

# Electronică Digitală

## Curs 16

*Iuliu Vasilescu*

Facultatea de Automatică și Calculatoare

Universitatea Politehnica București

1 iunie 2021

# Memorie

## 1 Clasificare

**RAM (Random Access Memory)** rămâne activă și păstrează datele cât timp este alimentată (cât are acel  $V_{cc}$  în prezent). Se poate clasifica în SRAM și DRAM. Ambele se pot citi și scrie, dar diferă în implementarea bitului de stocare, celula de memorie.

SRAM (static RAM) - celula statică de memorie își păstrează fără niciun ajutor informația pentru un timp nedefinit, cât timp este alimentată memoria.

DRAM (dynamic RAM) - celula dinamică are o durată limitată de ordinul zecilor de milisecunde (în jur de 64 ms, cât este garantată păstrarea informației, după care trebuie înprospătată), deci realizează un refresh periodic.

**ROM (Read Only Memory)** își păstrează datele odată cu întreruperea curentului electric.

ROM (memoria clasică) – scrisă de fabricant, și de utilizator poate fi doar citită. Păstrează informații care nu se schimbă în timp, e inflexibilă, și are utilizare limitată. Singurul motiv pentru care am prefera să o folosim este pentru că este ieftină, sau dorim posibilitatea de a nu putea fi schimbată memoria (ex : tabela de fonturi pentru imprimantă).

PROM (programmable ROM)/OTP (one time programmable) – poate fi scrisă o singură dată și după aceea doar citită. Este primită de la fabricant ștersă cu niciun fel de informație. Este puțin mai flexibilă.

EPROM (erasable programmable ROM) – poate fi programată și ștersă pentru a fi programată din nou. Seamănă cu RAM, dar partea de ștergere nu se face electric, ci prin expunere la lumina UV, iar partea de programare este mai lentă. Nu se

poate programa continuu în timpul rulării unui program, și de asemenea numărul de cicluri de programare și ștergere este limitat.

EEPROM (electricly erasable programmable ROM) – în mod normal read only, cel mai adesea folosim pentru citire, dar poate fi programată și ștearsă electric. Avantaj față de RAM – păstrează datele după ce oprim curentul. Dezavantaj : este mai greu de scris și de șters, procesele durează mai mult timp, numărul de cicluri până memoria se uzează este limitat. Din EEPROM face parte și memoria FLASH (folosită în SSD-uri, stick-uri USB, memoria telefoanelor etc.)

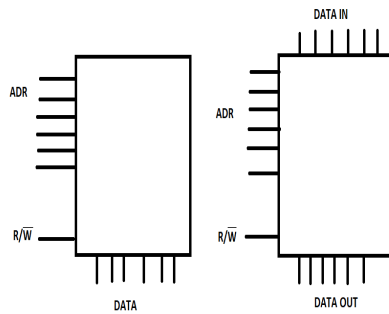
Cele mai folosite memorii:

- SRAM
- DRAM
- FLASH

FRAM (Ferroelectric RAM) - cât timp este alimentată funcționează ca o memorie RAM cu performanțe de viteză, scriere, citire similare, dar când este oprit curentul își păstrează datele ca și cum ar fi o memorie ROM.

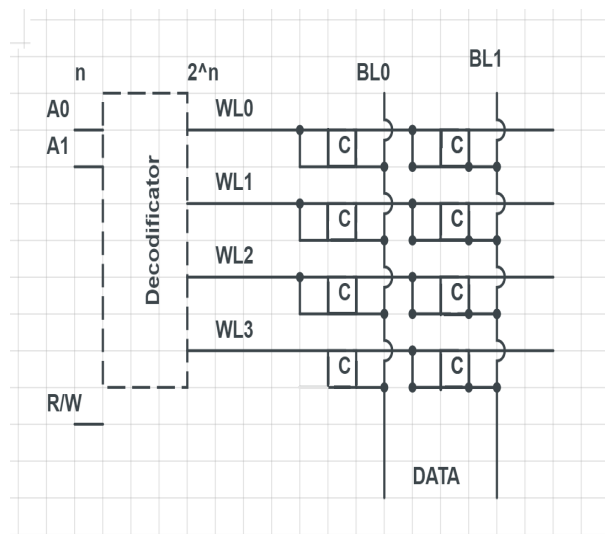
## 2 Structura generală a memoriei

E un blackbox cu o interfață foarte simplă. Îi dăm o adresă, o comandă (Read/Write negat) și memoria respectivă citește ce se află la acea adresă. O adresă pe  $n$  biți poate adresa  $2^n$  la  $n$  locații de memorie.



