

# Le memorie ad accesso casuale

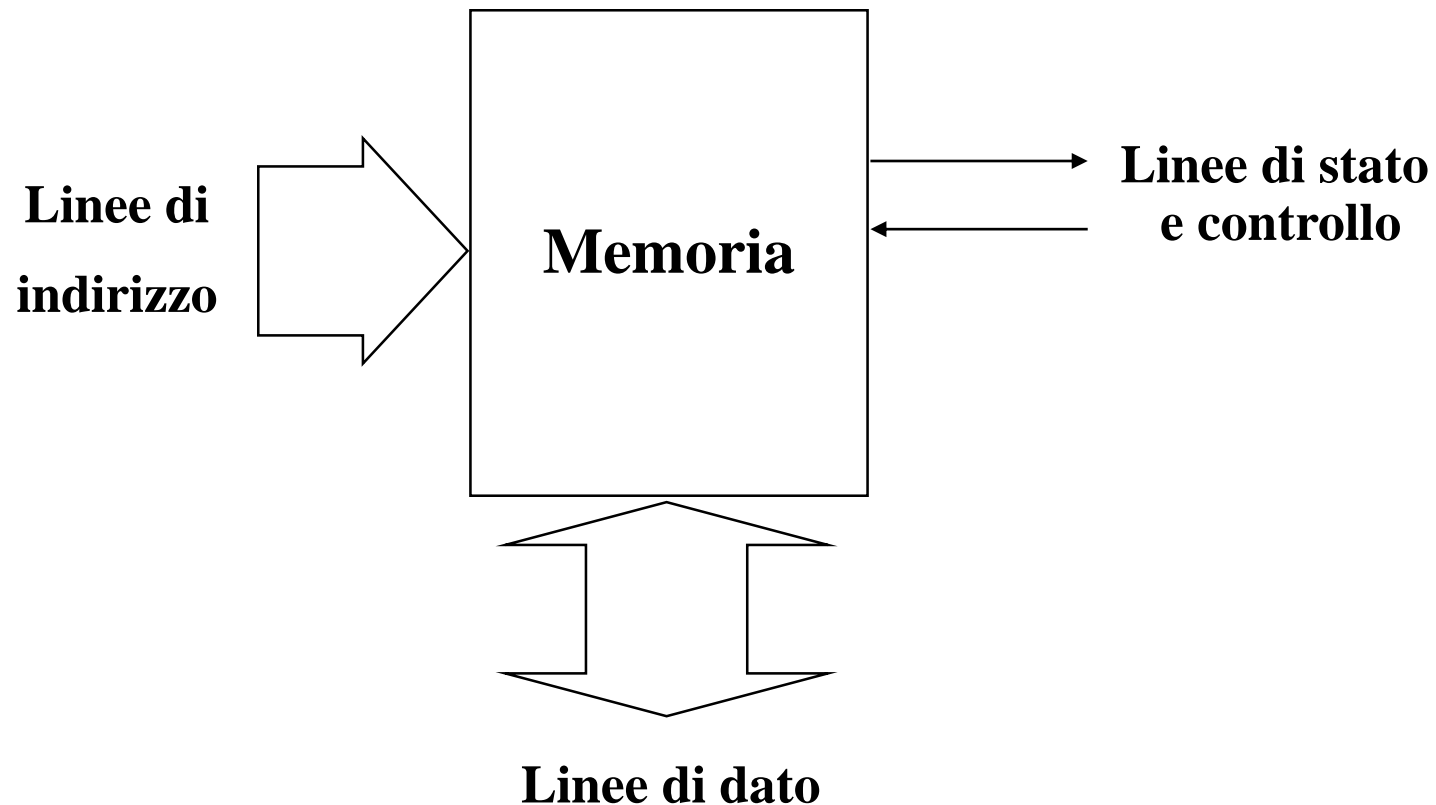
Matteo Sonza Reorda

Politecnico di Torino  
Dip. di Automatica e Informatica



# Caratteristiche generali

- Ogni cella può essere indirizzata indipendentemente
- I tempi di accesso sono uguali e costanti per ogni cella.



# Funzionamento

## **Lettura:**

- **Sull'Abus si scrive l'indirizzo della cella da leggere**
- **Sul Cbus si attivano i segnali di lettura**
- **Sul Dbus la memoria scrive il contenuto della cella.**

## **Scrittura:**

- **Sull'Abus si mette l'indirizzo della cella da leggere**
- **Sul Cbus si attivano i segnali di scrittura**
- **Sul Dbus si scrive il valore da memorizzare nella cella.**

# Segnali di controllo

- **Permettono alla memoria di sapere**
  - **il tipo di operazione richiesta (lettura o scrittura)**
  - **quando gli indirizzi sono disponibili sull'Abus**
  - **quando i dati sono disponibili sul Dbus (solo per la scrittura).**
- **Deve inoltre esistere un meccanismo con il quale è possibile sapere**
  - **quando i dati sono disponibili sul Dbus (solo per la lettura)**
  - **quando è possibile procedere con un nuovo ciclo di accesso alla memoria.**

# Segnali di cor

Soluzione sincrona: la memoria e chi la utilizza condividono un segnale di clock (o informazioni di tempo) e quindi sanno quando possono procedere con ciascuna operazione

- Permettono alla memoria di sapere
  - il tipo di operazione richiesta (lettura o scrittura)
  - quando gli indirizzi sono disponibili sull'Abus
  - quando i dati sono disponibili sul Dbus (solo per la scrittura).
- Deve inoltre esistere un meccanismo con il quale è possibile sapere
  - quando i dati sono disponibili sul Dbus (solo per la lettura)
  - quando è possibile procedere con un nuovo ciclo di accesso alla memoria.

# Segnali di controllo

**Soluzione asincrona:** la memoria e chi la utilizza non condividono nessun segnale di clock (o informazioni di tempo) e quindi necessitano di segnali di controllo che dicono quando è possibile procedere con ciascuna operazione

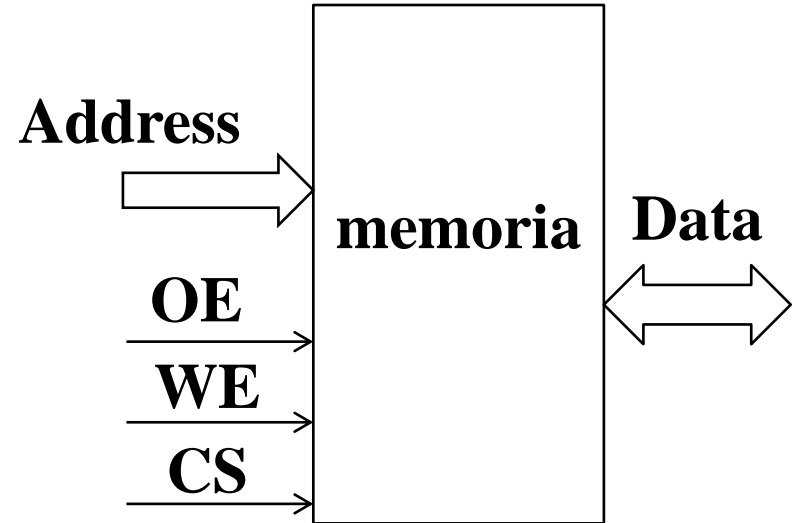
- Permettono alla memoria di sapere
  - il tipo di operazione richiesta (lettura o scrittura)
  - quando gli indirizzi sono disponibili sull'Abus
  - quando i dati sono disponibili sul Dbus (solo per la scrittura).
- Deve inoltre esistere un meccanismo con il quale è possibile sapere
  - quando i dati sono disponibili sul Dbus (solo per la lettura)
  - quando è possibile procedere con un nuovo ciclo di accesso alla memoria.

# Esempio di segnali di controllo

- **Chip Select (CS):**
  - da attivare per poter leggere o scrivere
- **Output enable (OE)**
  - da attivare per poter abilitare l'uscita del chip su un bus condiviso
  - serve a poter collegare molti chip di memoria ad un singolo bus
- **Write enable (WE)**
  - da attivare per poter effettuare una operazione di scrittura.

# Esempio

- **Per scrivere:**
  - **CS attivo**
  - **Indirizzo sulle linee di Address**
  - **Dato in input sulle linee di Data**
  - **WE attivo**
- **Per leggere:**
  - **CS attivo**
  - **Indirizzo sulle linee di Address**
  - **OE attivo**
  - **Dato in output sulle linee di Data**



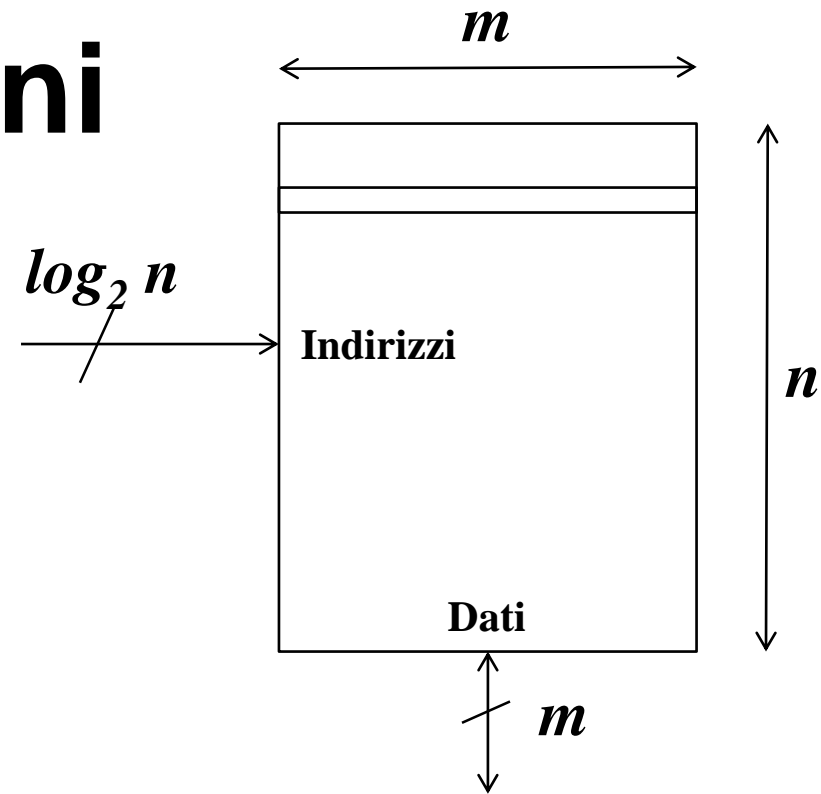


# Segnali di stato

## Esempi:

- **Errore**
  - **La memoria ha individuato una parola con un dato errato**
- **MFC**
  - **La memoria ha completato l'operazione di lettura/scrittura.**

