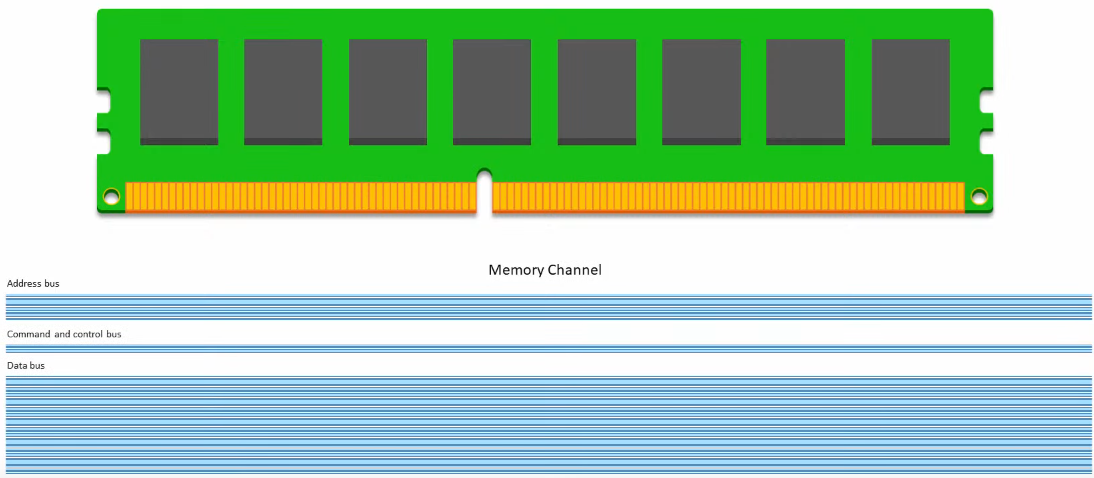
**MEMORIA RAM (RANDOM ACCESS MEMORY)**

La memoria RAM è la **memoria principale** **(MM)** del PC, sul quale la CPU va a cercare la maggior parte dei dati o programmi che gli servono in ogni istante.

È **volatile**, cioè perde il contenuto informativo se non c’è alimentazione.

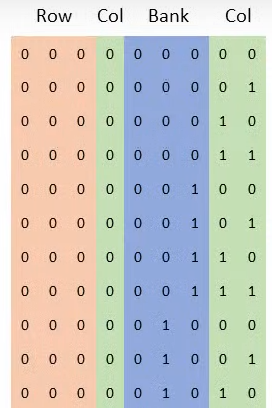


Il canale di memoria viene inviato alla **MMU (Memory Management Unit)**

**BUS ADDR = contiene gli indirizzi di RAW e COL = 17bits**

**Con 17bit è possibile indirizzare 2^17 Bytes**

**Grazie alla suddivisione invece è possbile definire 17 linee e 17 colonne, per un totale di 2^17 \* 2^17 celle**



**ROW** identifica la RIGA durante il RAS

**BANK** identifica il Banco di memoria da selezionare

2^BANKlength possibili banchi all’interno di un modulo

**COL** identifica la colonna da leggere

in base alla dimensione di BURST si leggono gli indirizzi successivi a quello appena letto