Pentru Read primim pe magistrala de date (DATA) informația aflată la adresa respectivă, pentru Write putem să îi dăm noi pe magistrala de date pentru a scrie în căsuța respectivă.

Uneori putem să separăm magistralele în DATA IN și DATA OUT, alteori sunt comune. Magistrala DATA OUT poate fi de diferite lățimi: 8, 16, 32 de biți. Este o lățime fizică dată de implementarea chipului, care poate fi diferită de interpretarea ei în procesor pentru a optimiza spațiul și viteza magistralei de date.



Structura:

1. Un decodificator de adresă, cu  $n$  linii de intrare ( $A_0, A_1$  etc) și  $2^n$  linii de ieșire, numite Word Line (WL) – care primește biții de adresă și activează doar o linie. De-a lungul Word Line-urilor sunt conectate celulele de memorie. Fiecare celulă stochează exact un bit.

În momentul în care avem o adresă la intrarea decodicatorului, exact un Word Line se activează cu celulele de memorie. Datele sunt conectate la Bit Line-uri ( $BL_0, BL_1$ ) și sunt disponibile în DATA.

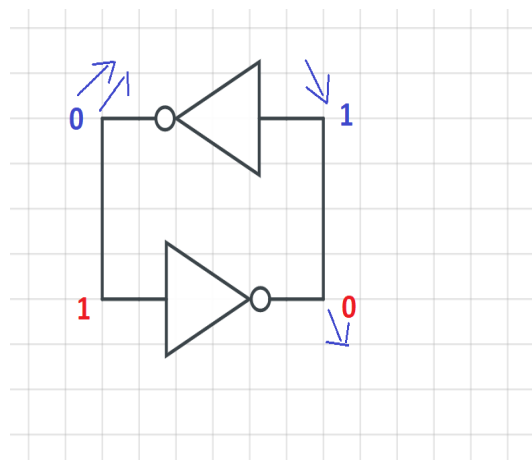
Rolul structurii este să activeze o linie din array. Celulele sunt apoi cuplate la Bit Lines, și exact o celulă este cuplată la o linie de date, iar celelalte celule sunt deconectate și doar păstrează informația.

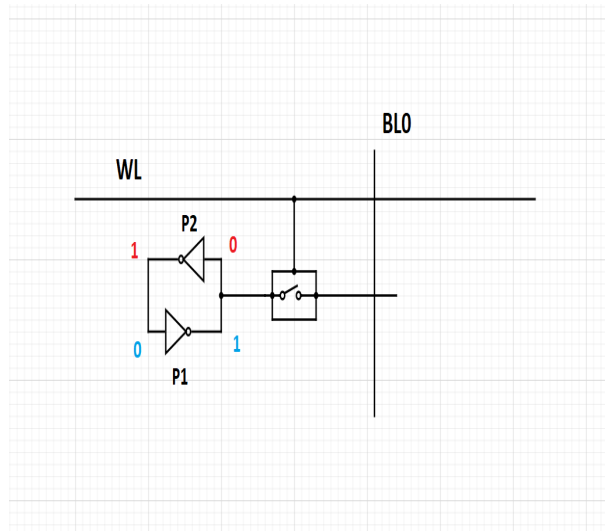
În exemplu sunt 2 biți per adresă, în mod clasic este un octet pe adresă, adică 8 astfel de celule.

2. Semnal de Read sau Write – este folosit de circuitul din jurul blocului de memorie pentru a comanda celulele pentru scriere și citire.

