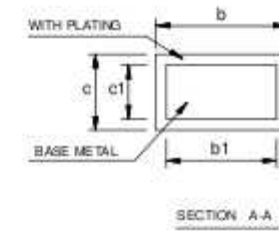
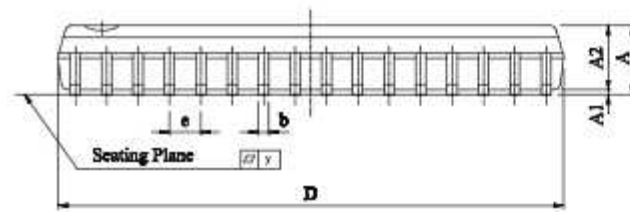
# Organizzazione di una SRAM

- Una memoria è in genere organizzata in un insieme di  $N$  parole, di  $W$  bit ciascuna
  - La capacità in bit della memoria è  $N \times W$
  - $N$  è generalmente una potenza del 2
    - $2^{10} = 1,024$                       1 k (Kilo)
    - $2^{20} = 1,048,576$                       1 M (Mega)
    - $2^{30} = 1,073,741,824$                       1 G (Giga)
  - Ogni parola è individuata da un indirizzo
    - Per  $N = 2^n$  indirizzi servono  $n = \log_2(N)$  linee
  - $W$  solitamente può valere 1, 4, 8
    - Ma si trovano anche valori diversi come 3 o 9

# Una SRAM 128 k × 8



**BS62LV1027-2.3**

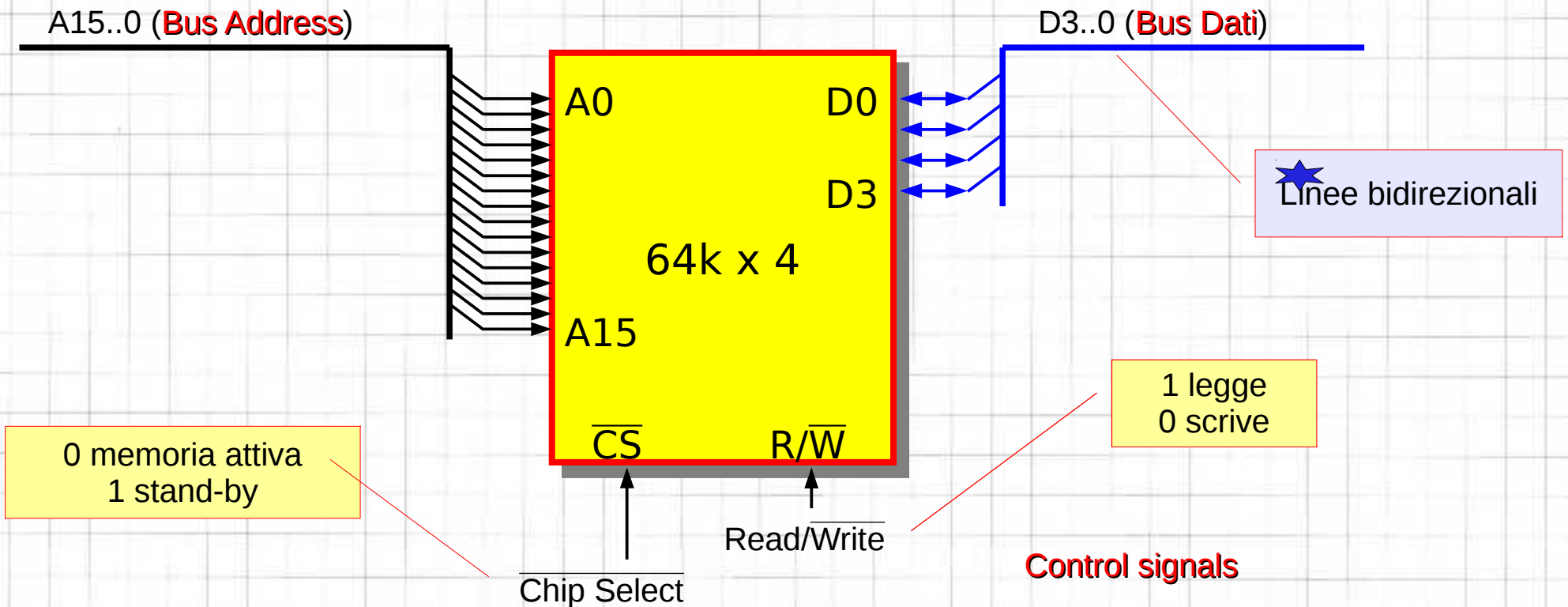


SECTION A-A

*SOP -32*

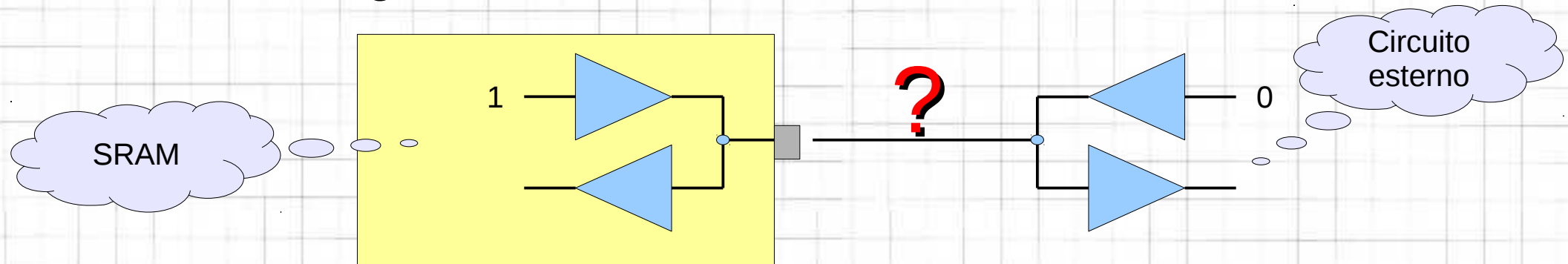
# Descrizione ai terminali

- Memoria SRAM 64k x 4



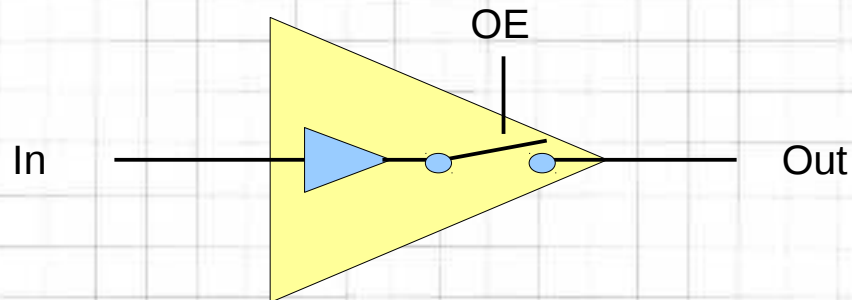
# Linee bidirezionali

- Le SRAM ( $W > 1$ ) hanno linee dati **bidirezionali**
  - Per risparmiare linee di interfaccia
    - Lettura e scrittura avvengono sempre in **momenti distinti**
- Occorrono elementi logici in grado di gestire il flusso di informazione nelle due direzioni
  - Evitando il conflitto tra i due driver che possono decidere il valore logico del nodo



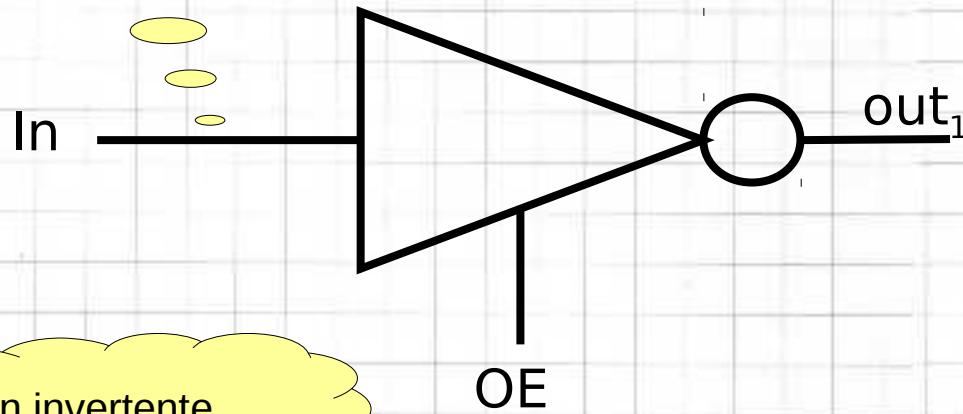
# Uscita tri-state

- Si introduce un nuovo stato logico **Z**
  - Sconnessione, alta impedenza o tri-state
- Più uscite tri-state si possono connettere insieme per pilotare un'unica linea
  - Le uscite tri-state hanno una **abilitazione**
  - Occorrono protocolli di abilitazione che garantiscano l'abilitazione di una sola uscita alla volta

