La memoria: tecnologie di memorizzazione

Proff. A. Borghese, F. Pedersini

Dipartimento di Informatica
Università degli Studi di Milano

Organizzazione della memoria

La memoria è organizzata in ***N = 2^K parole (word)*** di ***M bit***

M* : *ampiezza della memoria [bit/byte]

N=2^K* : *altezza della memoria [n. celle]

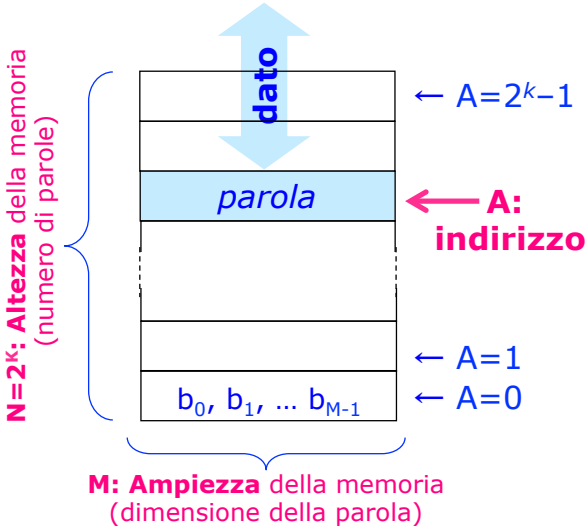
- La dimensione della parola di memoria coincide in MIPS32 con la dimensione dei registri della *CPU (**word**)*

C: Capacità della memoria

***C = N x M** [bytes]*

Esempio: M=32 bit, k=32

- ***C*** $= 2^{32}$ *x 32 bit*
  $=$ *4 Gwords x 4 bytes =* ***16 GB***



Memoria MIPS32:

- **Celle elem. indirizzate** di **1 byte**
- **Read/Write: 1 word (4 byte)**

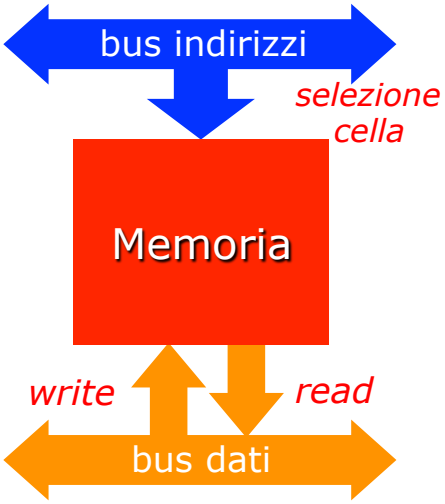
Struttura della memoria

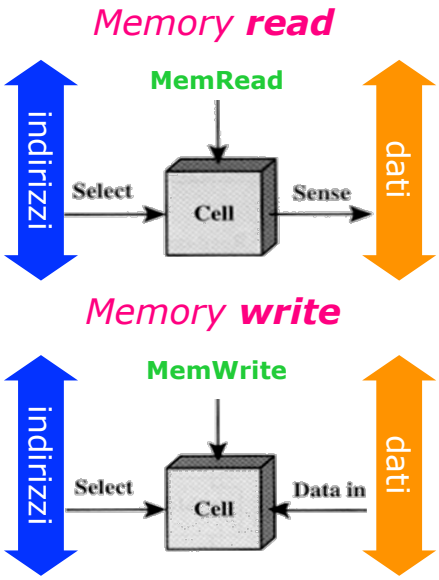
2 porte di comunicazione:

- **INDIRIZZO**: selezione di cella
- **DATI**: contenuto della cella

2 operazioni:

WRITE: scrittura nella cella
READ: lettura della cella

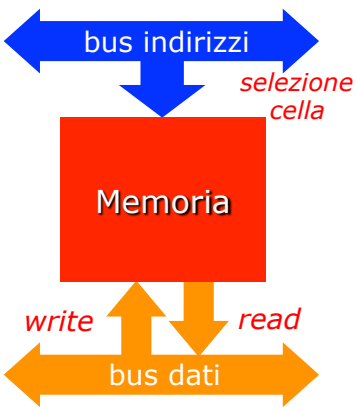


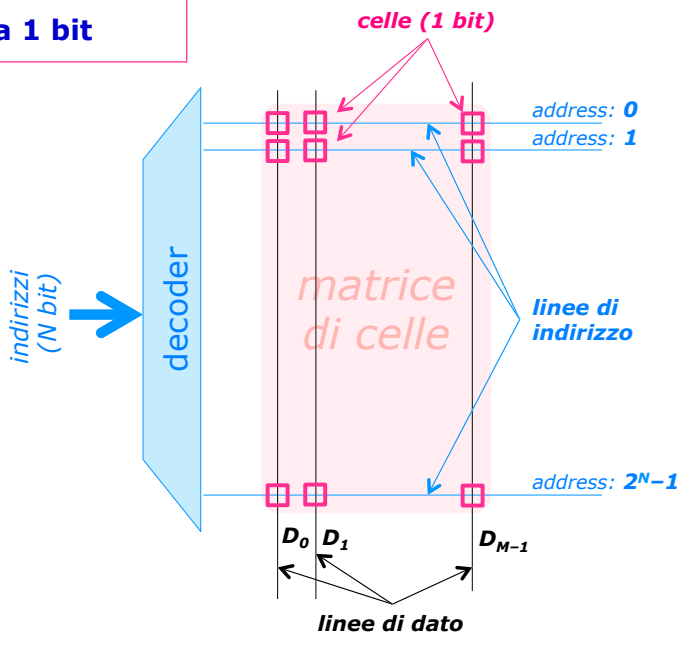


Struttura della memoria

Struttura generale di una memoria

Es: memoria di **2^N celle da M bit**:
struttura a **matrice** di **2^N x M celle da 1 bit**





