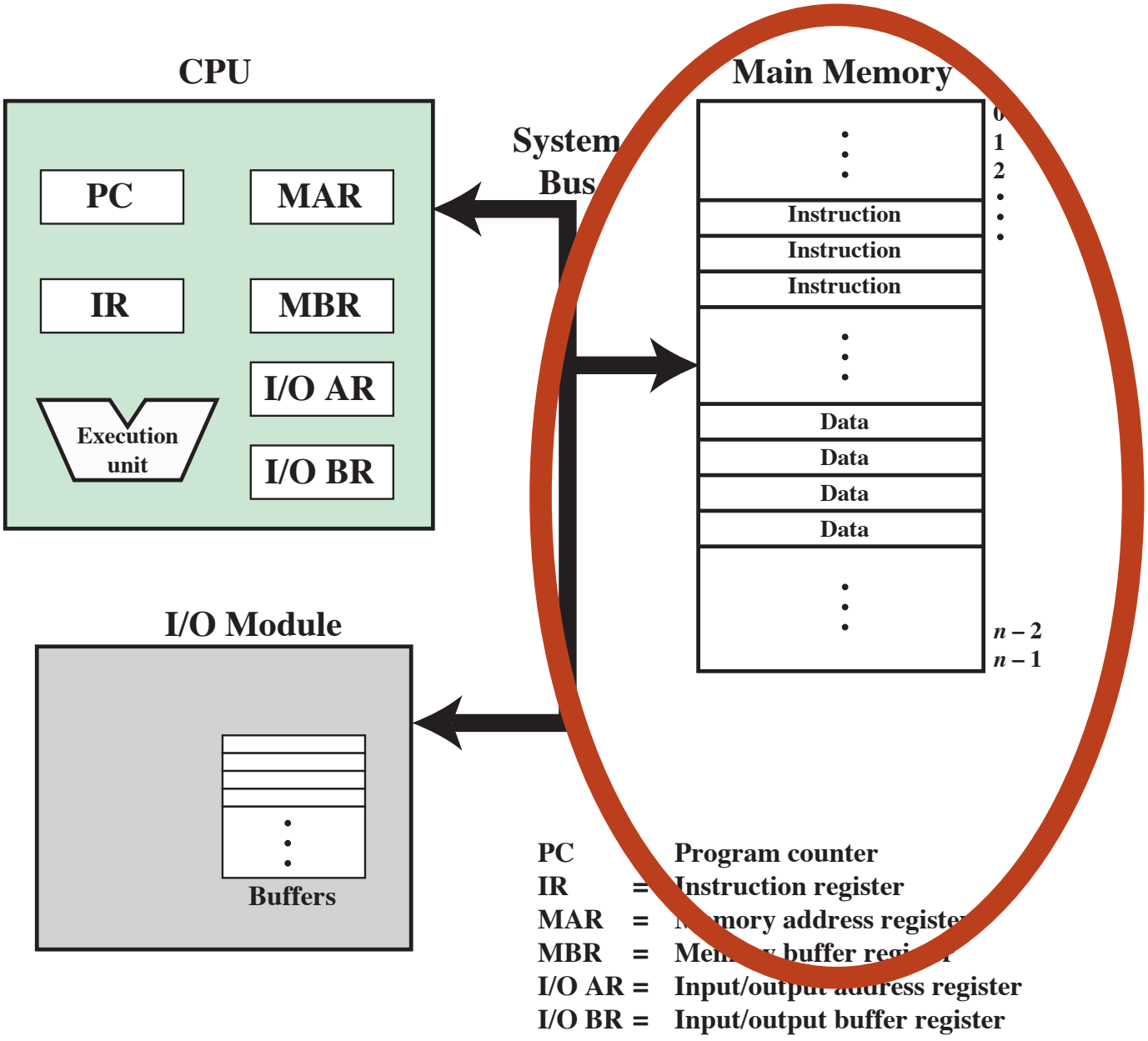
Memoria RAM

Argomento:

- Tipi di memorie
- Organizzazione della memoria

Materiale di studio

- Capitolo 5.1


CPU
PC
MAR
IR
MBR
I/O AR
Execution unit
I/O BR
System Bus
Main Memory
0
1
2
Instruction
Instruction
Instruction
Data
Data
Data
Data
n – 2
n – 1
I/O Module
Buffers
PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

Caratteristiche della memoria centrale

- La memoria è divisa in **locazioni**
- Ogni locazione è individuata da un indirizzo
- Con un indirizzo da **n bit** sono indirizzabili un totale di **2^n locazioni** (nota: 2^{10} = 1024 = 1K, 2^{20} = 1M, 2^{30} = 1G)
- Ogni locazione contiene **L** bits
  - In genere L = 8 bits = 1 byte, ma in alcuni casi L = 32 bits.
- Una memoria può essere vista come un vettore di **$N=2^n$** elementi, dove ogni elemento occupa **L** bytes

Word

La parola **word** è usata per denotare l'insieme di bit/byte che compongono un dato in un'architettura

La dimensione di una word è variabile

- Word è stato usato per indicare 16 bit (2 byte), 32 bit (4 byte), 64 bit (8 byte), …

Durante il corso, assumiamo che la dimensione di una word sia **32 bit (4 byte)**

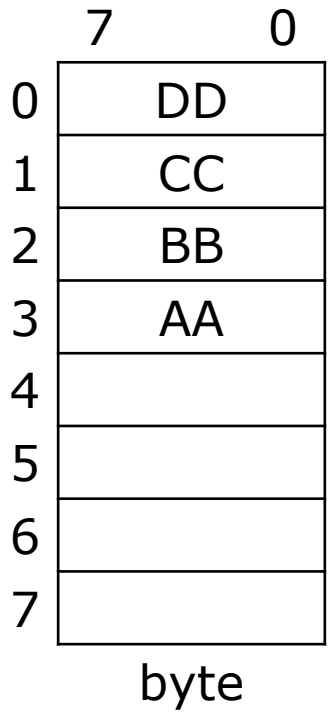
- Halfword: 2 byte
- Doubleword: 8 byte

Ordinamento dei byte in memoria

Problema: se la memoria è organizzata in byte (1 locazione = 1 byte), come salvo una word (4 byte) in memoria?

