



---

# Il Livello Logico-Digitale

## Blocchi funzionali sequenziali

22 -10 -2015

---



# Libreria di blocchi sequenziali

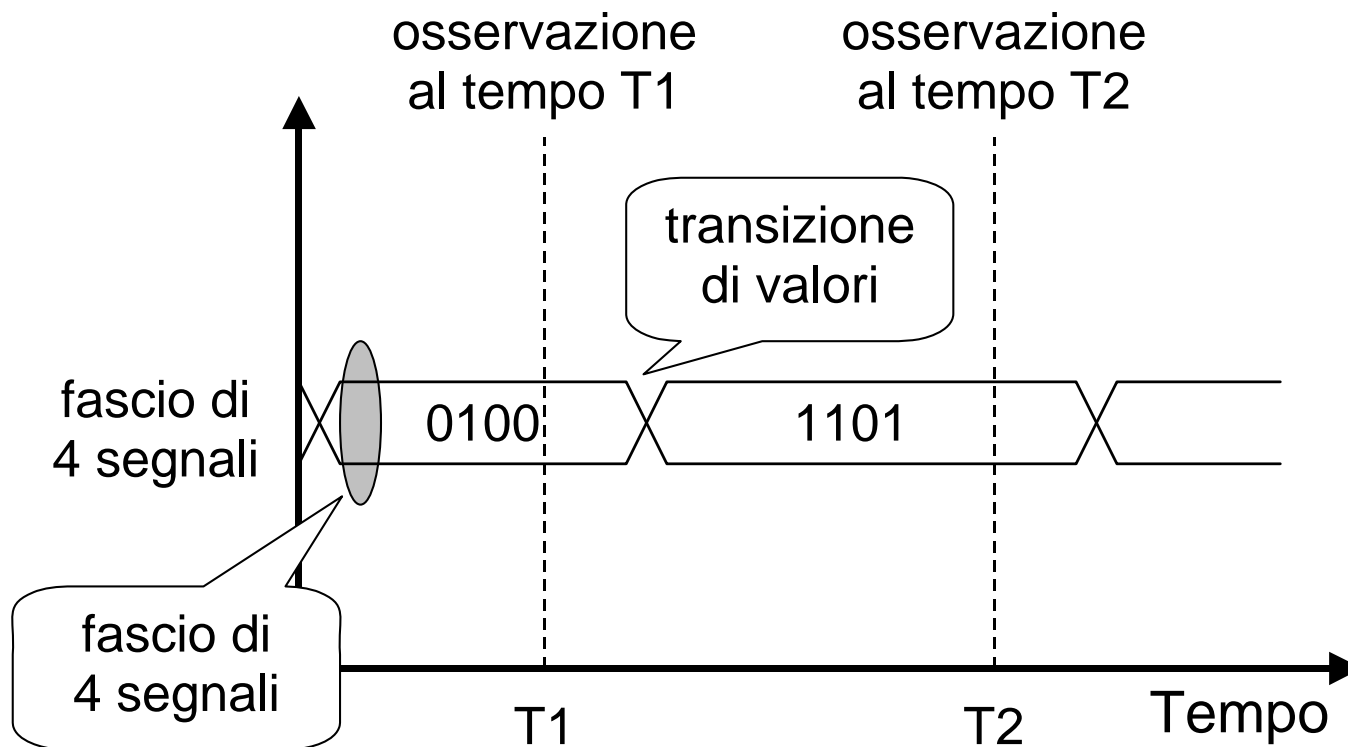
---

- Tipici principali componenti sequenziali di libreria:
  - Registro parallelo
  - Registro a scorrimento
  - Banco di registri
  - Memoria
- Ognuno di questi blocchi ammette numerose versioni e varianti



## Ancora diagramma temporale

Come rappresentare un **fascio di segnali**



- Al tempo T1 i 4 segnali valgono 0100, al tempo T2 i 4 segnali valgono 1101



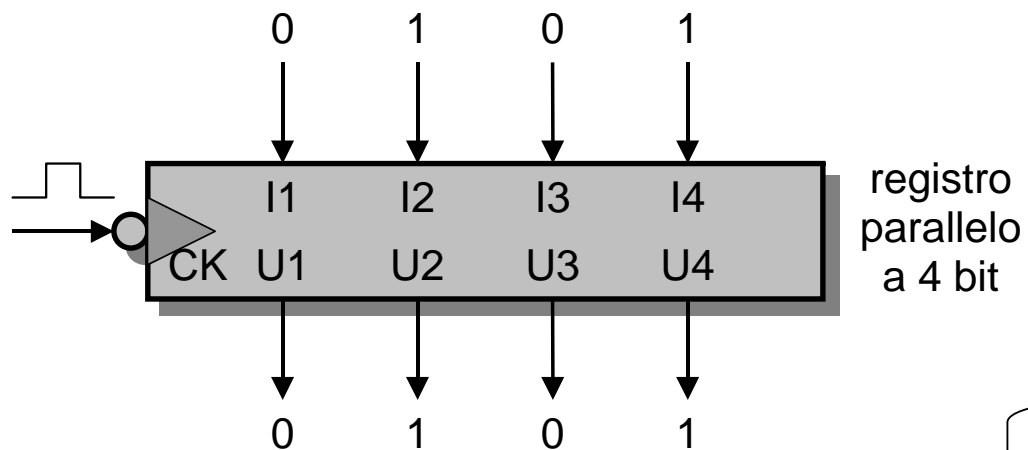
## Registro parallelo

---

- Il registro parallelo è un vettore di  $n \geq 1$  flip-flop di tipo D. Ha:
  - $n \geq 1$  ingressi  $I_1, \dots, I_n$
  - $n \geq 1$  uscite  $U_1, \dots, U_n$
  - e naturalmente l'ingresso di clock CK
  
- A ogni ciclo di clock, il registro legge e memorizza nel suo stato la parola di  $n$  bit presente in ingresso, e la presenta sulle  $n$  uscite nel ciclo successivo