Sommario

- Memorie ROM
- Memorie RAM
  - La RAM statica (SRAM)
  - La RAM dinamica (DRAM)
  - Memorie con controllo degli errori (ECC)

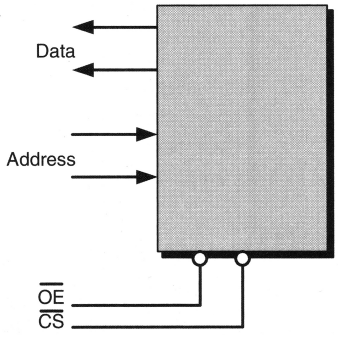
Memorie ROM / RAM

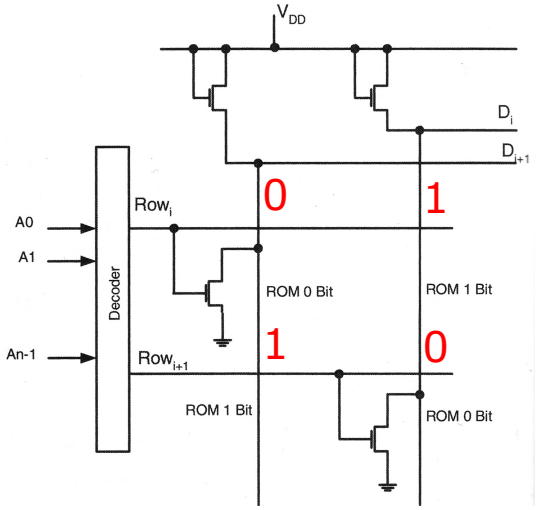
ROM: Read-Only Memory: memoria di **sola lettura**

- porta indirizzi (Address): INGRESSO
- porta dati (Data): USCITA

Tecnica di memorizzazione: **presenza/assenza** di MOS ➔ **0/1**

- ROM vergine: tutti i MOS presenti
- SCRITTURA: in corrispondenza di "0", il MOS viene bruciato.

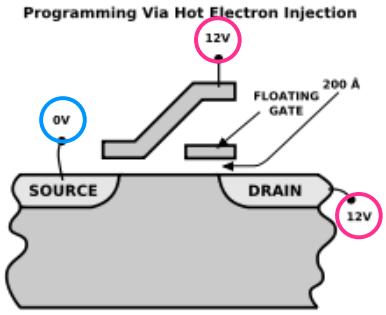




Memorie **FLASH**

Tecnologia FLASH

sta sostituendo le ROM e le memorie di massa (hard disk)

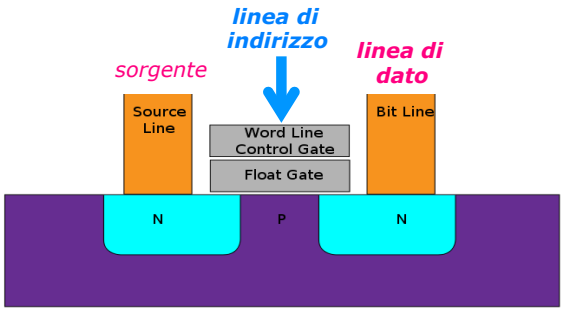


Cella: 1 transistore MOSFET con doppio GATE

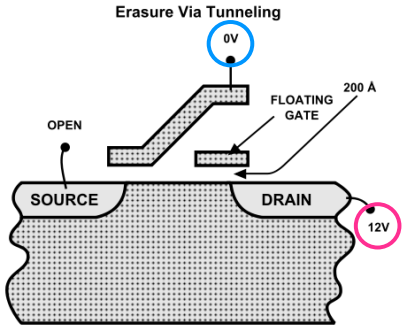
Control Gate e **Floating Gate** (isolato)

Operazioni di base:

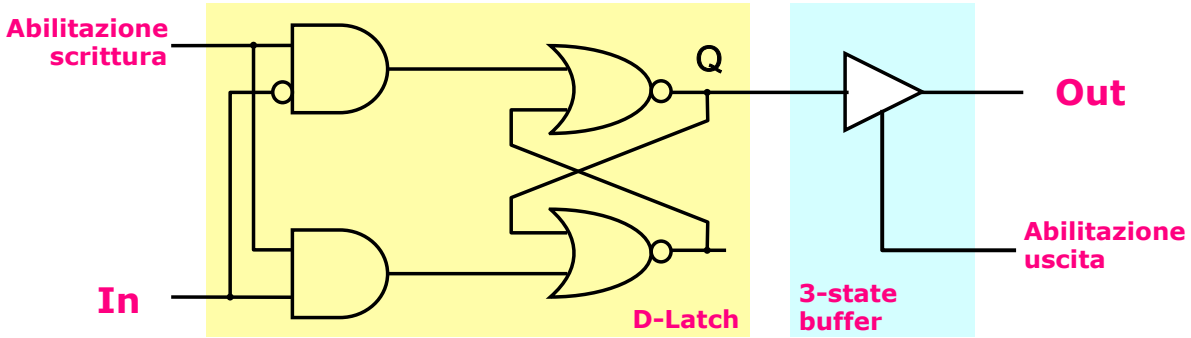
- **scrittura "1"**: iniezione di carica nel floating gate (*"hot charge injection"*)
- **cancellazione (scrittura "0")**: svuotamento della carica (per effetto *tunnel*)



MOSFET a doppio GATE



SRAM – RAM statica



Static RAM – SRAM

CELLA SRAM: un D-latch

- Qualità principale delle S-RAM: **velocità** (scrittura/lettura 1÷10 nsec)

Velocità di una SRAM:

- **Tempo di lettura**: tempo di abilitazione del buffer di uscita: t_{buf}
- **Tempo di scrittura**: tempo di transizione del Latch

Perché abbiamo bisogno di un buffer 3-state in uscita?