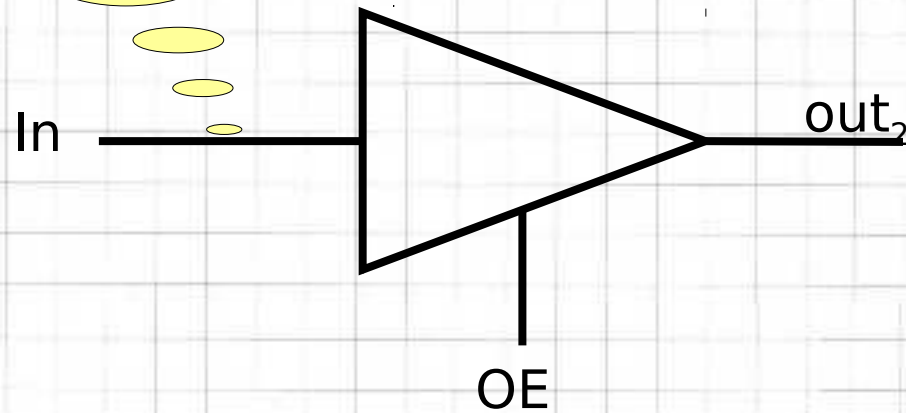


# Buffer tri-state

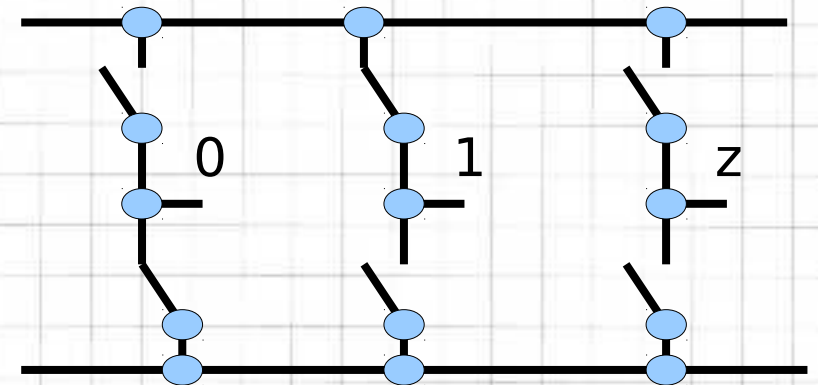
Invertente



Non invertente

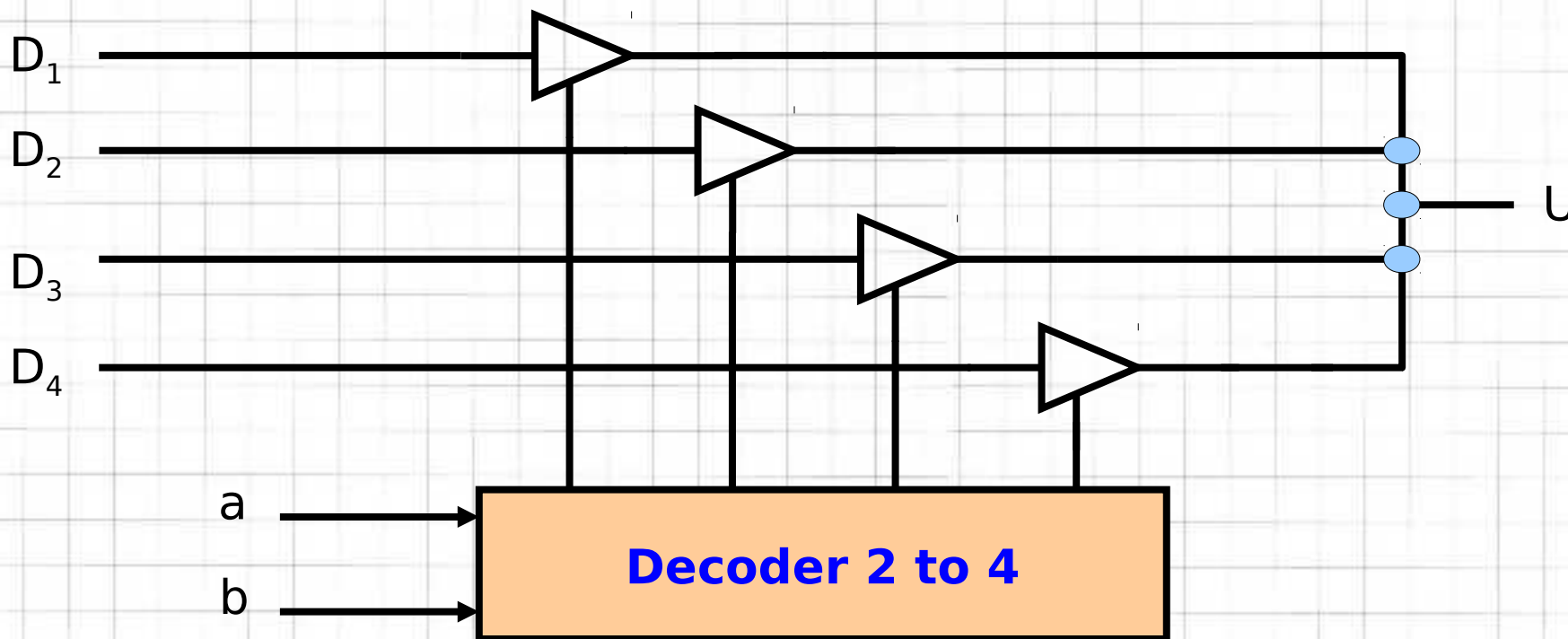


OE	In	out <sub>1</sub>	out <sub>2</sub>
0	0	Z	Z
0	1	Z	Z
1	0	1	0
1	1	0	1



# Un'altra applicazione

Multiplexer 4 a 1



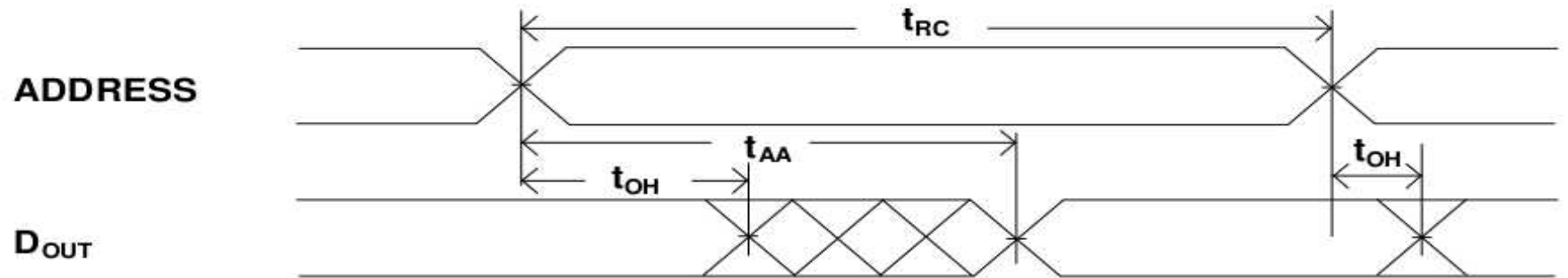
# Temporizzazione

- Le SRAM sono **reti sequenziali asincrone**
  - Non esiste un terminale di clock
  - Le operazioni di lettura e scrittura devono essere eseguite rispettando **tempi specifici**
  - I produttori indicano su grafici cartesiani le sequenze che devono rispettare i segnali all'interfaccia per leggere e scrivere correttamente
    - Cicli di lettura e scrittura
    - Cicli di letture o scritture multiple
    - Ciclo read-modify-write