SI NOTA che ogni 4 colonne lette viene incrementato il numero di banco

**BUS CONTROL**

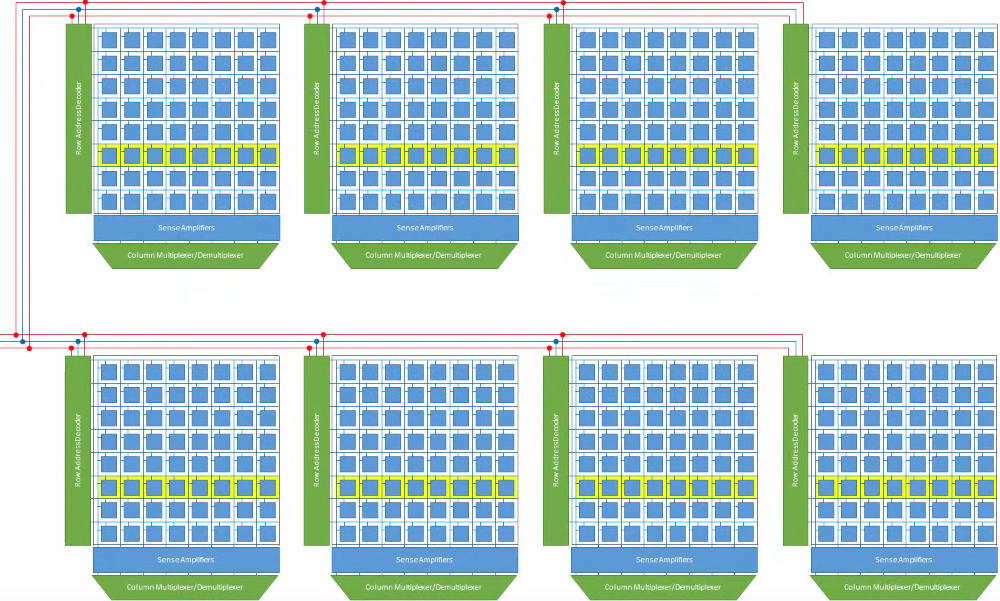
* **WE = Write Enable**
* **CAS = per indicare l’inzio del CAS**
* **RAS = per indicare l’inizio del RAS**

**BUS DATA**

**Contiene i dati in lettura o scrittura per la RAM**

**Può presentare dei bit aggiuntivi per l’implementazione dei codici di controllo errori**

**I dati al suo interno vengono inseriti in CACHE, dopo un’opportuna decodifica dell’indirizzo di memoria**



Ogni ARRAY di celle può leggere solo 1 bit alla volta, selezionato dal RAS e CAS

Per leggere dati multipli è necessario intervenire su diversi ARRAY insieme

I diversi ARRAY vengono letti e scritti contemporaneamente dallo stesso indirizzo.

Durante il **RAS** vengono **selezionate tutte le righe** di ogni **ARRAY**, le quali vengono copiate nei relativi **sense amplifier**

Durante il **CAS** vengono selezionate le colonne per ogni **ARRAY**, e i bit prelevati, 1 per ogni modulo, vengono inseriti nel BUS DATI.

In base alla **dimensione del BUS** si individua la **quantità** di **bit letti/scritti** in un **singolo accesso**

**L’unione di tutti gli array si chiama Banco di memoria.**

All’interno di un **singolo modulo** di memoria **vengono salvati più Banchi,** sempre in **quantità** esprimibili in potenzedi 2**, tipicamente 8 o 16.**

l’insieme di tutti i banchi prende il nome di CHIP.

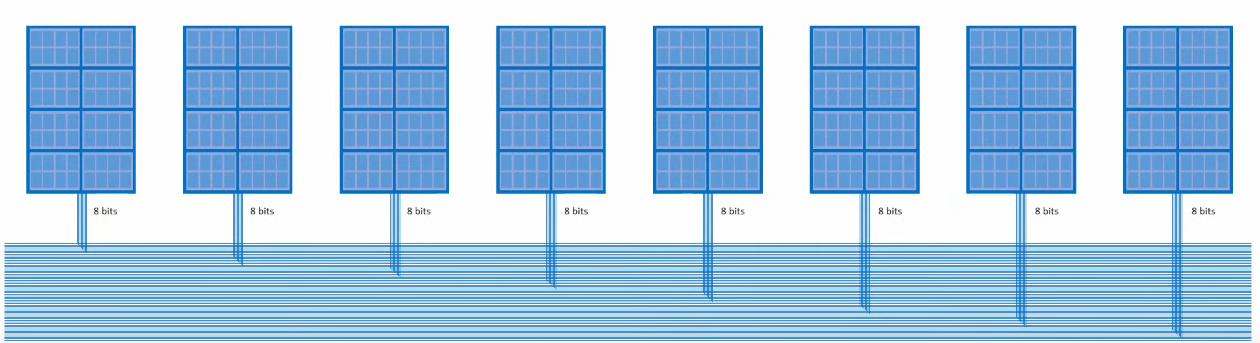
L’insieme dei CHIP prende il nome di RANK

All’interno di un modulo è possibile trovare differenti RANKS, gestiti separatamente l’uno dall’altro. Ogni Modulo ha accesso al canale di memoria, perciò anche i RANKS dovranno attendere il completamento di altri trasferimenti prima di avviare il proprio.