Word: 0xAABBCCDD

(Il problema esiste anche con halfwords e doublewords)



Little Endian

Big Endian

Big e little endian

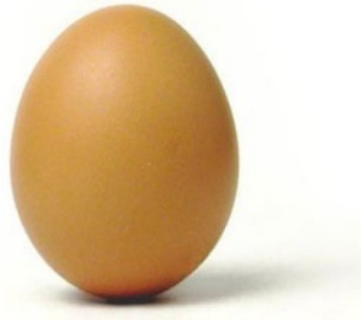
- **Big-endian** è l'ordine per cui la parte più significativa (BIG END) viene memorizzata per prima (all'indirizzo più basso di memoria).

- **Little-endian** è l'ordine per cui la parte meno significativa (LITTLE END) viene memorizzata per prima.

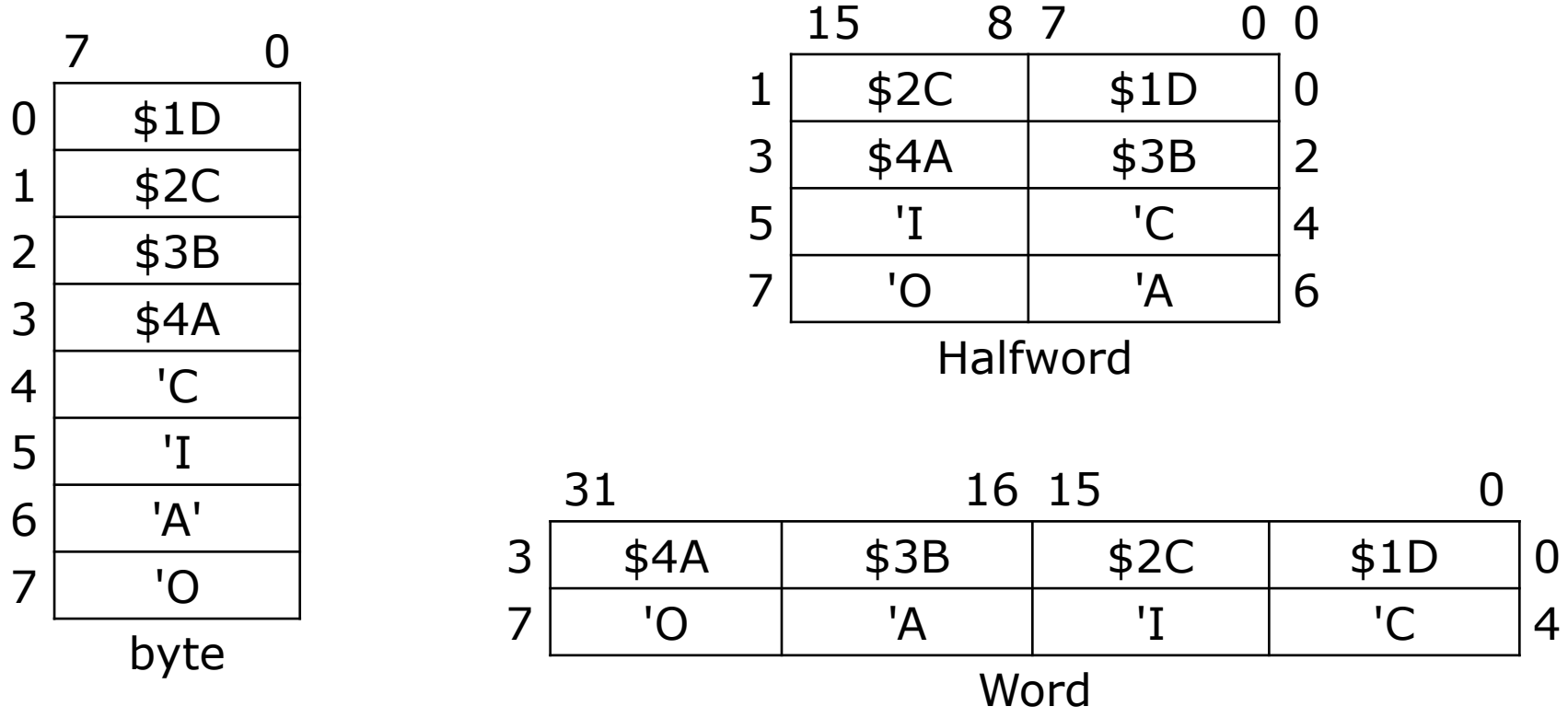
- Le due alternative sono entrambe utilizzabili e utilizzate.

Big e little endian (2)

- Nomi tratti dal libro "I viaggi di Gulliver" (Jonathan Swift)
- I Big Endians erano una fazione conservatrice che rompeva le uova sode dalla parte più larga del guscio, in contrapposizione al re dei Lillipuziani che richiedeva ai suoi sudditi (i Little Endians) di aprire le uova dalla punta.