# Read cycle (1)

## READ CYCLE 1 <sup>(1,2,4)</sup>

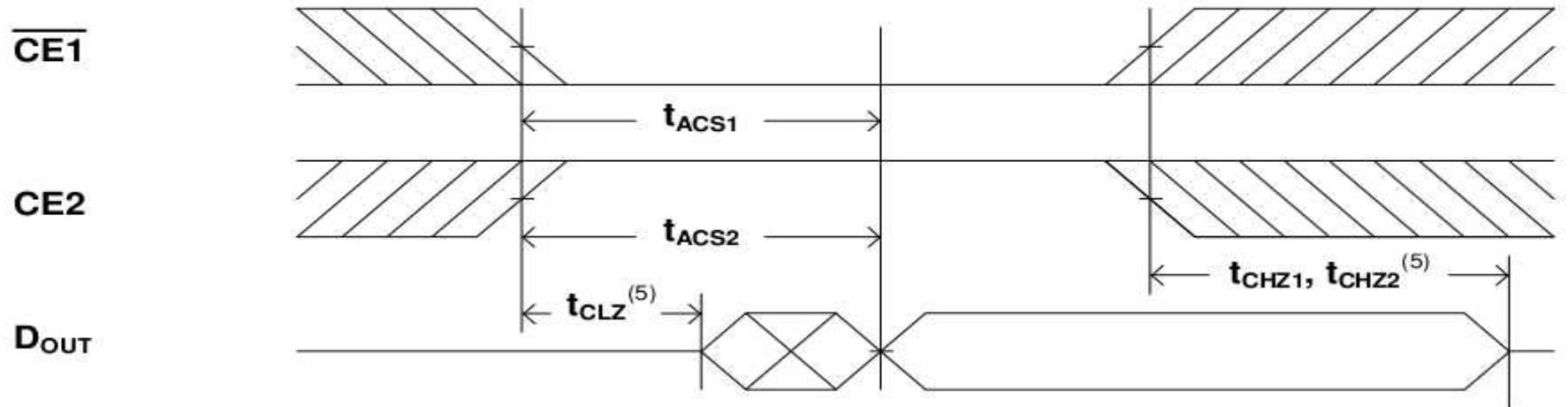


### NOTES:

1. WE is high in read Cycle.
2. Device is continuously selected when CE1 = VIL and CE2= VIH.
4. OE = VIL.

# Read cycle (2)

## READ CYCLE 2 <sup>(1,3,4)</sup>

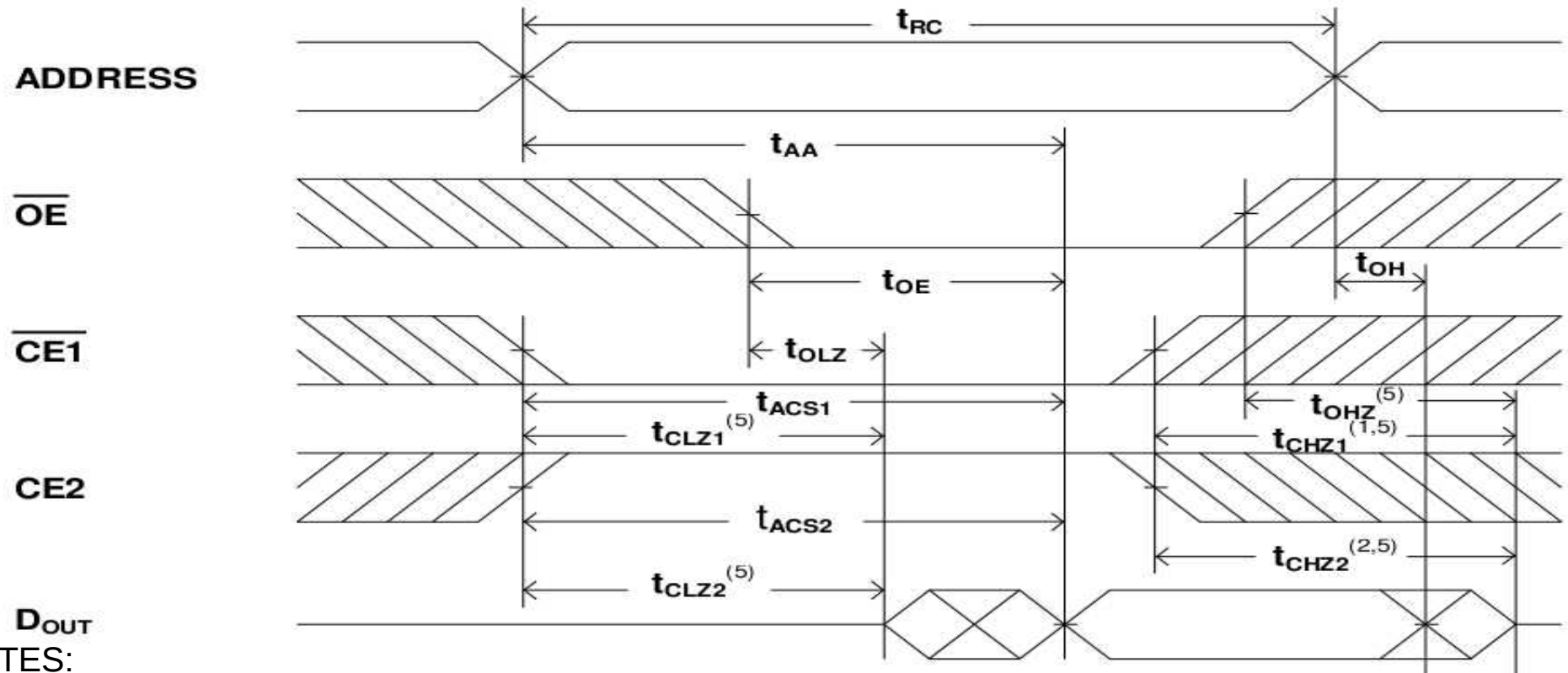


### NOTES:

1. WE is high in read Cycle.
3. Address valid prior to or coincident with CE1 transition low and/or CE2 transition high.
4. OE = VIL.
5. Transition is measured  $\pm 500\text{mV}$  from steady state with  $\text{CL} = 5\text{pF}$ .  
The parameter is guaranteed but not 100% tested.

# Read cycle (3)

## READ CYCLE 3 <sup>(1, 4)</sup>

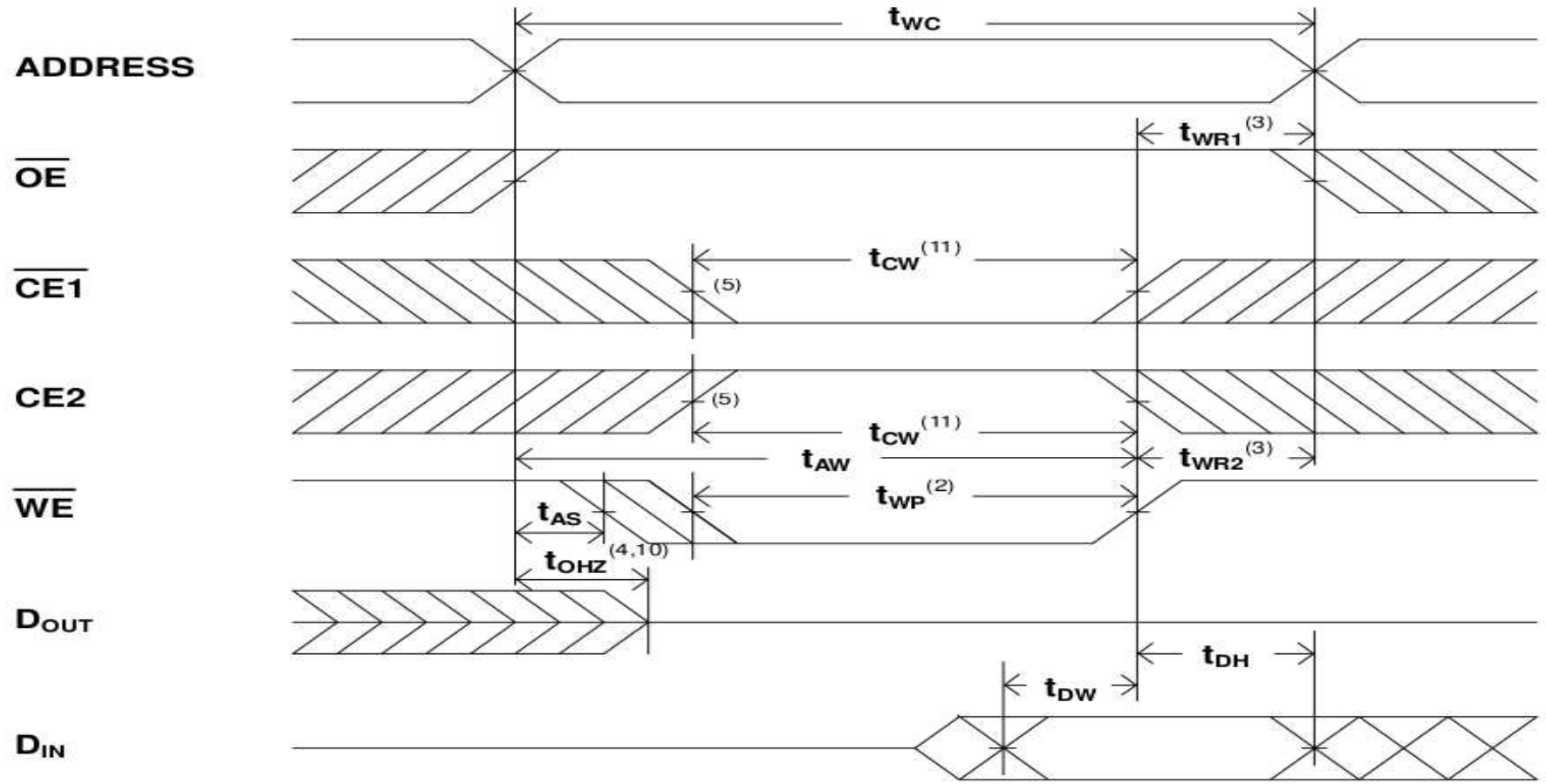


### NOTES:

1. WE is high in read Cycle.
2. Device is continuously selected when CE1 = VIL and CE2= VIH.
4. OE = VIL.
5. Transition is measured  $\pm 500\text{mV}$  from steady state with  $CL = 5\text{pF}$ .  
The parameter is guaranteed but not 100% tested.

