

Memoria centrale a semiconduttore

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

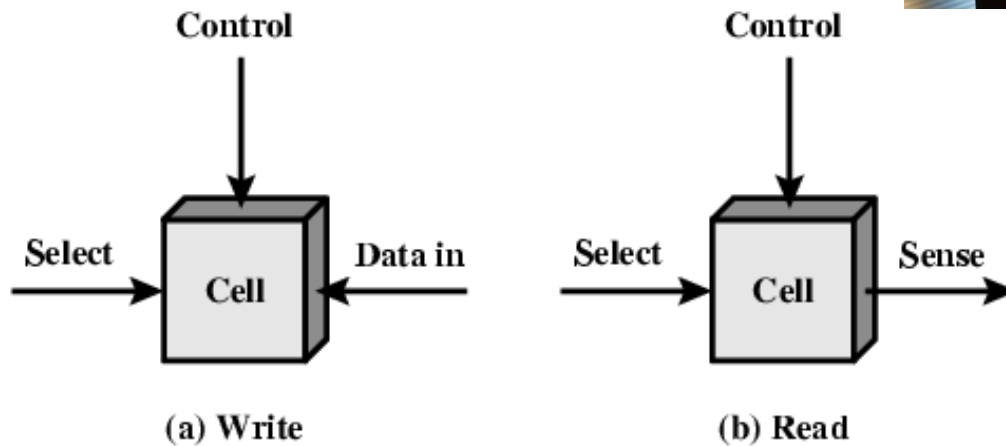
Memorie a semiconduttore



■ RAM

- ☐ Accesso casuale
- ☐ Read/Write
- ☐ Volatile
- ☐ Memorizzazione temporanea
- ☐ Statica o dinamica

Operazioni cella memoria

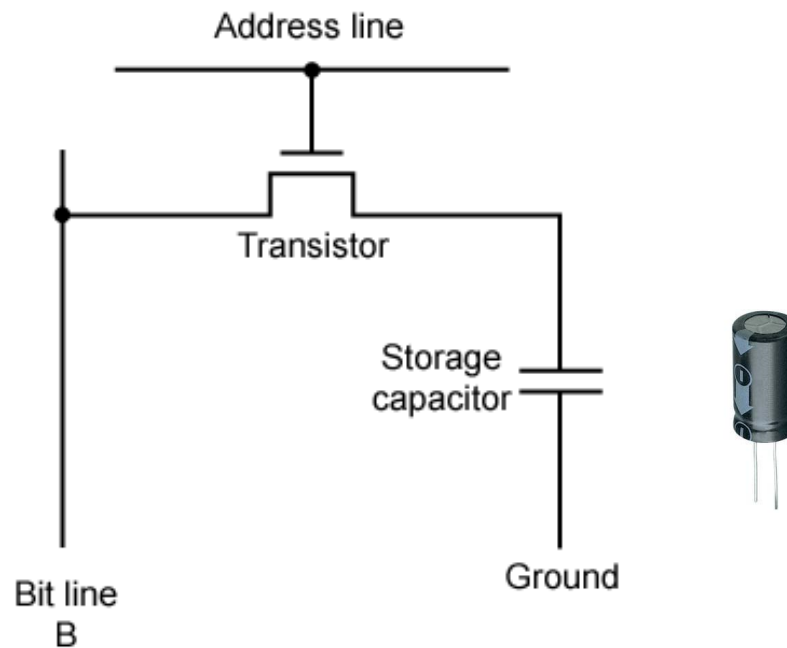


RAM Dinamiche (Dynamic RAM)



- Bit memorizzati come cariche in condensatori
- Decadimento delle cariche con il tempo
- Necessitano di refresh delle cariche, anche durante l'alimentazione
- Costruzione più semplice
- Un condensatore per bit
- Meno costose
- Necessitano di circuiti per il refresh
- Più lente
- Usate per la memoria principale
- In essenza operano in modo analogico
 - il livello di carica determina il valore digitale

Struttura RAM Dinamiche



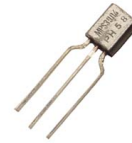
Funzionamento DRAM

- Linea indirizzo attivata quando si deve scrivere o leggere un bit
 - ☐ Transistor “chiuso” (la corrente fluisce)
- Write
 - ☐ Si applica tensione alla linea di bit
 - Tensione alta indica valore 1; tensione bassa indica valore 0
 - ☐ Poi si applica un segnale alla linea indirizzo
 - Trasferisce la carica al condensatore
- Read
 - ☐ Si seleziona la linea indirizzo
 - transistor si accende
 - ☐ La carica del condensatore fluisce attraverso la linea di bit verso un amplificatore
 - Valore di carica comparato con un segnale di riferimento per stabilire se vale 0 o 1
 - ☐ La carica del condensatore deve essere ristabilita (refresh)