### 3 Celula de memorie statică ( SRAM)

E un bistabil. Celula de memorie este simplă: două inversoare. Are reacție pozitivă. Circuitul va avea 2 stări stabile.

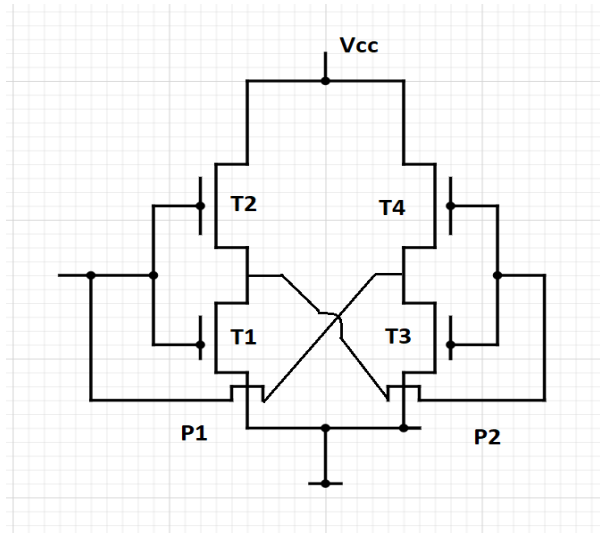




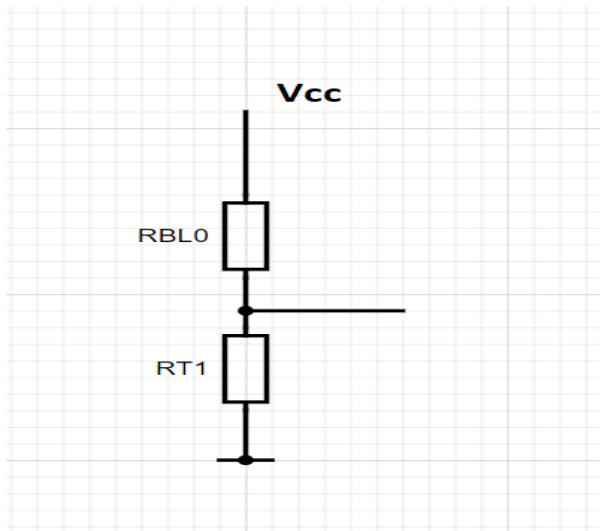
În momentul în care am conectat celula, ea va determina nivelul logic de pe bitline care este citit mai departe de restul sistemului. Citirea este trivială, se activează Word Line.

Dacă în dreapta avem 0 logic și vrem să scriem 1 în celula de memorie, atunci tranzistorul este activ și trage către 0, primind semnal pozitiv la intrare.

În practică, la scriere, tranzistoarele sunt de dimensiuni mici, astfel încât au rezistență mare la conducție. Deci sunt ca niște rezistențe destul de mari – suficient de mici cât să păstreze nivelul logic și să nu fie sensibile la zgomot, dar destul de mari astfel încât un semnal puternic pe bitline de 1 logic construit cu niște tranzistoare mai puternice, adică cu rezistențe mai mici, să poată să suprascrie informația.



În momentul scrierii, avem tranzistorul din celula de memorie, T1, mai mic cu o rezistență mai mare, și atunci tranzistoarele care comandă Bit Line-ul pentru scriere (exterioare circuitului de memorie) sunt mai mari, deci cu o rezistență internă de conducere mai mică (conduc mai bine curentul), astfel încât RBL0 e mai mică decât RT1.



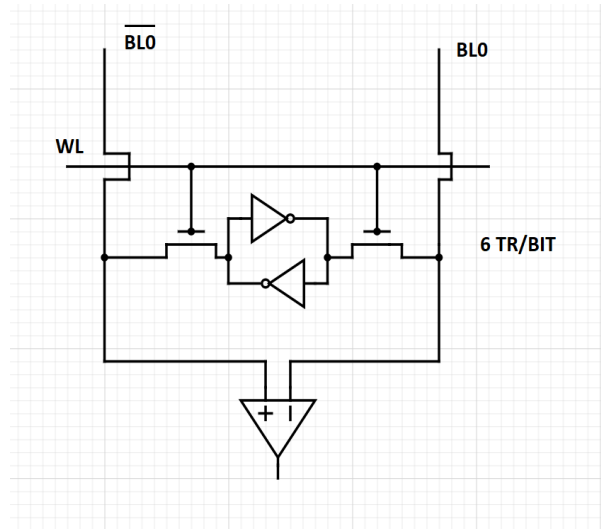
$V1 = RT1 / (RT1 + RBL0) > Vcc/2$  și asta va face ca circuitul bistabil să basculeze în starea cealaltă. Pentru scriere se folosește diferența de tărie între cele