# Write cycle (1)

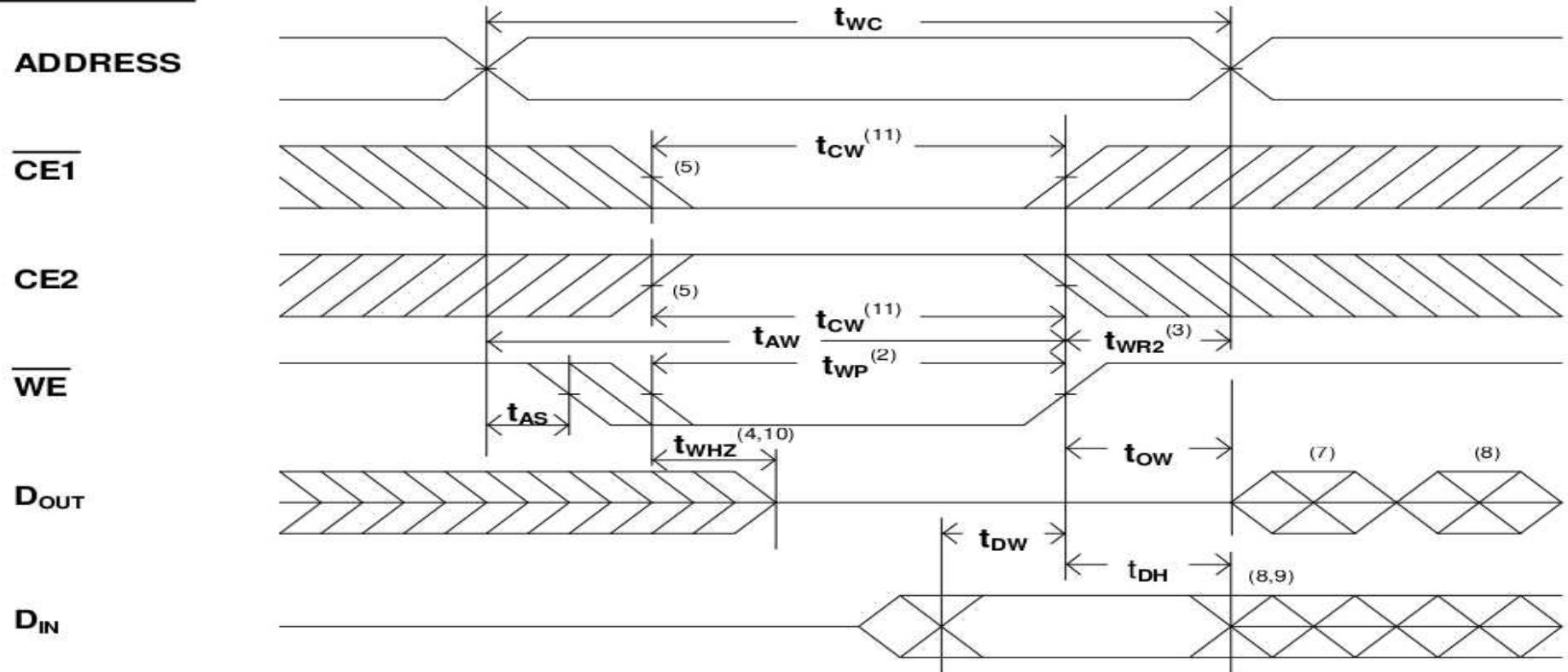
## WRITE CYCLE 1 <sup>(1)</sup>



NOTES:  
see later

# Write cycle (2)

**WRITE CYCLE 2** <sup>(1,6)</sup>



NOTES:  
see later

# Write cycle (notes)

## NOTES:

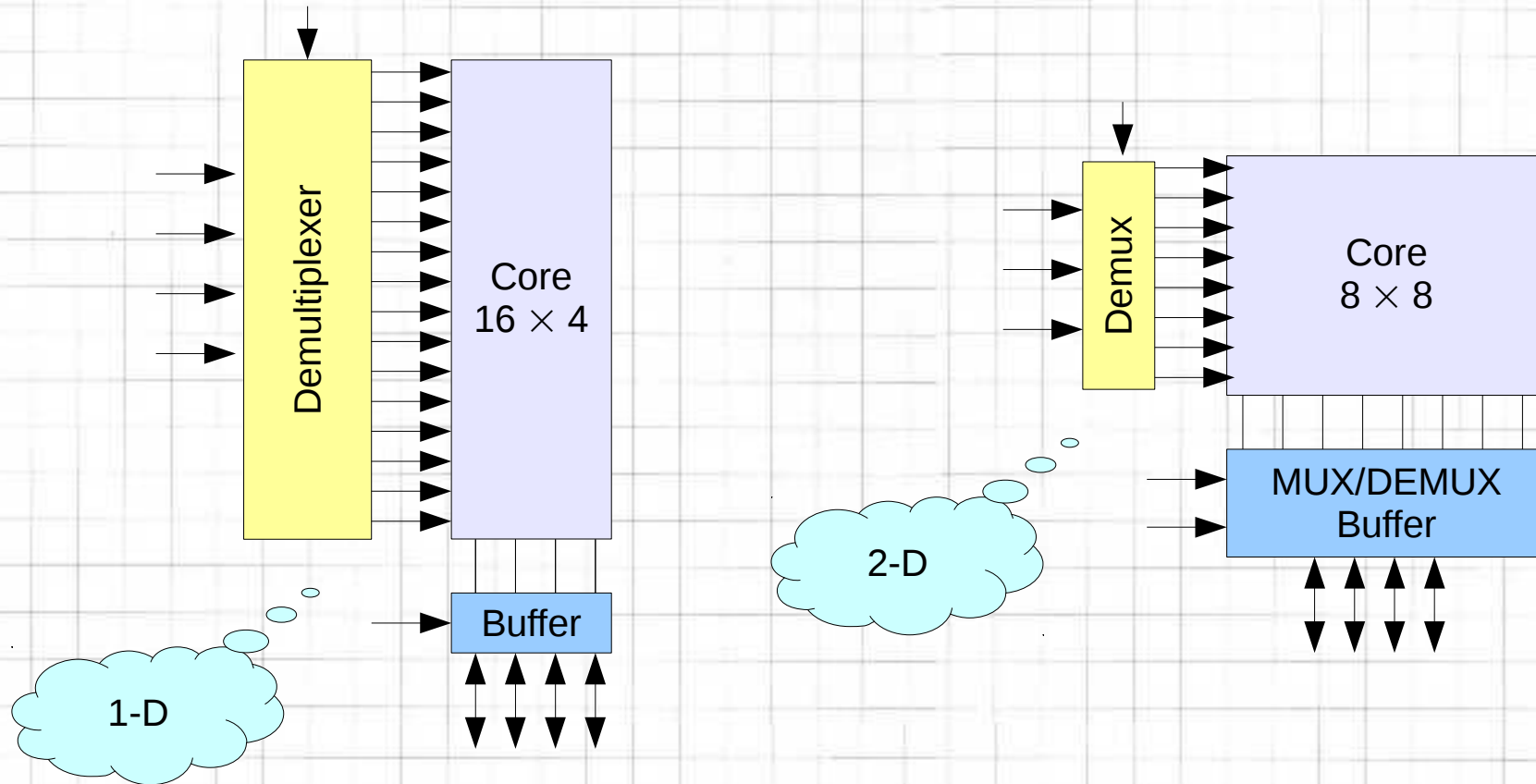
1.  $\overline{WE}$  must be high during address transitions.
2. The internal write time of the memory is defined by the overlap of  $\overline{CE1}$  and CE2 active and  $\overline{WE}$  low. All signals must be active to initiate a write and any one signal can terminate a write by going inactive. The data input setup and hold timing should be referenced to the second transition edge of the signal that terminates the write.
3.  $t_{WR}$  is measured from the earlier of  $\overline{CE1}$  or  $\overline{WE}$  going high or CE2 going low at the end of write cycle.
4. During this period, DQ pins are in the output state so that the input signals of opposite phase to the outputs must not be applied.
5. If the  $\overline{CE1}$  low transition or the CE2 high transition occurs simultaneously with the  $\overline{WE}$  low transitions or after the  $\overline{WE}$  transition, output remain in a high impedance state.
6.  $\overline{OE}$  is continuously low ( $\overline{OE} = V_{IL}$ ).
7. DOUT is the same phase of write data of this write cycle.
8.  $\overline{DOUT}$  is the read data of next address.
9. If  $\overline{CE1}$  is low and CE2 is high during this period, DQ pins are in the output state. Then the data input signals of opposite phase to the outputs must not be applied to them.
10. Transition is measured  $\pm 500\text{mV}$  from steady state with  $CL = 5\text{pF}$ . The parameter is guaranteed but not 100% tested.
11.  $t_{CW}$  is measured from the later of  $\overline{CE1}$  going low or CE2 going high to the end of write.

# Architettura di una SRAM

- Nucleo centrale
  - Array di celle basate sull'**elemento bistabile**
  - Accesso alla singola cella
    - Abilitazione di riga (e colonna)
- Interfaccia
  - **Decoder** con abilitazione
    - Per l'abilitazione di riga (e colonna)
  - **Multiplexer/demultiplexer** distribuito
    - Multiplexer per la generazione dell'uscita
    - Demultiplexer per la selezione del dato da scrivere
- Gestione della comunicazione bidirezionale
  - **Buffer** a tre stati