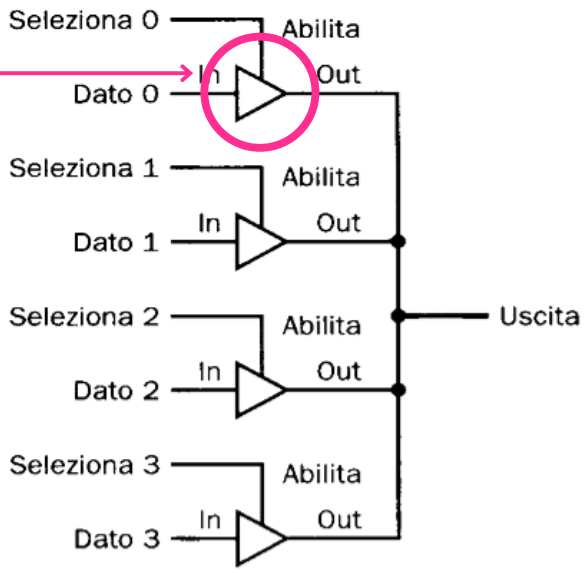
Memorie: *uscita three-state*

Problematico utilizzare un **array** di 2^k **celle** (come nel Register File)

- Ad es. memorie di 64 kB richiederebbero un **MUX a 64k ingressi**

Si utilizza la tecnologia **three-state:**
➔ **buffer three-state**

- Tutte le uscite delle celle sono collegate ad un'uscita comune
  - necessario evitare conflitti fra le uscite
  - → **uscite "isolate" con porte *three-state***
  - → **seleziono una sola cella alla volta**