două tranzistoare. Cele de jos sunt suficient de tari să le putem citi în siguranță, dar sunt mai slabe decât tranzistoarele care vor suprascrie informația. Din moment ce niciunul nu e aproape de un conductor perfect, în momentul scrierii nu se petrece niciun scurtcircuit.

La fel și în acest caz când T2 este în conducție. T2 va trage potențialul de ieșire la VCC cu RT2 care e ceva mai mare. Dacă dorim să scriem informații tragem potențialul către masă cu un tranzistor cu potențialul mai puternic

1 inversor și 4 tranzistoare => 8 tranzistoare pentru un bit de memorie

Putem optimiza numărul de tranzistoare pentru fiecare celulă de memorie. În practică, celula va conține tot același inversor.



Și în loc de poartă de transfer sunt puse două tranzistoare NMOS (deoarece au performanțe superioare, întrucât mobilitatea electronilor este mai bună decât mobilitatea gurilor) conectate la Word Line și două Bit Line-uri. Bit Line-urile sunt implementate în straturile de metal. Stratul de siliciu – tranzistoarele, peste care vine stratul de metal în care sunt implementate firele. Firele se suprapun, nu cresc cu mult dimensiunea fizică a unei celule.



Avantaj – simetria sistemului: Tranzistoarele NMOS conduc mai bine nivelul logic 0 cu cât diferența dintre potențialul de poartă și potențialul sursei este mai mare. Bit line și bit line negat garantează că una dintre ele este 0 în orice stare ar fi bistabilul, practic este conectat mai bine nivelul logic de 0. Pentru citirea bitului propriu zis punem un comparator care ne va da nivelul logic.

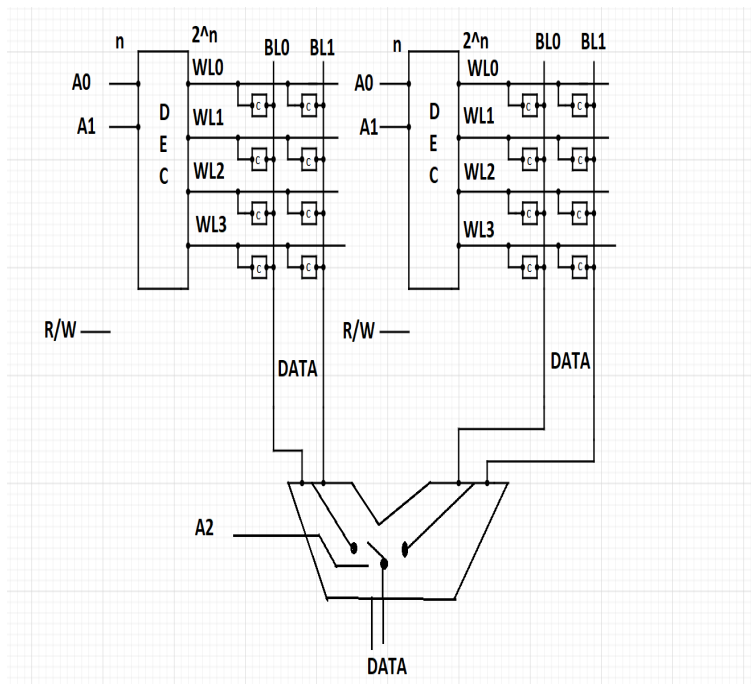
Avantaj: Este foarte rapidă și se poate implementa practic la limita tehnologică cu structura aceasta. Este folosită cel mai adesea în zona de cache a unui procesor.