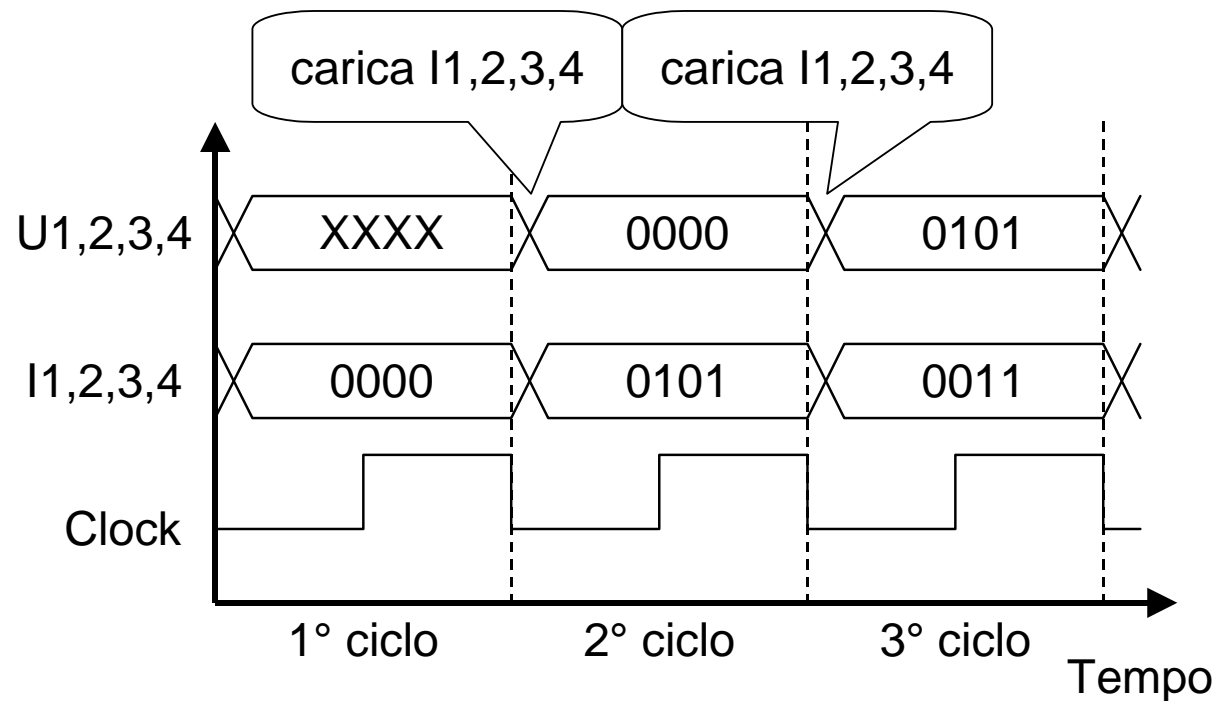


## Simbolo e funzionamento



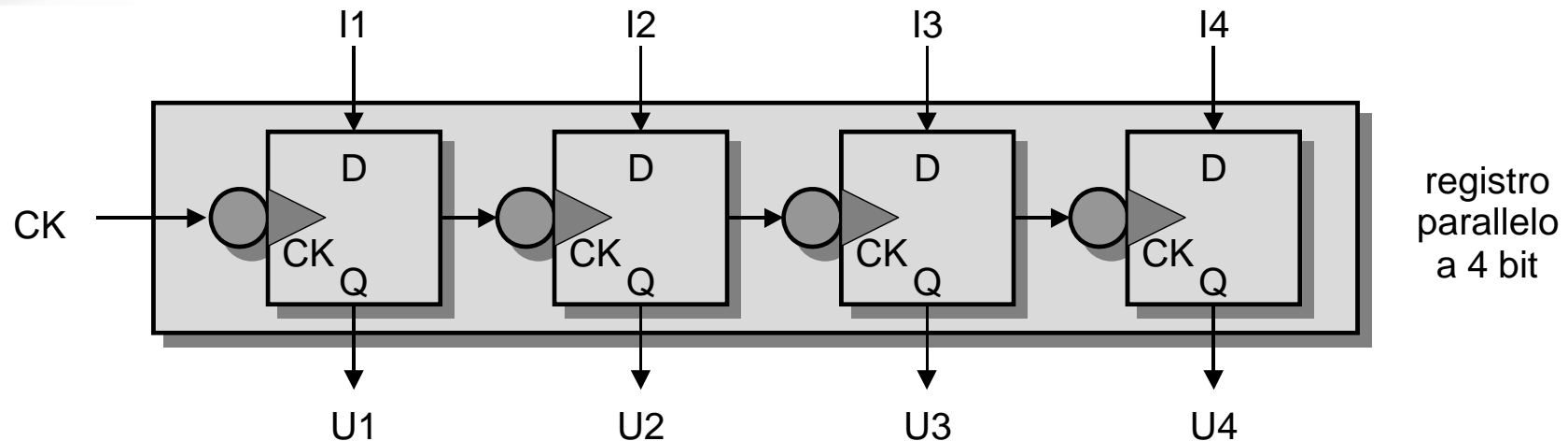
- ❑ carica 0000
- ❑ carica 0101
- ❑ ecc ...

### Diagramma temporale





## Progetto in stile funzionale



Registro parallelo progettato in stile funzionale, usando 4 **flip-flop** D sincroni sul fronte (di discesa)

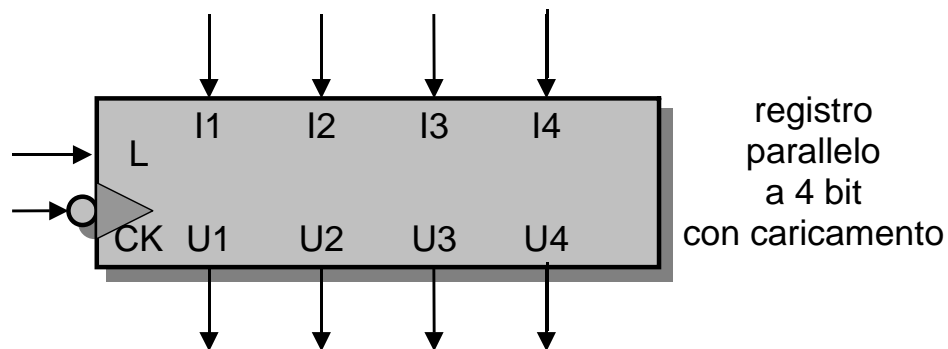
**Nota bene:**

se si usassero dei bistabili D trasparenti (sincronizzati sul livello), durante il livello alto del clock il registro sarebbe esso stesso del tutto trasparente, e dunque non si comporterebbe come un registro ...

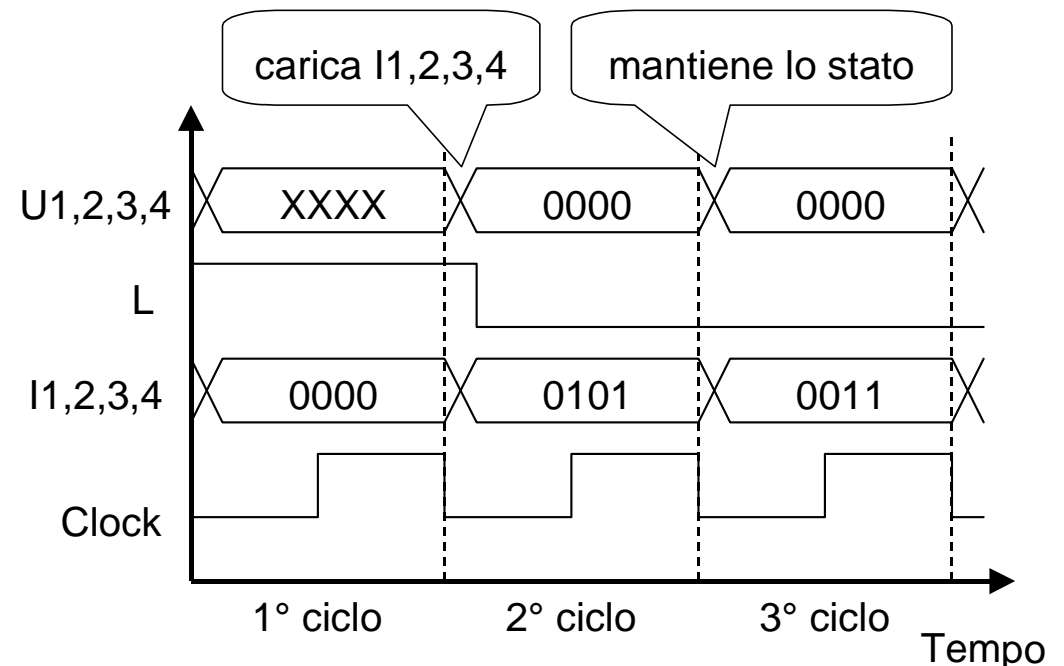


# Registro parallelo con comando di caricamento

- Funziona come il registro parallelo, ma ha in aggiunta un **ingresso** di comandi **di caricamento** (L, ingresso di **Load**):
  - Se il comando L è attivo (p. es.  $L = 1$ ), la parola in ingresso al registro viene memorizzata nel registro stesso e presentata in uscita nel ciclo successivo
  - Altrimenti (cioè  $L = 0$ ), il registro mantiene il suo stato corrente di memorizzazione

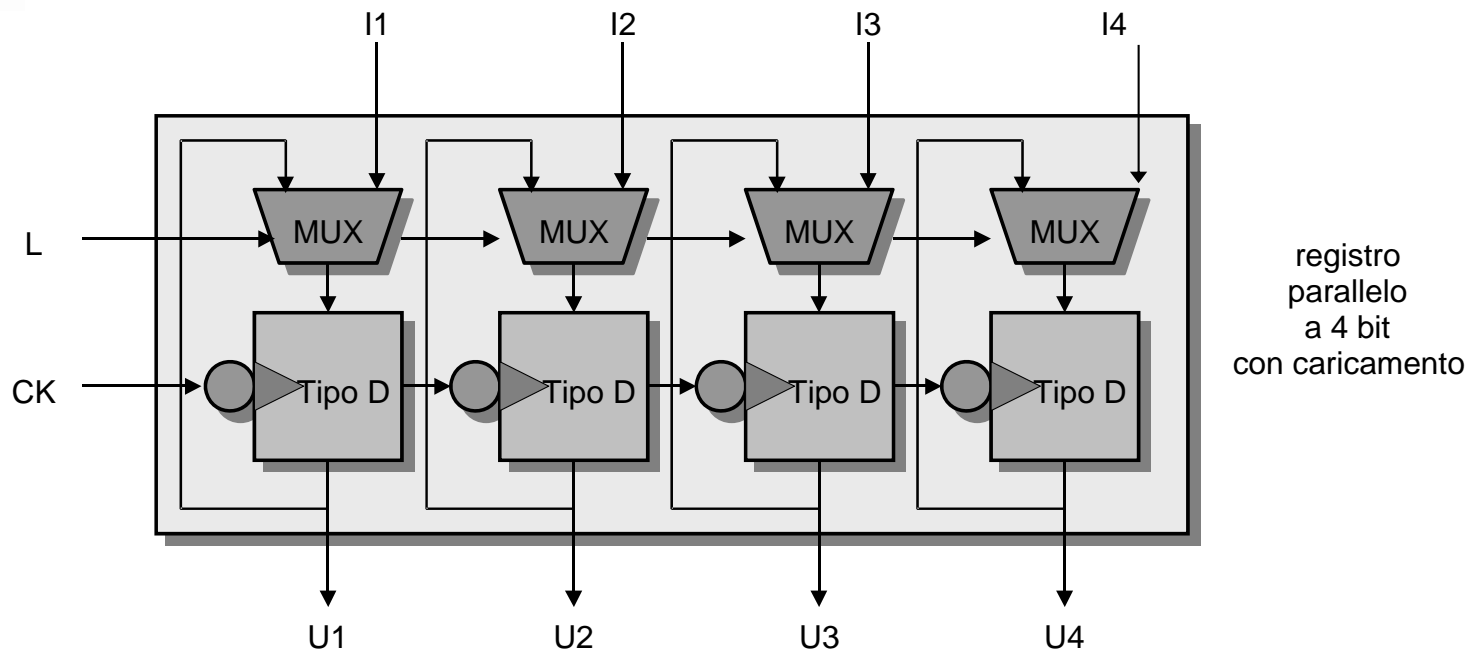


- carica 0000
- mantiene 0000
- ecc ...