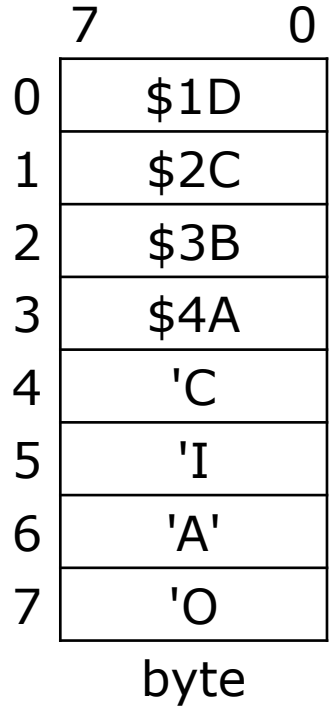


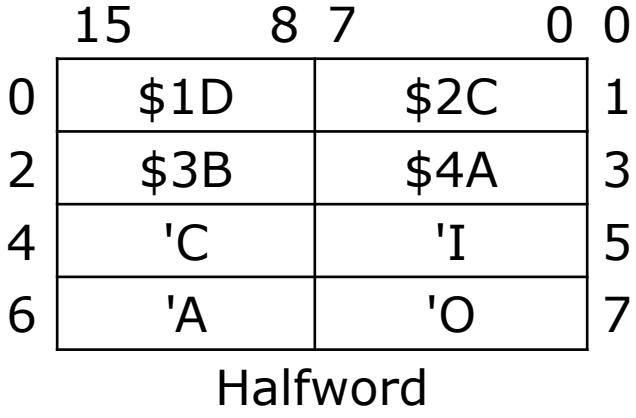
Ordinamento dei byte: little endian

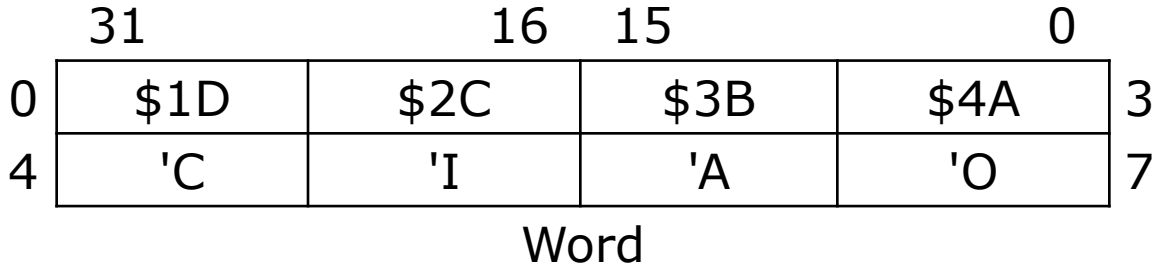


Organizzazione little endian: numero $4A3B2C1D e stringa "CIAO"

Ordinamento dei byte: big endian







Organizzazione big endian: numero $1D2C3B4A e stringa “CIAO”

Ordinamento dei byte: little o big endian?

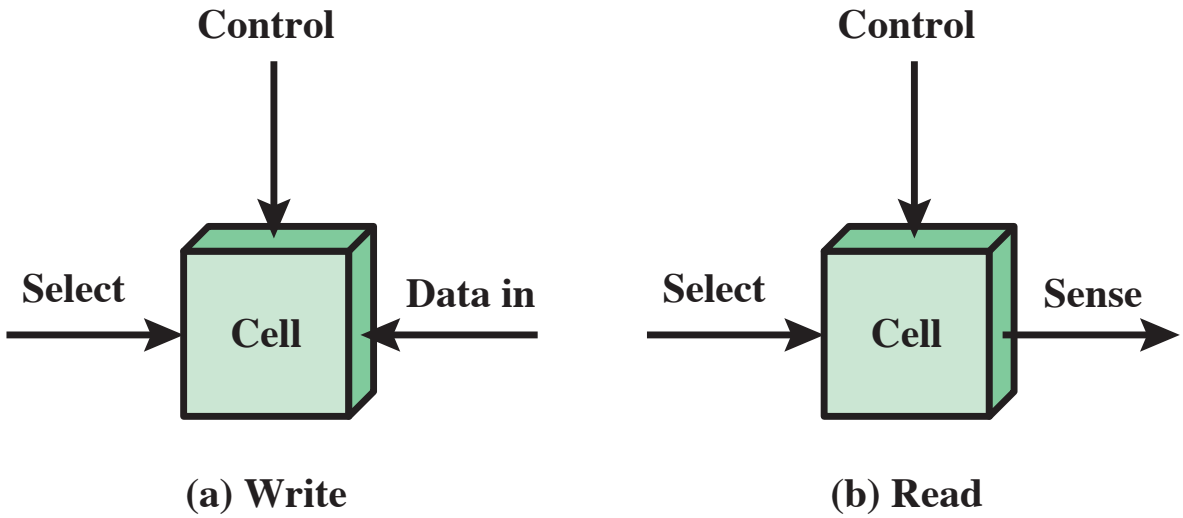
- I computer storici (IBM370), molti RISC e i processori Motorola usano il metodo big-endian.
- I processori INTEL adottano l'ordinamento little-endian, ritenuto più conveniente nella trasmissione dei dati, ove è trasmessa per prima la parte meno significativa.

- Il PowerPC e l'**ARM** possono funzionare in tutte e due le modalità.

- Anche nei formati grafici vi sono scelte diverse:
  - GIF → Little Endian
  - JPEG → Big Endian

Memoria

- Assume due stati: 0 e 1
- Può essere scritta (almeno una volta) per impostare il valore
- Può essere letta più volte



Tipi di memoria

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)				
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level	Electrically	
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

Read Only Memory (ROM)

- Il contenuto della memoria è permanente e non può essere cambiato
- Non è richiesta alimentazione per mantenere il valore in memoria
- I dati sono cablati nel chip durante il processo di fabbricazione
- Svantaggi:
  - Non ci possono essere errori perché non si possono correggere
  - Inserimento dati costoso

Programmable ROM (PROM)

- Il contenuto della memoria è permanente e non può essere cambiato
- La scrittura può essere effettuata dopo il processo di fabbricazione del chip tramite un'opportuna apparecchiatura elettrica.
- Non è richiesta alimentazione per mantenere il valore in memoria
- Maggior flessibilità e minor costo rispetto la ROM

Memorie persistenti con scrittura

- Le memorie persistenti permettono la modifica del loro contenuto
- Erasable programmable read-only memory (EPROM)
  - La cancellazione dell'intera memoria avviene tramite radiazione ultravioletta
- Electrically erasable programmable read-only memory (EEPROM)
  - E' possibile cancellare solo singoli byte tramite segnali elettrici
  - Più costosa dell'EPROM e con densità minore (meno bit per chip): ogni bit richiede due transistor

Memorie persistenti con scrittura (2)

- Flash Memory
  - Posizione intermedia tra EPROM e EEPROM per costi e funzionalità
  - La cancellazione avviene per settori
  - La densità è paragonabile a quelle di una EPROM: ogni bit richiede un transistor

RAM: random access memory

- Tutte le memorie permettono l'accesso casuale alle locazioni di memoria
  - Si contrappone alle memorie ad accesso sequenziale (nastri)
- Con RAM si denota tradizionalmente una memoria volatile in cui sia possibile sia leggere che scrivere efficientemente
- Memorie RAM principali:
  - **Dynamic RAM (DRAM)**
  - **Static RAM (SRAM)**

Dynamic RAM (DRAM)