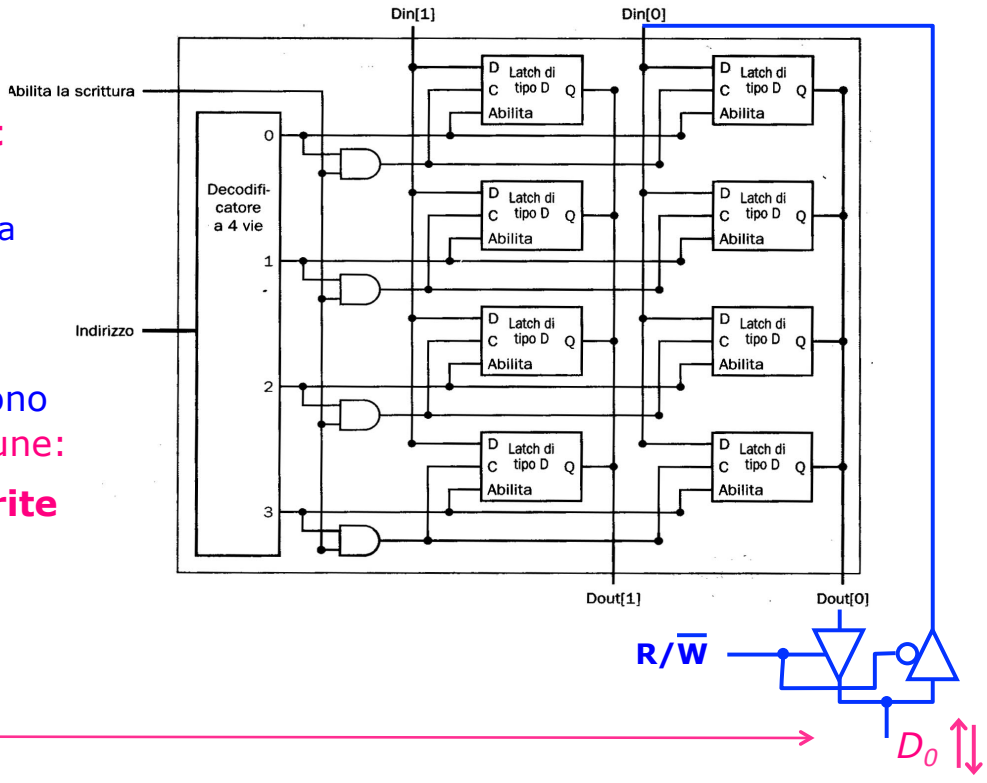
Esempio di SRAM

Esempio:
SRAM
4 celle x 2 bit

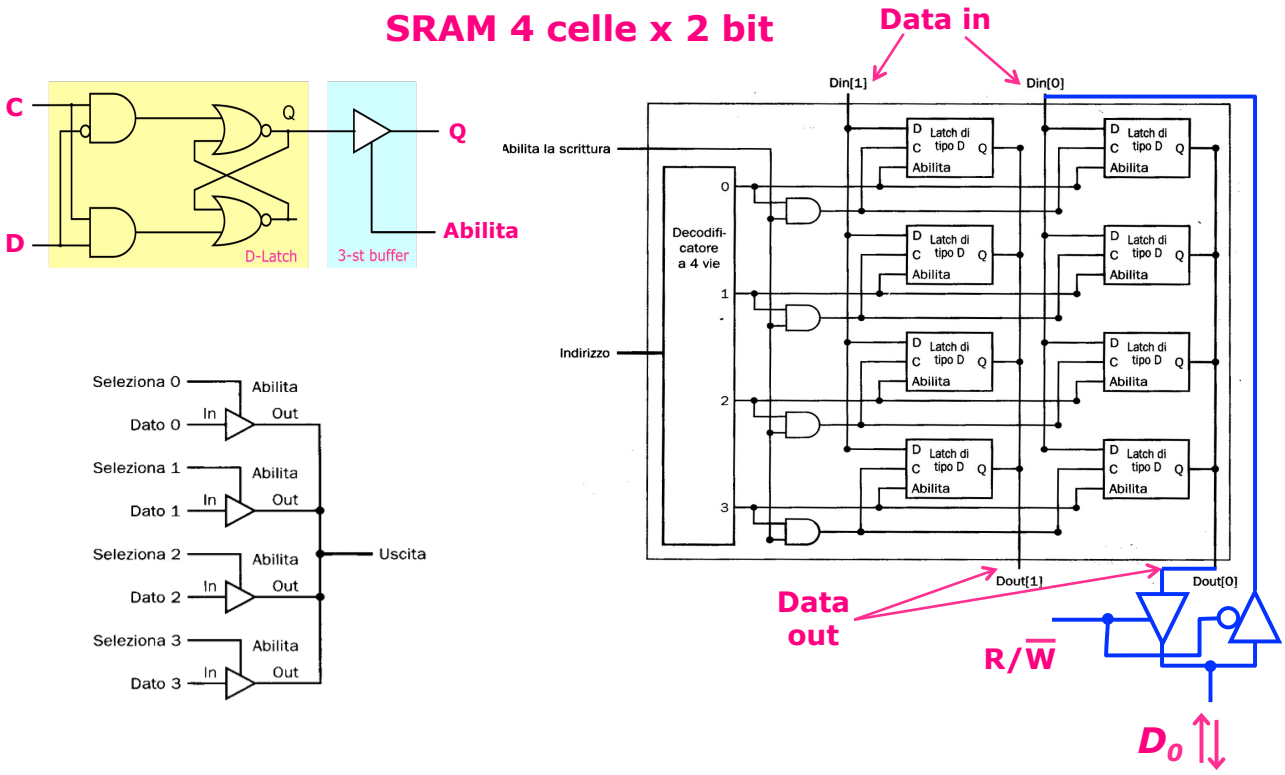
Struttura simile a quella di un Register File

Le linee dati sono spesso in comune:

Data read/write



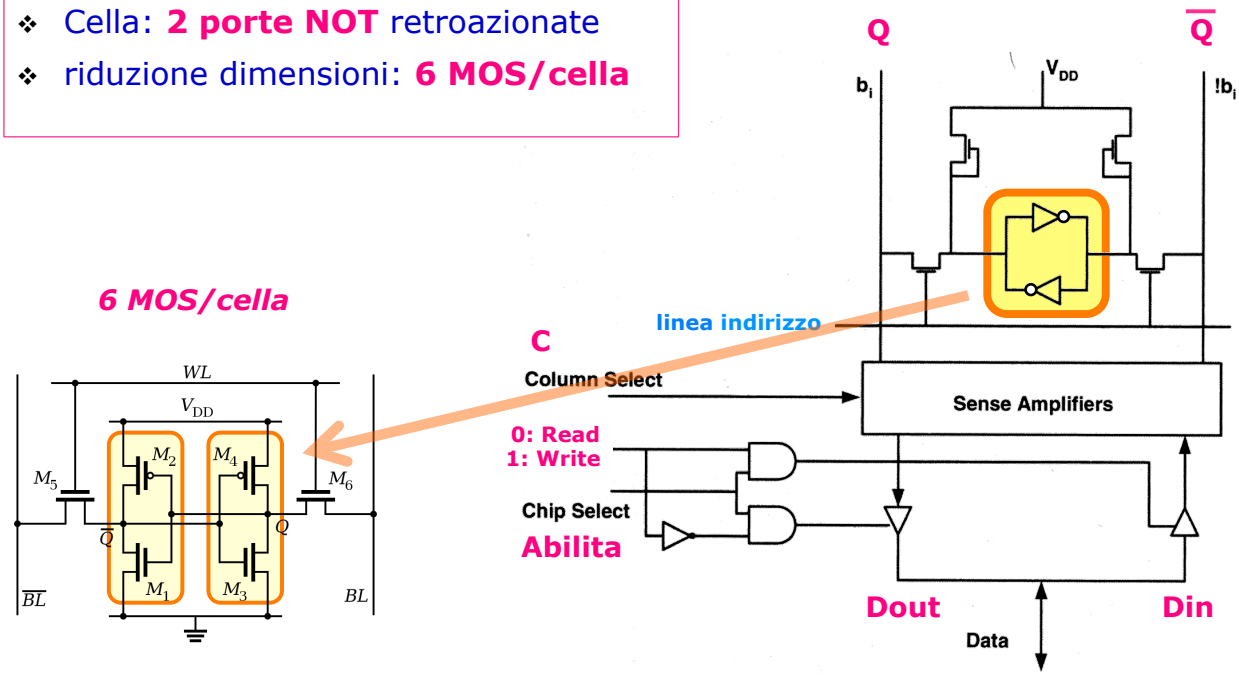
Esempio di SRAM



Memorie RAM statiche (SRAM)