# Dimensioni



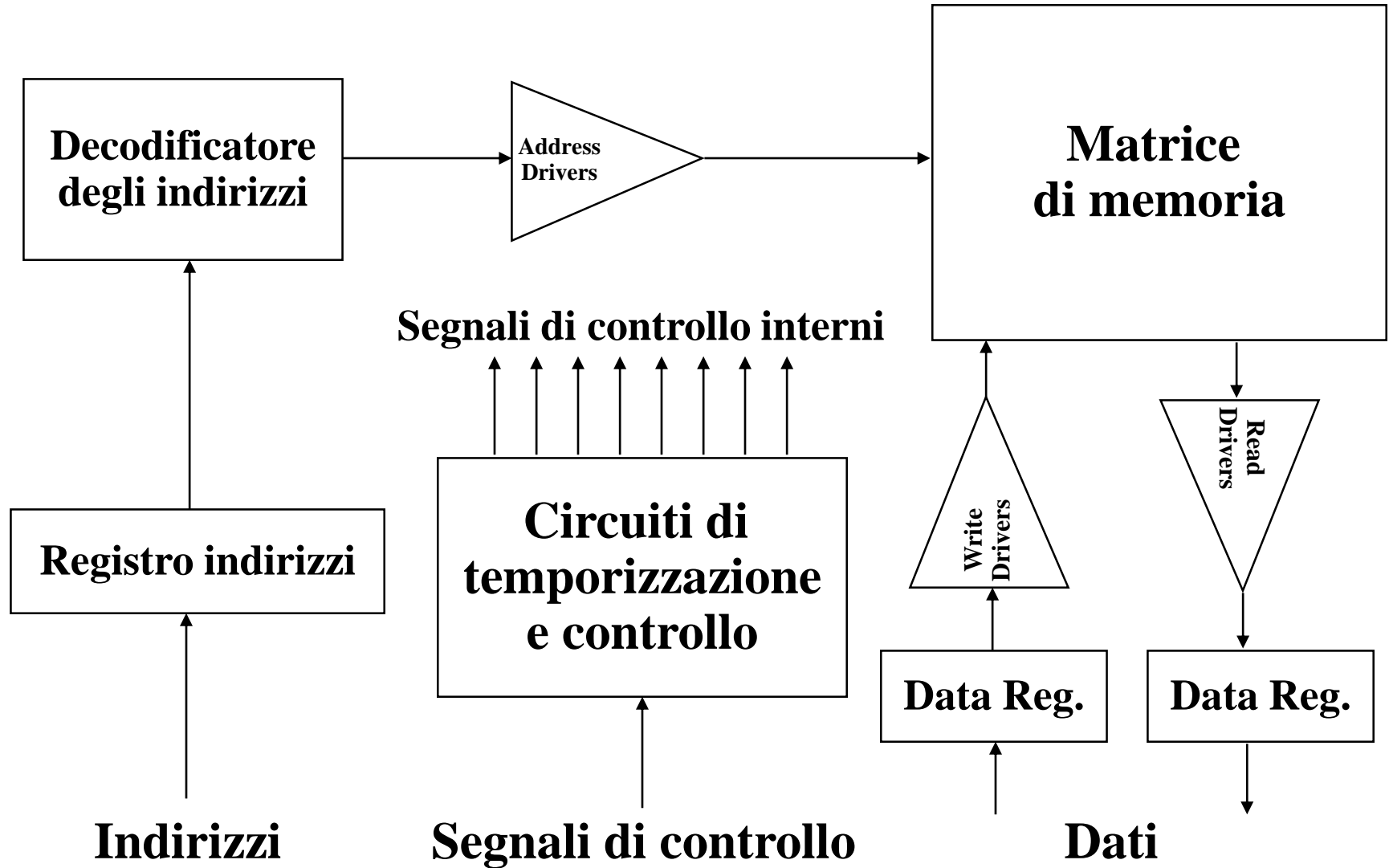
Sono caratterizzate da

- numero di parole ( $n$ )
- numero di bit per parola ( $m$ ).

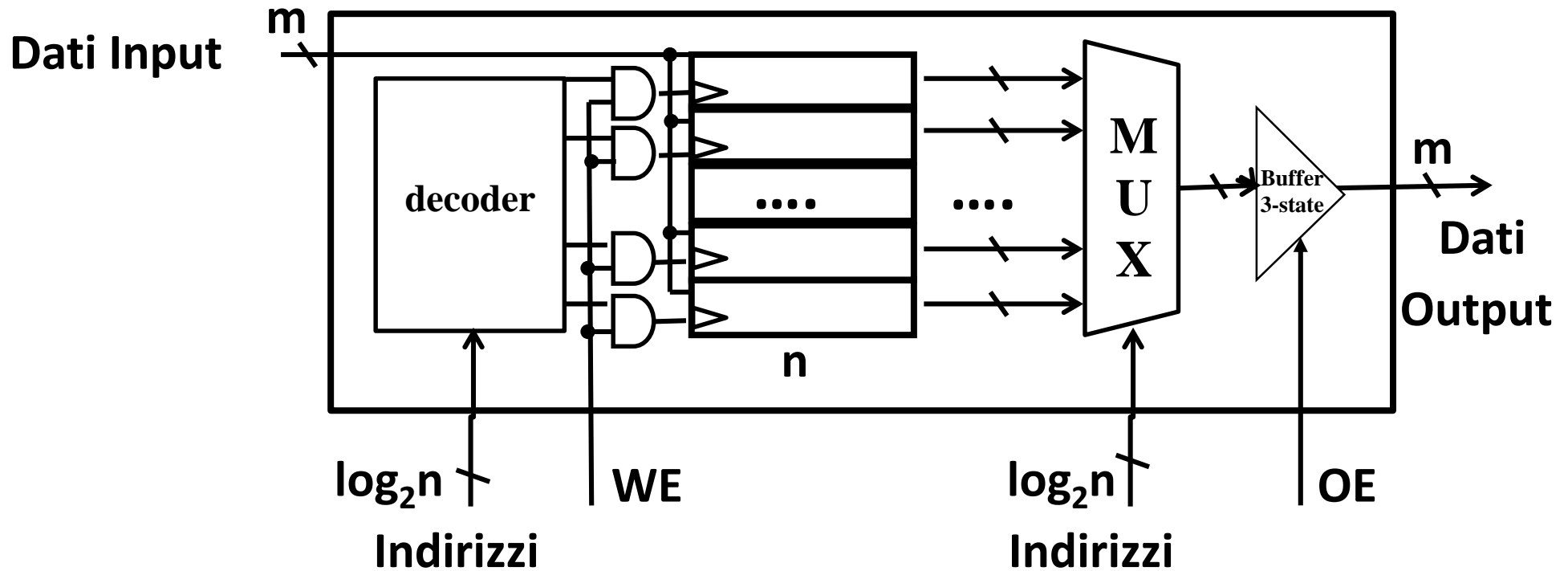
Il numero di segnali di ingresso/uscita è conseguentemente

- $\log_2 n$  per i segnali di indirizzo
- $m$  per i segnali di dato.

# Architettura



# Schema generale



# Organizzazione

**Il costo di una RAM dipende anche dalla complessità dei dispositivi di accesso.**

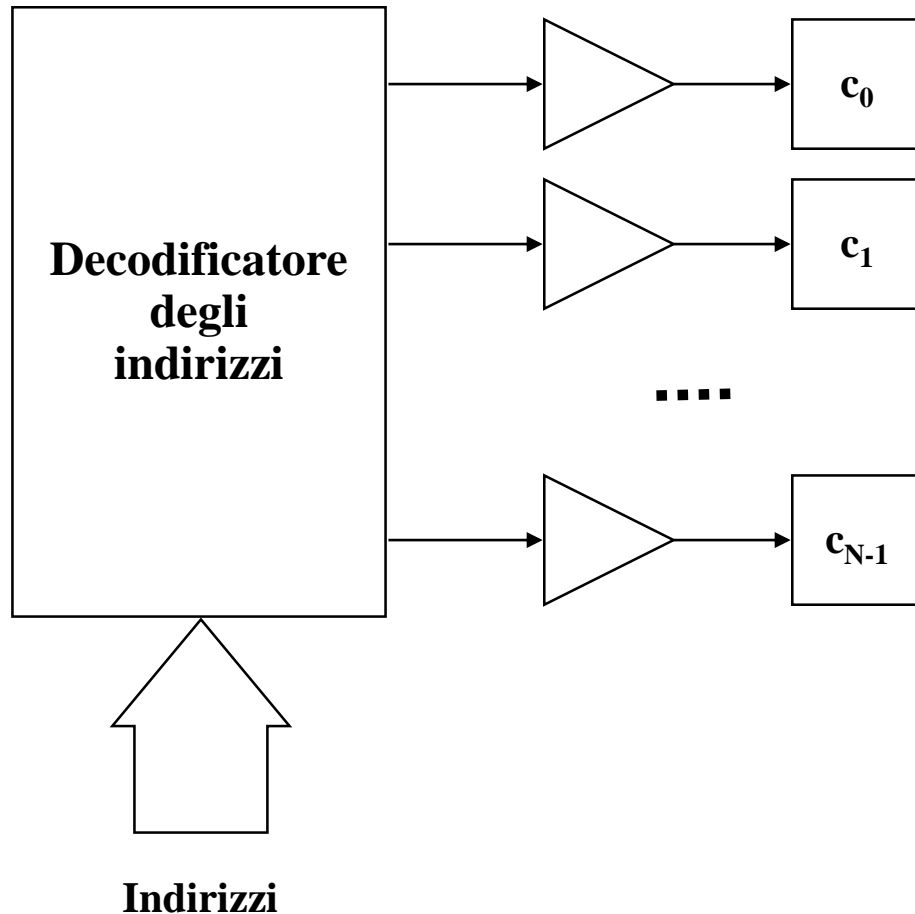
**Questa può essere ridotta tramite un'opportuna organizzazione delle celle di memoria.**