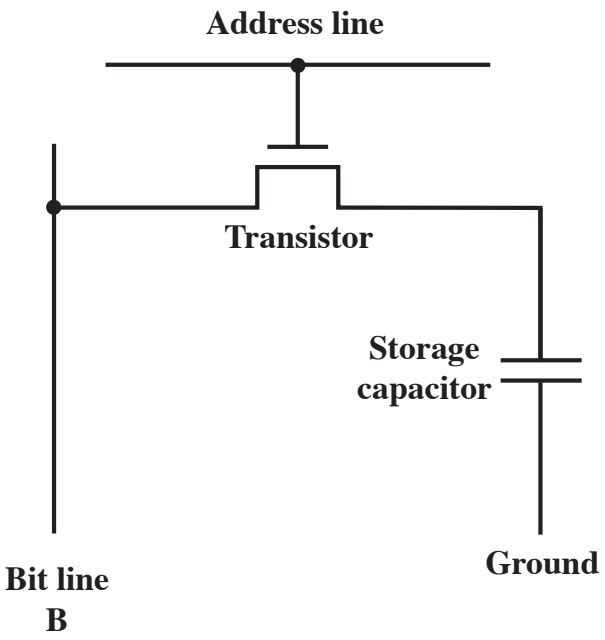
- Il valore di un bit viene rappresentato tramite la presenza o meno di una carica in un condensatore
- Richiede un refresh periodico per mantenere il valore
  - Un condensatore carico tende a disperdere la carica
  - Refresh: il dato viene letto e poi riscritto
- Alta densità, limitata velocità di accesso

Static RAM (SRAM)

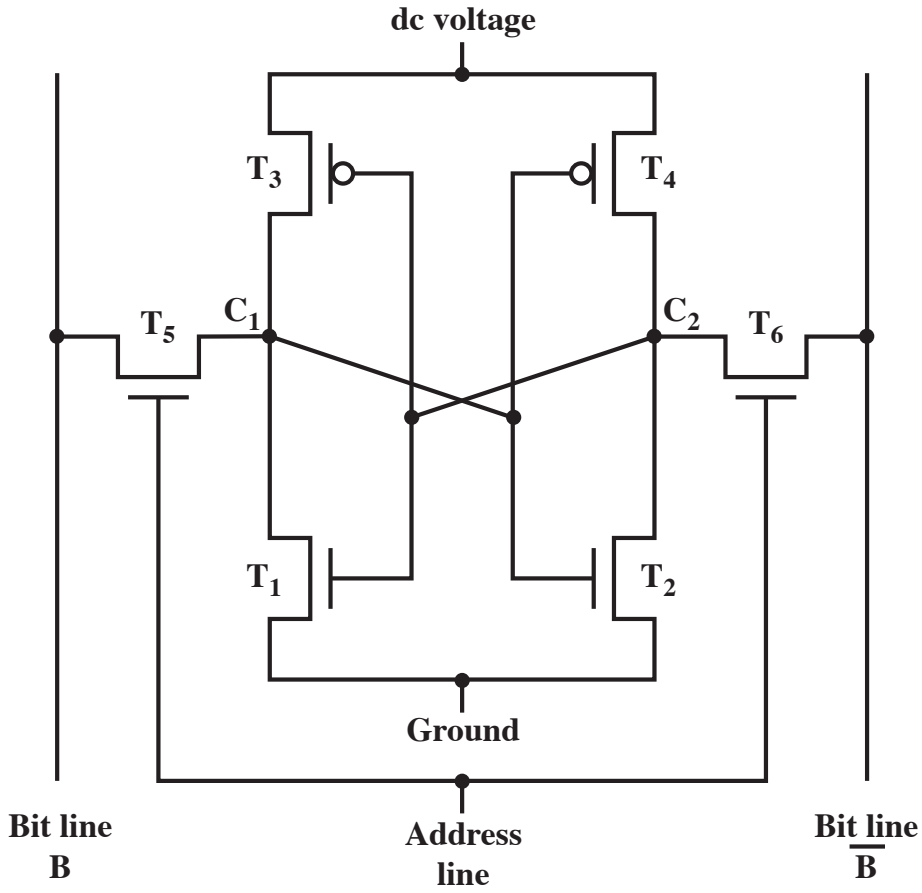
- Il valore di un bit viene salvato tramite flip-flop
- Mantengono il valore finché è presente l'alimentazione

- Bassa densità, alta velocità di accesso

DRAM vs SRAM



(a) Dynamic RAM (DRAM) cell



(b) Static RAM (SRAM) cell

DRAM vs SRAM (2)

DRAM

- Memoria volatile
- Facili da costruire, basso costo
- Alta densità: 2 componenti per bit
- Limitata velocità di accesso
- Richiede refresh
- Utilizzate come memoria principale

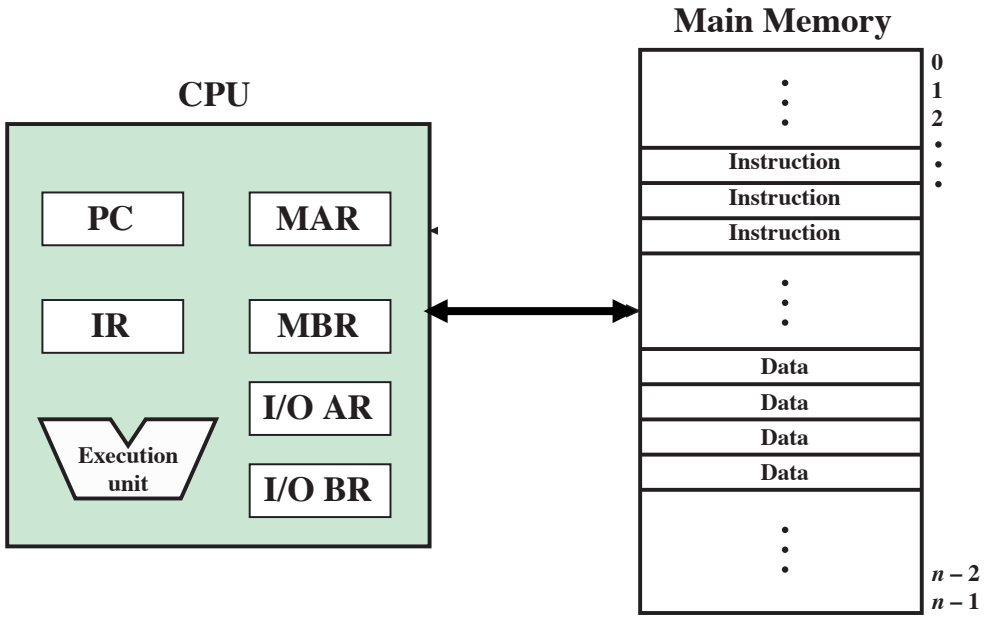
SRAM

- Memoria volatile
- Alto costo
- Bassa densità: 6 componenti per bit
- Alta velocità di accesso
- Non richiede refresh
- Utilizzata come cache

