



# La tecnologia delle memorie

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione  
[borgnese@dsi.unimi.it](mailto:borgnese@dsi.unimi.it)

Università degli Studi di Milano

Riferimento Patterson: B8; B9.



## Sommario

Gestione delle memorie cache.

SRAM.

DRAM.

Correzione degli errori.



## Gestione dei fallimenti di una cache



*Hit* – è quello che vorremmo ottenere, il funzionamento della CPU non viene alterato.

*Miss* – **in lettura** devo aspettare che il dato sia pronto in cache -> stallo.

Passi da eseguire in caso di Miss (fase Mem):

- 1) Ricaricare la pipeline (indirizzo dell'istruzione (PC-> PC-4), decodifica, esecuzione, memoria...)
- 2) Leggere il blocco di memoria dalla memoria principale.
- 3) Trasferire il blocco in cache, aggiornare i campi validita' e tag.
- 4) Avviare la fase di fetch, decodifica, esecuzione, memoria dell'istruzione con i dati corretti. .

NB Il programma non può continuare!!

**In scrittura?**



## Criteri di progettazione



*Cache primaria*: massimizzo Hit rate.

*Cache secondaria*: minimizzo Miss penalty (massimizzo transfer rate).



## Miss penalty: esempio



Tempi di accesso:

1 ciclo di clock per inviare l'indirizzo.

15 cicli di clock per ciascuna attivazione della Memoria (lettura di parola, dettata dal cammino critico in lettura).

1 ciclo di clock per trasferire una parola al livello superiore (cache).

Blocco di cache di 4 parole, blocco di memoria RAM di 1 parola

↓

$$\text{Miss\_Penalty} = 1 + 15 * 4 (\text{parole}) + 1 * 4 (\text{parole}) = 65 \text{ cicli\_clock}$$

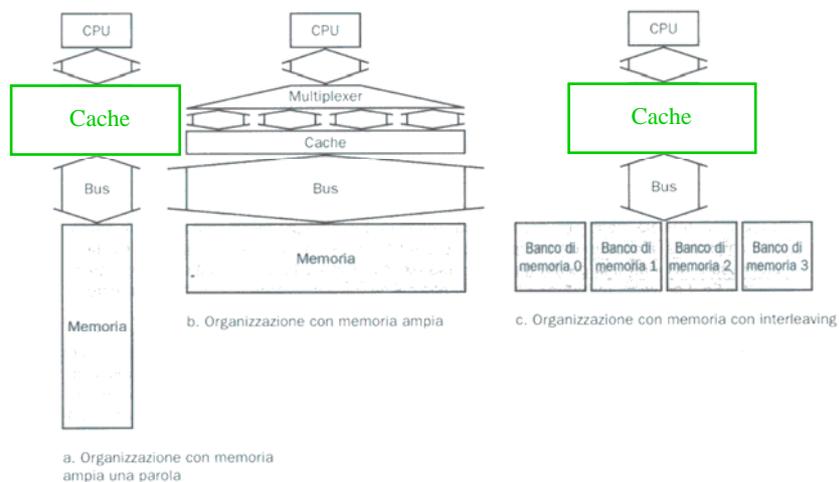
$$\# \text{byte} / \text{ciclo\_clock} = 4(\text{parole}) * 4(\text{byte/parola}) / 65(\text{cicli\_clock}) \cong 0,25(\text{byte} / \text{ciclo\_clock})$$

### Obiettivi:

- Diminuire la penalità di fallimento (miss\_penalty).
- Diminuire il tasso di fallimento (miss\_rate).