**Si hanno due tipologie principali:**

- **organizzazione a *vettore***
- **organizzazione a *matrice bidimensionale*.**

**La regolarità nell'organizzazione delle celle di memoria influenza pesantemente anche il costo del layout.**

# Organizzazione a vettore



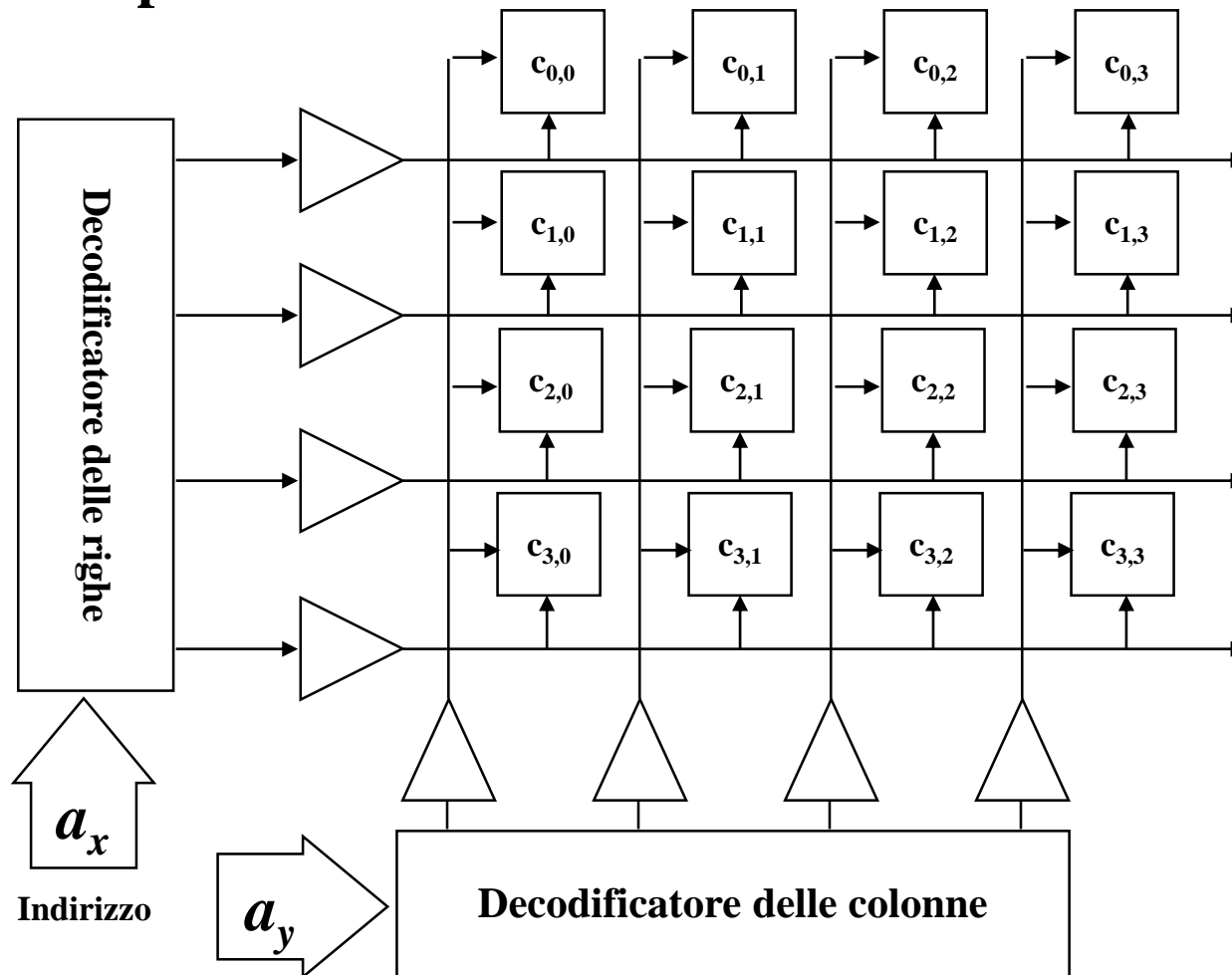
**Costo della circuiteria di accesso:**

- 1 decoder  $\log_2 N$ -to- $N$
- $N$  driver.

**Celle di memoria:**  
1 cella = 1 parola

# Organizzazione a matrice

L'indirizzo  $a$  è suddiviso in 2 parti  $a_x$  e  $a_y$ , che selezionano la riga e la colonna in cui si trova la cella di memoria, rispettivamente.



**Costo della  
circuiteria di  
accesso:**

- 2 decoder  
( $\log_2 \sqrt{N}$ )-to- $\sqrt{N}$
- $2\sqrt{N}$  driver.

# **Organizzazione a matrice: vantaggi**

- **Minor costo dell'HW**
- **Layout più compatto.**



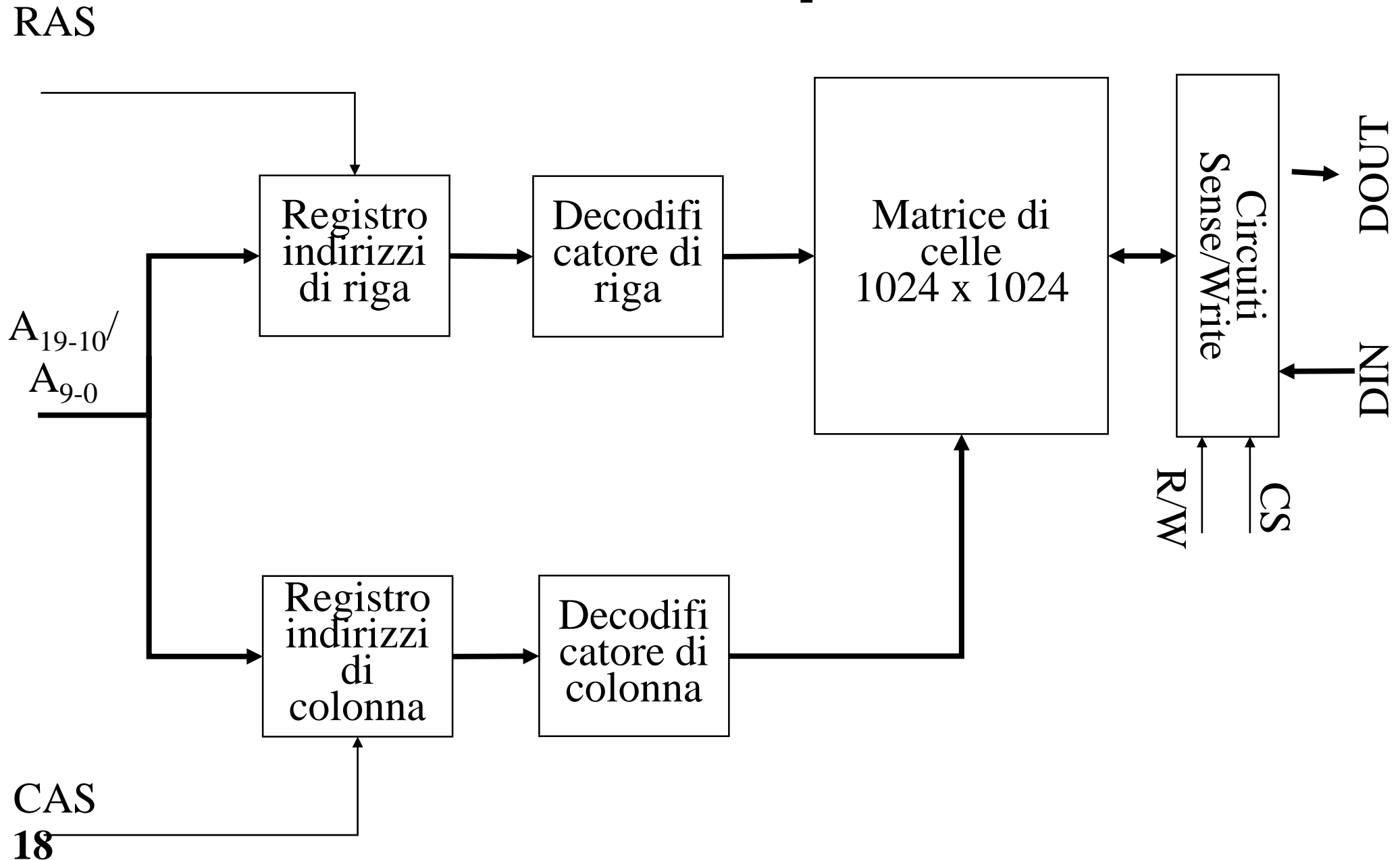
# Segnali RAS e CAS

**Per ridurre il numero di segnali di ingresso, le RAM organizzate a matrice talvolta prevedono un diverso protocollo di accesso e un numero ridotto di segnali di indirizzo (indicativamente la metà).**