## Progetto in stile funzionale



Registro parallelo progettato in stile funzionale, usando 4 flip-flop D sincroni sul fronte (di discesa) e 4 multiplexer a un ingresso di selezione e due ingressi dati





## Varianti e integrazioni

---

- ❑ Registro parallelo con comando di ripristino, per azzerare il contenuto
- ❑ Registro parallelo con comando di ripristino e di precarica
- ❑ Registro parallelo universale, riunisce le funzioni di tutti i registri precedenti: comandi di caricamento, comando di ripristino e comando di precarica



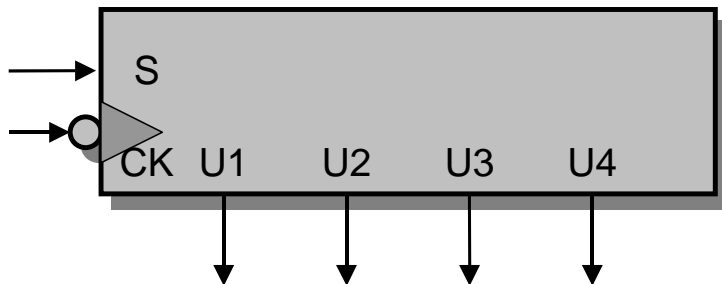
## Registro a scorrimento

---

- Il registro a scorrimento è una successione di  $n \geq 1$  flip-flop di tipo D collegati in cascata. Ha:
  - un ingresso seriale S
  - $n \geq 1$  uscite parallele  $U1, \dots, Un$
  - e naturalmente l'ingresso di clock
  
- A ogni ciclo di clock, fa scorrere di un bit verso DX la parola memorizzata, perdendo il bit più a DX e aggiungendo a sinistra il bit presente sull'ingresso seriale

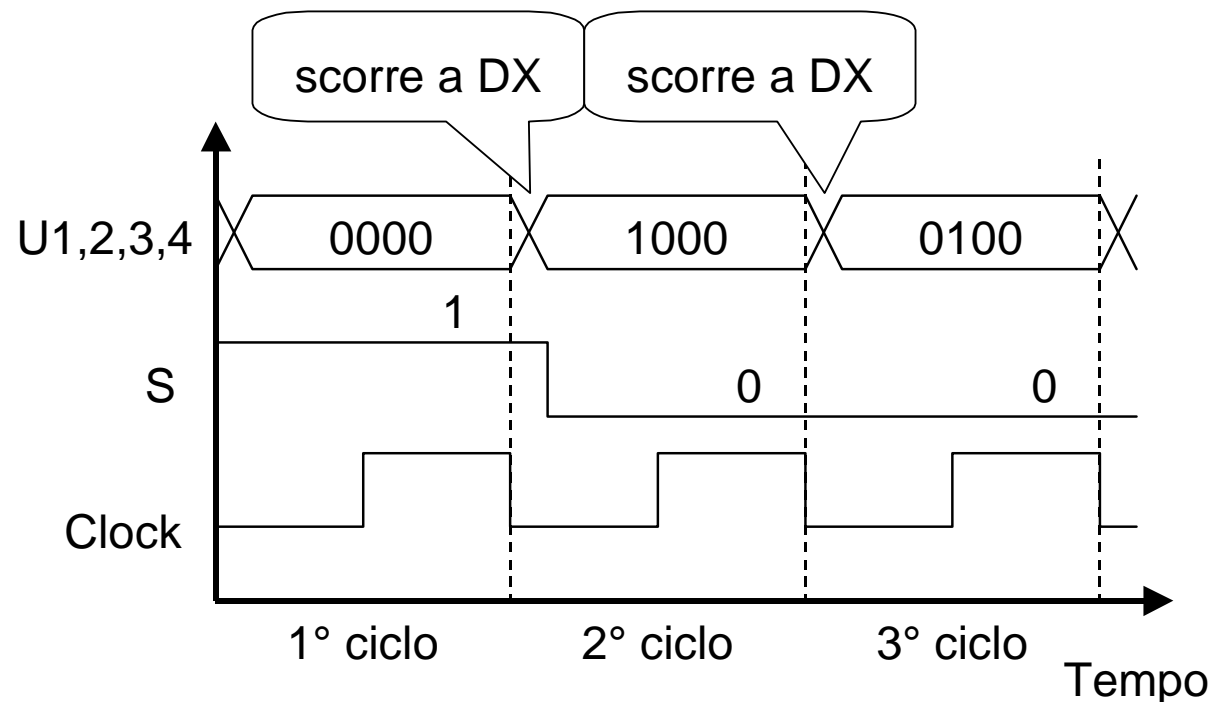


## Simbolo e funzionamento



registro  
a scorrimento  
a DX a 4 bit

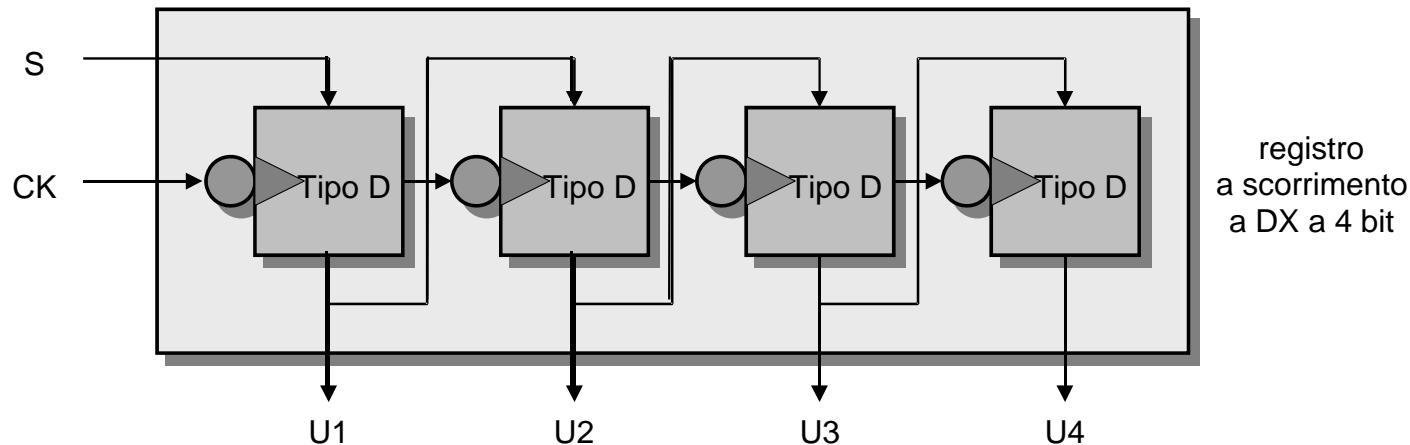
diagramma temporale



- scorre a DX
- scorre a DX
- ecc ...



## Progetto in stile funzionale



Registro a scorrimento a DX progettato in stile funzionale, usando 4 flip-flop D sincroni sul fronte (di discesa) collegati in cascata

### Nota bene:

se si usassero dei bistabili D trasparenti (sincronizzati sul livello), durante il livello alto del clock un bit potrebbe propagarsi lungo l'intera catena di bistabili ... non sarebbe un comportamento accettabile!



## Varianti e integrazioni

---

- ❑ Registro a scorrimento a SX
- ❑ Registro a scorrimento universale: DX e SX (è dotato di un comando di scelta del verso di scorrimento)
- ❑ Registro a scorrimento (DX o SX) con funzione di caricamento parallelo
- ❑ Registro parallelo / a scorrimento universale: riunisce le funzioni dei registri parallelo e a scorrimento universali
  
- ❑ Registro IN seriale / OUT seriale
- ❑ Registro IN parallelo / OUT seriale
- ❑ Registro IN parallelo/seriale OUT parallelo/seriale



---

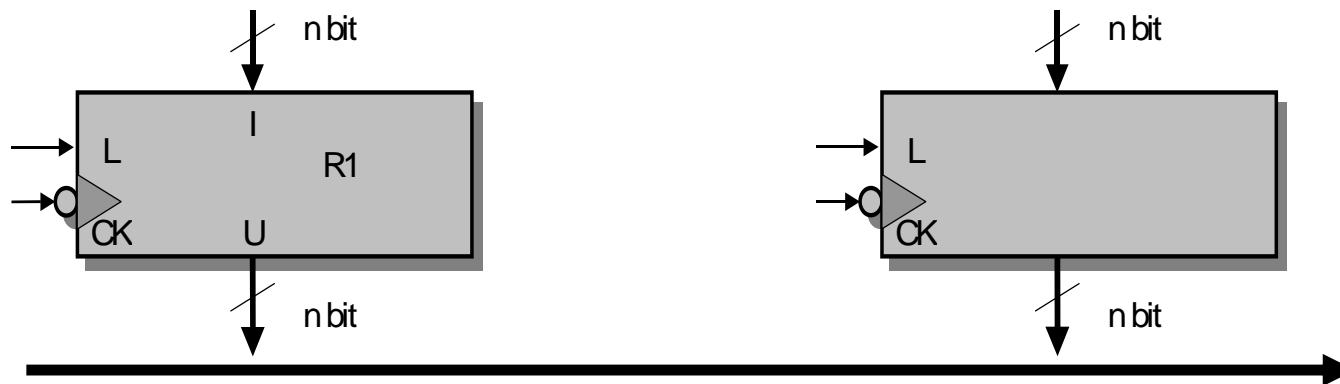
## Register File e memoria

- Componenti di memoria e uscite condivise
  - Register File (banco di registri): struttura e funzionamento
  - Memoria: struttura e funzionamento
  - Banco di memoria
  - Tecnologie di memoria
-