## Riduzione del miss penalty



**Interleaving** (interallacciamento). Banche che potrebbero essere trasferiti in parallelo alla cache.



## Valutazione della riduzione della “miss\_penalty”



**Architettura standard:** penalità di miss è di 65 cicli\_clock.

### Maggiore ampiezza della memoria:

- Organizzazione della Memoria Principale per blocchi.
- Bus più ampio (bus dati largo un blocco, 4 parole).
- Per blocchi di Memoria di 4 parole, blocchi di cache di 4 parole:

$$\text{Miss\_penalty} = 1 + 15 * 1 + 1 * 1 = 17 \text{ cicli\_clock.}$$

$$\#byte / ciclo\_clock = 4(parole) * 4(byte/parola) / 16(cicli\_clock) = 0,94(byte / ciclo\_clock)$$

### Interleaving:

- Organizzazione della Memoria Principale per **banchi** con accesso indipendente alla memoria (interleaving).
- Bus standard (trasferimento di 1 parola alla volta).
- Per blocchi di Memoria di 1 parola, blocchi di cache di 4 parole:

$$\text{Miss\_penalty} = 1 + 15 * 1 + 1 * 4 = 20 \text{ cicli\_clock.}$$

$$\#byte / ciclo\_clock = 4(parole) * 4(byte/parola) / 20(cicli\_clock) = 0,80(byte / ciclo\_clock)$$



## Criteri di progettazione



**Cache primaria:** massimizzo Hit rate.

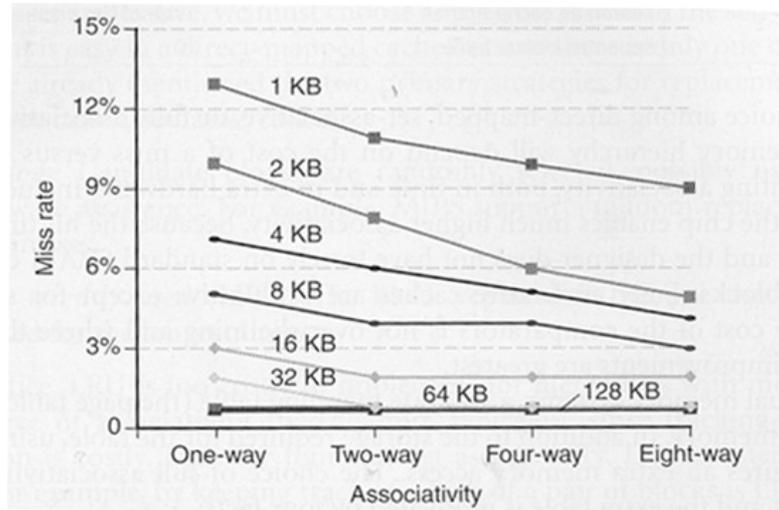
**Cache secondaria:** minimizzo Miss penalty (massimizzo transfer rate).



## Miss rate funzione del grado di associatività



Aumenta la associatività → Aumenta il tempo di trasferimento alla CPU.



## Miss rate in funzione della lunghezza del blocco



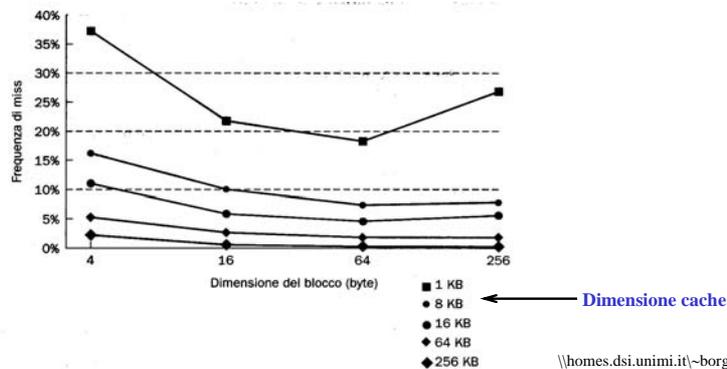
La parola di cache (blocco) è un multiplo della parola della macchina.

*Vantaggi:* per la località spaziale, diminuisco la frequenza di miss.

*Svantaggi:* per le dimensioni del blocco rispetto alla dimensione totale della cache aumenta la penalità di miss: competizione per le poche linee di cache. La penalità di miss è influenzata dalla lunghezza della linea di cache.

La località diminuisce all'aumentare della dimensione della linea.

La lunghezza della linea di cache dipende dalla parola del processore. Oggi si va verso 64-128byte.