**In tal caso l'indirizzo è fornito in due fasi distinte e successive, utilizzando gli stessi segnali di indirizzo:**

- in una prima fase vengono forniti i segnali di indirizzo che vanno al decodificatore di riga, accompagnati dal segnale RAS (*Row Address Strobe*)**
- in un'altra fase vengono forniti i segnali di indirizzo che vanno al decodificatore di colonna, accompagnati dal segnale CAS (*Column Address Strobe*).**

# Esempio



# Page Mode

**Qualora sia necessario accedere consecutivamente a celle poste ad indirizzi successivi, e queste risiedano nella stessa riga della matrice, è possibile attivare il cosiddetto *Page Mode*:**

- 1 si invia alla memoria la parte di indirizzo relativo alla riga, accompagnata da RAS**
- 2 si invia alla memoria la parte di indirizzo relativo alla colonna, accompagnata da CAS**
- 3 si accede al dato**
- 4 si ripete dal punto 2.**

**In questa maniera è possibile ridurre i tempi di accesso alla memoria.**

# Memorie a semiconduttore

- **Le memorie sono attualmente i dispositivi a più alta densità realizzati su IC**
- **Al crescere dell'integrazione è possibile**
  - **aumentare la capacità (a parità di dimensione)**
  - **aumentare la velocità**
  - **ridurre il consumo.**

# Classificazione

- **ROM** (*Read Only Memory*)
- **PROM** (*Programmable Read Only Memory*)
- **EPROM** (*Electrically Programmable Read Only Memory*)
- **EEPROM** (*Electrically Erasable Programmable Read Only Memory*)
- **Flash**
- **RAM.**

# ROM

## Applicazioni:

- librerie di procedure frequentemente usate
- programmi di sistema
- tavole di funzioni.

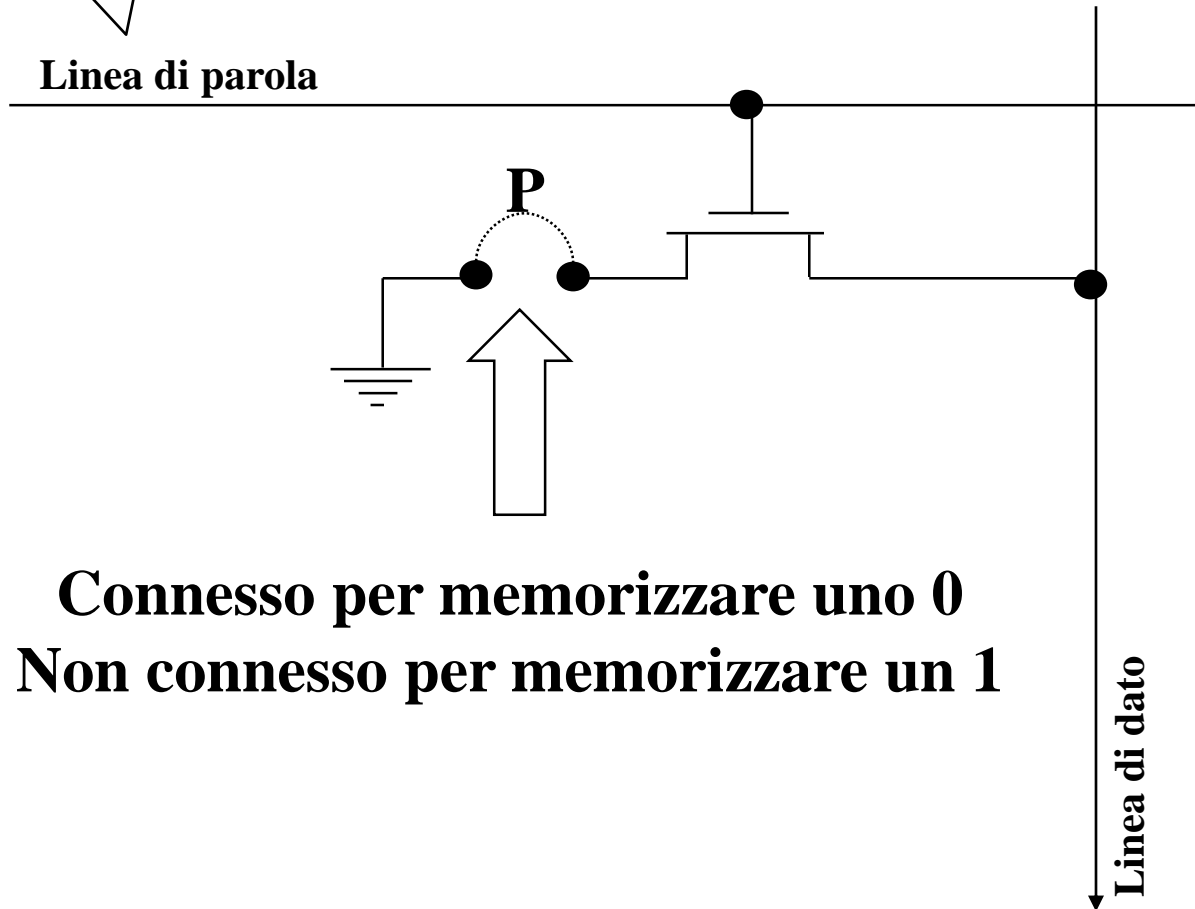
La definizione del contenuto avviene prima della realizzazione del silicio, ed il contenuto non è in alcun modo modificabile in seguito.

Il costo è dominato dal costo fisso per allestire la linea di produzione del dispositivo, ed è quasi indipendente dal numero di chip prodotti.

L'impatto di qualsiasi errore di progetto è drammatico.

La linea di parola è  
pilotata dal  
decodificatore degli  
indirizzi

# Cella ROM



Connesso per memorizzare uno 0  
Non connesso per memorizzare un 1

# PROM

**La scrittura è eseguita a valle del processo di produzione tramite speciali attrezzature denominate *programmatori*.**

**Fisicamente, sono realizzate ponendo nel punto P del disegno precedente un diodo, che può essere *bruciato* durante la programmazione, creando un circuito aperto.**

**La scrittura può avvenire una volta sola.**

**La PROM è non volatile.**

**Sono preferibili alle ROM per bassi volumi.**



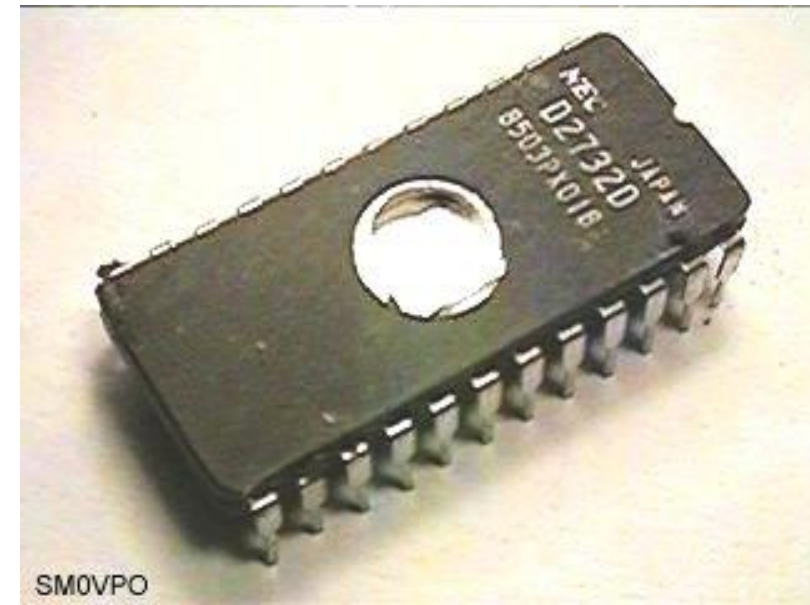
# EPROM

**Possono essere riprogrammate, previa precedente cancellazione tramite esposizione prolungata (~20 min) a luce ultravioletta.**

**La scrittura può avvenire un numero indefinito di volte.**

**La scrittura può avvenire anche dopo il montaggio sulla scheda.**

**Sono più costose delle PROM.**



# EEPROM

**Le EEPROM (o E<sup>2</sup>PROM) possono essere riprogrammate byte per byte anche dopo il montaggio sulla scheda, ma l'operazione richiede più tempo di quella di lettura (centinaia di  $\mu$ s).**

**La scrittura può essere eseguita tramite i normali canali (bus) e segnali.**

**Sono più costose e meno dense delle EPROM.**

# Flash

**Il costo è intermedio tra quello di EPROM ed EEPROM.**

**Usano un solo transistor per bit, e sono quindi relativamente dense.**

**In lettura si comportano come le DRAM, mentre le operazioni di scrittura**

- **sono più lente (di almeno un ordine di grandezza)**
- **vanno eseguite a blocchi**
- **richiedono una precedente operazione di cancellazione.**

**Le Flash sono inoltre in grado di eseguire un numero limitato di cicli di scrittura.**

# Flash (II)

**Esistono due tipi di memorie Flash:**

- **Flash di tipo NOR:** hanno lo stesso comportamento di una RAM, ma sono più lente e resistono per un numero limitato di cancellazioni/scritture
- **Flash di tipo NAND:** possono essere lette e scritte solo a blocchi (da centinaia a migliaia di bit). Sono più dense, più veloci e più resistenti.