# Indirizzamento 1-D o 2-D

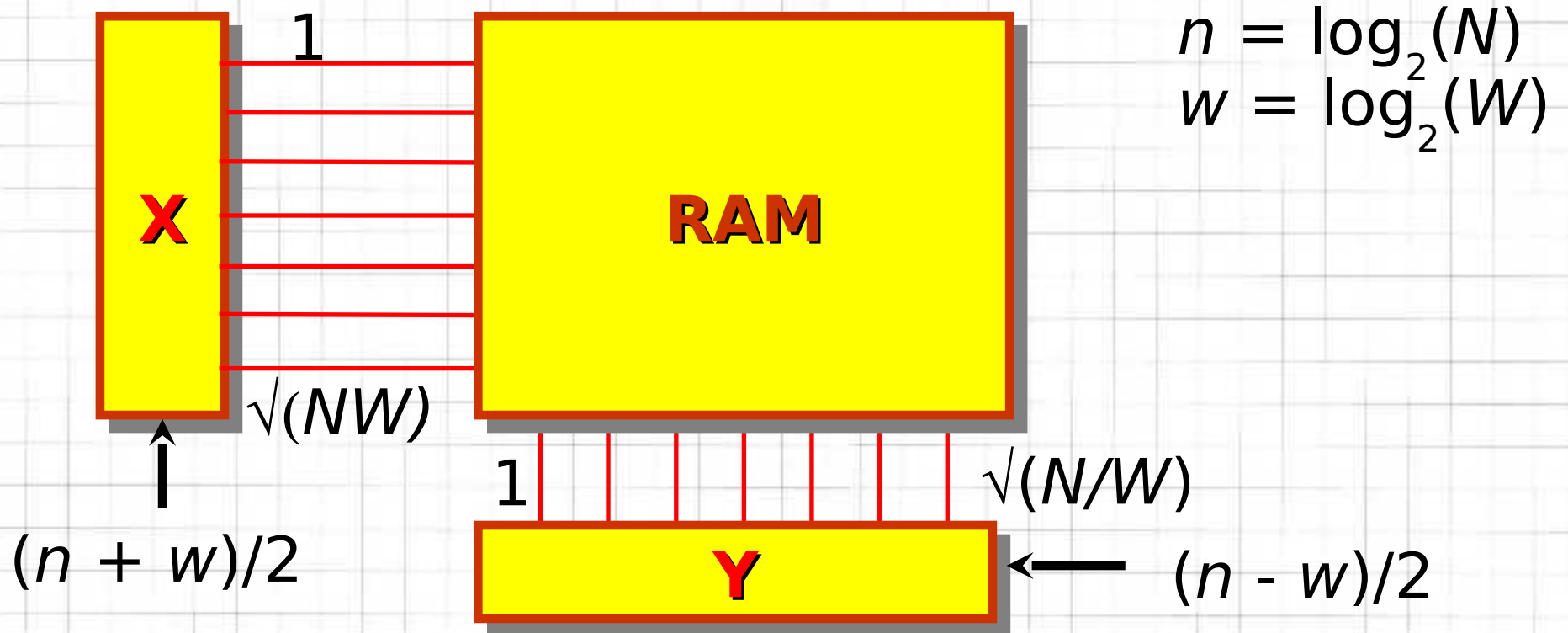
Esempio: memoria da  $16 \times 4$





# Indirizzamento 2-D

Celle di memoria indirizzata 2-D

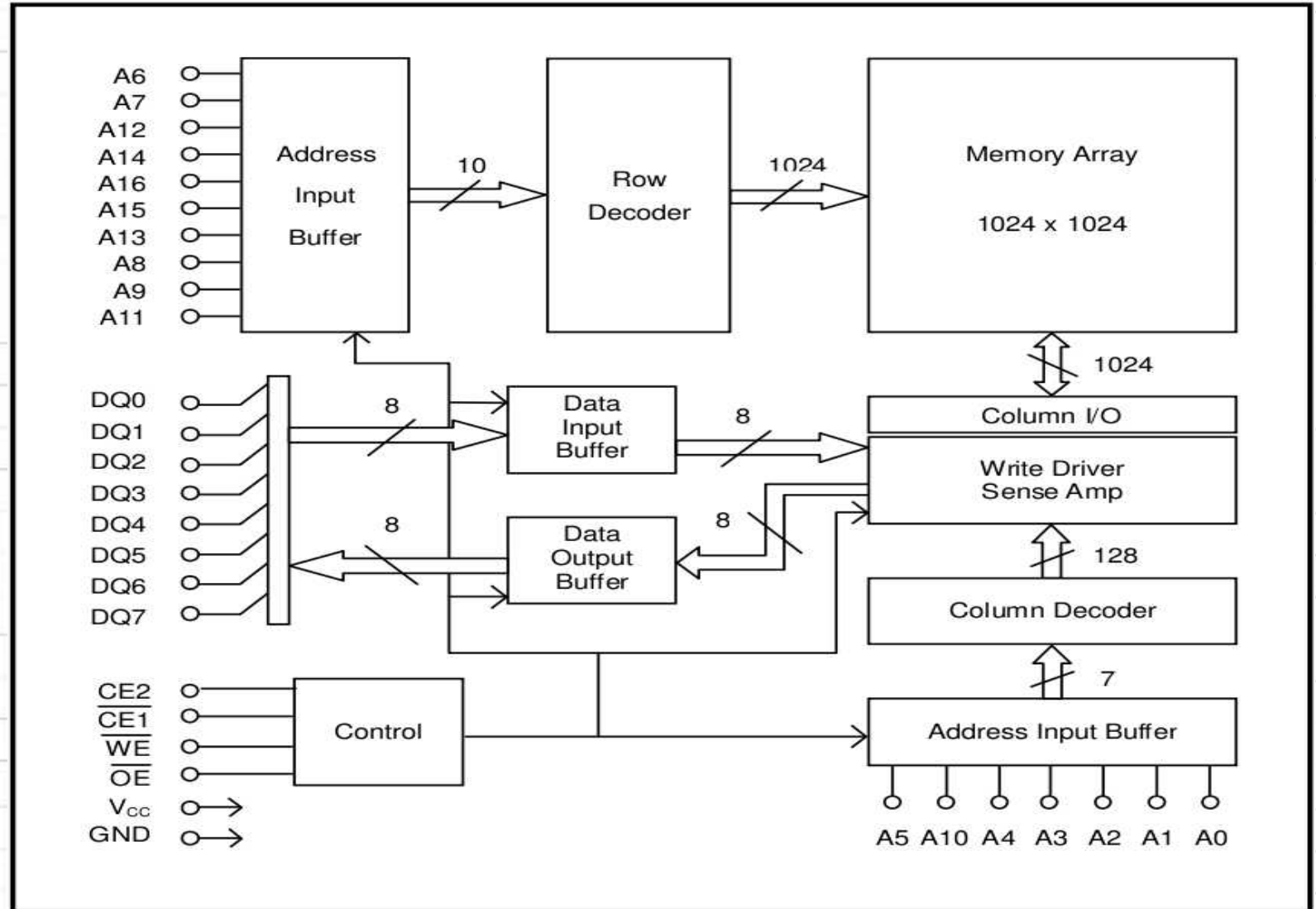


$$n2^n \gg (n + w) 2^{(n+w)/2} + (n - w) 2^{(n-w)/2}$$

# Osservazioni

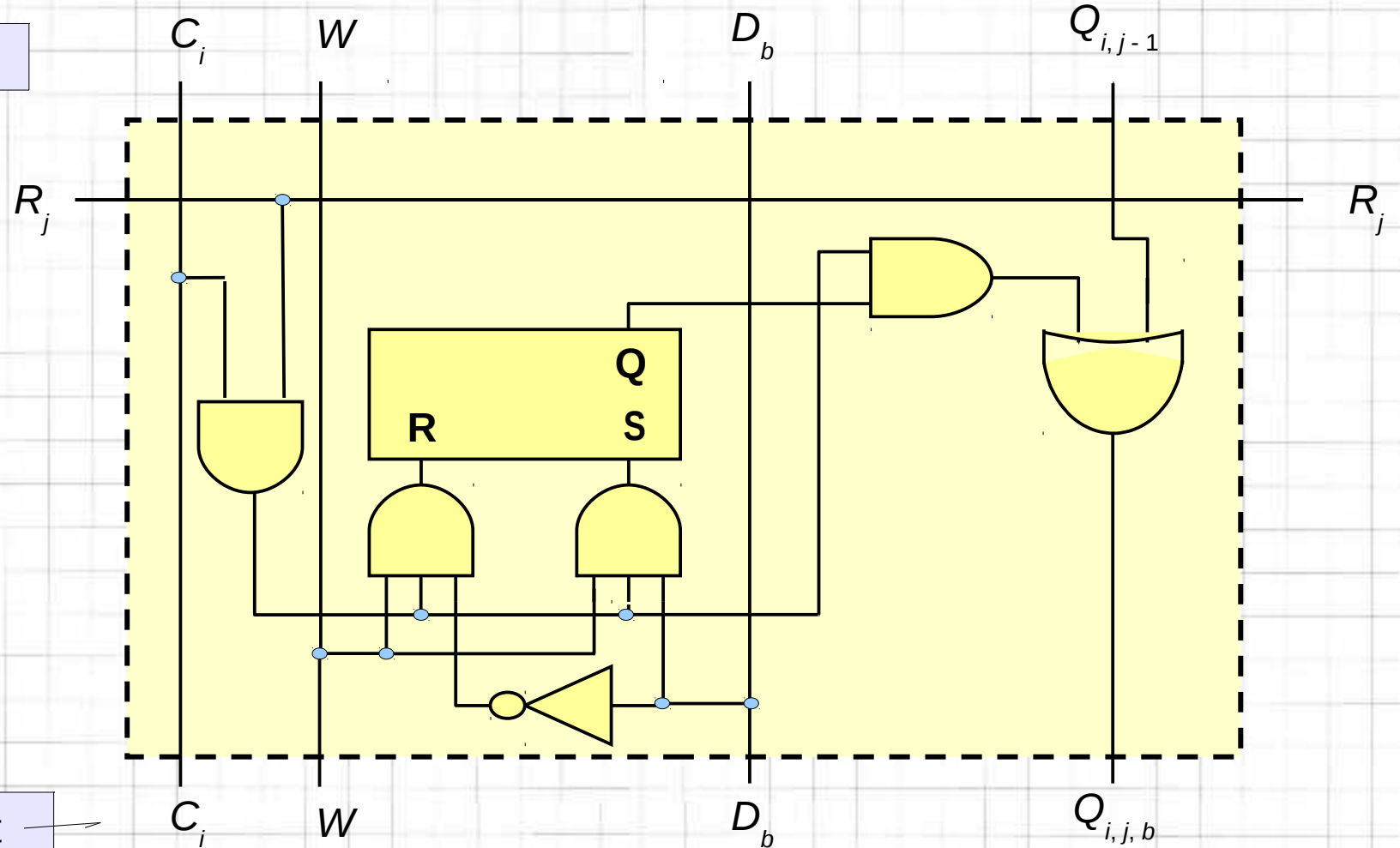
- Vantaggi dell'indirizzamento 2-D
  - Architettura con **fattore di forma unitario**
    - Conviene realizzare chip di forma quadrata
  - **Ridotta complessità** della decodifica
    - Due demux da  $\sqrt{Nk}$  e  $\sqrt{N/k}$  uscite contro uno da  $N$
- Esempio: memoria da 1 M ( $128 \text{ k} \times 8 \text{ b}$ )
  - Confronto tra complessità dei decoder
    - Caso 1-D: 2,228,224 letterali
    - Caso 2-D:  $10240 + 896 = 11,136$  letterali
  - Per contro, la cella elementare sarà leggermente più complessa