## Split cache



**Split-cache: Cache dati e cache istruzioni.**

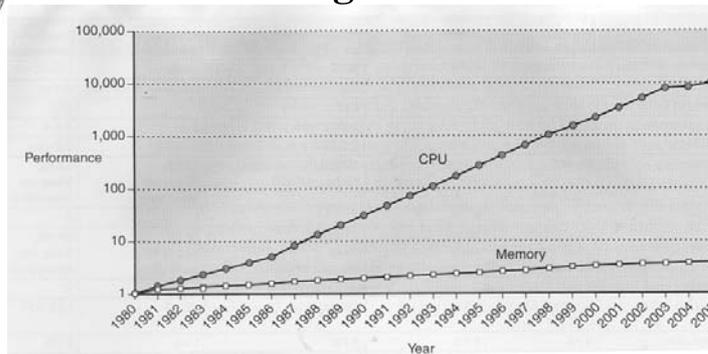
**Vantaggi.** Possibilità di analizzare le istruzioni in coda (contenute nella cache istruzioni) mentre si eseguono altre istruzioni (che lavorano su dati contenuti nella cache dati), senza dovere competere per l'accesso alla cache. Efficiente per le architetture superscalari.

**Svantaggi.** Minore hit rate, perchè non si sfrutta al meglio la memoria cache. Si potrebbe riempire un'unica cache maggiormente con dati od istruzioni a seconda del frammento di codice correntemente in esecuzione.

Il register spilling e le miss sono inevitabili → Come fare vedere al processore una memoria sufficientemente veloce?



## I trend delle gerarchie di memoria



Cache primaria nel processore.

Cache secondaria con un bus dedicato per il trasferimento al(la cache) del processore.

I criteri di progetto delle cache secondarie (e di terzo livello) possono essere diversi dai criteri di progetto della cache primaria (min miss\_penalty, vs min miss\_rate).

Si può utilizzare la fase di *idle* della cache secondaria per cercare attivamente quei blocchi che serviranno (pre-fetch).

Riorganizzazione efficiente del codice, in modo da soddisfare efficacemente il principio di località.



## Le dimensioni di alcune cache



Processor	Type	Year of Introduction	L1 cache <sup>a</sup>	L2 cache	L3 cache
IBM 360/85	Mainframe	1968	16 to 32 KB	—	—
PDP-11/70	Minicomputer	1975	1 KB	—	—
VAX 11/780	Minicomputer	1978	16 KB	—	—
IBM 3033	Mainframe	1978	64 KB	—	—
IBM 3090	Mainframe	1985	128 to 256 KB	—	—
Intel 80486	PC	1989	8 KB	—	—
Pentium	PC	1993	8 KB/8 KB	256 to 512 KB	—
PowerPC 601	PC	1993	32 KB	—	—
PowerPC 620	PC	1996	32 KB/32 KB	—	—
PowerPC G4	PC/server	1999	32 KB/32 KB	256 KB to 1 MB	2 MB
IBM S/390 G4	Mainframe	1997	32 KB	256 KB	2 MB
IBM S/390 G6	Mainframe	1999	256 KB	8 MB	—
Pentium 4	PC/server	2000	8 KB/8 KB	256 KB	—
IBM SP	High-end server/ supercomputer	2000	64 KB/32 KB	8 MB	—
CRAY MTA <sup>b</sup>	PC/server	2001	16 KB/16 KB	96 KB	4 MB
Itanium	PC/server	2001	16 KB/16 KB	96 KB	4 MB
SGI Origin 2001	High-end server	2001	32 KB/32 KB	4 MB	—



## Principi sulla Memoria Virtuale



*Due motivazioni:*

- Estensione della memoria fisica. Maggiore quantità di memoria.
- Gestione del multi-tasking. Negli anni '90 *overlay* definito nel linker, ora trasparente tramite il gestore della memoria virtuale.

Ogni programma ha il suo spazio di indirizzamento.  
Mappatura dello spazio di indirizzamento nella memoria fisica  
(*memory mapping* tramite la *page table*).

Memoria virtuale (estensione su disco) è concettualmente analoga alla cache.

Blocco di memoria → Pagina.  
Miss → Page Fault.



## Sommario



Gestione delle memorie cache.