# Quadro di sintesi

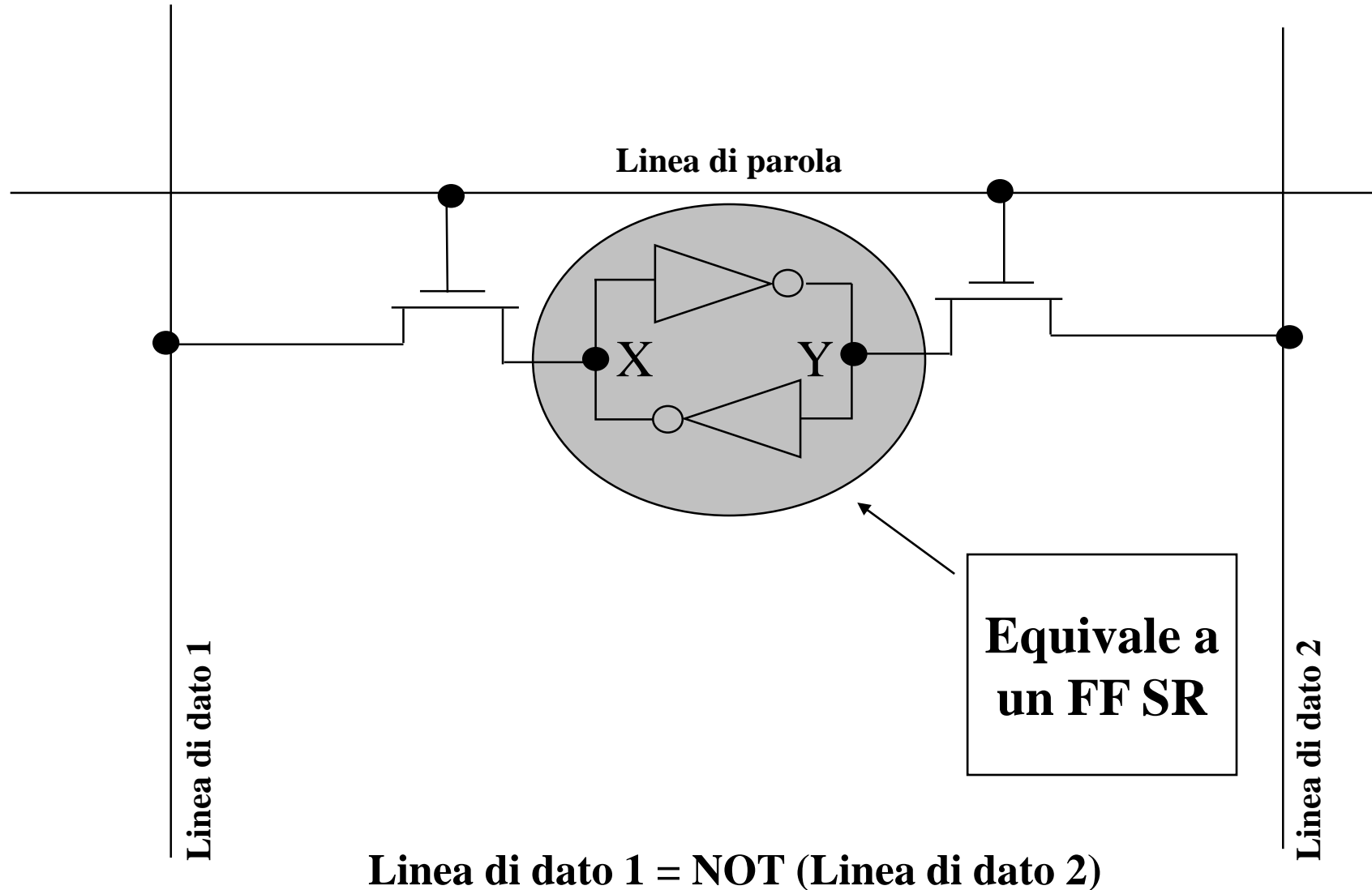
<b>Tipo</b>	<b>Categoria</b>	<b>Cancellazione</b>	<b>Scrittura</b>	<b>Volatilità</b>
<b>RAM</b>	<b>read/write</b>	<b>elettricamente</b>	<b>elettricamente</b>	<b>volatile</b>
<b>ROM</b>	<b>read-only</b>	<b>impossibile</b>	<b>in fase di produz.</b>	<b>non-volatile</b>
<b>PROM</b>	<b>read-only</b>	<b>impossibile</b>	<b>elettricamente</b>	<b>non-volatile</b>
<b>EPROM</b>	<b>read-mostly</b>	<b>luce UV</b>	<b>elettricamente</b>	<b>non-volatile</b>
<b>EEPROM</b>	<b>read-mostly</b>	<b>elettricamente</b>	<b>elettricamente</b>	<b>non-volatile</b>
<b>Flash</b>	<b>read-mostly</b>	<b>elettricamente</b>	<b>elettricamente</b>	<b>non-volatile</b>

# RAM

**Sono di 2 tipi:**

- **memorie *statiche* (o SRAM):**
  - **la singola cella corrisponde a un flip flop**
- **memorie *dinamiche* (o DRAM):**
  - **la singola cella corrisponde a un condensatore e a un transistor**
  - **l'informazione è memorizzata sotto forma di carica del condensatore**
  - **richiedono un rinfresco periodico dell'informazione**
  - **la lettura è di tipo distruttivo (*Destructive Read-Out*).**

# Cella di RAM statica



# Funzionamento

**Per leggere o scrivere una parola si deve attivare la relativa linea di parola.**

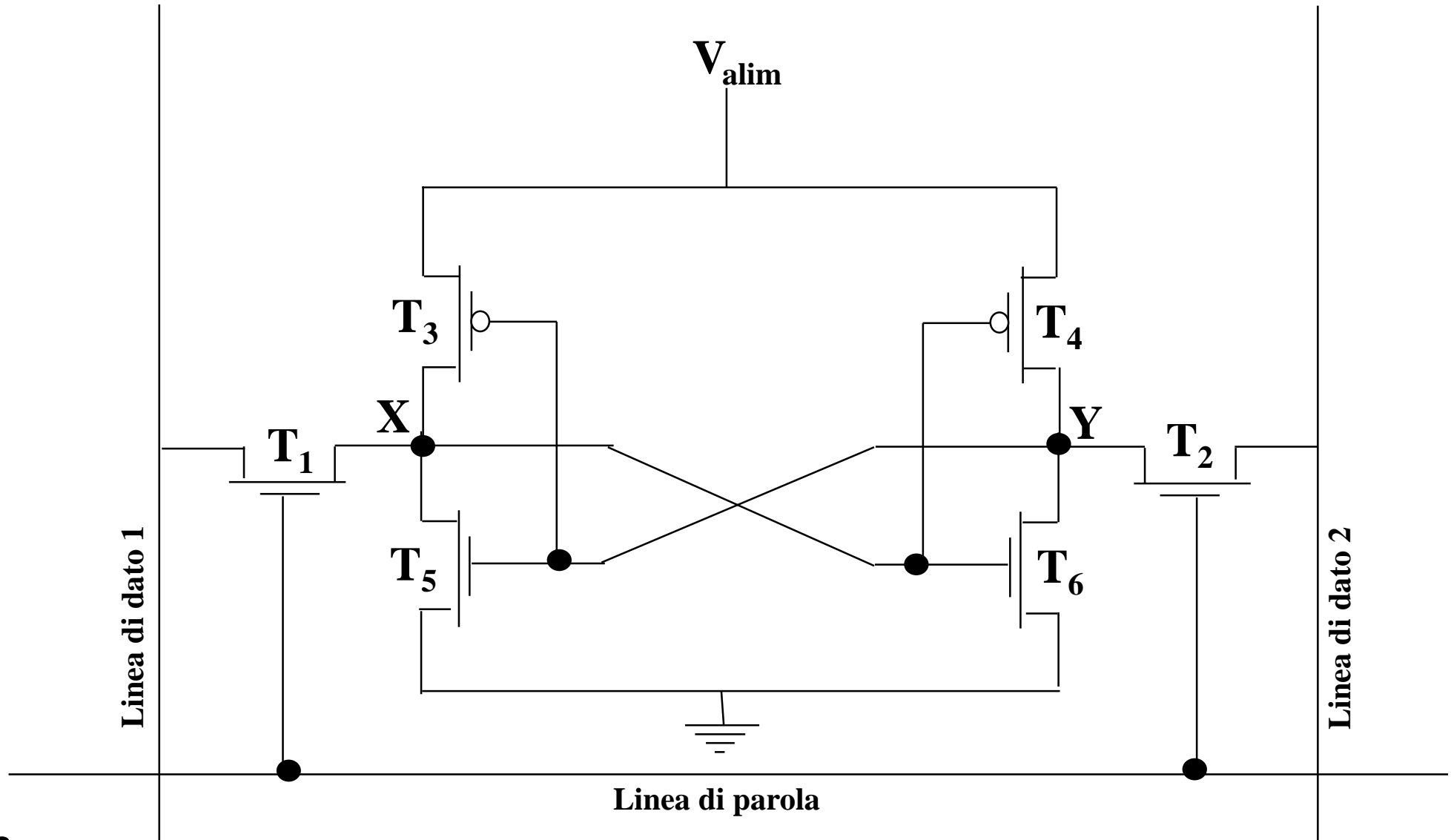
**Quando la linea di parola non è attivata, il relativo flip flop mantiene il proprio valore.**