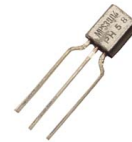
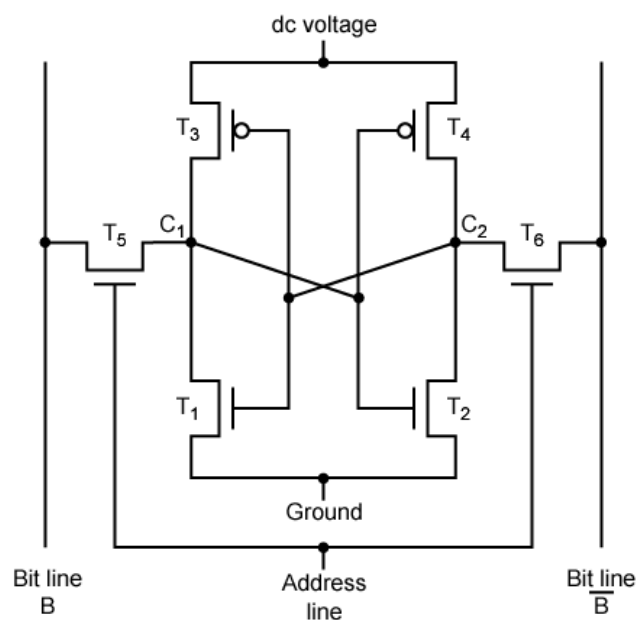


RAM Statica

- Bit memorizzati tramite porte logiche
- Nessuna perdita di carica
- Nessuna necessità di refresh
- Costruzione più complessa
- Più elementi per bit
- Più costosa
- Non ha bisogno di circuiti di refresh
- Più veloci
- Usate per la cache
- Digitale
 - usa flip-flop

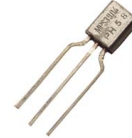


Struttura RAM Statica



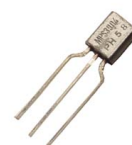
Funzionamento RAM Statica

- La disposizione dei transistor garantisce stati stabili
- Stato 1
 - C_1 alto, C_2 basso
 - T_1 T_4 “spenti”, T_2 T_3 “accesi”,
- Stato 0
 - C_2 alto, C_1 basso
 - T_2 T_3 “spenti”, T_1 T_4 “accesi”,
- La linea indirizzo controlla i transistor T_5 T_6 (accesi con presenza di segnale)
- Write – si applica il valore da scrivere alla linea B ed il complemento del valore alla linea \bar{B}
- Read – il valore viene letto tramite la linea B



SRAM e DRAM a confronto

- Entrambe sono volatili
 - Alimentazione necessaria per preservare i dati
- celle dinamiche
 - Più semplici da costruire, più piccole
 - Più dense
 - Meno costose
 - Necessitano di refresh
 - Unità di memoria più capienti
- celle statiche
 - Più veloci
 - Cache





Read Only Memory (ROM)

- Memorizzazione permanente
 - ☐ Non volatili
- Usate per memorizzare:
 - ☐ microprogrammi
 - ☐ subroutine di libreria
 - ☐ programmi di sistema (BIOS)
 - ☐ funzioni tabulate