SRAM.

DRAM.

Correzione degli errori.



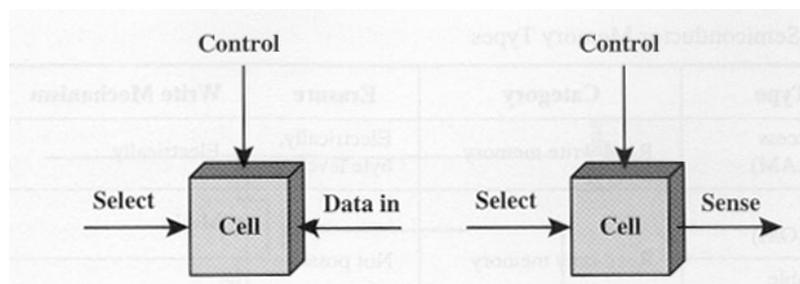
## Cella di memoria



La memoria è suddivisa in celle, ciascuna delle quali assume un valore binario stabile.

Si può scrivere il valore 0/1 in una cella.

Si può leggere il valore di ciascuna cella.



Quale struttura di memoria abbiamo già incontrato?

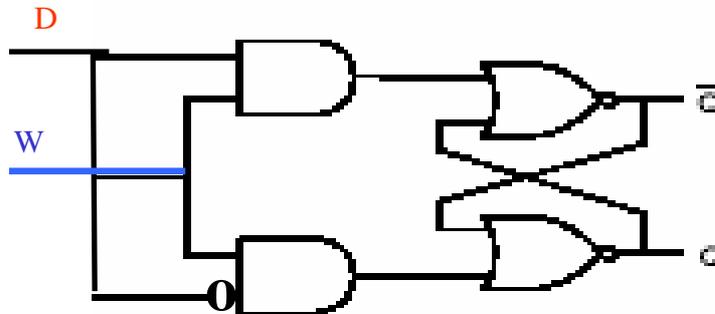
Control (lettura – abilitazione; scrittura)

Select (cf. dataport)

Data in & Sense (Data in & Data out).



## Cella SRAM



Selezione – non si applica  
 Lettura .- sempre disponibile in uscita  
 Scrittura – segnale esplicito (in AND con il clock in caso di cella sincrona).



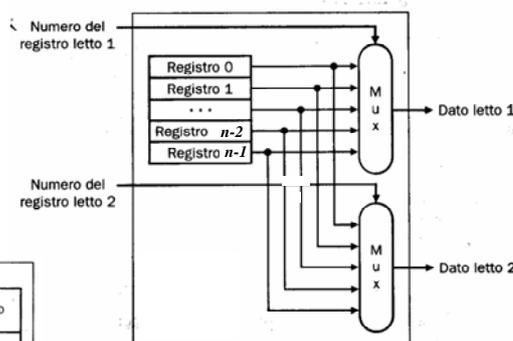
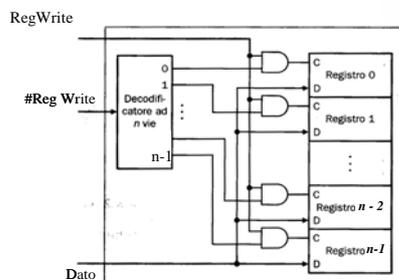
## Register file



Il tempo di lettura dipende dal cammino critico dei Mux.

Il tempo di scrittura dipende dal cammino critico del Decoder.

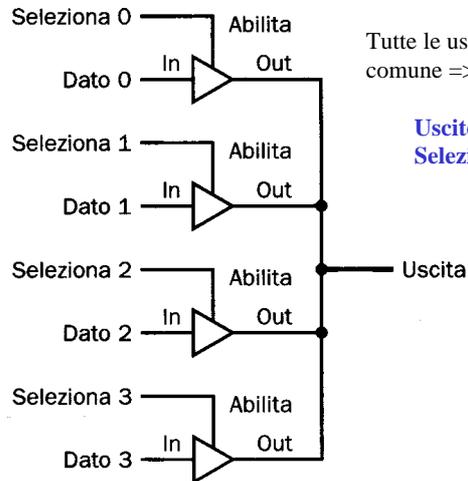
Numero\_registro = selettore.