# Schema a blocchi di SRAM



# Cella di SRAM

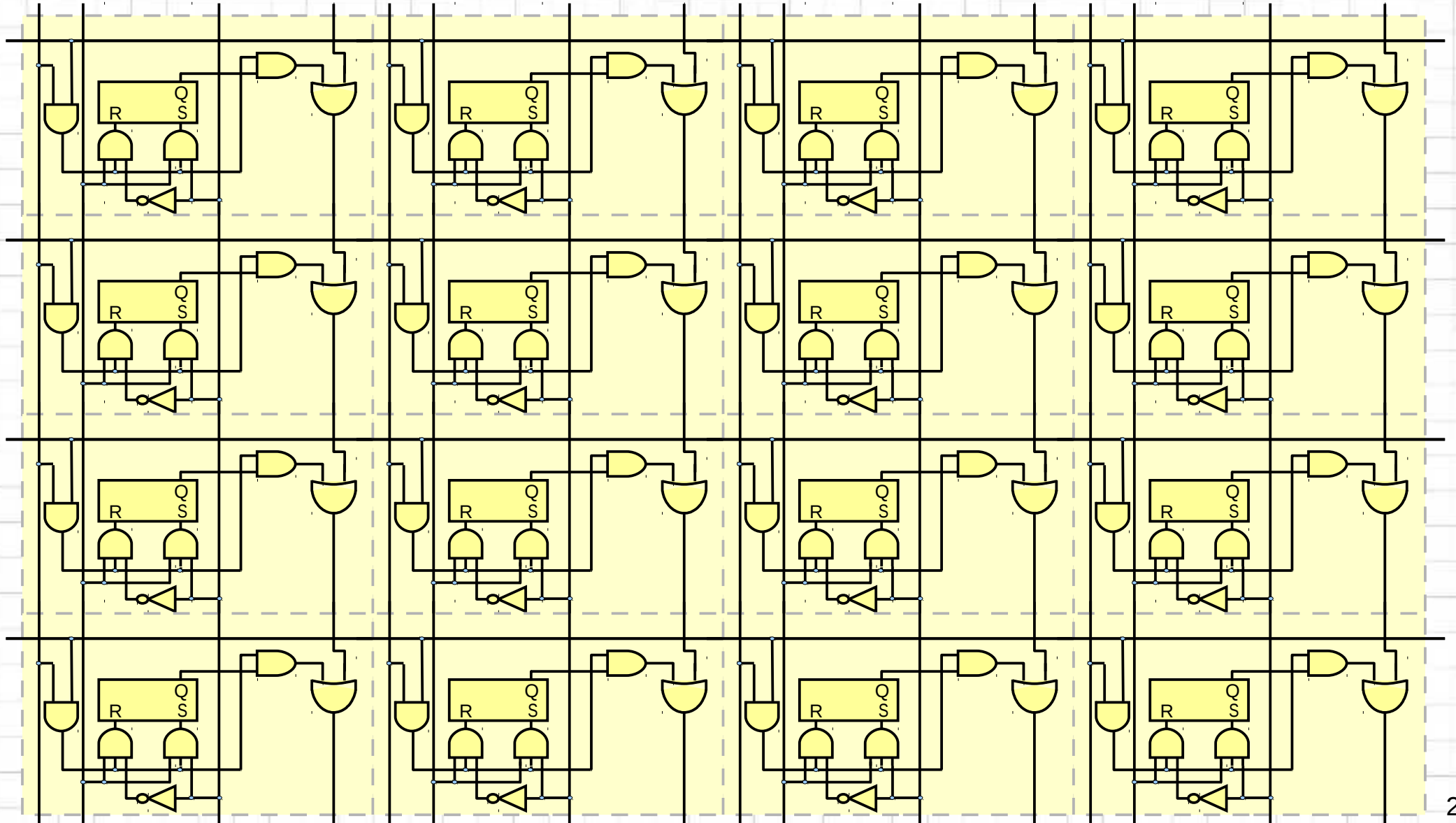
Row select



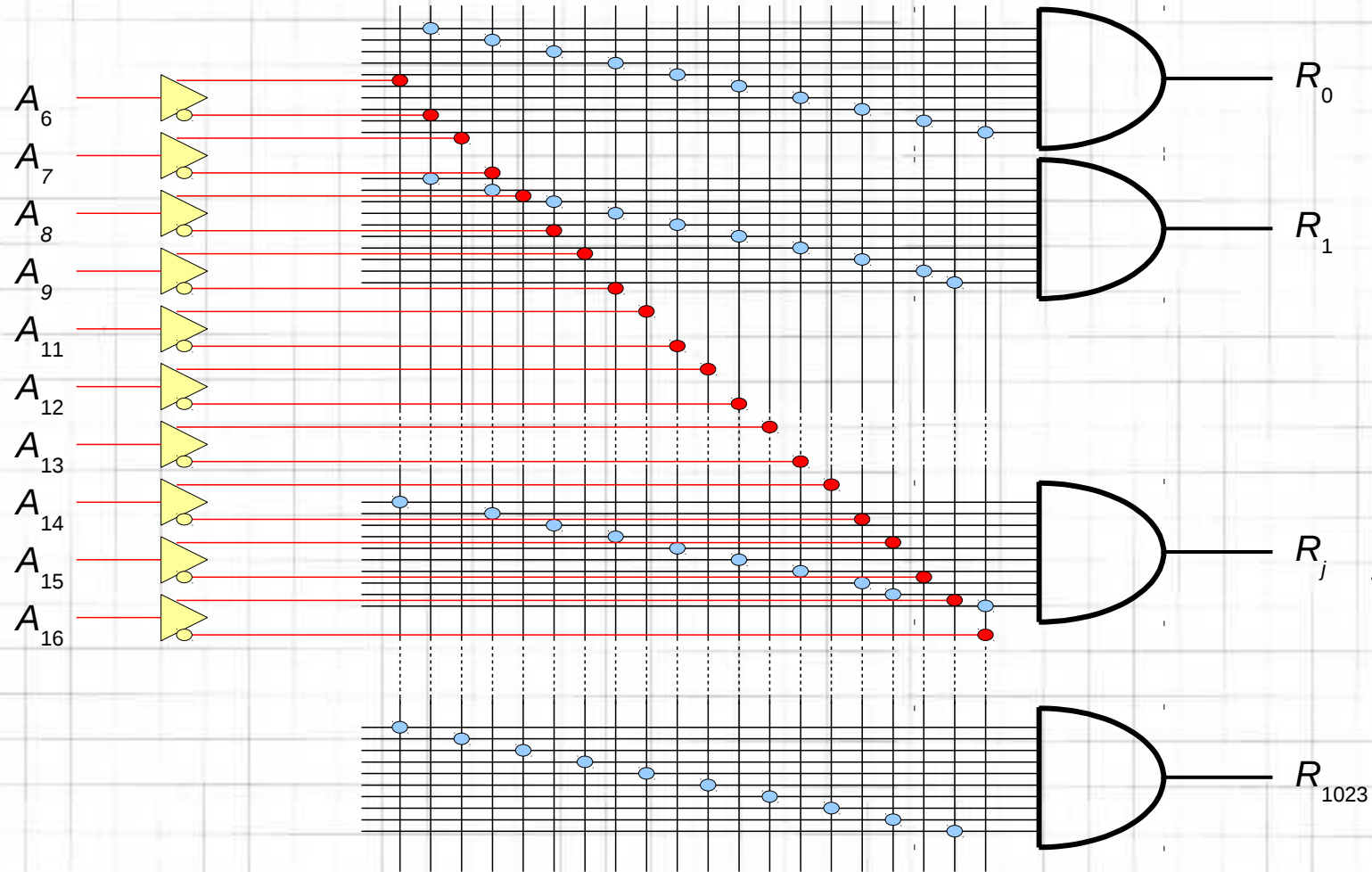
Column select

$b$ : peso del bit nella parola

# Matrici di memoria

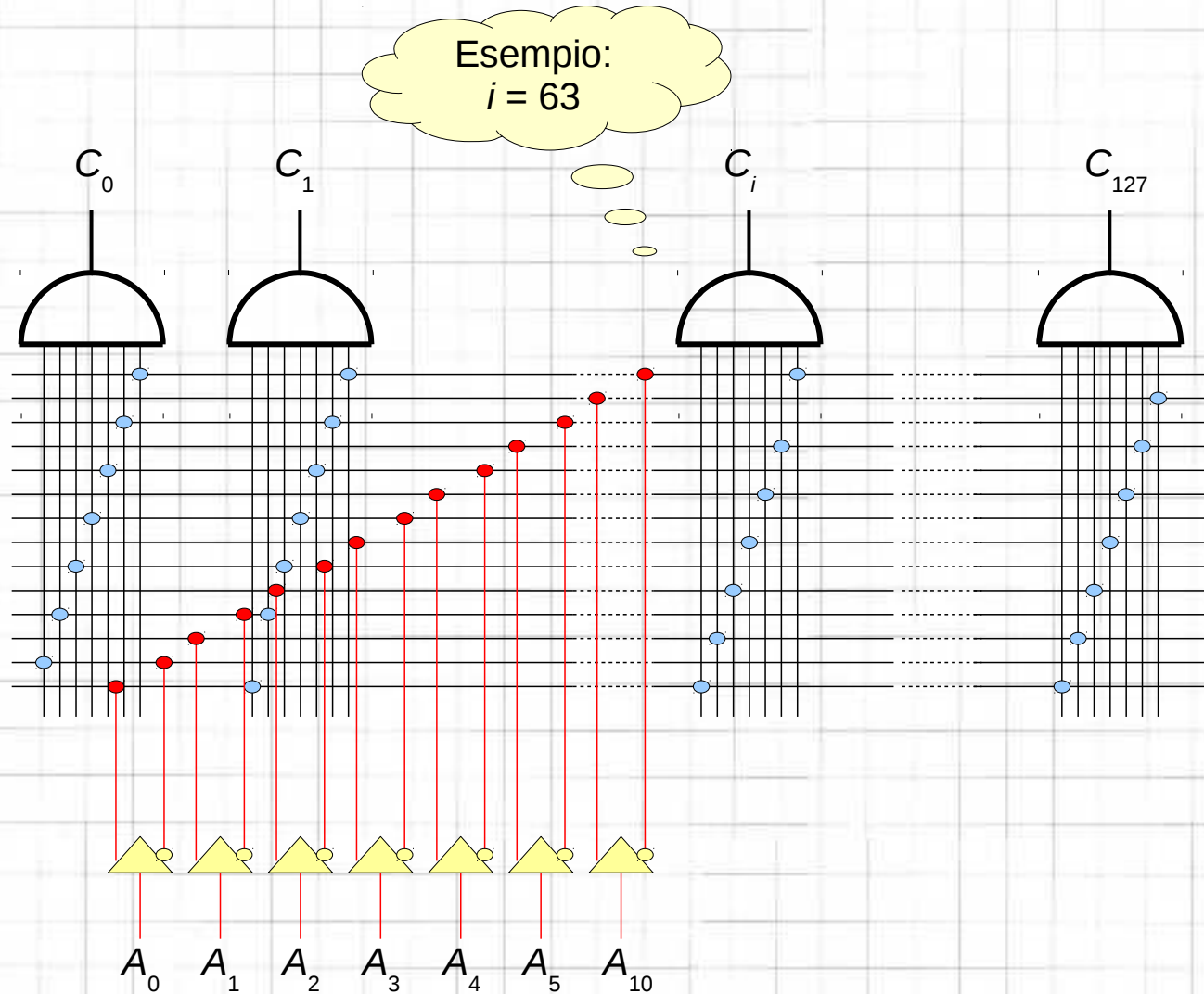


# Decoder di riga

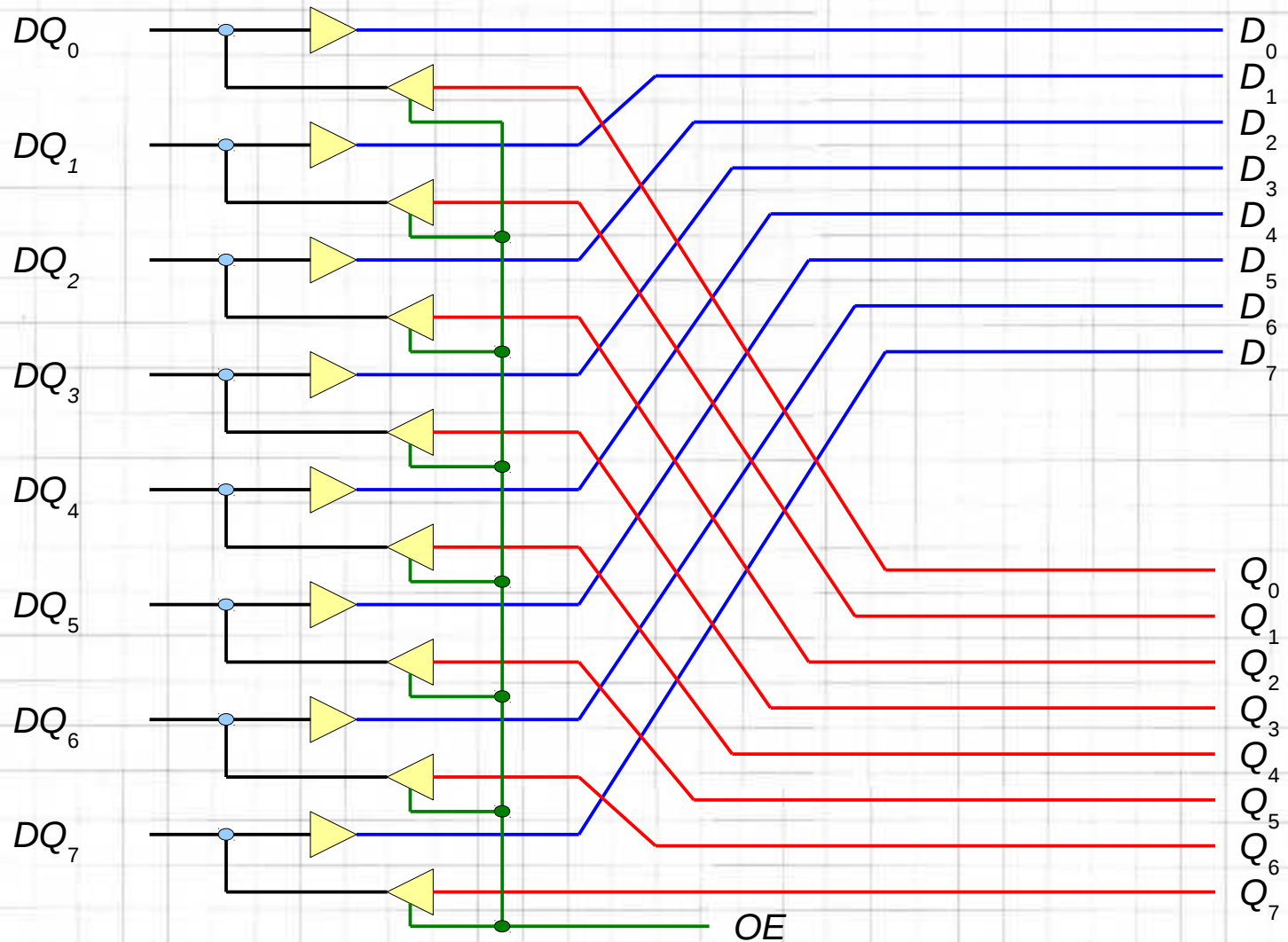


Esempio:  
 $j = 514$

# Decoder di colonna



# Buffer dei dati

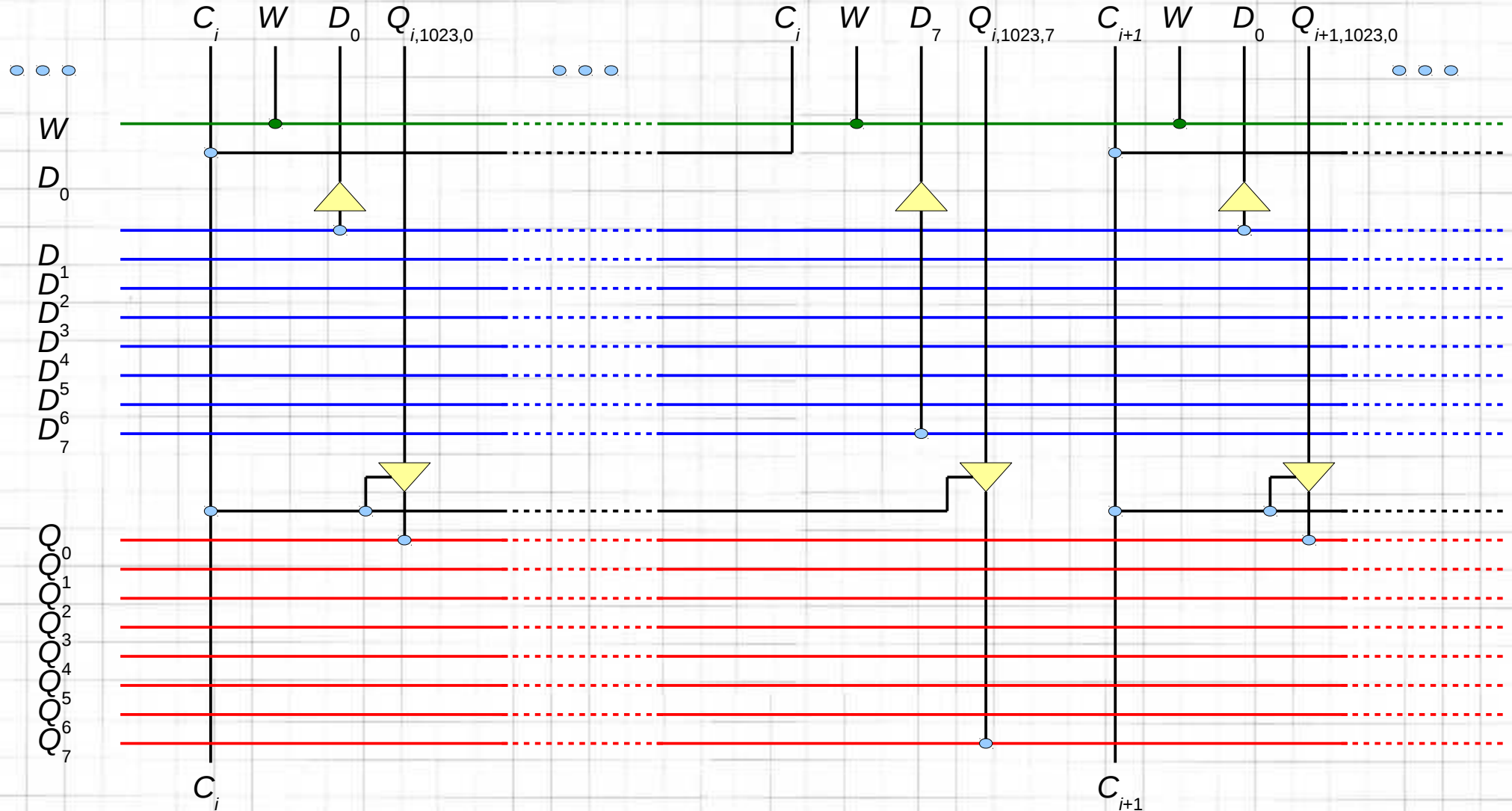


Al  
"write driver"