## Linee di uscita condivise

- L'organizzazione interna della memoria e la struttura dei banchi di memoria e dei banchi di registri prevedono generalmente che le **uscite di 2 o più componenti** siano **collegate alle stesse linee di uscita** (bus)
- Sono necessari opportuni "elementi funzionali" (**circuiti di pilotaggio delle uscite** del componente) che garantiscano la NON interferenza (dei segnali) tra i moduli che condividono le stesse linee di uscita





## Circuiti di pilotaggio delle uscite: Buffer tri-state

---

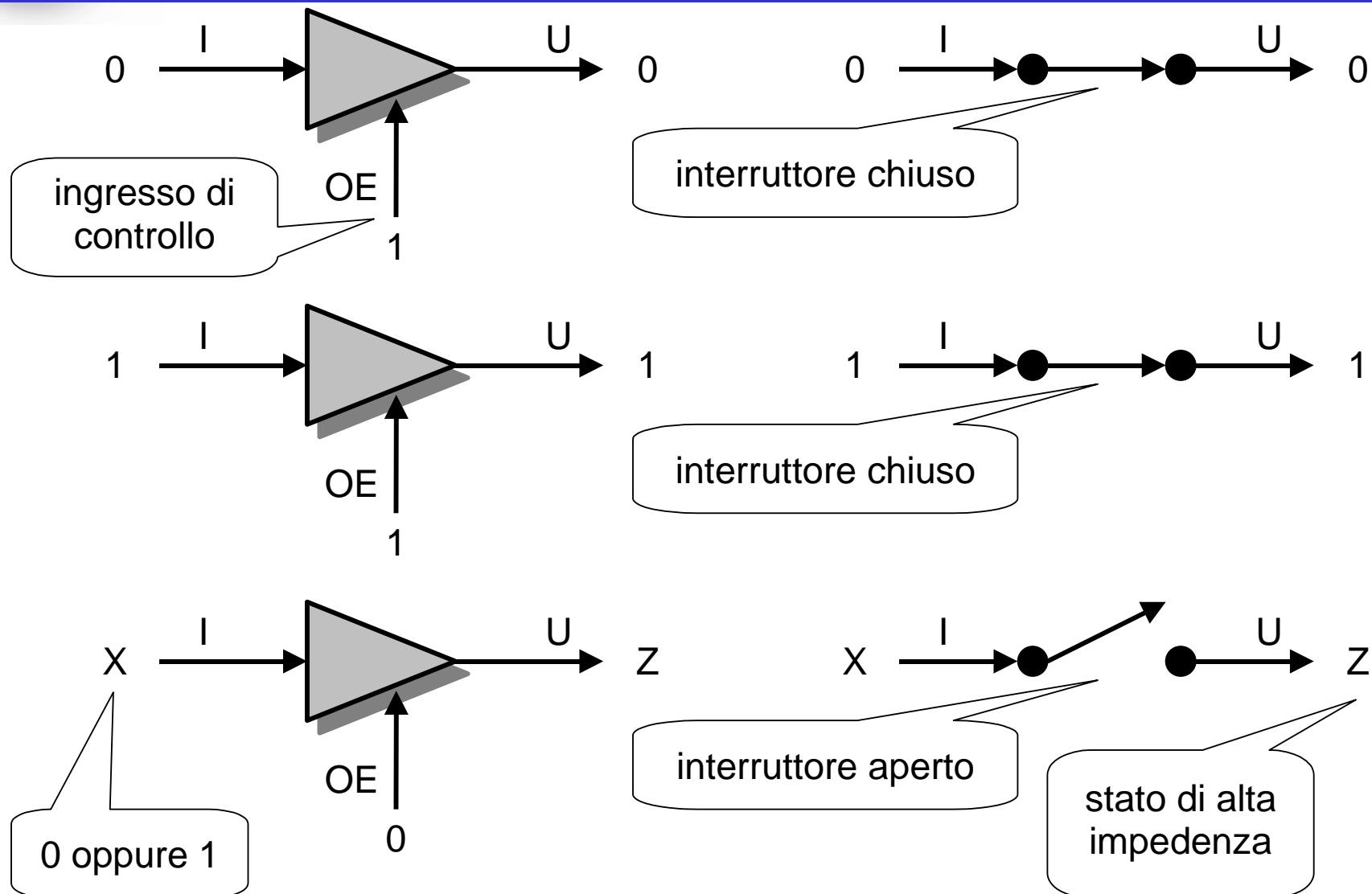
E' il circuito elementare modellabile come un contatto a tre posizioni:

- ❑ in stato di **bassa impedenza** consente di avere in uscita o il **livello alto** (1) o il **livello basso** (0)
- ❑ in stato di **alta impedenza** (Z) **isola elettricamente** l'uscita
- ❑ l'uscita tri-state viene gestita da un apposito ingresso di controllo (Output Enable) che, se non attivo, forza lo stato di alta impedenza



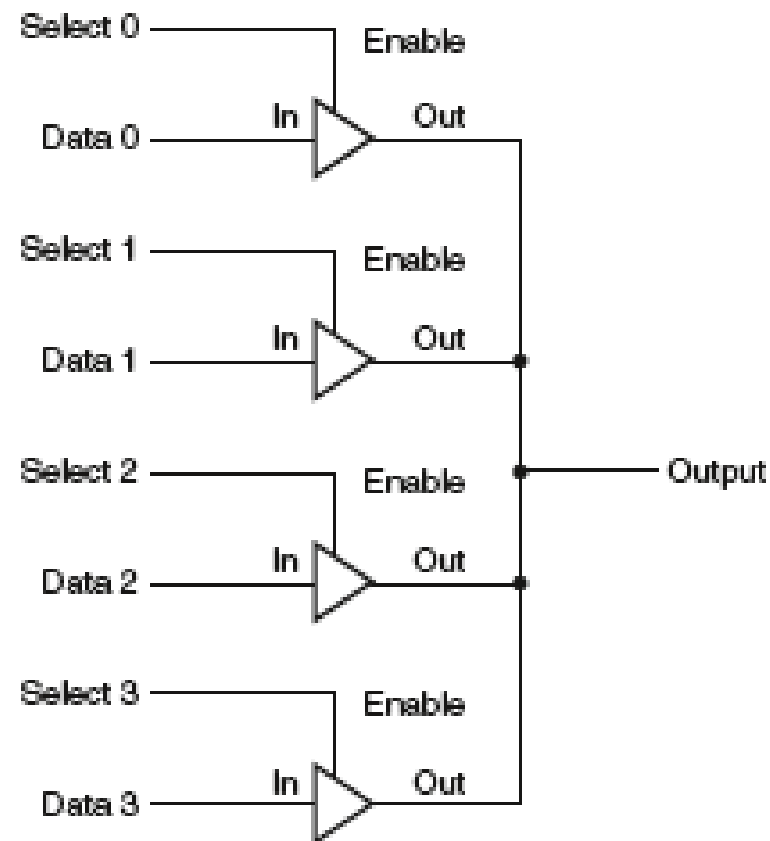


# Funzionamento





## Buffer tri-state: multiplexer sulle linee di uscita

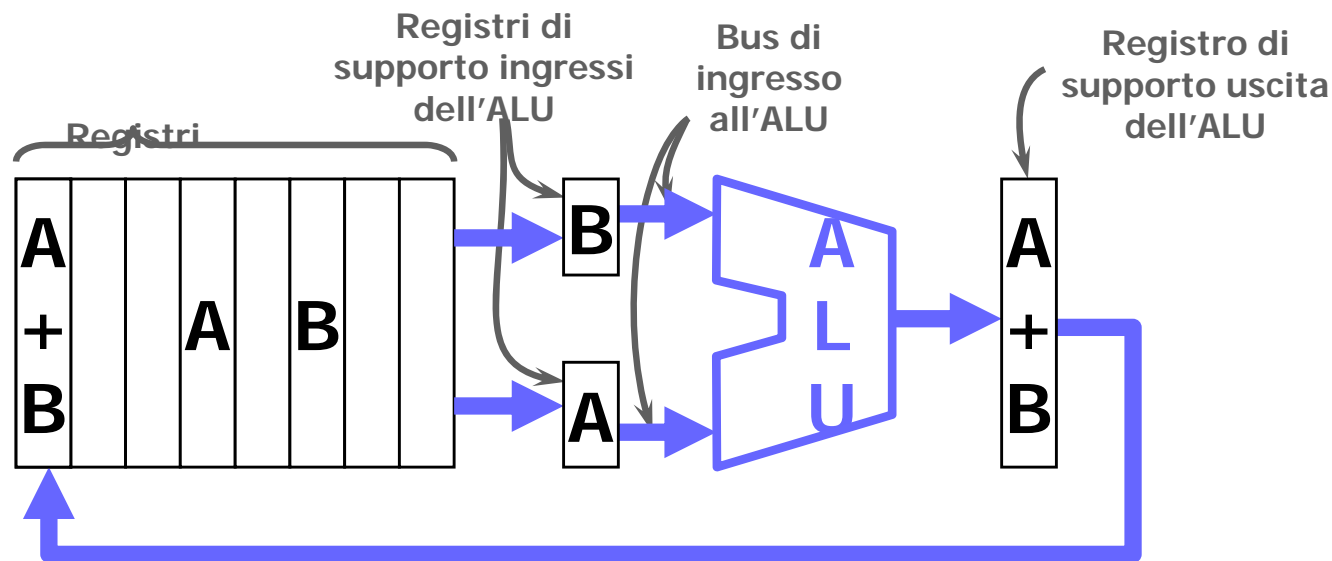


**FIGURE B.9.2** Four three-state buffers are used to form a multiplexor. Only one of the four Select inputs can be asserted. A three-state buffer with a deasserted Output enable has a high-impedance output that allows a three-state buffer whose Output enable is asserted to drive the shared output line.