**Quando la linea di parola è attivata, è possibile**

- **leggere i valori (opposti) forzati dal flip flop sulle due linee di dato (*lettura*)**
- **scrivere un nuovo valore, forzando due valori opposti sulle linee di dato (*scrittura*).**

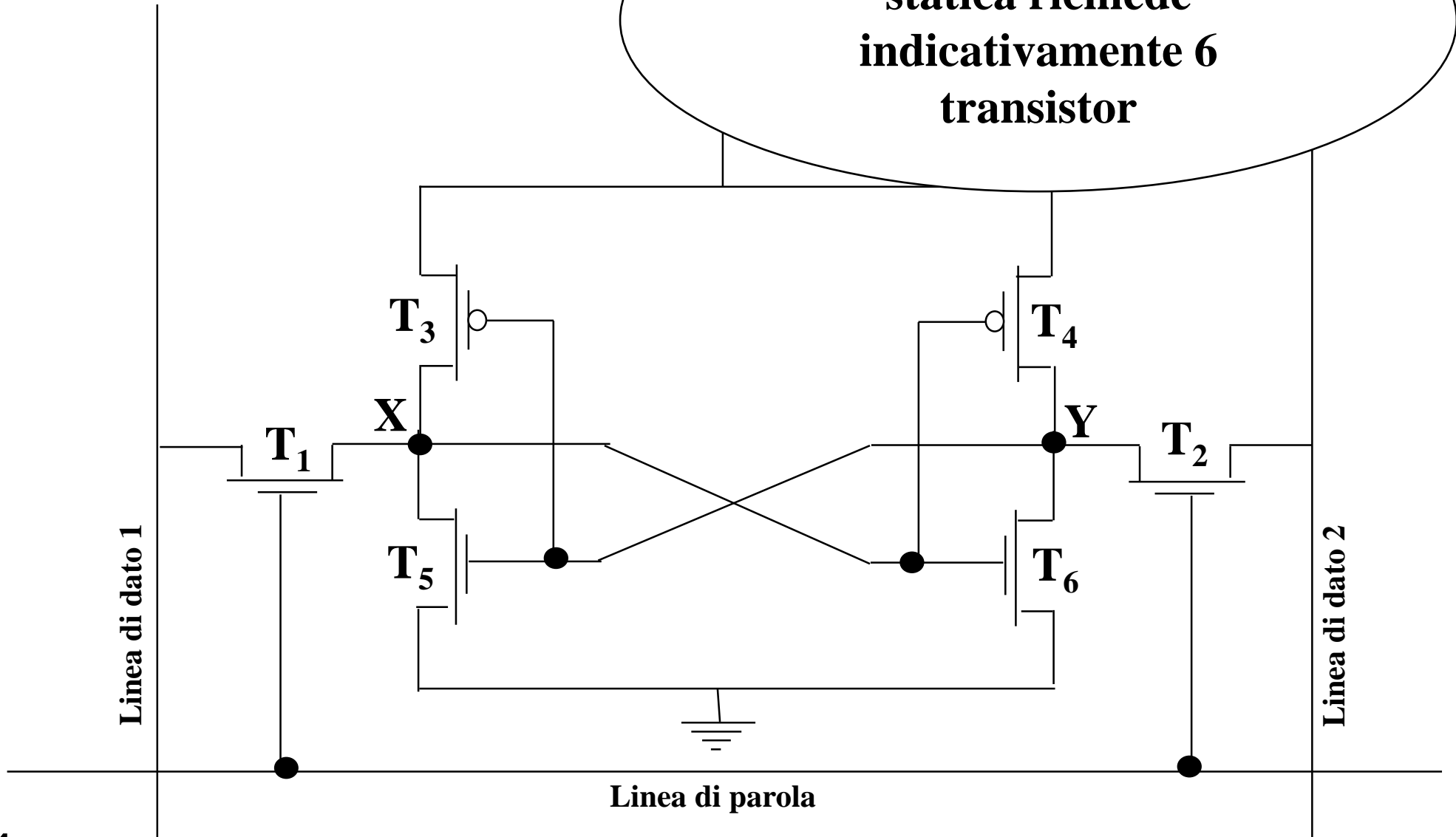


# Implementazione CMOS

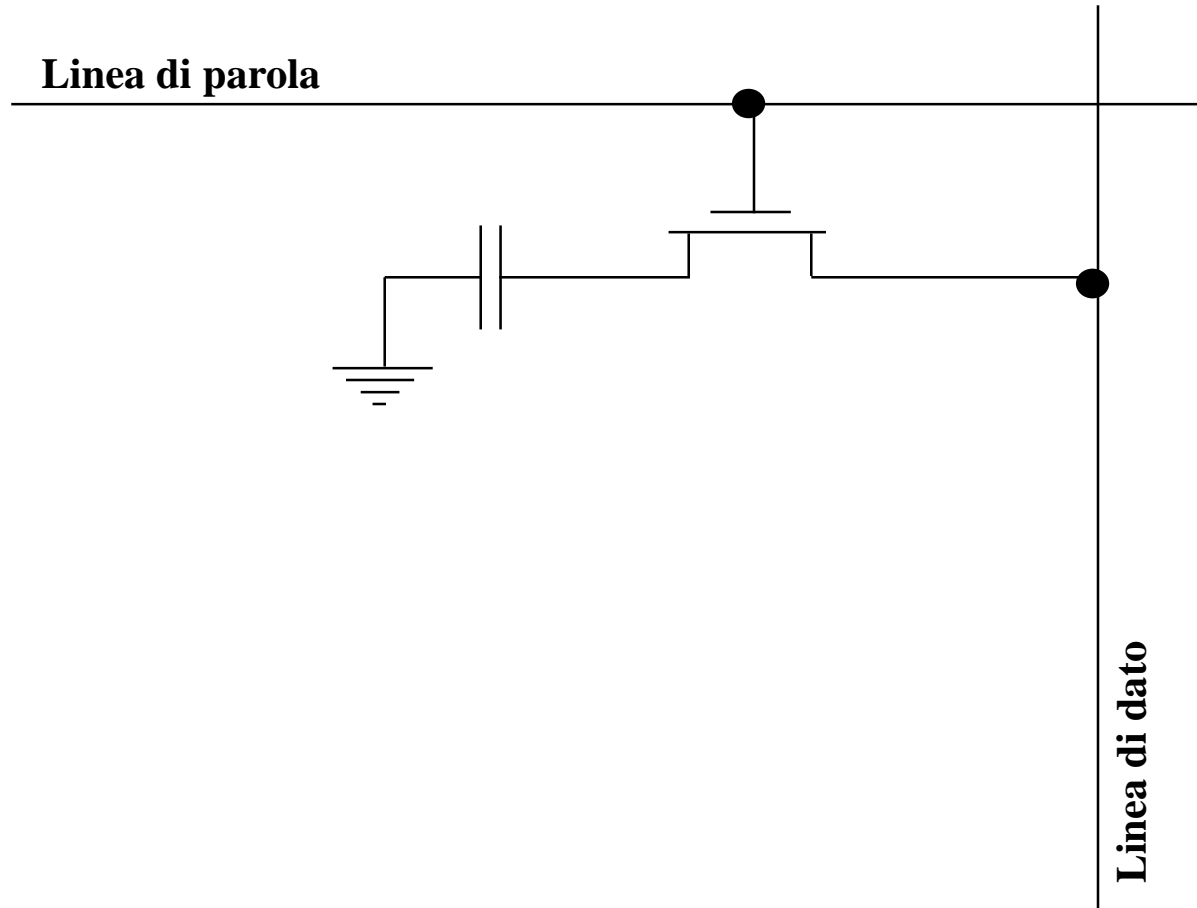


# Implementa

Ciascuna cella di RAM  
statica richiede  
indicativamente 6  
transistor



# Cella di RAM dinamica



# Funzionamento

**Nelle operazioni di lettura, attivando la linea di parola il condensatore viene collegato alla linea di dato, che assume quindi un valore 0 o 1 a seconda del valore memorizzato.**

**Nelle operazioni di scrittura, l'attivazione della linea di parola provoca il collegamento della linea di dato con il condensatore, che viene quindi caricato o scaricato, a seconda del valore di questa.**

**Un apposito sensore collegato alla linea di dato è in grado di rilevare l'eventuale cambiamento di tensione causato dalla carica/scarica del condensatore, e di produrre il corrispondente bit.**

# **Destructive Readout**

- **L'operazione di lettura comporta l'annullamento della carica elettrica immagazzinata nel condensatore con la conseguenza della perdita dell'informazione binaria (ossia da 1 si passa a 0)**
- **Ad ogni lettura la memoria, in modo autonomo, deve forzare un ciclo di scrittura per ripristinare il valore letto.**

# Rinfresco

**Consiste nell'operazione di amplificazione (verso 1) della eventuale carica contenuta nel condensatore, che tende a 0 per l'esistenza di inevitabili correnti di dispersione.**

**Si basa su operazioni di lettura fittizie, nelle quali il valore letto non viene trasmesso all'esterno.**

**È indispensabile per poter mantenere indefinitamente il contenuto di ciascuna cella di DRAM.**

**Le operazioni di rinfresco occupano una RAM per tempi molto brevi (dell'ordine di qualche %).**

# Circuiteria per il rinfresco

**In passato doveva venir gestito da circuiteria apposita esterna ai chip di DRAM. Attualmente è quasi sempre realizzato con il supporto di apposita circuiteria interna. È quindi quasi trasparente all'utente.**

**È possibile che le operazioni di rinfresco (prioritarie) e quelle di accesso normale siano attivate contemporaneamente: in tal caso è necessario che l'operazione normale sia temporaneamente sospesa.**