Dai  
"sense amp"



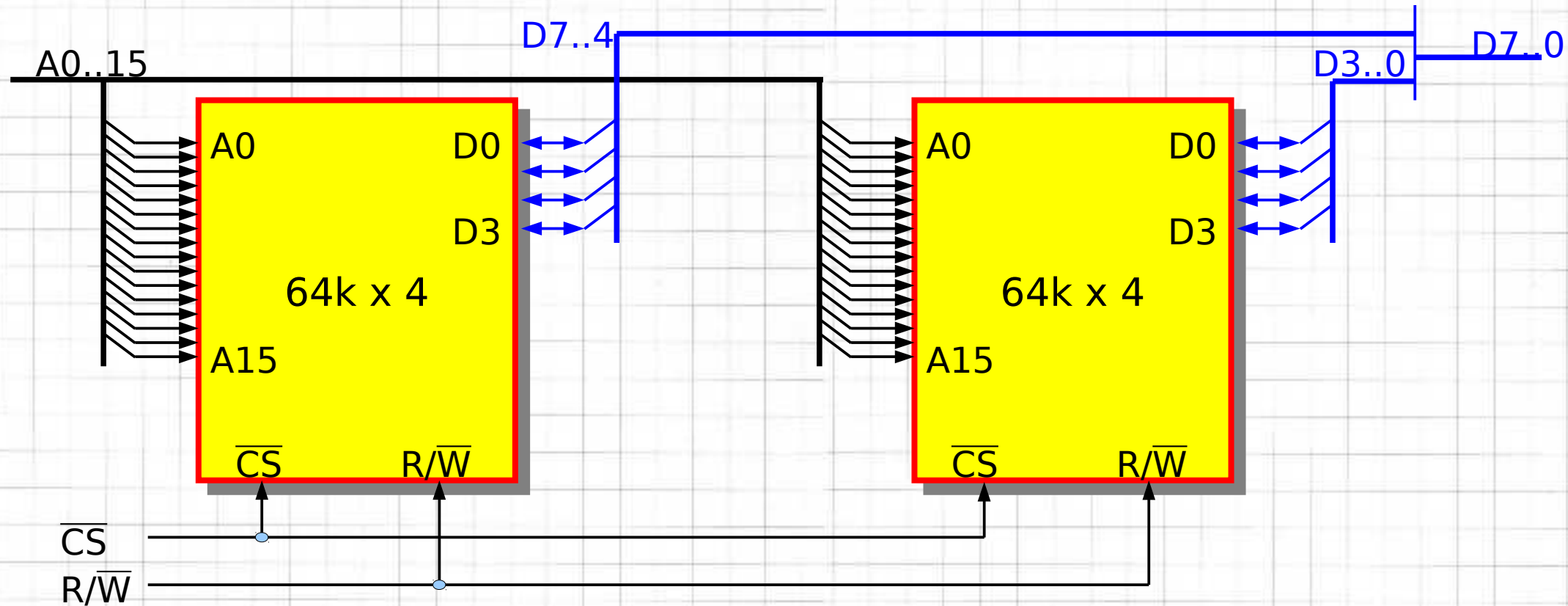
# Write driver, sense amplifier



# Realizzazione di moduli

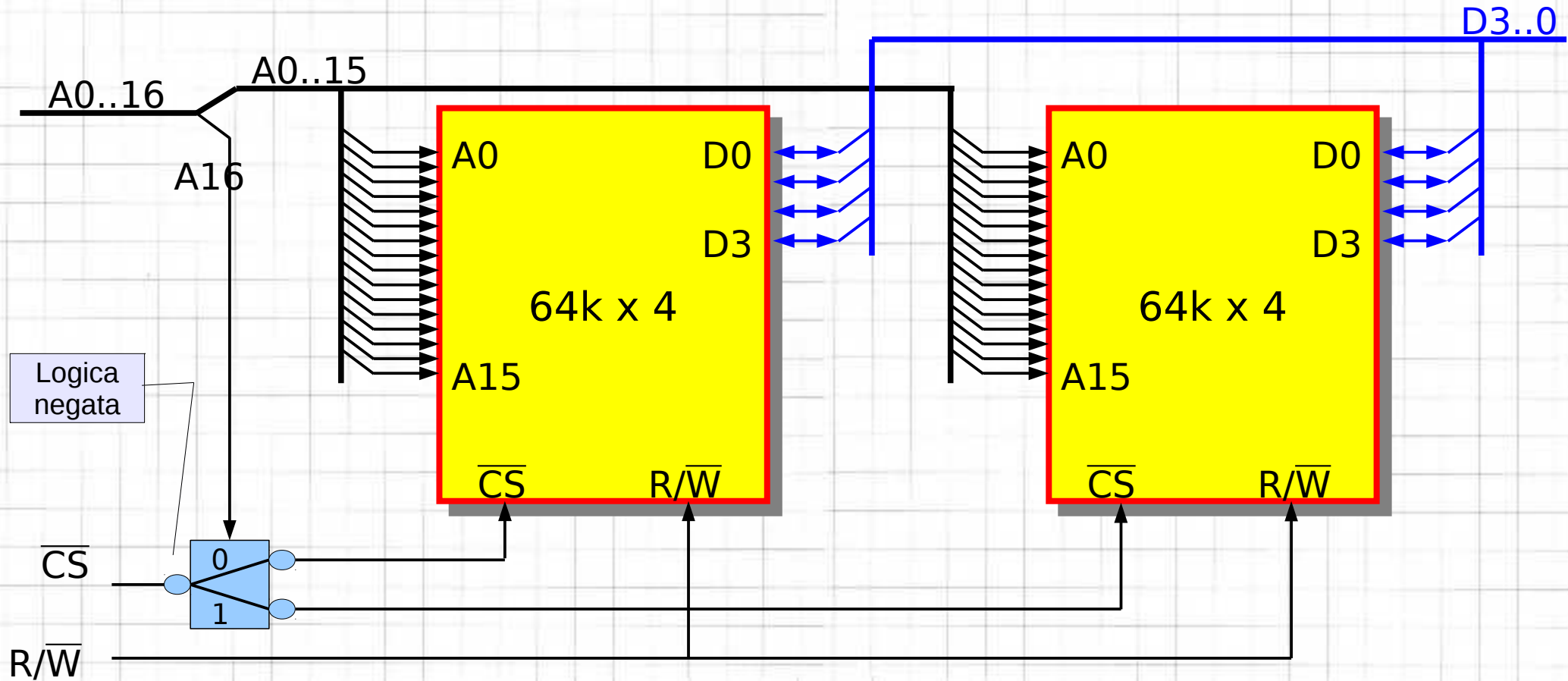
- Usare più SRAM per ottenerne una più grande
  - Per aumentare la dimensione di parola  $W$
  - Per aumentare il numero di parole  $N$
- **Aumento di  $W$** 
  - Si usano memorie di uguale numero di parole  $N$
  - Si controllano in parallelo (indirizzi,  $\overline{CS}$ ,  $\overline{R/W}$ )
  - Si affiancano le linee dei dati
- **Aumento di  $N$** 
  - Si usano memorie con la stessa dimensione  $W$
  - Si pilotano i  $\overline{CS}$  per abilitare una SRAM alla volta
    - Viene usato un demultiplexer pilotato dalle linee di indirizzo più significative

# Raddoppiare i bit di parola $W$



Da 2 memorie 64k x 4 → 1 memoria 64k x 8

# Raddoppiare le parole $N$



Da 2 memorie 64k x 4  $\rightarrow$  1 memoria 128k x 4