Il RANK trasmetterà numeroBANKS \* bit/bank = numero di bank presenti in un chip \* bit trasmessi da un singolo banco.

I banchi da 8 arrays possono trasmettere 8bits contemporaneamente, siccome ogni array ne fornisce solo 1.

Il numero di banchi varia a seconda dell’architettura.



Per identificare ogni Banco all’interno del modulo sono necessari dei bit dedicati nel BUS ADDR

Per ridurre notevolmente il traffico dati in RAM una volta effettuato l’accesso è possibile trasferire più di 1 bit per ogni banco.

**BURST =** numero di bit massimi che possono essere letti durante un singolo accesso.

La lunghezza del BURST dipende dalla architettura della RAM

DDR2 → BURST = 4

DDR3 → BURST = 8

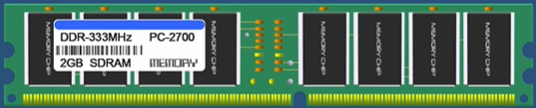
DDR4 → BURST = 16

Usando i BURST viene implementata la tecnica del PREFETCHING, o meglio località spaziale.

Caricando più locazioni contigue è probabile che i dati contigui vengano anch’essi richiesti dalla CPU

La RAM si divide in:

* **Statica SRAM**
* **Dinamica DRAM**



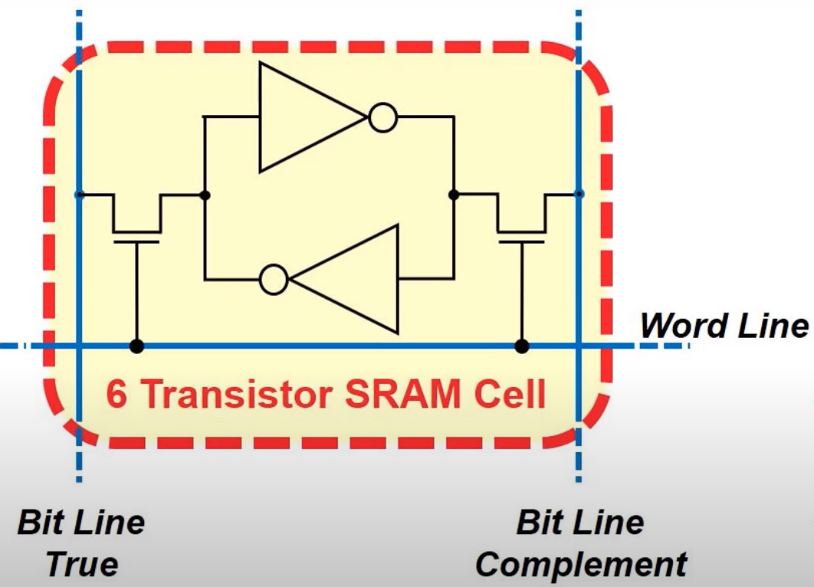
Il numero dopo DDR indica la frequenza di operatività, quello dopo PC indica il larghezza di Banda = bitrate

**SRAM**

Una cella di memoria è composta da 2 transistor e 1 LATCH/FLIP-FLOP (4 transistor).

Ogni cella memorizza 1 BIT.