# Banco di registri (register file)

- Spesso occorre utilizzare un certo numero di registri paralleli con funzione di caricamento, tutti aventi le stesse dimensioni e le stesse funzioni (ad es. nel data-path della CPU)



- I registri vengono organizzati in una struttura a vettore, chiamata **banco di registri** o **register file (RF)**



## Banco di registri: caratteristiche

---

- ❑ Consideriamo come esempio di riferimento un banco di 32 registri da 32 bit ciascuno (Register File del MIPS)
- ❑ Ogni registro è identificato da un **indirizzo** (numero di registro) specificato su 5 bit
  - con  $n \geq 1$  registri occorrono  $\lceil \log_2 n \rceil$  bit
- ❑ Le operazioni eseguibili sul banco sono:
  - **lettura**: si presentano in uscita i 32 bit memorizzati nel registro indirizzato
  - **scrittura**: si memorizzano 32 bit acquisiti in ingresso nel registro indirizzato
- ❑ **Porta di lettura/scrittura**: insieme di segnali che consentono la lettura e la scrittura dei registri
  - 1 porta di lettura e scrittura
  - 1 porta di lettura e 1 di scrittura
  - 2 porte di lettura e 1 di scrittura (vedi ALU)



# Register File del MIPS

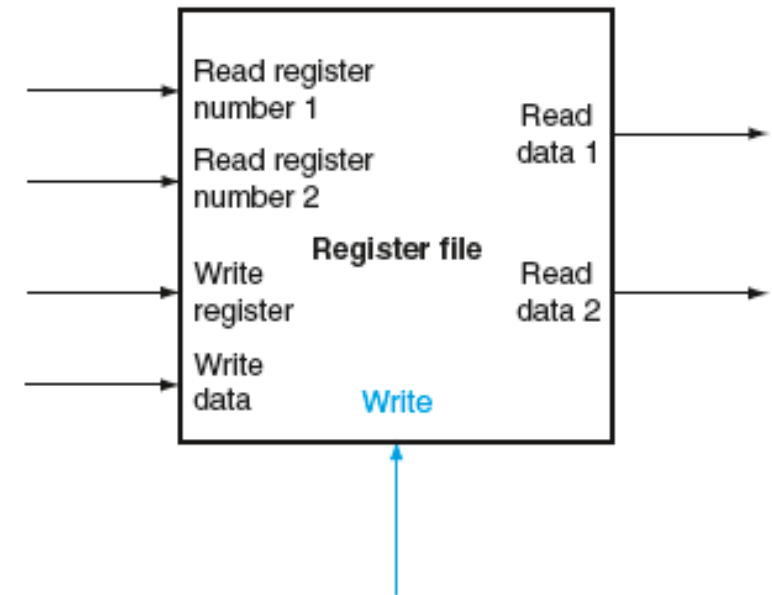
- 2 porte di lettura indipendenti
  - 1 porta di scrittura
- con le opportune temporizzazioni è possibile accedere in parallelo a 3 registri distinti

## *Accesso in lettura:*

- lo stato del registro non viene modificato
- poiché le porte di lettura e scrittura sono distinte e indipendenti non è necessario un comando di lettura esplicito: è sufficiente fornire l'indirizzo dei registri coinvolti e la lettura avviene ogni ciclo di clock
- la realizzazione del banco può richiedere buffer tri-state le uscite devono essere scollegate elettricamente

## *Accesso in scrittura:*

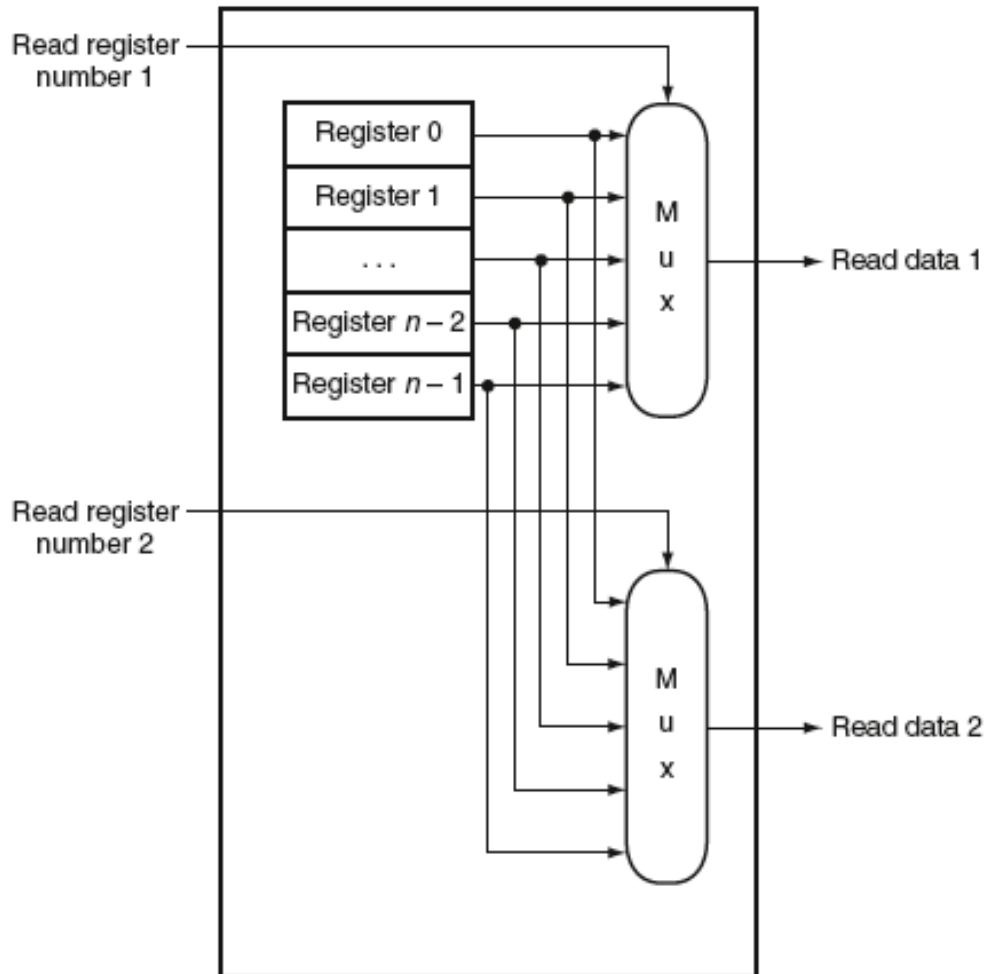
- indirizzo del registro
- dato da scrivere
- segnale di scrittura esplicito (**Write**)



Il segnale di clock è sottointeso



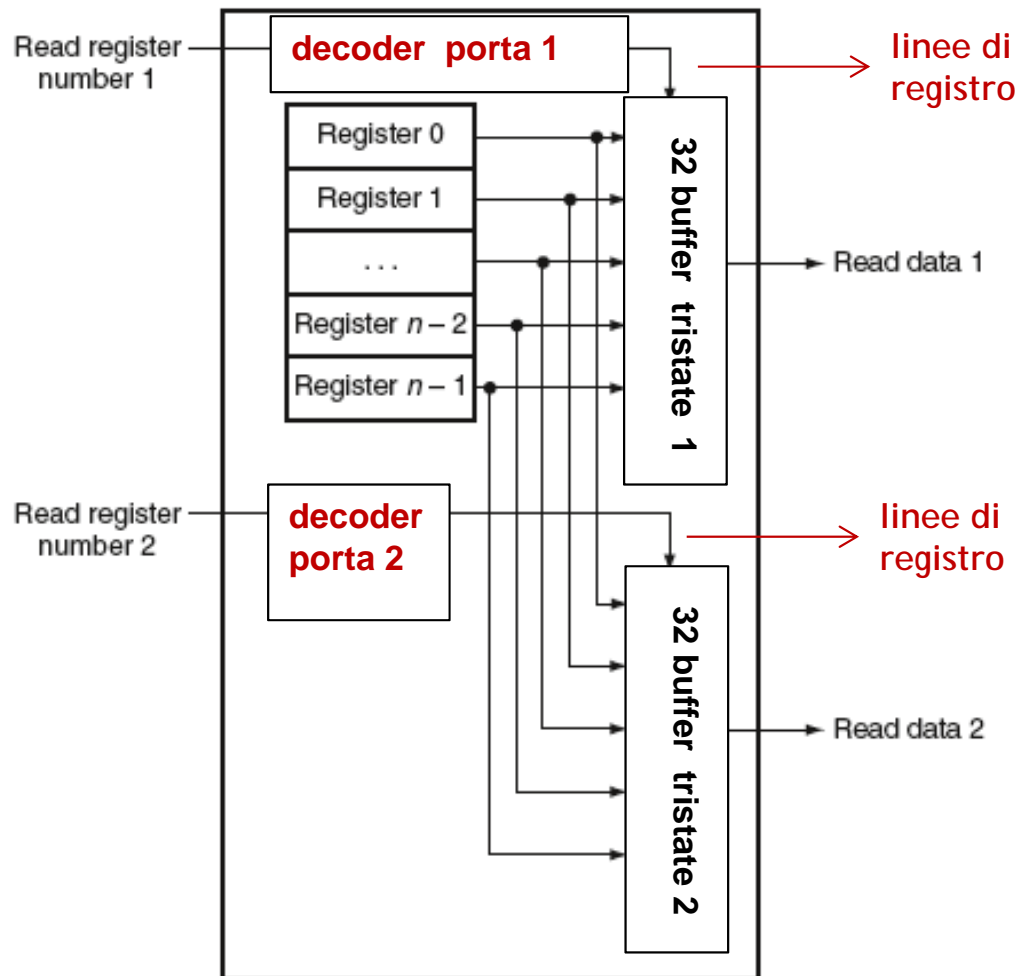
## Register File del MIPS: porte di lettura