SRAM moderne

- Cella: **2 porte NOT** retroazionate
- riduzione dimensioni: **6 MOS/cella**



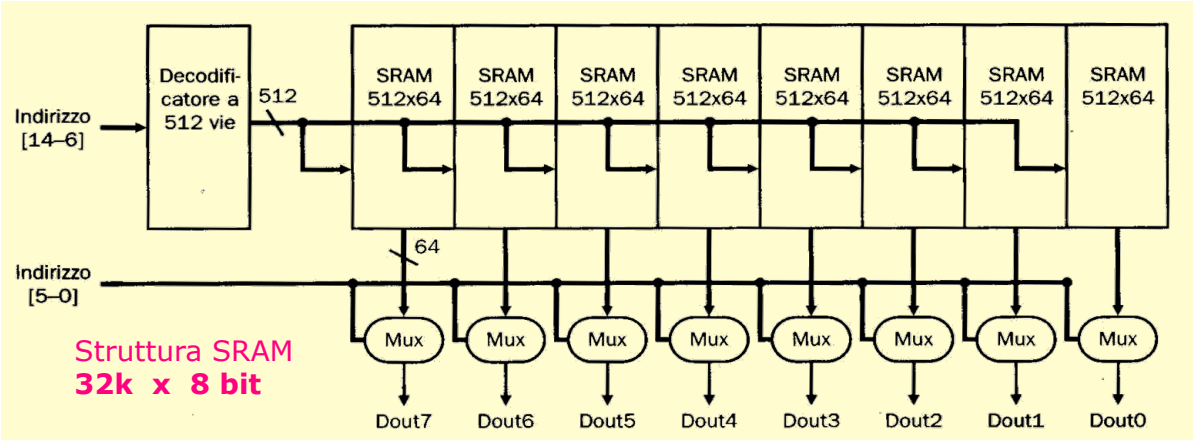
Indirizzamento SRAM a banchi

Problema: **RAM 32 k x 8 bit** → 8 x (32 k x 1 bit)

- Decodificatore a 15 bit → **32 K** linee di abilitazione, **32 K** uscite

Considero che: **32k x 8 bit = 512 x 512 bit**

- Per ogni bit di ampiezza, ho 512 banchi di 64 bit
  - **1 DEMUX a 15 bit (32 K uscite)**
  - **1 DEMUX a 9 bit (512 uscite) + 8 MUX a 6 bit → (64 ingressi)**



Sommario

- Memorie ROM
- Memorie RAM
  - La RAM statica (SRAM)
  - La RAM dinamica (DRAM)
  - Memorie con controllo degli errori (ECC)

Memorie RAM dinamiche (DRAM)

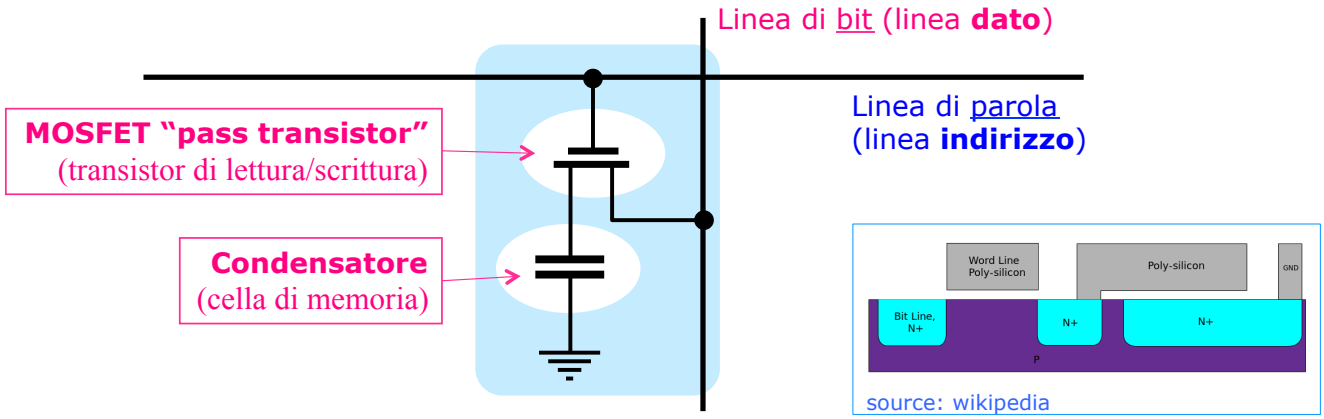
Funzionamento:

Elemento di memoria: **carica di un condensatore**

- condensatore CARICO → 1
- condensatore SCARICO → 0

Cella DRAM: **1 transistor + 1 condensatore**

- La lettura scarica la memoria che deve essere ricaricata
- Necessario un refresh periodico (gestito autonomamente dal controllore della memoria)