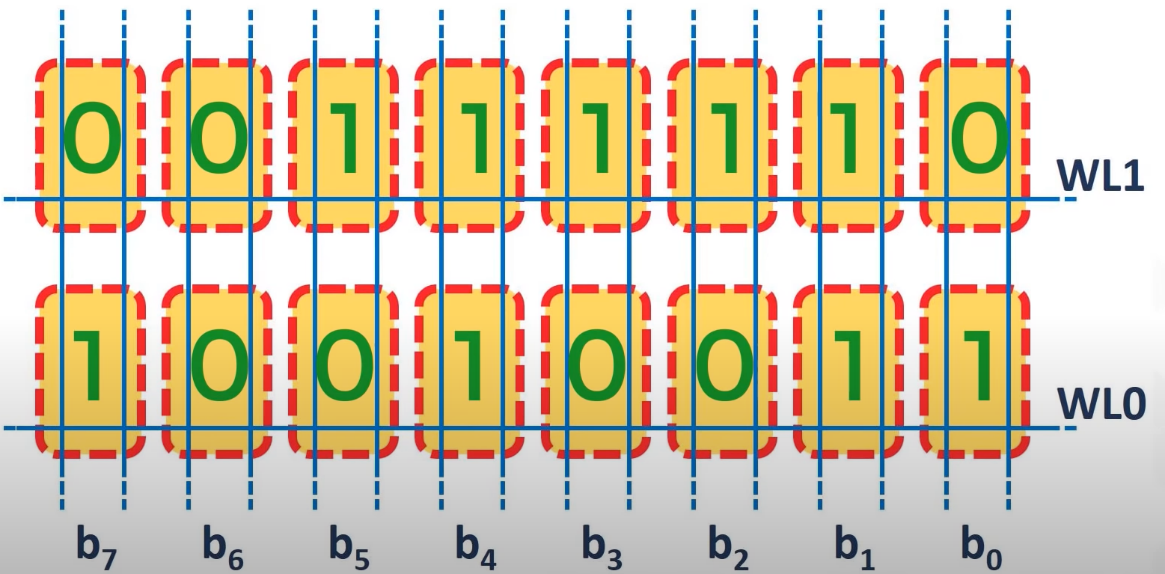
È molto dispendioso dal punto di vista economico.



**WORD LINE:** si attiva quando viene decodificato l’indirizzo di memoria, corrispondente a una locazione.

**BIT LINE**: contiene il contenuto informativo da LEGGERE o SCRIVERE in memoria.

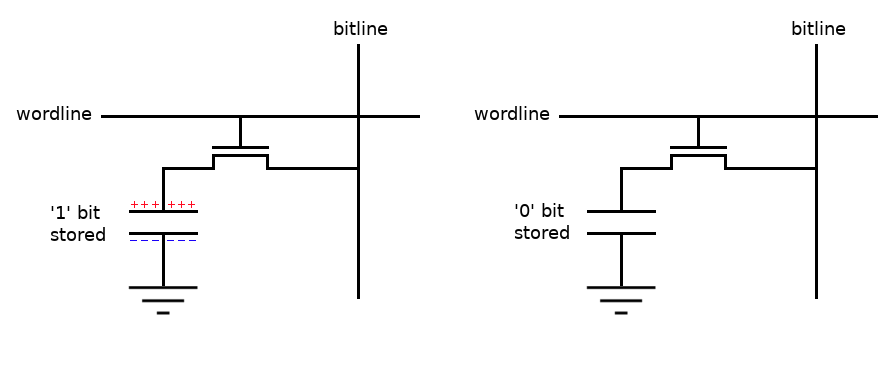
Quando la WL = 1 i transistor si chiudono.

* **Lettura:** la BL riceve il valore contenuto nella cella
* **Scrittura:** la BL invia il valore nella cella

**Le operazioni di lettura e scrittura possono avvenire all’infinito senza causare danni ai circuiti interni. Tempo Lettura = Tempo Scrittura**

* **Synchronous RAM:** architettura basata sulle pipeline, cioè in parallelo.
  + **BUS** grandi con **trasferimenti** fino a **Gb/s**
  + Molto **costose** per essere sviluppate a causa del **grande** **numero** di **componenti** richiesto (6 transistor = 1 bit)
* **Asynchronus RAM:** 
  + **velocità** di trasferimento **modesta**
  + **Risparmio energetico**

**DRAM**



Come nelle SRAM sono presenti la WORD LINE e la BIT LINE, e hanno la stessa funzione.

In questo caso **1 bit** viene **salvato** all’interno di un **condensatore**, che se è **carico = 1**, **scarico = 0**.

Un condensatore però non mantiene la sua carica all’infinito, ma si scarica in un tempo di 5 = 5 \* (R\*C).

Esso può scaricarsi sulla BL in caso di LETTURA, oppure sul transistor quando esso è aperto, quindi la WL = 0. Ciò vuol dire che non è stata selezionata la locazione.