Tempi di accesso

- Memorie statiche (SRAM):
  - 2-3 ns
- Memorie dinamiche (DRAM):
  - 20-35 ns
  - + latenza: per il primo dato ci vuole un tempo 4-5 volte quello per i dati successivi

Lettura/scrittura di un dato in memoria



Organizzazione della memoria

Una memoria RAM con N locazioni e L bit per locazione può essere implementata in molti modi tramite una struttura gerarchica

- Al livello più basso troviamo i **chip di memoria** che consistono di N_1 locazioni e L_1 bit per locazione
  - $N_1 < N$ e $L_1 < L$
- I chip di memoria vengono uniti in un **modulo** per ottenere N_1 locazioni e L bit per locazione
  - si aumenta la dimensione di una locazione
- I moduli vengono uniti in **banchi** per ottenere N locazioni e L bit per locazione
  - Si aumenta il numero di locazioni

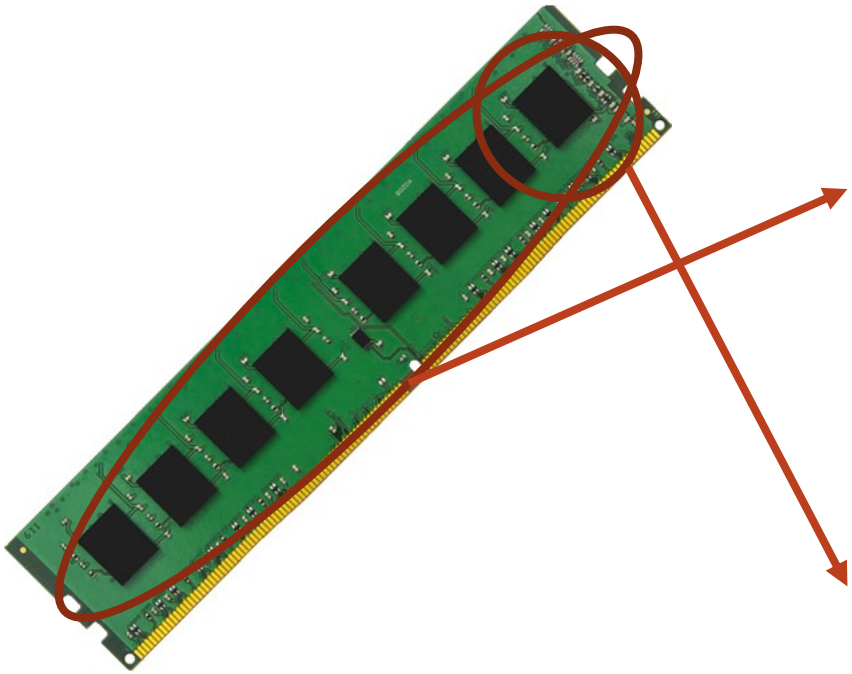
Esempio

Come implementare una memoria RAM di 16 GB con $2G=2^{31}$ locazioni da 64 bit (8 byte)?

- L'unità base è dato da un chip con 1G locazioni da 8 bit
- Uniamo 8 chip in un modulo per ottenere 1G locazioni da 8*8 = 64 bit (8 byte)
- Uniamo 2 moduli per ottenere 2G locazioni da 8 byte

Esempio (2)

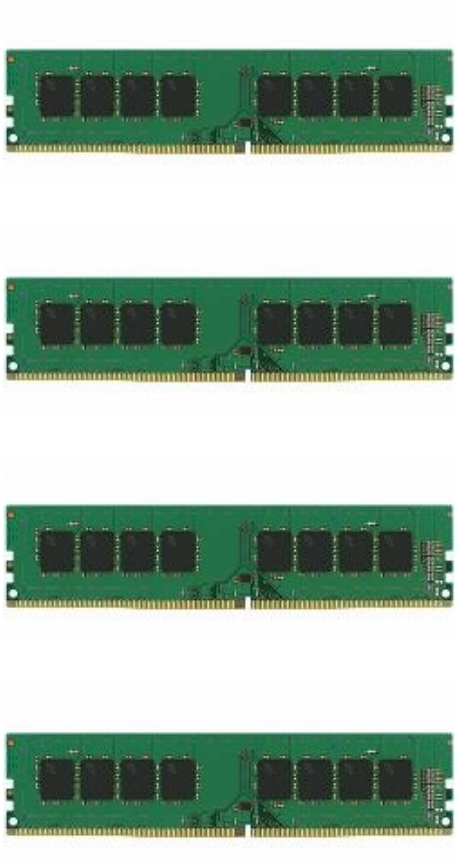
In negozio comprate un banco di memoria RAM da 16 GB: 2^{31} locazioni da 64 bit (in foto Micron 16GB, DDR4, 2133 MHz memory module)



Il banco consiste di due moduli (uno per lato): ogni modulo consiste di 2^{30} locazioni da 64 bit

Ogni modulo consiste di 8 chip: ogni chip consiste di 2^{30} locazioni da 8 bit

Esempio (3)



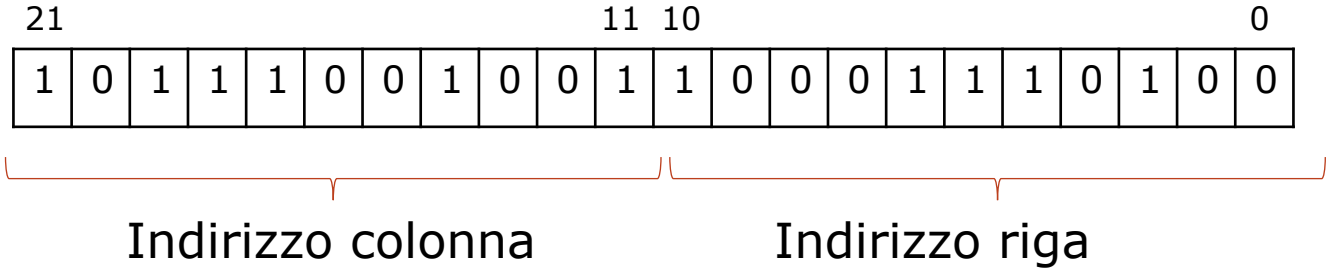
Se volete estendere a 64 GB la memoria RAM del vostro computer, comprate 4 componenti