Tipi di ROM

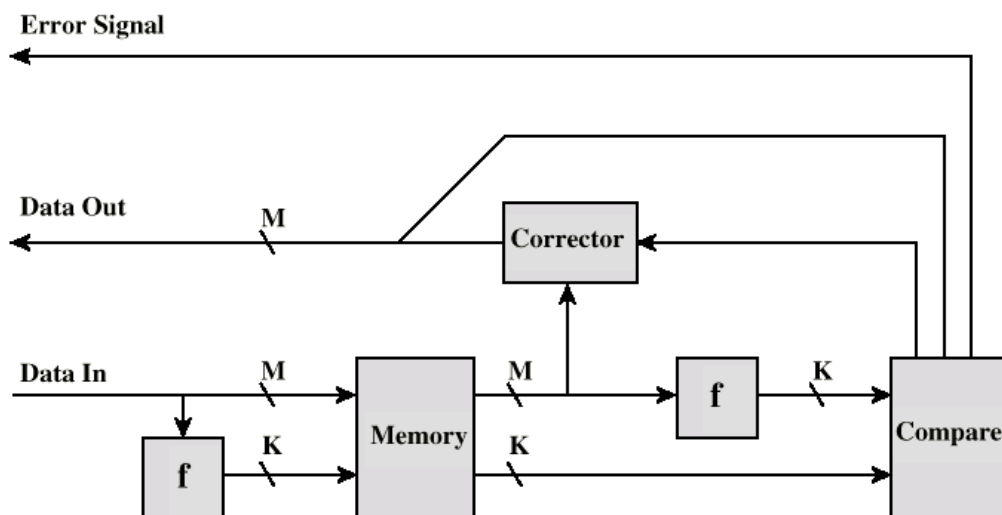
- Scritte in produzione
 - ☐ Molto costoso per pochi “pezzi”
- Programmabili (una sola volta)
 - ☐ PROM
 - ☐ Necessitano di strumentazione speciale per la programmazione
- Principalmente di lettura (Read “mostly”)
 - ☐ Erasable Programmable (EPROM)
 - Si cancellano (per intero) tramite raggi ultravioletti
 - ☐ Electrically Erasable (EEPROM)
 - Impiegano molto più tempo per la scrittura che per la lettura
 - ☐ Memorie Flash
 - Cancellazione elettrica di blocchi di memoria

Correzione Errori

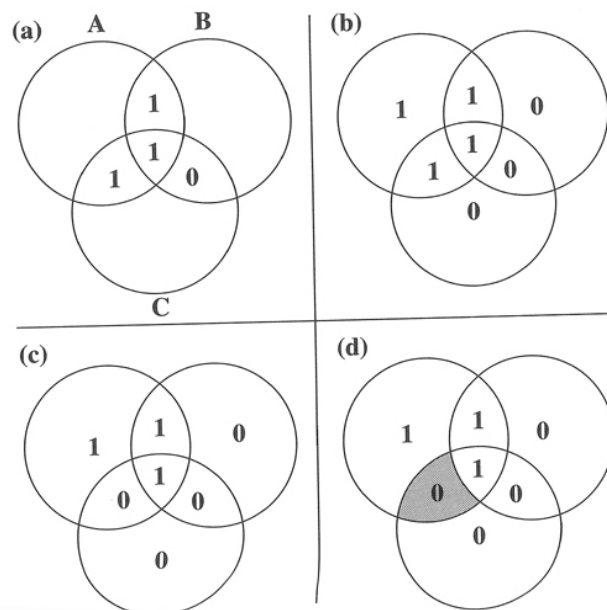


- Guasti Hardware (Hard Failure)
 - Guasti permanenti
- Errori Software (Soft Error)
 - Random, non-distruttivi
 - Danni alla memoria non permanenti
- Errori rilevati ed eventualmente corretti usando, ad esempio, codici correttori di Hamming

Schema di funzionamento del codice a correzione di errore



Esempio di codice a correzione di errore di Hamming



Correzione degli errori: disposizione bit

Quanti bit di controllo servono ? $2^K - 1 \geq M + K$

Bit di dati	Bit di controllo	% incremento
8	4	50
16	5	31,25
32	6	18,75
64	7	10,94

Bit Position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data Bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check Bit					C8				C4		C2	C1

Codice correzione Hamming

Esempio generazione bit di controllo

12	11	10	9	8	7	6	5	4	3	2	1
1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
D8	D7	D6	D5		D4	D3	D2		D1		
				C8				C4		C2	C1
1	1	0	0		0	0	1		0		

$$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$$

$$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$$

$$C4 = D2 \oplus D3 \oplus D4 \oplus D8$$

$$C8 = D5 \oplus D6 \oplus D7 \oplus D8$$

si ha

$$C1 = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0$$

$$C2 = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

$$C4 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$$

$$C8 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$$

Correzione degli errori: disposizione bit

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1
Word stored as	0	0	1	1	0	1	0	0	1	1	1	1
Word fetched as	0	0	1	1	0	1	1	0	1	1	1	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Check Bit					0			0		0	0	1

risultato XOR 0

1

1

0

Correzione degli errori: SEC-DED

correzione di singolo errore, rilevazione di 2 errori