Se si scarica si perde il contenuto informatico, perciò è necessaria una operazione di REFRESH circa 16 volte al secondo = 16Hz -> 1 / 16 = 64ms

**VANTAGGI E SVANTAGGI DRAM**

* **Più** **lente** delle **SRAM**
* **Più** **economiche**, richiedono **meno** **componenti**: 1 transistor e 1 Condensatore
* **Complessità** della **conservazione** dello stato: **operazione** di **REFRESH** **onerosa**

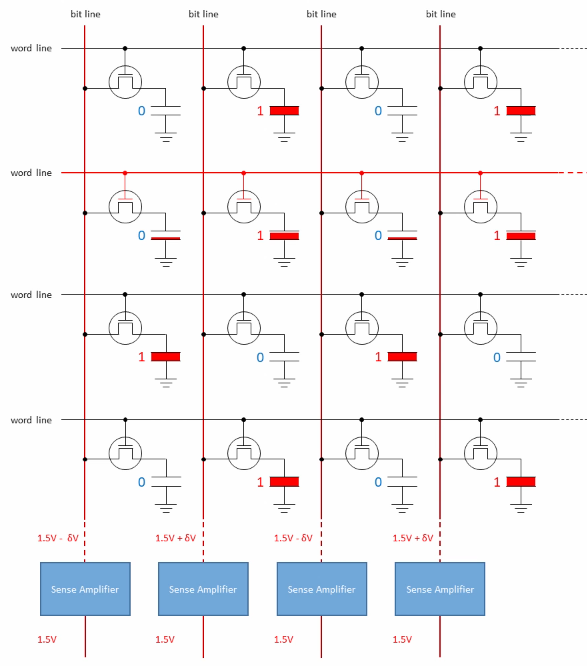
**TIPOLOGIE DI DRAM**

* **DDR = Double Data Rate:** i dati vengono trasmessi anche quando CLK è basso

Immagine che contiene testo, orologio

Descrizione generata automaticamente

REFRESH



Durante un accesso le **BITLINE** vengono impostate con un valore di V intermedio a quello tra i valori logici 0 e 1

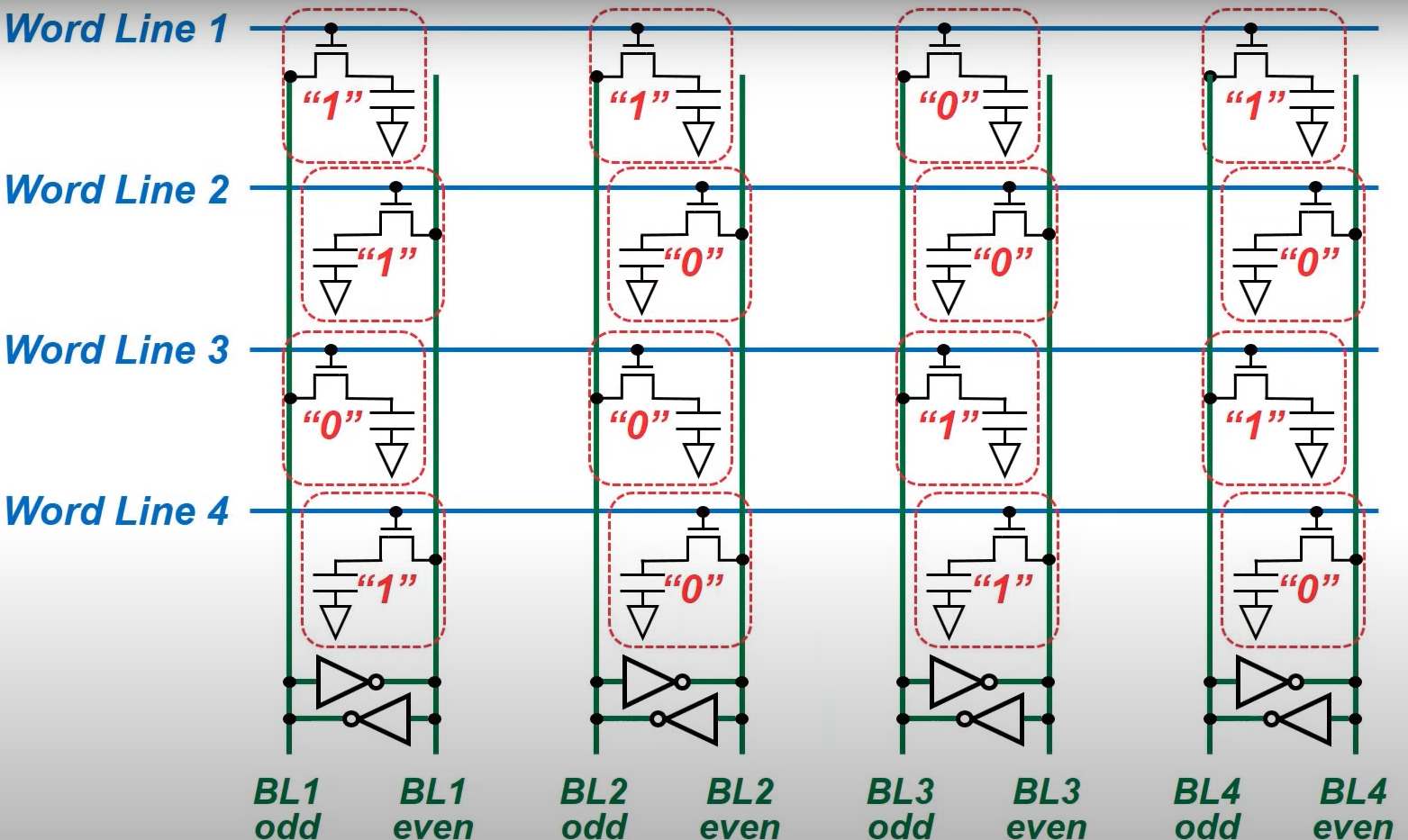
* **CELLA = 1**, il **condensatore** si **scarica** sulla **BL**, **caricandola** leggermente in **+**
* **Cella = 0**, il **condensatore** si **carica** grazie alla tensione fornita dalla BITLINE, la **BL si scarica in -**