Selezione – #registro  
 Lettura .- sempre disponibile in uscita  
 (dopo tempo di commutazione del MUX)  
 Scrittura – segnale esplicito (in AND con il clock in caso di cella sincrona).

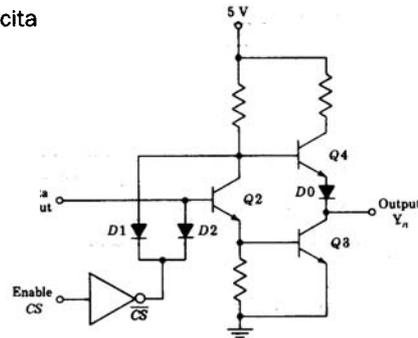


## Memoria three-state

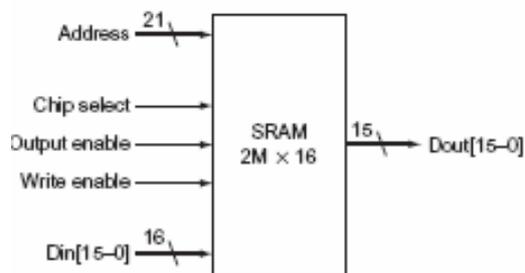


Tutte le uscite delle celle sono collegate ad un'uscita comune => E' necessario evitare conflitti fra le uscite.

Uscite "isolate" con porte *three-state*  
 Selezione una sola cella alla volta



## Chip di SRAM



Tempo di accesso:  
 da Address a Dout.

Selezione – indirizzo + Chip select.  
 Scrittura – Write enable.  
 Lettura – Output enable.

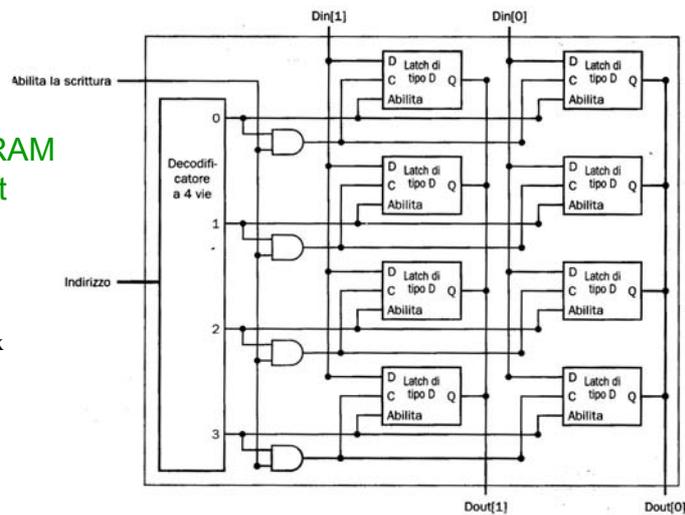


## Esempio di SRAM



Esempio: SRAM  
4 celle x 2 bit

Risparmio il Mux



Problemi con il crescere del numero di linee. Esempio: SRAM 2M x 16.

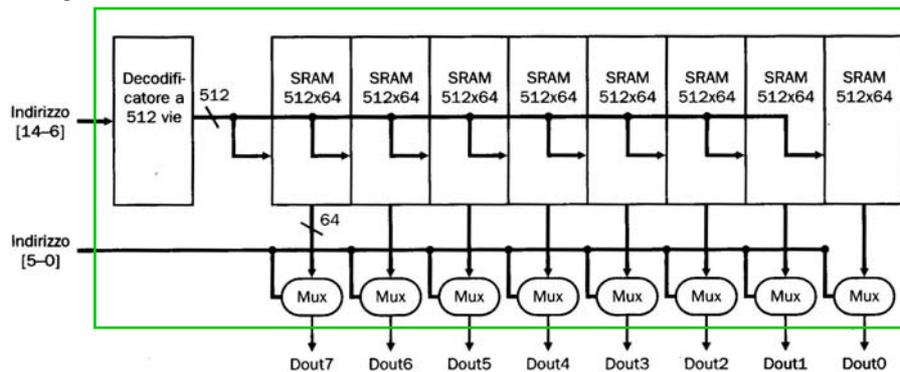
Decodificatore a 21 ( $\log_2 2M$ ) bit e 2M uscite per 2M linee di abilitazione e di selezione (ingresso C) dei bistabili.