- L'indirizzo del registro da leggere è usato come segnale di controllo del multiplexer
- I due multiplexer vengono controllati in modo indipendente



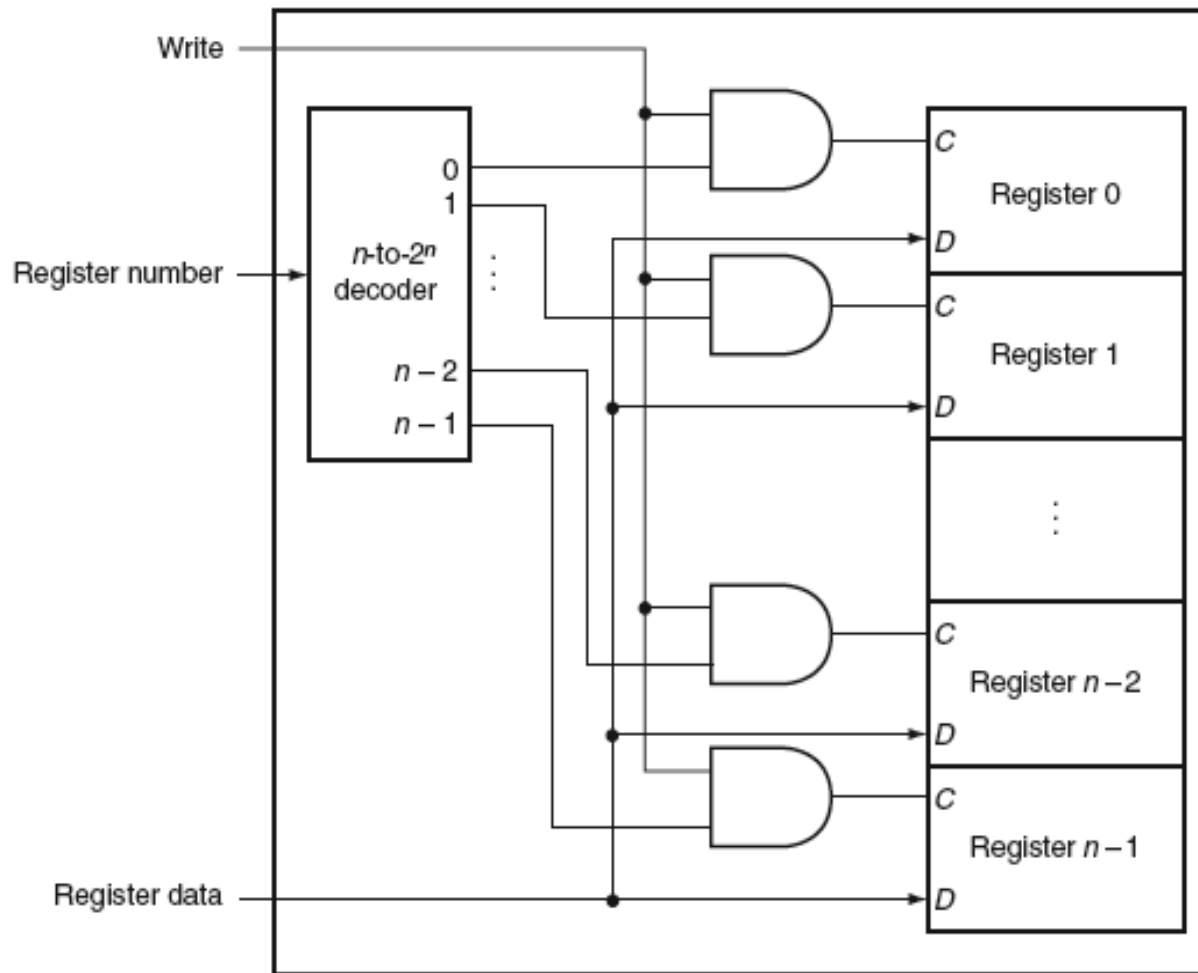
# Register File del MIPS: porte di lettura (un'altra versione)



- Tutte le uscite dei registri condividono le stesse linee di uscita
- L'indirizzo del registro da leggere è usato come segnale di ingresso al decodificatore (5:32)
- Le 32 linee di registro (uscite del decoder) sono attive in mutua esclusione e sono usate come linee di *Enable* per il buffer tristate della porta di lettura considerata
- I decodificatori sono controllati in modo indipendente



## Register File del MIPS: porta di scrittura



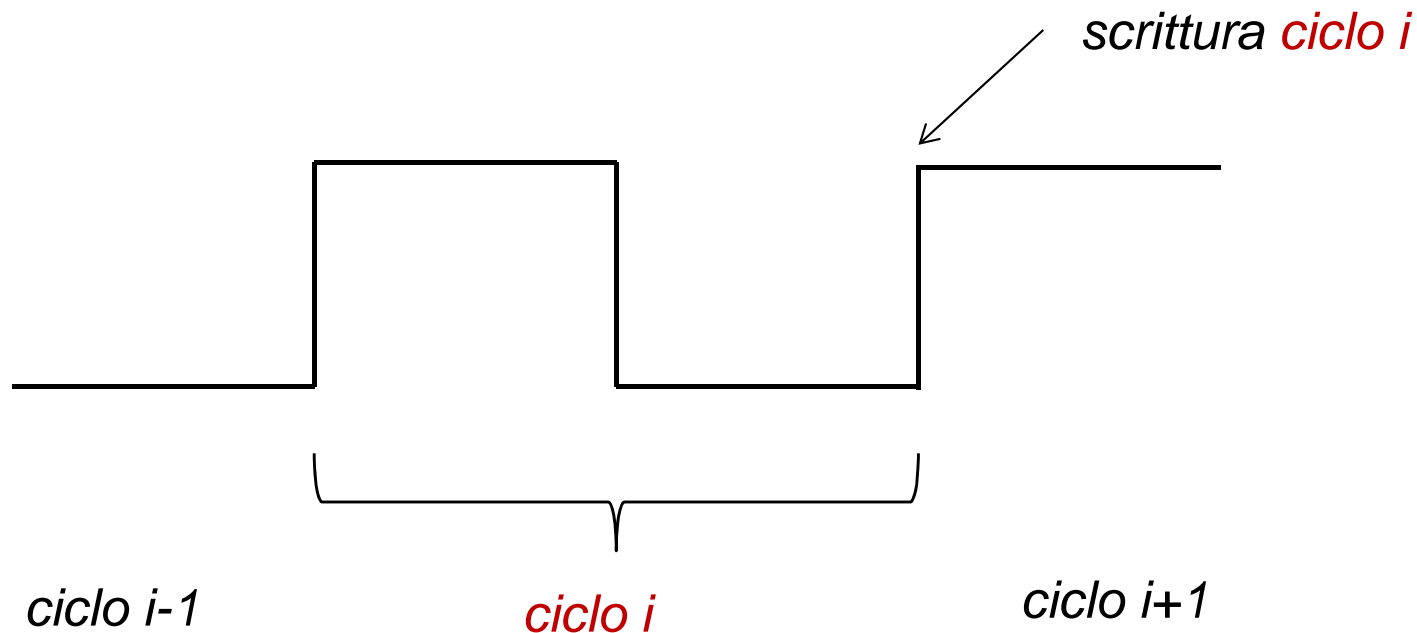
- Un solo registro deve modificare il proprio contenuto
- L'indirizzo del registro da scrivere è fornito come ingresso al decodificatore
- La *linea «di registro»* e il segnale di *write* abilitano un solo registro in scrittura
- Sul fronte attivo del clock (segnale non mostrato) il dato da scrivere *D* viene memorizzato nel registro e si presenta in uscita dopo un opportuno ritardo di propagazione





## Lettura e scrittura di un registro nello stesso ciclo di clock

La scrittura avviene sul fronte di salita del ciclo e quindi la lettura fornisce il valore scritto al ciclo precedente