Le variazioni di tensione vengono rilevate dai **SENSE AMPLIFIER**

* **circuiti** **logici** **SRAM** che **mantengono** il **dato** prelevato dalla cella
* **codificano** la **BL** in **uscita** con il **valore** **logico** di **V** appena **letto** dalla **cella**

**Dopo ogni operazione di lettura deve avvenire una di scrittura, per non perdere i dati appena letti.**

**I dati vengono riforniti dai SENSE AMPLIFIER**



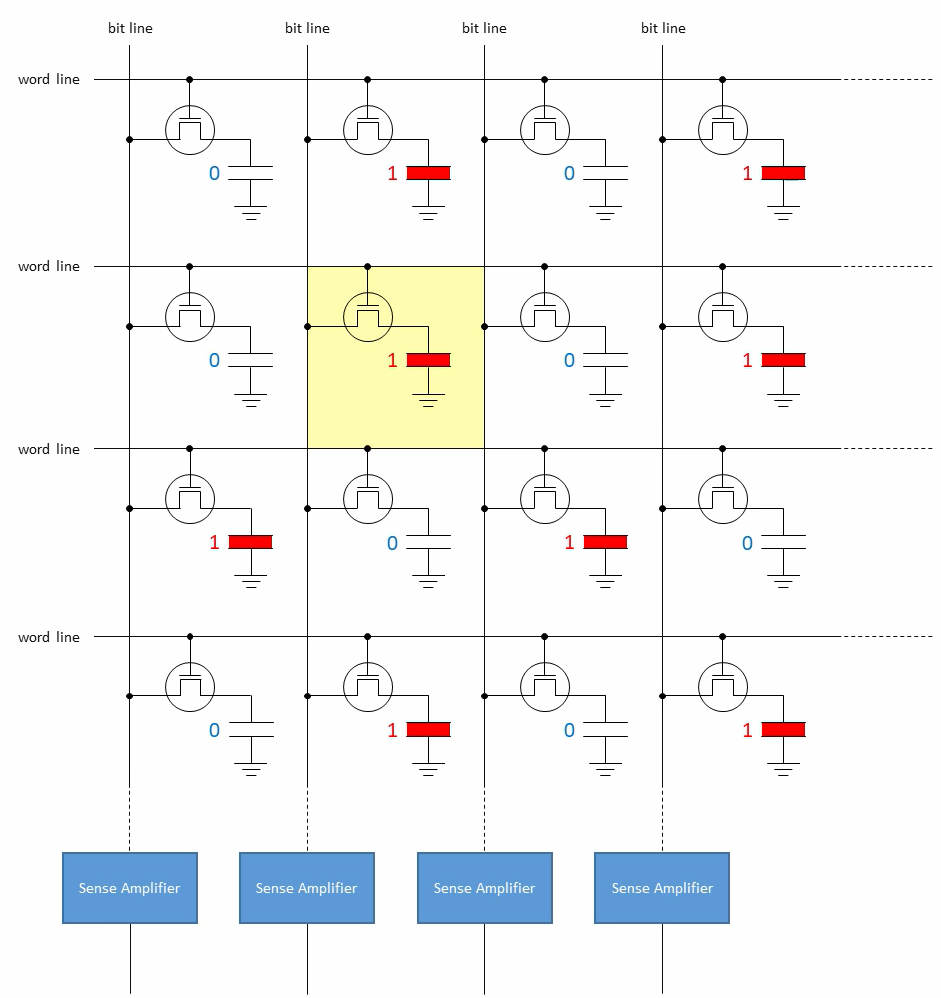
Ci sono **2 BIT LINE**, una per le **locazioni** **DISPARI** e una per quelle **PARI**.