- Avete esteso il banco di memoria

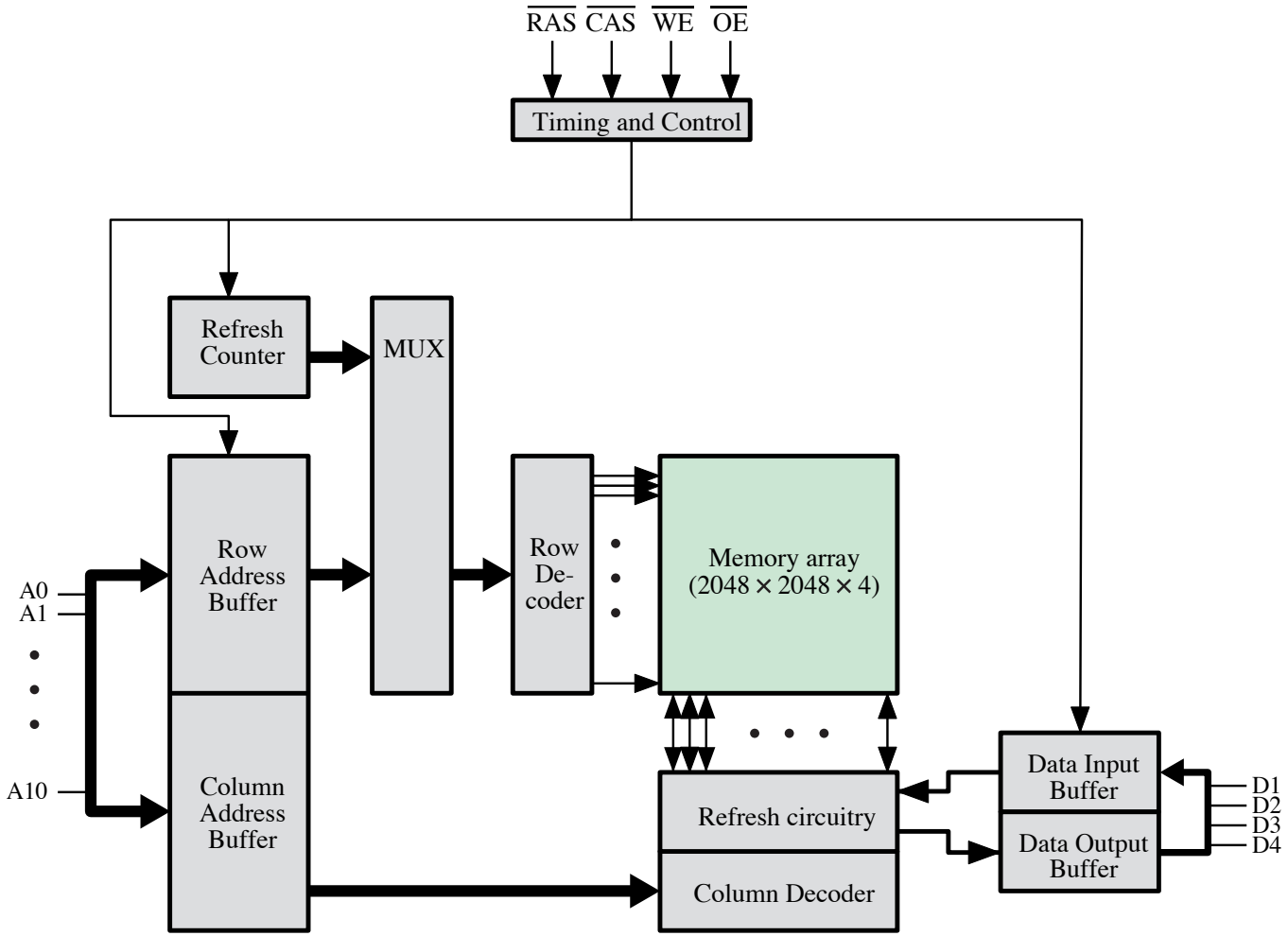
Organizzazione DRAM

- Come organizzare un chip 4M x 4 (2^{22} locazioni da 4 bit)?

- Servono 22 bit per indicare una locazione
- Posizioniamo le locazioni in una matrice 2^{11} x 2^{11}
  - Ogni elemento della matrice contiene 4 bit
  - Servono 11 bit per indicare la riga e 11 bit per indicare una colonna



Organizzazione DRAM (2)



Organizzazione DRAM (3)

- I 22 bit di indirizzo non vengono trasferiti contemporaneamente ma in due parti
  - Prima l'indirizzo di riga, attivando il segnale RAS (row address select)
  - Seguito dall'indirizzo di colonna, attivando il segnale CAS (column address select)
- Questa tecnica permette di ridurre il numero di pin del chip
  - Aggiungendo un pin la memoria aumenta di un fattore 4
- I segnali WE (write enable) e OE (output enable) denotano una scrittura o una lettura

Refresh della memoria

DRAM 64Kx1 (256 righe x 256 colonne)

Periodo di refresh di ciascun bit = 4 ms

t_a = 60ns

Se si operasse il refresh un bit alla volta:

t_r = 4 ms / ($64 \cdot 2^{10}$) ≈ **61 ns**

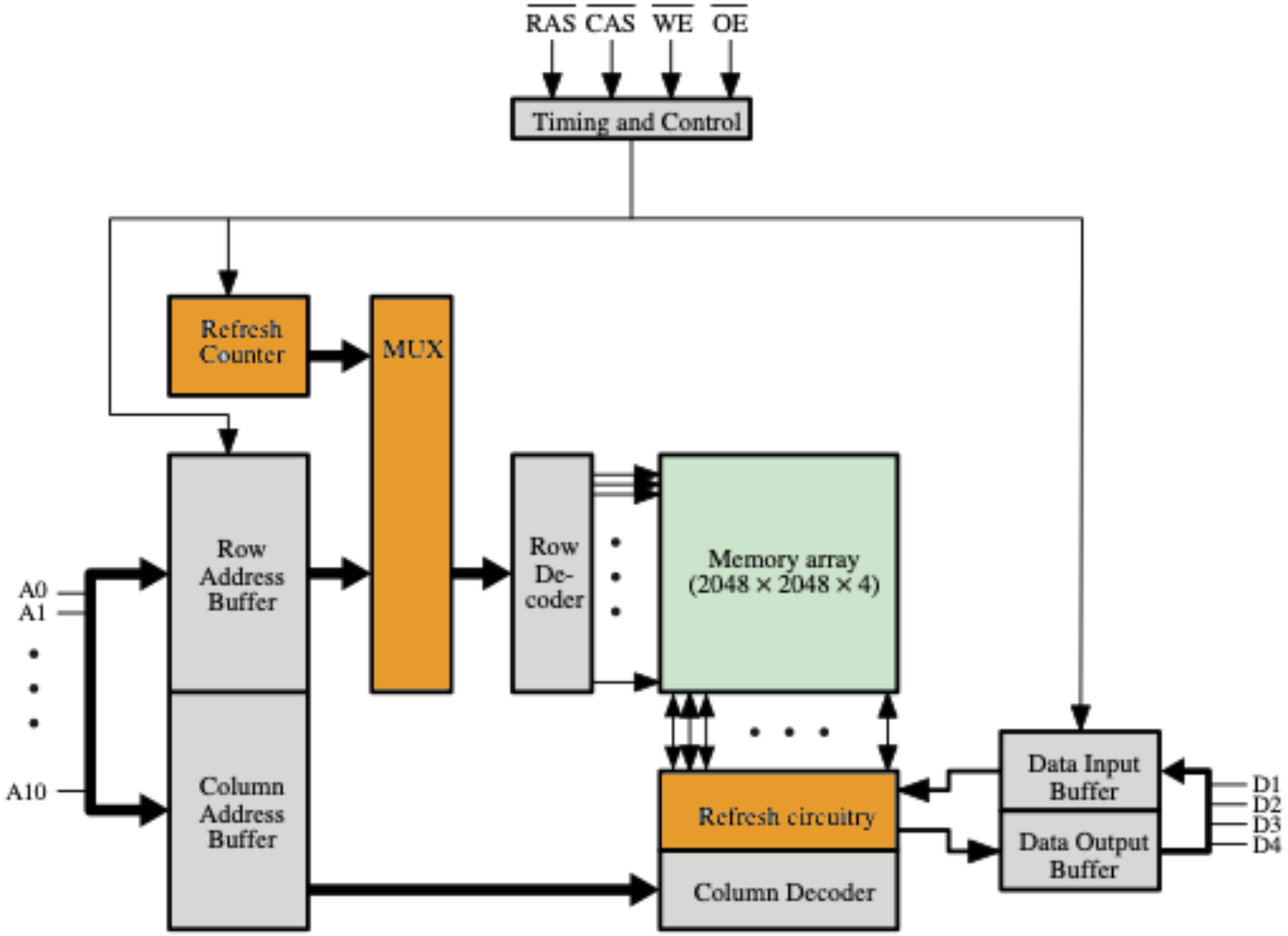
l'impegno percentuale sarebbe ≈ **100%**

Refresh di un'intera riga alla volta:

t_r = 4 ms / 256 ≈ **16 μs**

impegno percentuale=(60 ns/16 μs)*100 ≈ **0.4%**

Refresh della memoria (2)

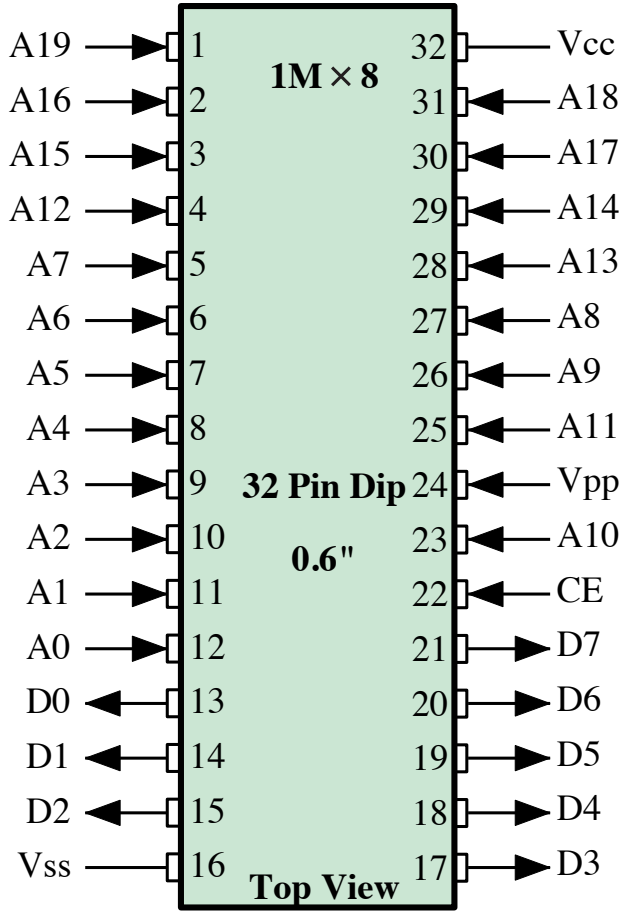


Organizzazione SRAM

E' equivalente all'organizzazione di una memoria DRAM ma senza le componenti per il refresh

PIN di un chip 1Mx8 (EPROM)

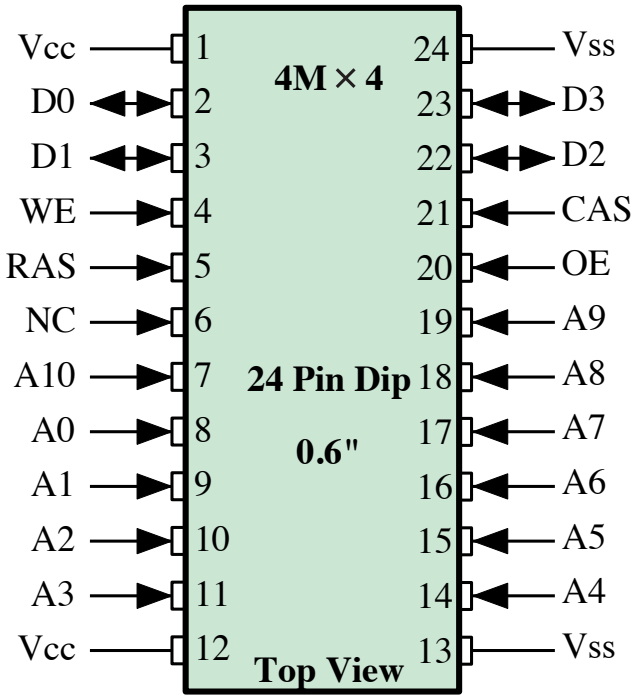
- D0…D7: 8 bit per il dato
- A0...A19: 20 bit per l'indirizzo. Indirizzi riga (10 bit) e colonna (10 bit) inviati contemporaneamente
- Vcc, Vss, Vpp: pin per funzionamento chip