# Stima di costo

**Periodo di rinfresco = 64 ms**

**Tempo minimo di accesso a parola = 50 ns**

**Numero di cicli di rinfresco = 8k**

**Durata del ciclo di rinfresco = 8k x 50 ns = 0,41 ms**

**Costo complessivo = 0,41 / 64 ms = 0,64%**



# Affidabilità

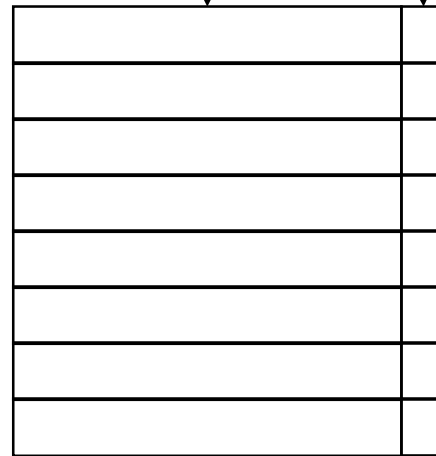
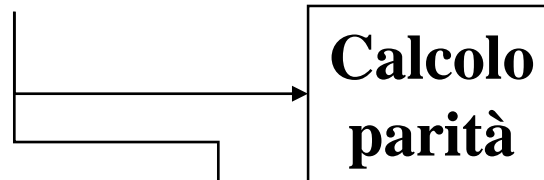
- Se una cella di memoria dinamica è colpita da una particella carica è possibile che la carica immagazzinata cambi, facendo cambiare il valore memorizzato (guasto transitorio)
- Al crescere della densità di integrazione, la dimensione della carica immagazzinata in ciascuna cella tende a diminuire in maniera significativa, rendendo quindi la memoria sempre più sensibile (e meno affidabile)
- Molte memorie dinamiche sono per questo protette, ad esempio tramite l'uso di codici di protezione.

# Codice di protezione

- Per aumentare l'affidabilità delle memorie, a ciascuna parola può essere associato un codice di protezione
- Il caso più semplice di codice di protezione è il *codice di parità* (1 bit)
- Funzionamento:
  - quando si scrive un valore nella parola, si calcola il relativo bit di parità, e lo si memorizza insieme al nuovo valore in un'apposita colonna
  - quando si legge la parola, si calcola il codice di parità associato al valore letto, e lo si confronta con quello memorizzato
  - in caso di diversità, si invia una segnalazione di errore.

# Codice di parità: architettura

Valore scritto



Valore letto

Calcolo  
parità

È un comparatore,  
implementato ad  
esempio attraverso  
un EXOR

Errore

# Codici di Hamming

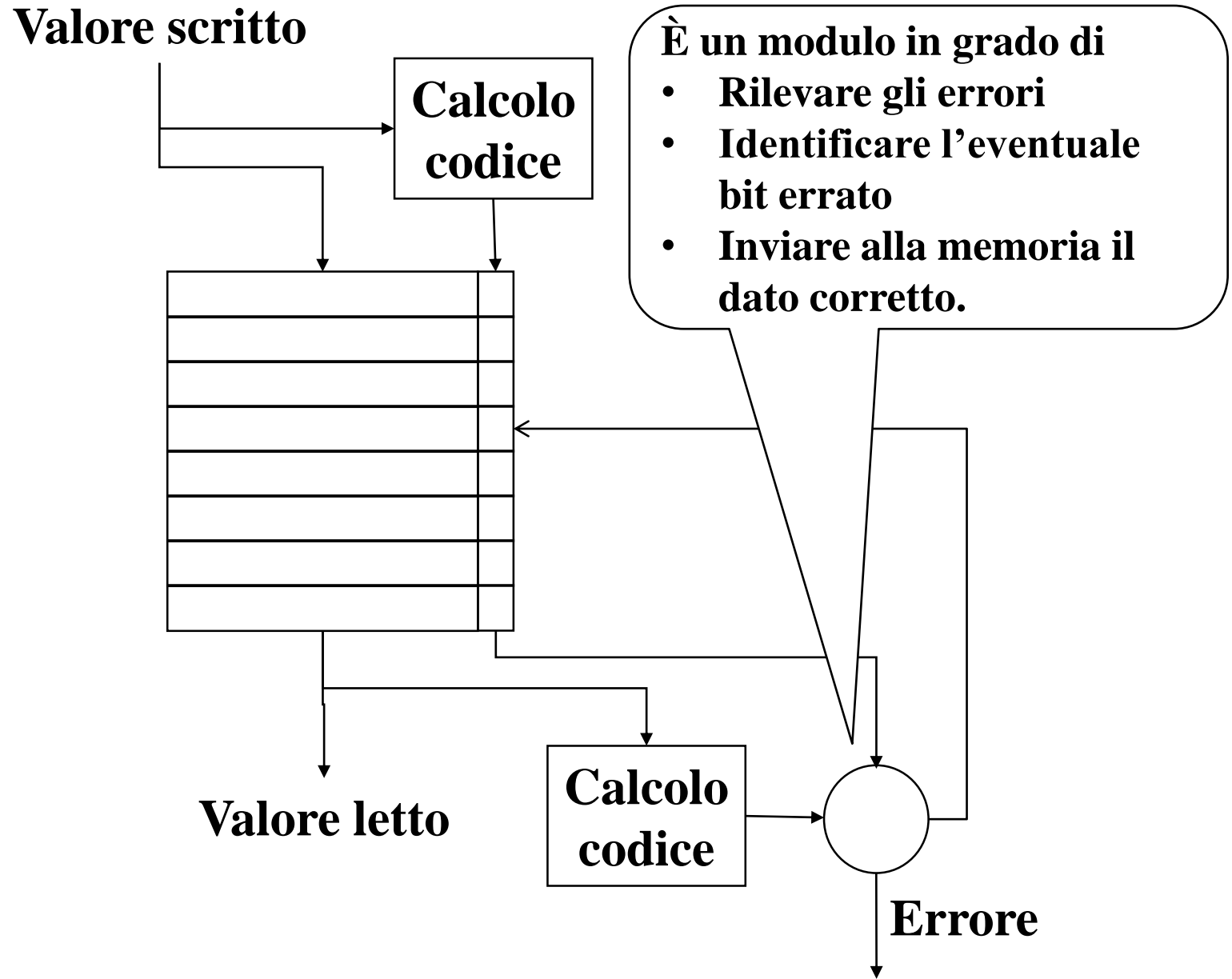
Attraverso i codici di Hamming è possibile non solo rilevare, ma anche correggere eventuali errori verificatisi in una memoria.

Detto  $n$  il parallelismo della memoria, tali codici

- Richiedono  $1 + \log_2 n$  bit di codice
- permettono di rilevare e correggere tutti gli errori singoli
- permettono di rilevare (ma non correggere) tutti gli errori doppi.

Per questo tali codici sono detti SECDED (*Single Error Correction Double Error Detection*).