Il circuito sotto è un **Inverter** **LOOP**.

Durante l’operazione di **REFRESH** i due **inverter** sono a una **tensione** **intermedia** **tra** i valori **0 e 1**, circa 0.5V. Se **BLodd = BLeven** si è in una **condizione** di **equilibrio**.

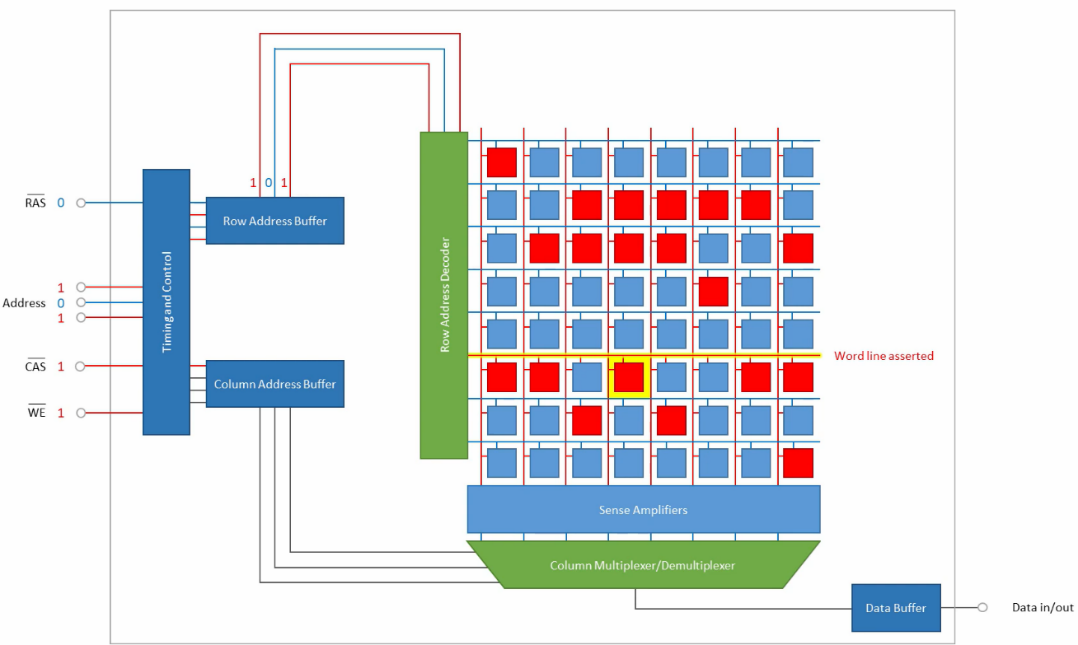
* Se **C = 1**, la sua tensione sarà maggiore di 0.5V, perciò **comincerà** ad **aumentare** quella sulla **BITLINE**. Il **LOOP** **incrementa** sempre di più **la tensione** fino ad arrivare a una condizione in cui una BL vale 1V e l’altra 0V.  
  Quando la **tensione** sulla **BL = 1V** **ricaricherà** il **C** che nel mentre si stava scaricando, **ripristinando il valore logico 1**, presente in precedenza.
* Se **C = 0**, la tensione sulla **BL = 0V**, perciò la tensione viene **subito ripristinata**.