(a) 8 Mbit EPROM

PIN di un chip 4Mx4 (DRAM)

- D0…D3: 4 bit per il dato
- A0...A10: 11 bit per l'indirizzo. Indirizzi riga (11 bit) e colonna (11 bit) inviati in due istanti distinti
- RAS, CAS pin per indicare indirizzo riga/colonna
- WE, OW: pin per indicare se lettura o scrittura
- Vcc, Vss: pin per funzionamento chip

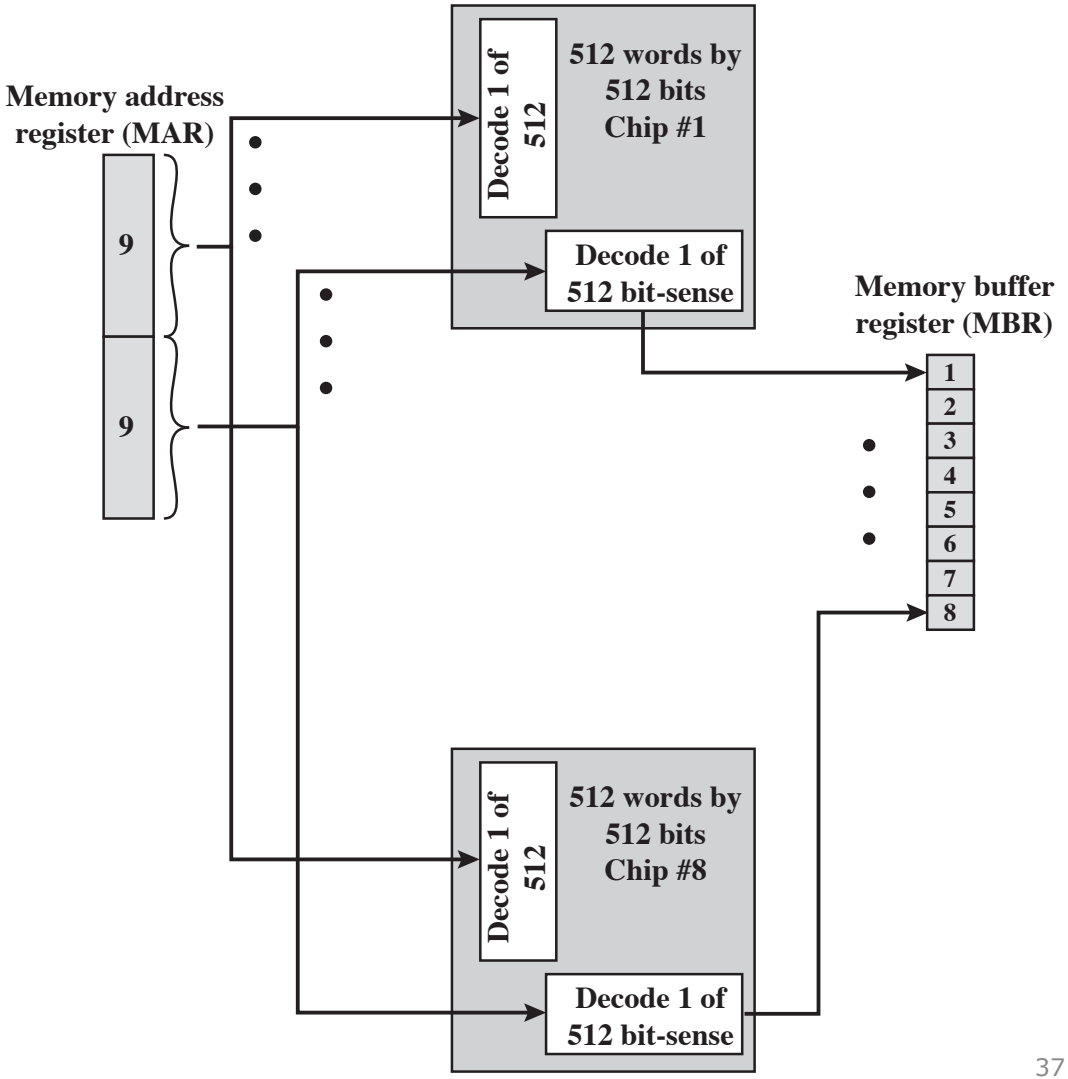


(b) 16 Mbit DRAM

Organizzazione in moduli

Come ottenere una memoria da 256K 8-bit da chip con 256K 1-bit

L'indirizzo viene propagato a tutti i chip

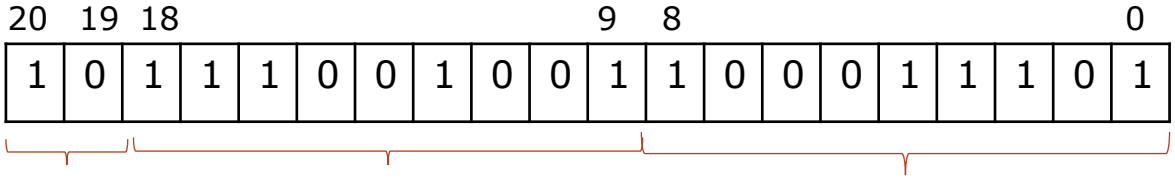


Organizzazione in moduli e banchi

Come ottenere una memoria da 1M 8-bit da chip con 256K 2-bit?

- Prima costruisco moduli 256k 8-bit unendo 4 chip
- Unisco 4 moduli in un banco per ottenere una memoria 1M 8-bit

L'indirizzo consiste di tre parti: indirizzo modulo, indirizzo riga, indirizzo colonna



Organizzazione in moduli e banchi (2)

Esempio dal libro: 1MByte realizzato con 4 moduli, ciascuno costituito da 8 chip del tipo 256 x 1