## Indirizzamento SRAM a BANCHI



Esempio: SRAM 32K x 8. Trasformo 32K linee in una matrice: 512 linee x 64 colonne (bit)



Il decodificatore sarà a 9 bit ( $\log_2 512$ ) per selezionare una delle 512 linee. Ciascuna linea fornisce 64bit. Ne seleziono uno con il Mux.

Nell'approccio non a banchi avrei avuto bisogno di un decodificatore a 15 bit ( $\log_2 32K$ ).



## Sommario



Gestione delle memorie cache.

SRAM.

**DRAM.**

Correzione degli errori.



## Memorie DRAM



Dynamic RAM. Condensatore che viene caricato.  $\forall$ bit (1 condensatore contro 4-6 SRAM).

Linea di parola (indirizzamento)

Pass transistor  
(transistor di lettura / scrittura)

Condensatore  
(cella di memoria)

Linea di bit  
(linea dato)

Scrittura  
Lettura -> scrittura

1 Pass transistor + 1 condensatore.  
La lettura scarica la memoria che deve essere ricaricata.



## I problemi delle DRAM



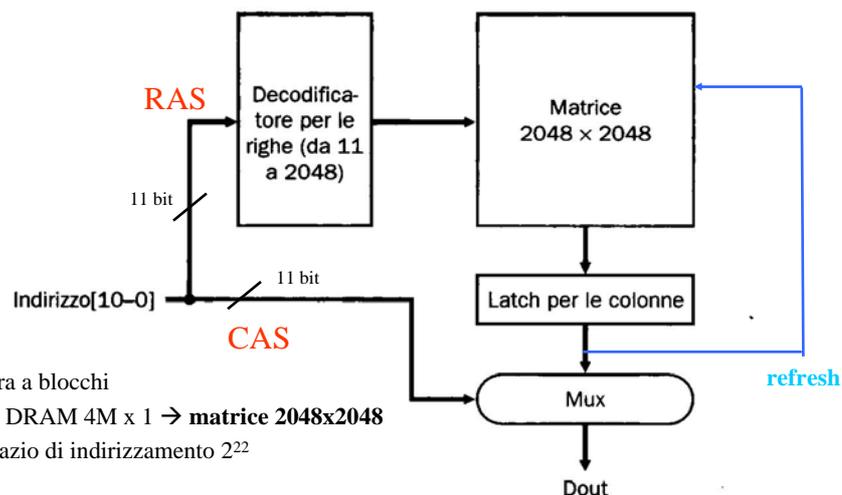
I condensatori .... si scaricano (qualche millisecondo)

**refresh** gestito autonomamente dal controllore della memoria mediante ciclo lettura/scrittura

Cosa leggo/scrivo? Quale/i bit?



## Struttura a 2 livelli (matrice) di una DRAM



- Struttura a blocchi
  - es. DRAM 4M x 1 → matrice 2048x2048
  - Spazio di indirizzamento  $2^{22}$

•Accesso:

selezione riga (RAS) + selezione colonna (CAS)

Efficiente per il refresh (refresh di riga) – (35-70ms, tempo di carica dei condensatori).

Utilizzo per la Memoria Principale.



## Evoluzione delle SRAM e DRAM



Synchronous version.

Trasferimento *a burst* o a pagina: trasferimento consecutivo di parole ad indirizzi consecutivi (e.g. RAM EDO).

### SDRAM

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 (NB il processore può essere la CPU o il controllore della memoria, o altro: il dispositivo che controlla la memoria).

**DDR-SDRAM.** Riescono a trasferire 2 bit per ciclo di clock. Frequenza doppia rispetto alla frequenza del clock del bus.