# Codice di Hamming: architettura



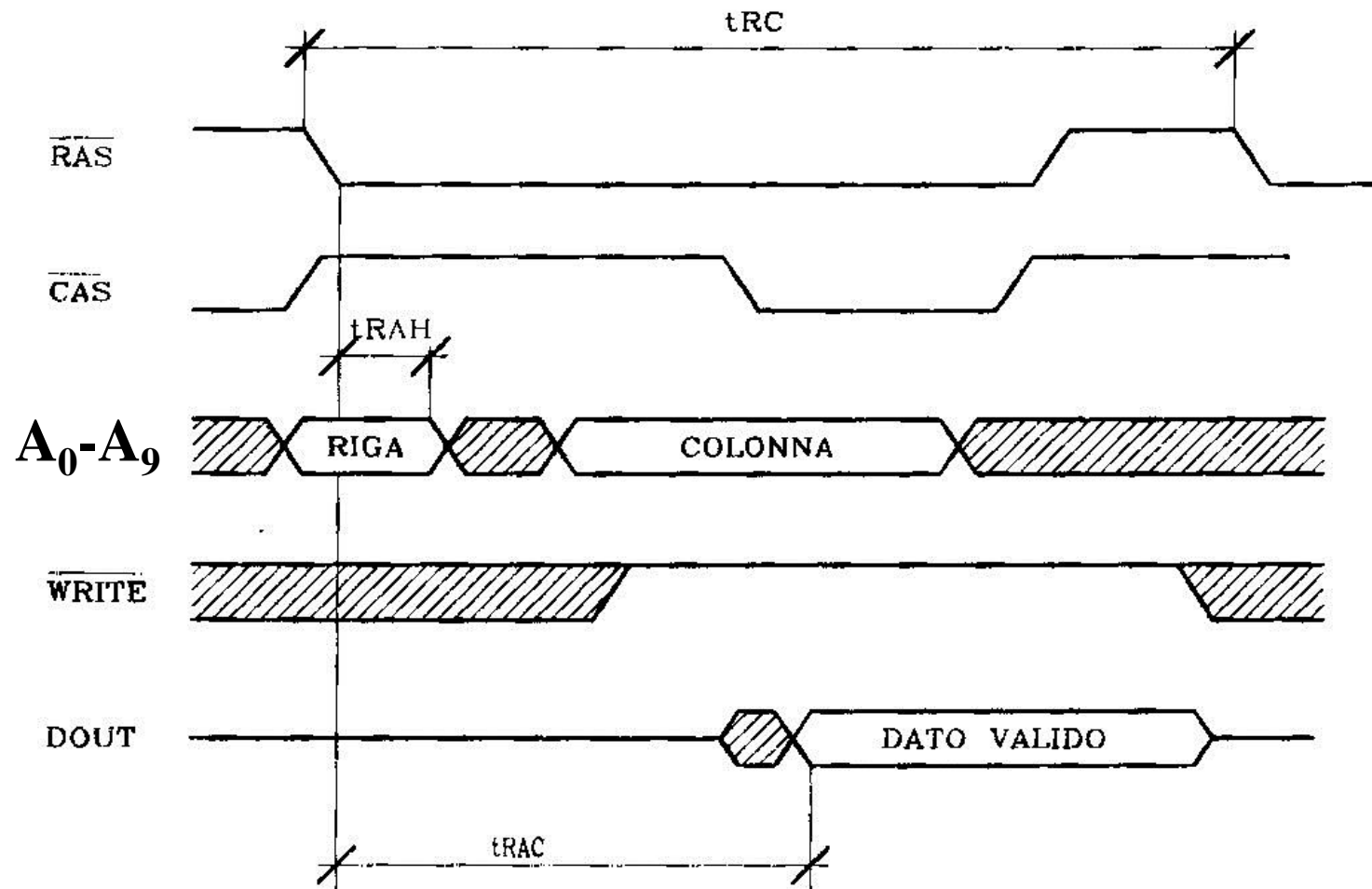
# Esempio

**Si consideri la DRAM AS4C1M16E5 prodotta da Alliance Semiconductor.**

**Caratteristiche:**

- **1 M × 16 bit**
- **Organizzazione a matrice quadrata**
- **Pin:**
  - **A<sub>0</sub> – A<sub>9</sub>: indirizzo**
  - **RAS, CAS (attivi bassi)**
  - **DQ1 – DQ16: dati**
  - **WE: abilitazione a scrittura (attivo basso)**
  - **OE: abilitazione uscite (attivo basso).**

# Temporizzazioni



# Lettura

- Si mette su  $A_0 - A_9$  l'indirizzo di riga
- Si attiva RAS
- Si aspetta il tempo  $t_{RAH}$
- Si mette su  $A_0 - A_9$  l'indirizzo di colonna
- Si attiva CAS
- Trascorso il tempo  $t_{RAC}$  i dati sono disponibili su DOUT.



# Note

- **Non si può iniziare un nuovo ciclo di lettura o scrittura se non dopo che è passato un tempo  $t_{RC}$  (pari a 100 ns).**
- **Ogni 16 ms è necessario eseguire un rinfresco completo, ossia eseguire 1024 operazioni con indirizzo di riga crescente.**

# RAM statiche e RAM dinamiche

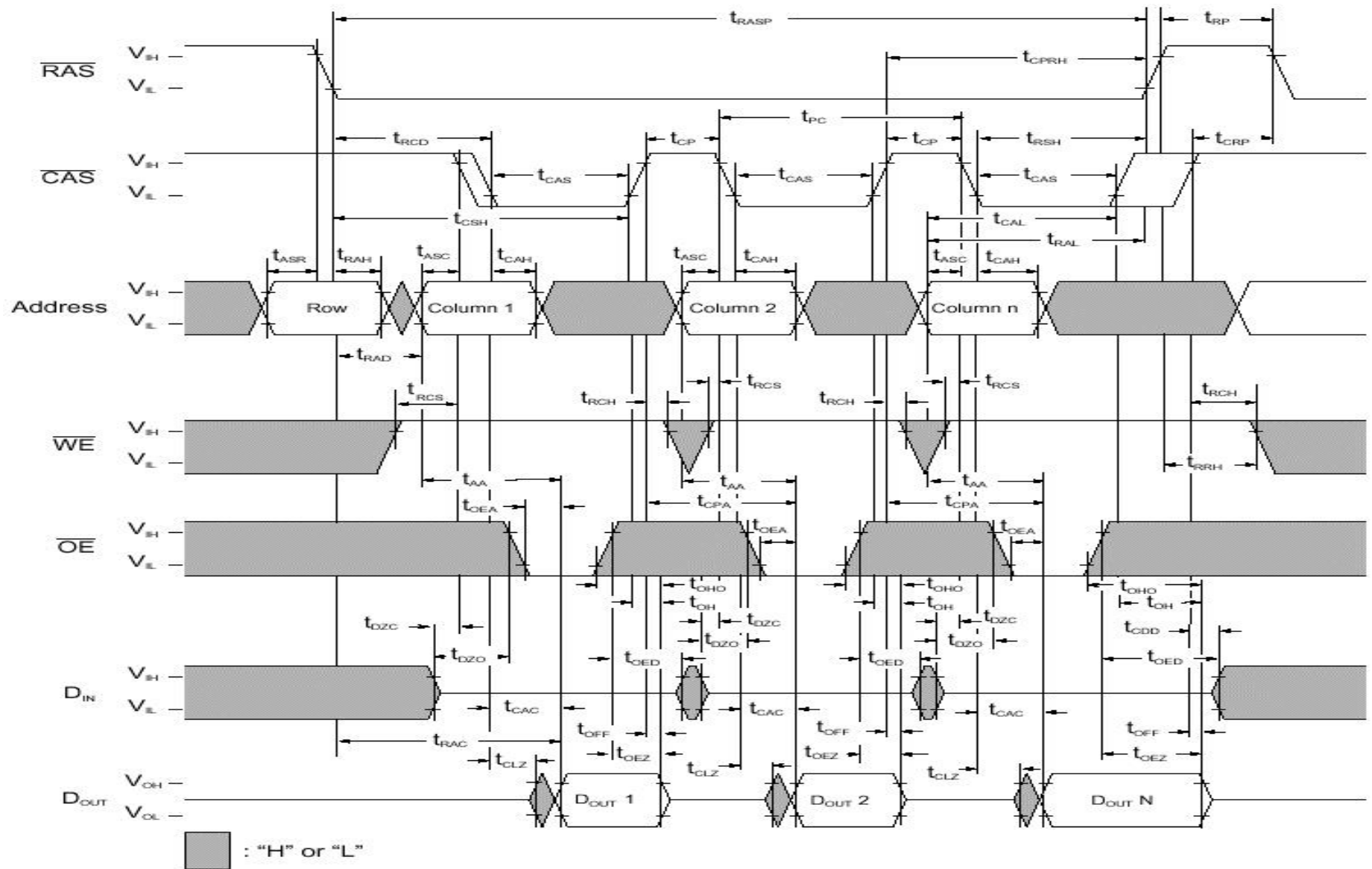
**Le RAM statiche sono (rispetto a quelle dinamiche):**

- **più veloci (tempi di accesso dell'ordine di 10 ns, contro 100 ns)**
- **più costose (in termini di area di silicio richiesta, quindi meno dense)**
- **più semplici da utilizzare**
- **più affidabili.**

# Page Mode DRAM

- Durante la lettura il valore di RAS rimane attivo ed è possibile leggere un blocco di parole appartenenti alla stessa riga (detta *pagina*) attraverso cicli successivi in cui vengono caricati esclusivamente gli indirizzi di colonna (attivando il segnale CAS).

# Ciclo di lettura in Fast Page Mode



# **Memorie interlacciate**

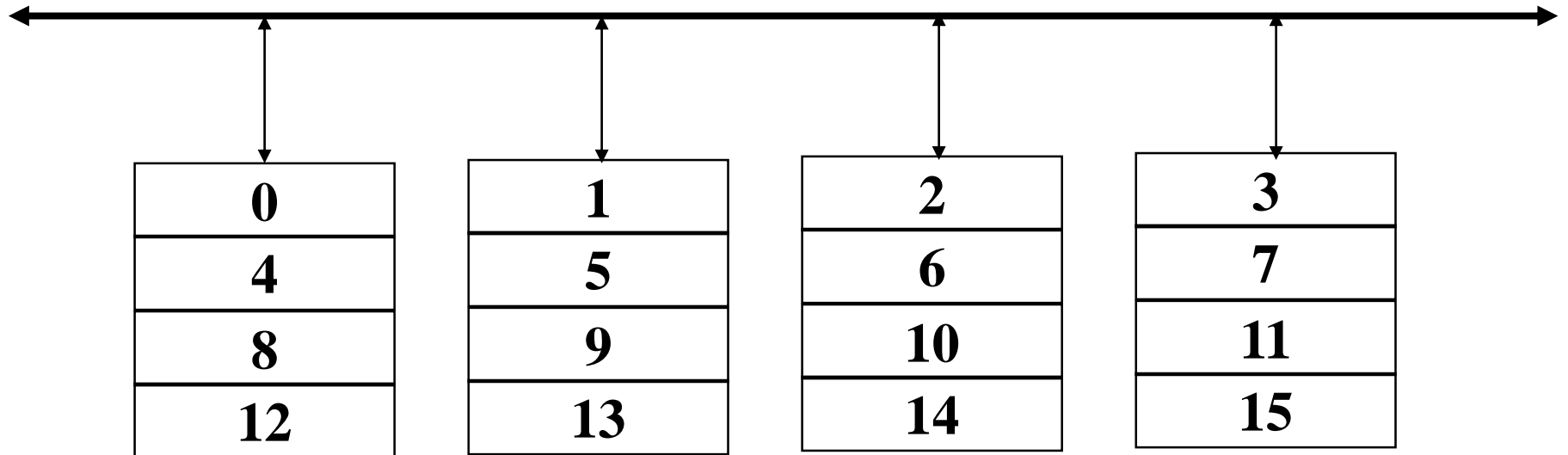
**Sono composte da più moduli di memoria.**

**Le parole sono disposte in maniera alternata tra i vari moduli.**

**Quando si fa accesso ad una serie consecutiva di parole, queste si trovano in moduli diversi, ed il loro accesso può essere fatto in parallelo, utilizzando un bus più ampio.**

**Permettono quindi di ridurre il tempo di accesso complessivo nel caso di accessi a blocchi, come quelli che si verificano nel caso sia presente una cache.**

# Memorie interlacciate: esempio



# Banchi di memoria in 8086