# Memoria

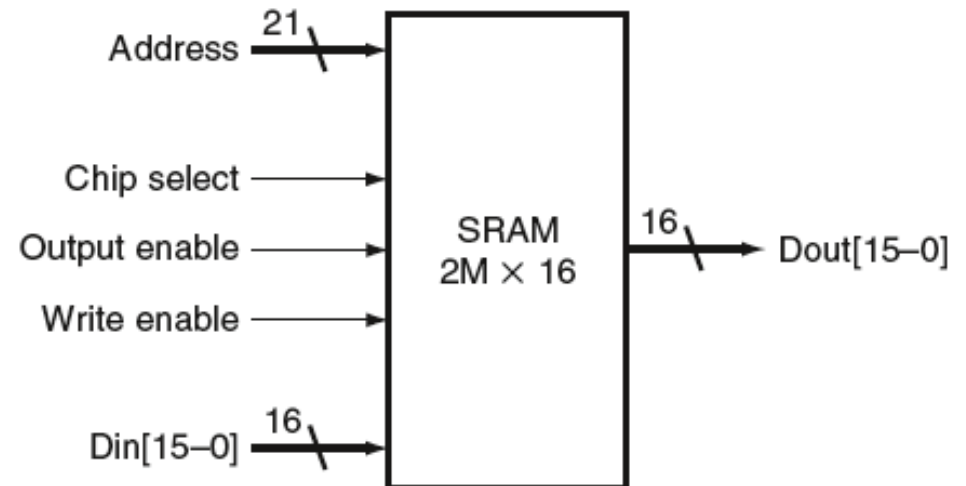
---

- ❑ La memoria è un **blocco funzionale** di tipo sequenziale complesso
    - Mantiene a tempo indefinito se alimentata (SRAM) le informazioni memorizzate e permette l'accesso in lettura o in scrittura
  - ❑ Ha una **struttura a vettore** (almeno in termini di segnali di accesso esterni) i cui elementi sono le parole di memoria di una certa lunghezza
  - ❑ Un **componente integrato** (chip) di memoria si caratterizza specificando:
    - la capacità, misurata in numero totale di bit memorizzabili: di solito si esprime come prodotto del numero di parole per il numero di bit per parola
    - le funzioni: lettura e scrittura, solo lettura
    - il numero di porte di accesso
    - e il tempo necessario per l'accesso
-



# Interfaccia di memoria

- Il contenuto della memoria viene letto o scritto una parola per volta, in un ciclo di clock (più cicli in memorie lente)
- Si accede a una parola di memoria tramite la **porta di accesso alla memoria**
- La porta di accesso alla memoria può funzionare in lettura e scrittura (è il caso più frequente), solo in lettura e teoricamente anche solo in scrittura (caso poco frequente)





## Segnali dell'interfaccia di memoria

---

- ❑ Gli **ingressi di indirizzo**, che codificano in binario l'indirizzo della parola su cui si deve operare
  - ❑ Le **uscite/ingressi di dato**, che servono per leggere/scrivere una parola
  - ❑ Per le linee di dato e indirizzo sono da rispettare i tempi di set\_up e hold, quindi questi segnali vengono forniti per primi in modo che siano stabili quando le linee di comando sono attivate a seconda dell'operazione
  - ❑ il comando di scrittura, **Write enable**. Se Write enable è un impulso deve avere una durata minima che consenta di soddisfare quanto sopra
  - ❑ il comando di abilitazione delle uscite dati, **OE** (output enable): OE = 1 le uscite sono abilitate; OE = 0 le uscite sono isolate
  - ❑ il comando di abilitazione del componente, **CS** (chip select): CS = 1 chip attivo, si può accedere al contenuto; CS = 0 chip non attivato
-



# Cicli di lettura e scrittura

---

## Ciclo di lettura

- ❑ indirizzo della parola da leggere
- ❑ comando di lettura (WE a livello 0)
- ❑ non isolare le uscite dati (OE = 1)
- ❑ abilitare il componente (CS = 1)
- ❑ contenuto della parola disponibile sulle uscite. Ritardo di lettura: 8 - 20 ns

## Ciclo di scrittura

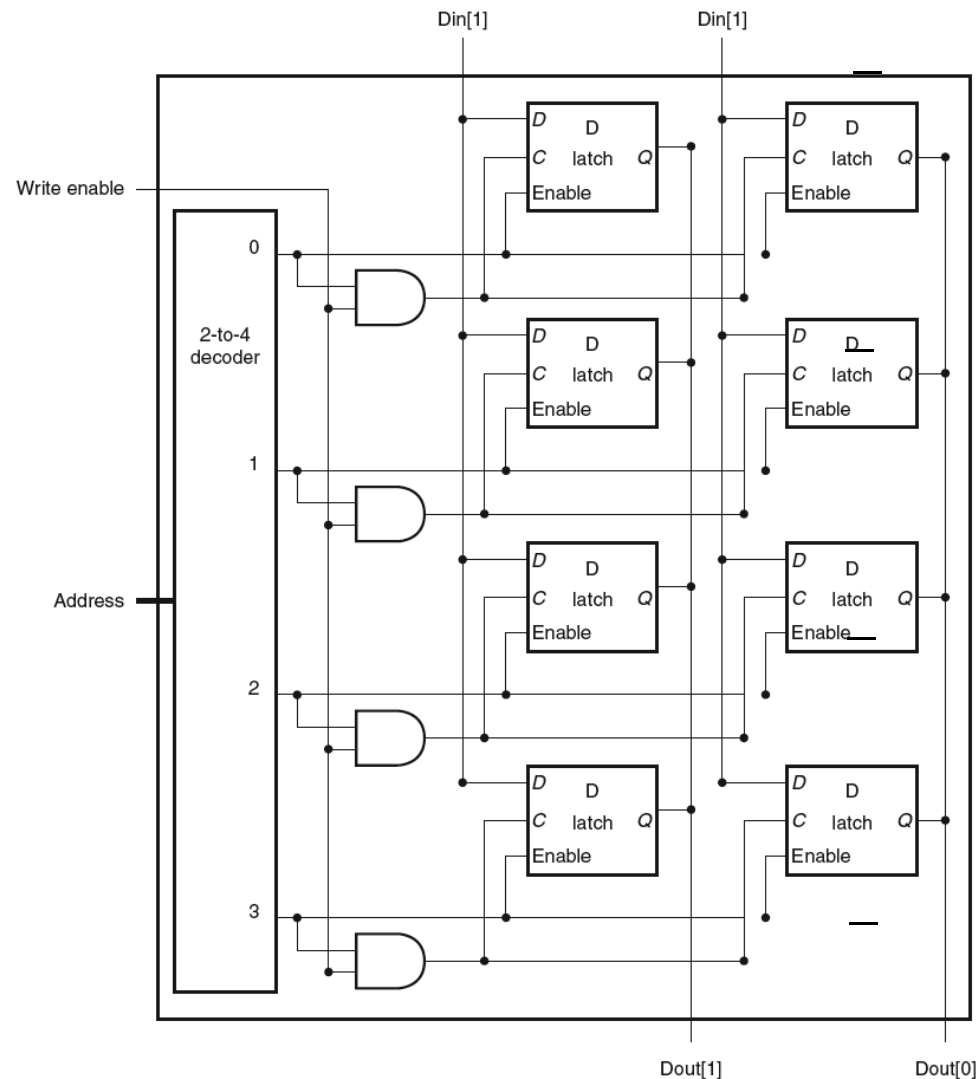
- ❑ indirizzo della parola da scrivere
- ❑ dato da scrivere in ingresso
- ❑ comando di scrittura (WE a livello 1)
- ❑ isolare le uscite dati (OE = 0)
- ❑ abilitare il componente (CS = 1).  
Ritardo di scrittura: 8 - 20 ns



# Struttura della memoria

## *organizzazione a matrice di bistabili*

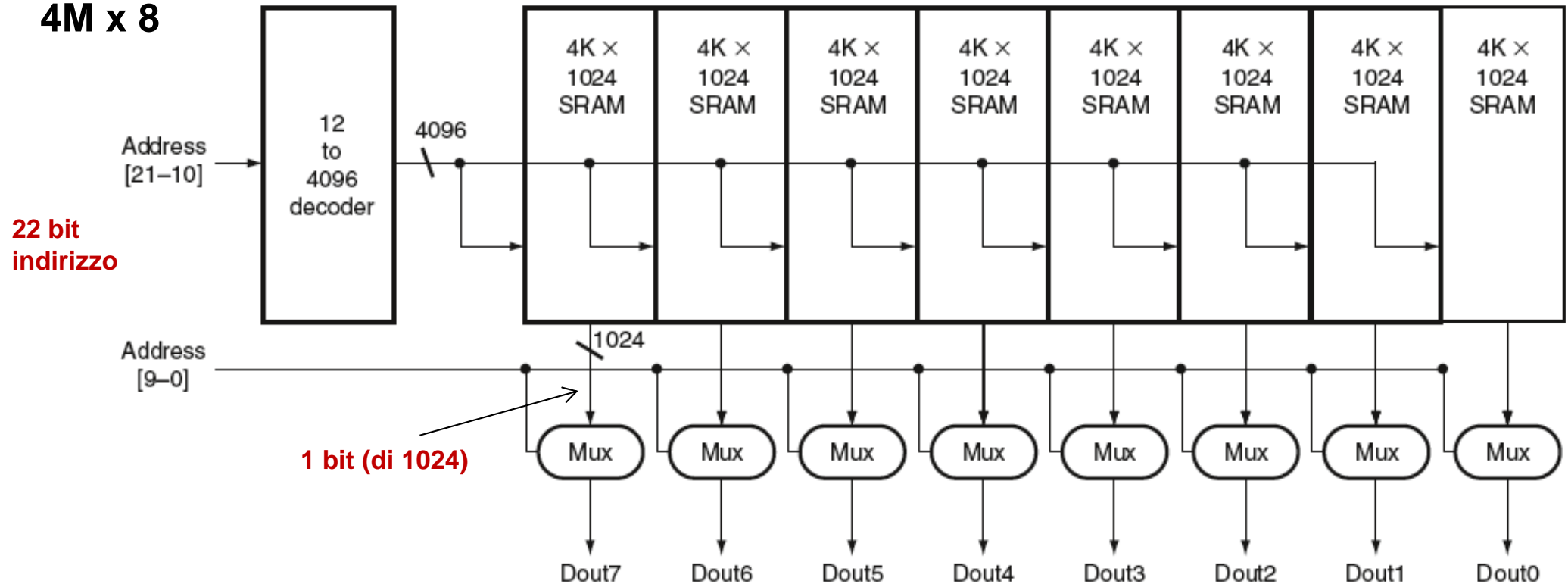
- ❑ Matrice di bistabili (righe=parole, colonne = bit della parola), completandola con reti combinatorie di controllo per gestire l'accesso alle parole
- ❑ Le uscite del decoder sono le linee di parola
- ❑ Ogni bistabile ha la sua uscita in tri-state perché condivisa con gli altri in posizione omologa
- ❑ Non adatta a dimensioni significative di memoria (decoder e linee di parola)