## Alcuni dati: aprile 2004



<http://www.samsung.com/Products/Semiconductor/>

### SRAM

Low power, 8M x 16, tempi di accesso: 70ns. Disponibilita' Page e Burst.

Sincrone, 1M x 36, 2M x 18, tempi di accesso: 2.6ns. Controllo di parità.

Sincrone, high speed, 1M x 18 o 512K x 36, tempi di accesso: 1.6ns.

Asincrone: 8M x 16, tempi di accesso: 10ns.

### DRAM (DDR)

256M x 4, rate 266Mb/s (133Mhz). Tempo di refresh 30-40ms. (DDR).

### SDRAM (DDR)

128M x 8, rate 266Mb/s (133Mhz).

16M x 16, rate 400Mb/s (200Mhz). 3 clock di latenza, 2-4-8 larghezza del burst.



## Sommario



Gestione delle memorie cache.

SRAM.

DRAM.

Correzione degli errori.



## ECC



- 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 gestirli.
- **Codici rivelatori d'errore**
  - Es: codice di parità.
  - Consente di individuare errori singoli in una parola.
  - Non consente di individuare su quale bit si è verificato l'errore.
- **Codici correttori d'errore (error-correcting codes – ECC)**
  - Consentono anche la correzione degli errori.
  - Richiedono più bit per ogni dato (più ridondanza)
    - Per la correzione di 1 errore per parole e l'individuazione di 2 errori, occorrono 8bit /128 bit.



## Codici rivelatori d'errore



- **Es: 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



- **Es: 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
  - **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  $\rightarrow d = 2$
- **Distanza minima** di un codice,  $d_{MIN}$ 
  - il valor minimo di  $d$  tra tutte le coppie di sequenze di un codice
- **Capacità di rivelazione** di un codice:  $t = d_{MIN} - 1$
- **Capacità di correzione** di un codice:  $r = (d_{MIN} - 1) / 2$
- **Esempi:**
  - Codice a bit di **parità**:  $d_{MIN} = 2 \rightarrow t = 1, r = 0$
  - Codice a **ripetizione (3,1)**:  $d_{MIN} = 3 \rightarrow t = 2, r = 1$



## Applicazioni nelle memorie



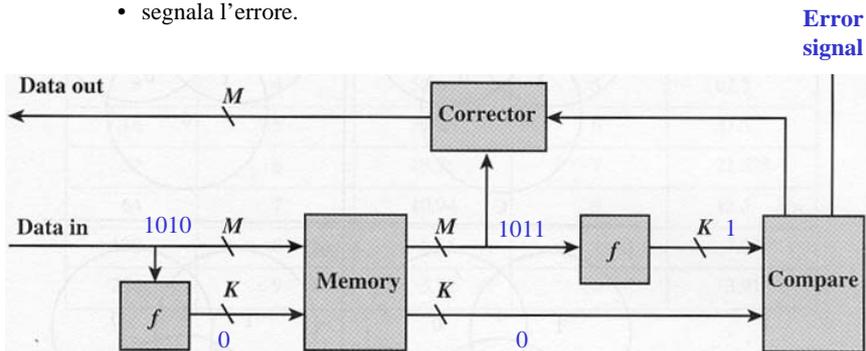
- **RAM con controllo di parità**
  - Aggiungo un bit di parità ad ogni byte
  - Es: RAM 1 M x 9 bit (8+1)
- **RAM con codice correttore di errori (ECC)**
  - si usa nelle memorie cache
  - codici ECC evoluti (alta efficienza)
    - Hamming, CCITT-32, Reed Solomon, ...



## Correzione degli Errori



- **OUT possibili:**
  - No errors detected
    - I dati letti possono essere inviati in uscita così come sono.
  - 1 errore è stato 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
    - segnala l'errore.



## Dimensione di codici ECC



- Conviene applicare ECC a parole più lunghe possibile → aggiungo meno ridondanza → maggiore efficienza del codice
  - A costo di complessità maggiori di codifica/decodifica

Data Bits	Single-Error Correction		Single-Error Correction/ Double-Error Detection	
	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



## Sommario



Gestione delle memorie cache.

SRAM.

DRAM.

Correzione degli errori.