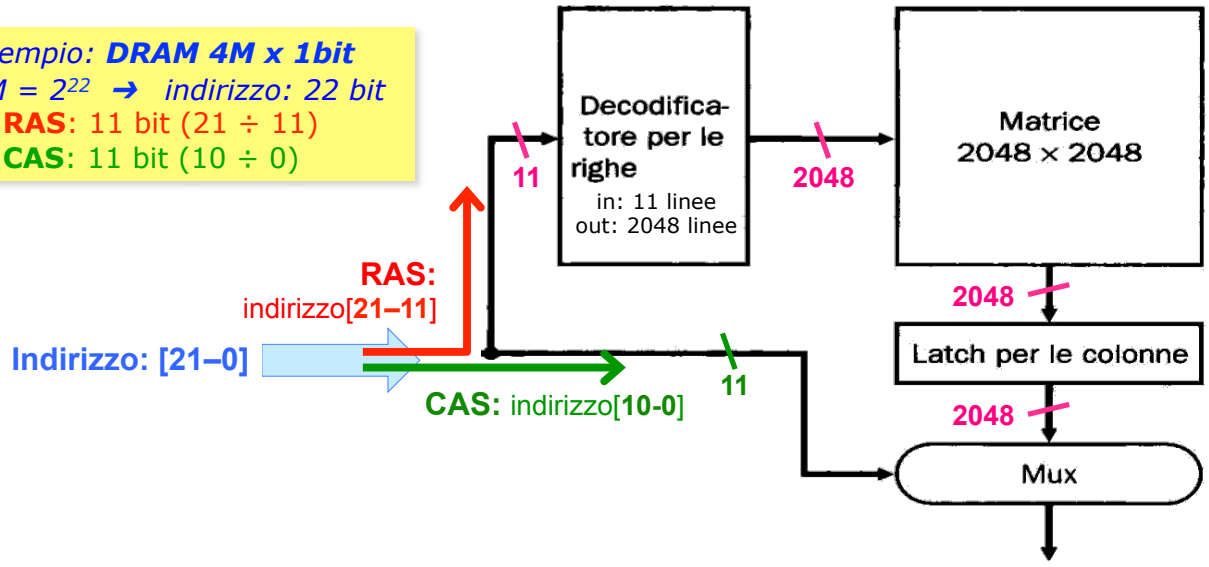
Struttura di una DRAM

Struttura a blocchi "quadrata" (come SRAM):

- **Capacità DRAM: 2^{2N} x 1 bit → matrice quadrata di 2^N x 2^N bit**

Accesso al bit: selezione **riga (N bit):** segnale **RAS** (**R**ow **A**ddress **S**trobe)
selezione **colonna (N bit)**: segnale **CAS** (**C**olumn **A**ddress **S**trobe)

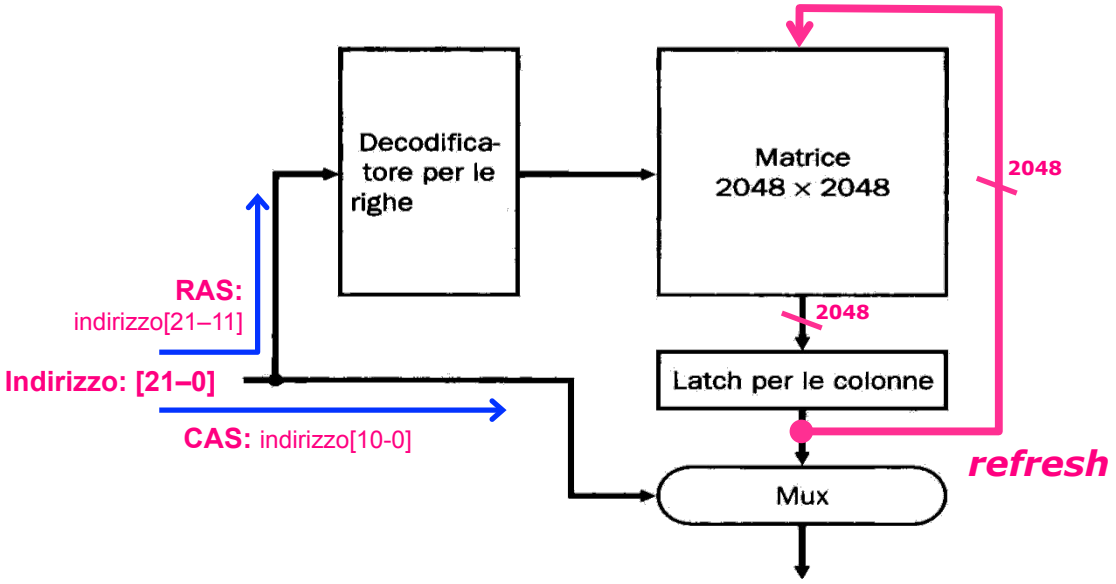
*Esempio: **DRAM 4M x 1bit***
4M = 2^{22} ➔ indirizzo: 22 bit
→ **RAS**: 11 bit (21 ÷ 11)
→ **CAS**: 11 bit (10 ÷ 0)



Struttura di una DRAM: refresh

Refresh di una RAM dinamica:

- Problema: il condensatore della cella si scarica in **30÷70 msec**!!!
  - entro tale tempo devo riscrivere ("rinfrescare") il dato nella DRAM
  - REFRESH: ad ogni lettura di riga



Struttura DRAM: esempio

Frequenza di refresh *di una RAM dinamica:*
frequenza di ripetizione delle operazione di refresh

Esempio:

In una RAM dinamica di 4 M x 1 bit il tempo di scarica dei condensatori è di 64 millisecondi.
Calcolare la minima frequenza di refresh.

- Una memoria di 4 Mbit ($\mathbf{4M = 2^{22}}$) è organizzata come matrice di $\mathbf{2^{11} \times 2^{11} = 2048 \times 2048}$ celle.
- Ogni ciclo di refresh rigenera una riga; la stessa riga deve essere di nuovo "rinfrescata" dopo al più 64 ms.
- Quindi in 64 ms devo rigenerare 2048 righe.

$$T_{REFRESH,\,MAX} = 64\ ms / 2048 = 31.25\ \mu s$$

$$f_{REFRESH,\,MIN} = 1 / T_{REFRESH,\,MAX} = 32\ kHz$$

Evoluzione delle SRAM e DRAM

Evoluzioni RAM dinamiche:

- Trasferimento a **burst** o a pagina: trasferimento consecutivo (ad alta velocità) di parole ad indirizzi consecutivi.