## Banco di memoria: organizzazione a matrice di componenti e decodifica dell'indirizzo a due livelli

4M x 8



**FIGURE B.9.4 Typical organization of a 4M × 8 SRAM as an array of 4K × 1024 arrays.** The first decoder generates the addresses for eight 4K × 1024 arrays; then a set of multiplexors is used to select 1 bit from each 1024-bit-wide array. This is a much easier design than a single-level decode that would need either an enormous decoder or a gigantic multiplexor. In practice, a modern SRAM of this size would probably use an even larger number of blocks, each somewhat smaller.



## DRAM: qualche considerazione

---

- ❑ RAM **dinamiche**: il singolo bit di informazione è «memorizzato» nella carica di un condensatore il cui accesso avviene tramite un transistor che può «leggere» o «scrivere» il suo valore (durata della carica alcuni millisecondi)
- ❑ Usano un singolo transistor per cella contro i 4-6 delle SRAM, quindi hanno un costo inferiore per bit e sono più dense. Tempi di accesso maggiori
- ❑ E' necessario il *refresh* delle parole di memoria (lettura e riscrittura del contenuto) per la degradazione della carica
- ❑ Sono organizzate sempre con decodifica dell'indirizzo a due livelli per diminuire i tempi di refresh
- ❑ Le operazioni di refresh consumano meno del 2% dei cicli di memoria



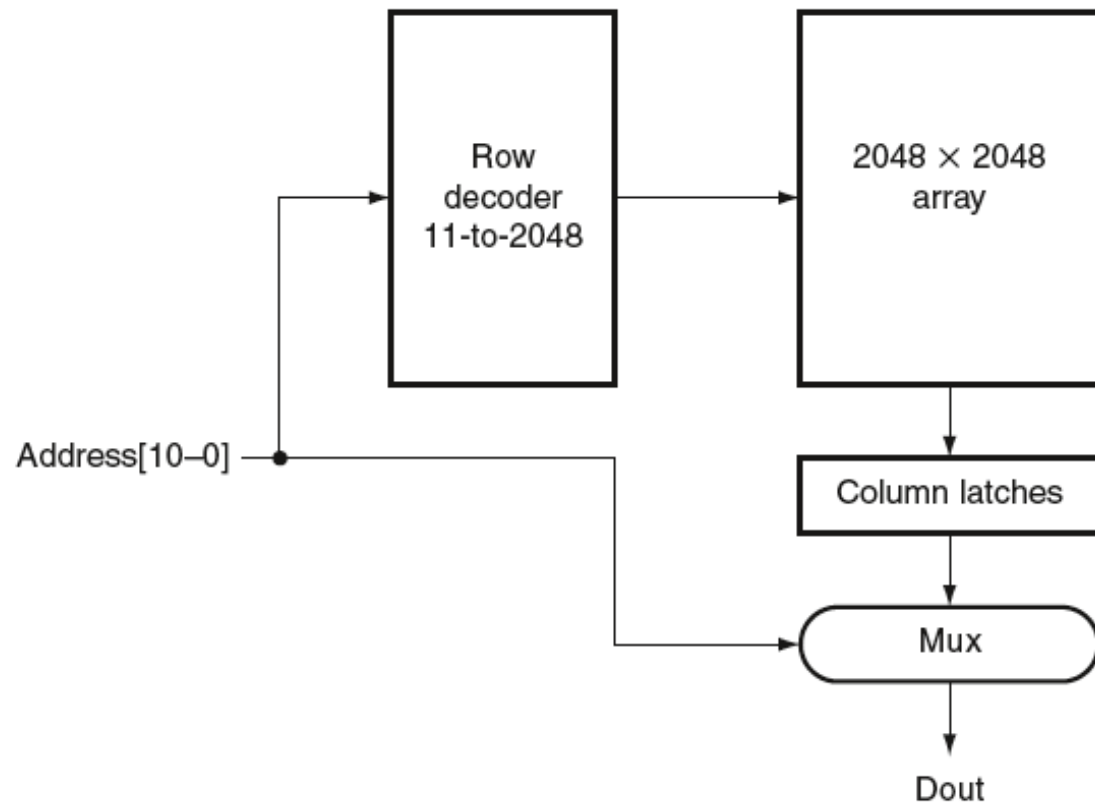


## Struttura di DRAM

Linee indirizzo multiplate:  
segnali RAS e CAS per  
utilizzare correttamente i  
bit indirizzo

Latch di colonna: contiene i  
contenuto di una riga letta

La singola operazione di  
refresh riscrive un'intera  
riga



**FIGURE B.9.6 A  $4M \times 1$  DRAM is built with a  $2048 \times 2048$  array.** The row access uses 11 bits to select a row, which is then latched in 2048 1-bit latches. A multiplexor chooses the output bit from these 2048 latches. The RAS and CAS signals control whether the address lines are sent to the row decoder or column multiplexor.



## RAM Statica (SRAM)

---

- ❑ Memoria RAM (Random Access Memory) realizzata con bistabili
  - ❑ Capacità medio-piccola
  - ❑ Tempo di accesso molto breve
  - ❑ Funziona in lettura e scrittura
  - ❑ Volatile: senza alimentazione il contenuto della memoria svanisce
  - ❑ Usi: svariati, in particolare come cache
- 
- ❑ La memoria SRAM consuma parecchi transistor per bit memorizzato (circa 6 transistor per bit)



## RAM Dinamica (DRAM)

---

- ❑ La tecnologia **DRAM** usa circa 1 transistor per bit memorizzato
  - Sfrutta il fenomeno dell'accumulo temporaneo di carica sul transistor (capacità parassita)
  - Internamente contiene un circuito di rinfresco che rigenera le cariche
- ❑ Memoria RAM (matrice di transistor) ad altissima densità
- ❑ Capacità grande-grandissima
- ❑ Tempo di accesso medio
- ❑ Funziona in lettura e scrittura
- ❑ Volatile: senza alimentazione il contenuto della memoria svanisce
- ❑ Usi: numerosissimi, la memoria centrale dei calcolatori normalmente è DRAM



# ROM

---

- ❑ Memoria ROM (Read Only Memory), realizzata come matrice di transistor
- ❑ Capacità grande
- ❑ Tempo di accesso medio
- ❑ Funziona in sola lettura
- ❑ Persistente: il contenuto permane anche in assenza di alimentazione
- ❑ Usi: per memorizzare programmi permanenti, non modificabili; grandi volumi di produzione



# PROM, EPROM, EEPROM

---

- ❑ Capacità e tempo simili alla ROM
- ❑ Sola lettura e persistenti
- ❑ Sono programmabili sul campo, tramite un apposito programmatore:
  - PROM: programmabile una volta sola
  - EPROM: cancellabile con raggi UV
  - EEPROM: cancellabile elettricamente (si può anche scrivere un solo byte per volta)
- ❑ Usi: piccoli volumi di produzione, prototipi



## Memoria FLASH

---

- ❑ Capacità e tempo simili alla DRAM (o solo di poco inferiori)
- ❑ Funziona in lettura e scrittura (la scrittura però è a blocchi di byte)
- ❑ Persistente: il contenuto permane anche in assenza di alimentazione
- ❑ Usi: dati multimediali (p. es. immagini statiche, sequenze video), programmi fissi ma periodicamente aggiornabili



## Tabella riassuntiva

Tipo	Categoria	Modalità di cancellazione	Scrittura byte	Volatile	Usi specifici
SRAM	lett/scritt	elettrica	si	si	cache
DRAM	lett/scritt	elettrica	si	si	mem. centrale
ROM	sola lett	nessuna	no	no	grandi vol.
PROM	sola lett <sup>*</sup>	nessuna	no	no	piccoli vol.
EPROM	sola lett <sup>*</sup>	luce UV	no	no	prototipi
EEPROM	sola lett <sup>*</sup>	elettrica	si (lenta)	no	prototipi
FLASH	lett/scritt	elettrica	a blocchi	no	multimedia

\*Le memorie cancellabili vengono talvolta qualificate come “memorie prevalentemente a sola lettura” (read-mostly), invece che “a sola lettura” (read-only)