**ACCESSO IN MEMORIA**



**Dopo ogni** operazione di **LETTURA** il **dato** viene **perso**, perciò è **necessaria** una operazione di **REFRESH** immediata.

**LETTURA E SCRITTURA IN DRAM**



not(RAS) = flag che indica se è il momento di effettuare la fase RAS. Attivo BASSO

not(CAS) = flag che indica se è il momento di effettuare la fase CAS. Attivo BASSO

not(WE) = flag che permette o meno la scrittura in memoria. Attivo BASSO

ADDRESS = numero di bit che servono a codificare un indirizzo di memoria

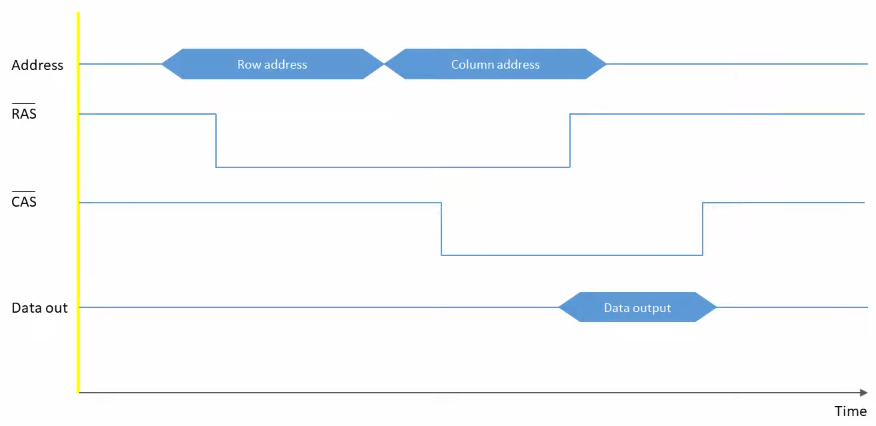
**N bit -> 2^N allocazioni possibili**

La fase di accesso in memoria si divide in 2 fasi principali:

* **RAS (Raw Address Strobe)** : individua la RIGA (WORDLINE) da attivare
* **CAS (Coloumn Address Strobe)** : individua la cella nella Locazione

Il modulo di controllo determina le fasi dal punto di vista tempistico e logico, in base ai flag attivi.

* **Indirizzi** entrano nel Modulo. Quando **RAS è attivo** **parte** la **fase di RAS**
* Nel **Raw Address Decoder** vengono **inseriti n bit** per la **codifica dell’indirizzo**.
  + **DECODER** = **n ingressi, 2^n uscite** -> **attiva l’uscita** corrispondente alla **codifica in binario degli ingressi**
  + Viene **attivata la RIGA** (WORDLINE) corrispondente alla codifica
  + **Tutta la LOCAZIONE** viene **salvata** all’interno dei **SENSE AMPLIFIER**
  + La **LOCAZIONE** viene **inserita** nel **Coloumn Address DE/Multiplexer**
* **LETTURA**:
  + **CAM**: viene **mandato** in **uscita** l’**ingresso** **corrispondente** alla **codifica** in binario **degli** ingressi **selettori** (n bit degli indirizzi)
* **SCRITTURA**
  + **CAD**: il dato in ingresso viene **mandato** in **una** delle 2^n **uscite**, **selezionata** dai **selettori**
  + **Una volta inviato il dato nei sense amplifier essi effettueranno il refresh, scrivendo in memoria il bit appena inserito**
* **CAS (Coloumn Access Strobe):**
  + **viene fornito l’indirizzo della colonna da leggere**

**DIAGRAMMA TEMPORALE DI UNA LETTURA**

**DIAGRAMMA TEMPORALE DI UNA SCRITTURA**