- **Synchronous** DRAM (**SDRAM**)
  - L'accesso alla memoria è sincrono con il clock dato dalla CPU (mem. bus)
  - La fase di indirizzamento e di recupero dei dati vengono separate in modo da ridurre al minimo l'impatto della latenza.
  - Tra l'indirizzamento ed il recupero dei dati, il processore può eseguire altri compiti (il processore può essere la CPU o il controllore della memoria, o altro: il dispositivo che controlla la memoria).

- **DDR-SDRAM** (Double-Data-Rate SDRAM)
  - Riescono 2 trasferimenti per ciclo di clock.
  - Data-rate doppio rispetto alla frequenza del clock del bus.

Prestazioni di una memoria

- **Parola di memoria** vs. **unità indirizzabile**
  - **Parola di memoria**:
    L'unità naturale in cui la memoria viene organizzata (MIPS: 32 bit)
  - **Unità indirizzabile**: il minimo numero di unità contigue indirizzabili.
    In quasi tutti i sistemi si tratta del byte.

- Tempo di accesso (***access time***):
  - tempo richiesto per eseguire una lettura/scrittura:
    dall'istante in cui l'indirizzo si presenta alla porta di lettura...
    ...all'istante in cui il dato diventa disponibile.

- Tempo di ciclo (***cycle time***):
  - per memorie ad accesso casuale: è il tempo di accesso più il tempo necessario perchè possa avvenire un secondo accesso a memoria.

- ***Transfer Rate***: quantità di informazione trasferita nell'unità di tempo [MB/s]
  - Random-access memory: $R = 1 / Memory\_cycle\_time$
  - Sequential memory: $R = 1 / [TA + N\ T_{TR}]$ T_{TR} : tempo di trasferim. N bytes

Alcuni dati: aprile 2004

- *Da: http://www.samsung.com/Products/Semiconductor*
- **SRAM**
  - Sincrone, 1M x 36, 2M x 18, tempo di accesso: 2,6ns
  - High speed, 1M x 18 o 512K x 36, tempo di accesso: 1,6ns
  - Asincrone: 8M x 16, tempo di accesso: 10ns
  - Low power, 8M x 16 tempo di accesso: 70ns
- **DDR–SDRAM**
  - 128M x 8, rate: **266Mb/s** (133Mhz → T_C = 7,5 ns)
  - 16M x 16, rate: **400Mb/s** (200Mhz).
  - 3 clock di latenza → 7,5*3 = **22,5 ns**
  - 2-4-8: larghezza del burst

  Attualmente:
  - DDR3 2133 MHz burst rate: > 17 GB/s (128 M x 64 bit)
  - GDDR5 burst rate: ~ 24 GB/s

Sommario

- Memorie ROM
- Memorie RAM
  - La RAM statica (SRAM)
  - La RAM dinamica (DRAM)
  - Memorie con controllo degli errori (ECC)

ECC – Error Correction Codes

- Errori dovuti a malfunzionamenti HW o SW
  - Date le dimensioni delle memorie (**10^{10} celle**) la probabilità d'errore non è più trascurabile
  - Per applicazioni sensibili, è di fondamentale importanza gestire gli errori di memorizzazione di bit.
- Codici di controllo errore
  - Permettono di riconoscere se si è verificato un errore, al costo dell'introduzione di bit in più, oltre ai dati (ridondanza)
  - **Codici rivelatori** d'errore
    - Consentono di individuare errori in una parola, ma non consentono di individuare dove si è verificato l'errore.
  - **Codici correttori** d'errore (error-correcting codes – ECC)
    - Consentono anche la correzione degli errori.
    - Richiedono bit in più rispetto ai codici rivelatori (es. per la correzione di 1 errore e l'individuazione di 2 errori in una parola di 128 bit, occorrono 8 bit in più)

Codici rivelatori d'errore

*Esempio di **codice rivelatore**:*

- **Bit di parità (even):**

  Aggiungo un bit ad una sequenza in modo da avere un n. pari (*even*) di "1"
  - `0000 1010 0` ← `bit di parità`
  - `0001 1010 1`
  - Un errore su uno dei bit porta ad un n. dispari di "1"
- Prestazioni del codice
  - mi accorgo dell'errore, ma non so dov'è
  - **rivelo ma non correggo errori singoli**

COSTO: 1 bit aggiuntivo ogni 8 → 9/8 = +12,5%

Codici correttori d'errore

*Esempio di **codice correttore**:*

- **Codice a ripetizione**

  Ripeto ogni singolo bit della sequenza originale per altre 2 volte
  → **triplico ogni bit**

  0 00 1 11 1 11 0 00 1 11 0 00 0 00 1 11 ...

  Un errore su un bit di ciascuna terna può essere corretto:

  000 → 010 → 000

  111 → 110 → 111

- Prestazioni del codice
  - mi accorgo dell'errore, ma non so dov'è
  - **rivelo e correggo errori singoli**

  COSTO: 2 bit aggiuntivi ogni 1 → 3/1 = +200%

Definizioni

Distanza di Hamming, **d** (tra 2 sequenze di N bit)

- il numero di cifre differenti, giustapponendole

  01001000

  01000010 → d = 2

Distanza minima di un codice, d_{MIN}

- il valor minimo di **d** (d_{MIN}) tra tutte le coppie di parole di codice

- **Capacità di rivelazione** di un codice: $r = d_{MIN} - 1$
- **Capacità di correzione** di un codice: $t = (d_{MIN} - 1)/2$

Esempi:

- Codice a bit di **parità**: $d_{MIN} = 2$ → r=1, t=0
- Codice a **ripetizione (3,1)**: $d_{MIN} = 3$ → r=2, t=1

Applicazioni nelle memorie

- RAM con controllo di parità
  - Aggiungo un bit di parità ad ogni byte

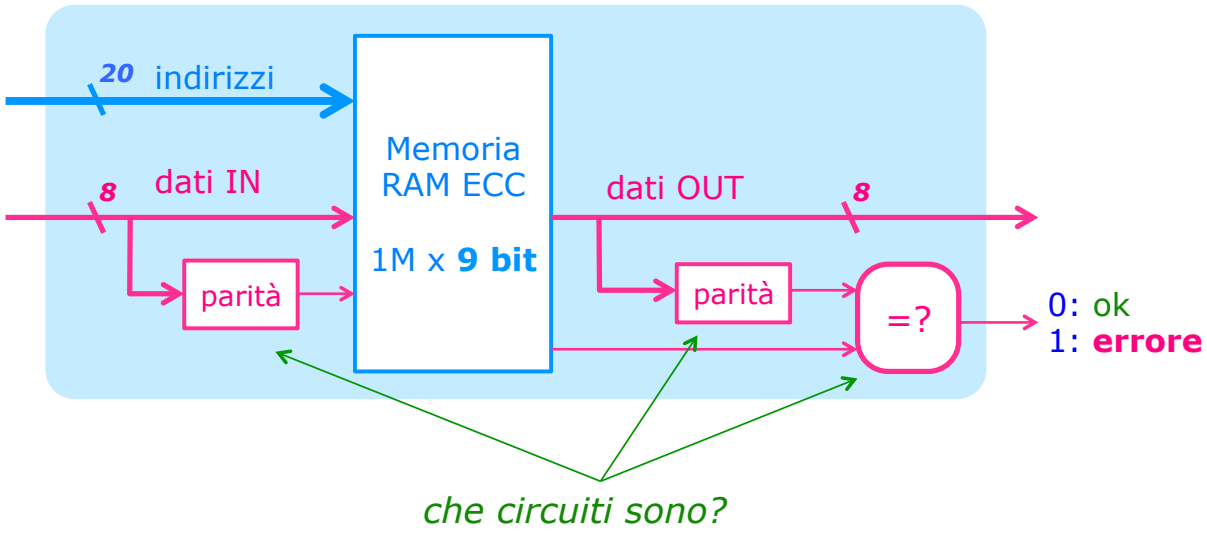
  Es: **RAM 1 M x 9 bit (8+1)**

  Detta p la probabilità di errore su un bit (ad es. $p=10^{-10}$),
  - Probabilità di **un errore** in un byte (+ parità): $P_1 = 9p(1-p)^8 \approx 9\,p$
  - Probabilità di **due errori** nel byte (+ parità): $P_2 = 36p^2(1-p)^7 \approx 36\,p^2$

- RAM con codice correttore di errori (ECC)
  - si usa nelle memorie cache
  - codici ECC evoluti (alta efficienza):
  - Hamming, CCITT-32, Reed-Solomon,
  - Turbo-codes...

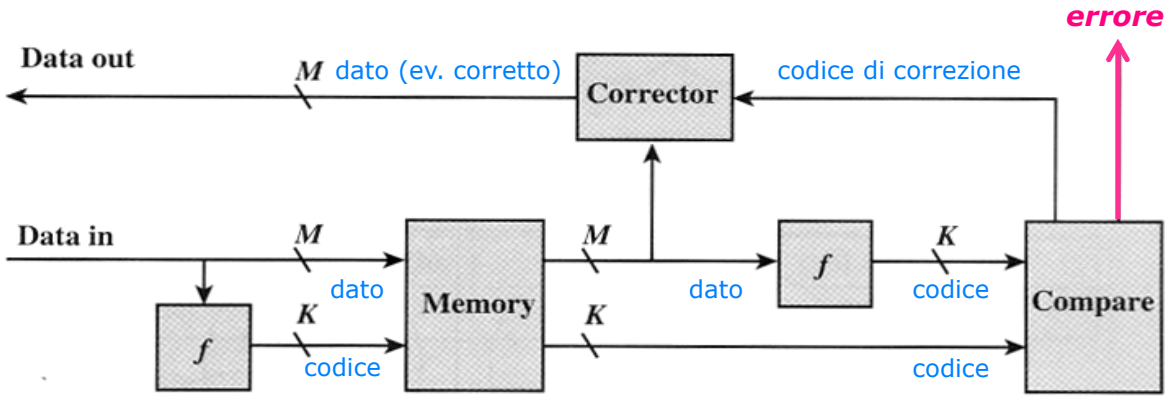
RAM con controllo di parità

- Schema RAM-ECC con controllo parità
  - parole di **9 bit**: 8 bit **dati** + 1 bit **parità** (o doppio: 18 bit)



Correzione degli Errori

- **ECC memory –** tre possibili casi:
  - **No errors detected**: il dato letto può essere inviato in uscita così com'è.
  - **1 errore individuato e corretto**: i bit del dato, più il codice associato vengono inviati al correttore, il quale provvede a correggere il dato.
  - **1 errore individuato, ma impossibile da correggere**: impossibilità di recupero – si segnala la condizione d'errore (genero una **eccezione**).



Dimensione di codici ECC

- Conviene applicare ECC a parole più lunghe possibile → aggiungo meno ridondanza → maggiore efficienza del codice

  Maggiore efficienza → maggiore complessità di codifica/decodifica

	Single-Error Correction		Single-Error Correction/ Double-Error Detection	
Data Bits	Check Bits	% Increase	Check Bits	% Increase
8	4	50	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91