Dezavantaj: Dimensiune mare. Avem 6 tranzistoare pe bit.

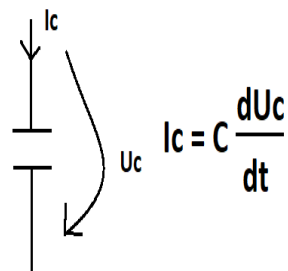
De remarcat: în momentul în care nu se scrie și nu se citește informație din memorie, ea nu consumă energie electrică. Cele două porți care formează un bistabil nu comută la un anumit nivel logic. Întotdeauna unul din tranzistoare este blocat. Poate rămâne cu consum minim de energie la infim, se consumă totuși doar din rezistențe parazite (dacă porțile nu sunt complet izolate, dacă avem mici zone de rezistență foarte mare care pot face curentul să scurgă), dar curentul e neglijabil de ordinul mA.

Un loc unde se mai consuma energie este în capacitățile parazite. În momentul în care avem o memorie de dimensiune mai semnificativă (ex de 1MB), Bit Line-urile devin lungi. Ele reprezintă linii de date, zone de cupru desenate deasupra unui izolator, sub care sunt alte linii încărcate electric. Asta formează condensatori, ai căror capacitate crește cu lungimea liniei de date. Cu cât e un condensator mai mare, cu atât se încarcă mai mult și se cheltuie mai multă energie. Pe scurt, cu cât crește dimensiunea array-ului de memorie, cu atât crește și energia necesară pentru a citi datele.

În implementarea fizică se preferă două decodificatoare identice în locul unuia foarte mare cu mai multe linii de intrare, legate la un comutator care conectează ieșirea circuitului la unul dintre cele două blocuri de memorie. Se reduce aproximativ la jumătate lungimea liniilor de date și se reduce capacitatea parazită, și implicit timpul de răspuns.



## 4 Celula de memorie dinamica (DRAM)



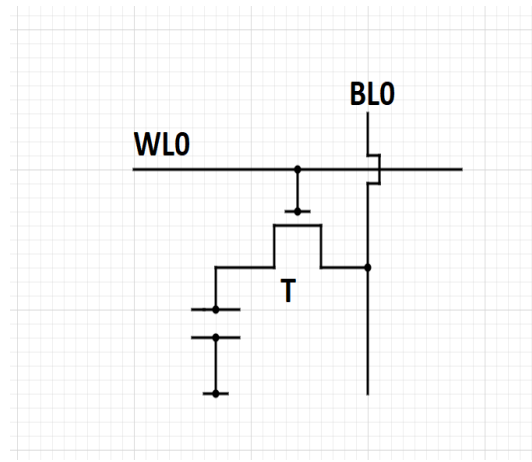
Informația e stocată cu un condensator. Dacă nu trece curent prin el și dacă nu există rezistențe parazite în sistem, poate fi încărcat până la o anumită tensiune și își păstrează tensiunea respectivă.

Metoda de stocare a informației este să încărcăm condensatorul fie la  $V_{cc}$  pentru a stoca 1 logic, fie să îl descărcăm la 0 volți pentru a stoca 0 logic.

Cu cât condensatorul e mai mic, cu atât vom avea o densitate mai mare de cellule de memorie pe mm patrat, și implicit un cost mai redus pentru bitul de memorie. De asemenea, poate fi încărcat și descărcat mai repede.

Pentru o memorie rapidă și densă, vrem o capacitate cât mai mică. Dar se stochează o cantitate mai mică de sarcină și atunci niște curenți paraziți pot schimba în timp valoarea logică stocată pe condensator.

### Structura memoriei DRAM



1 celula de memorie conectată la un bit line (BL0) în funcție de starea unei linii de cuvânt (WL0)

Dacă WL este pe 0 logic, tranzistorul este blocat și nu ajunge niciun curent prin condensator, care își va pastra tensiunea. Când WL este 1, tranzistorul este în conducție și condensatorul este cuplat la Bit Line și implicit potențialul de pe condensator este egal cu potențialul de pe Bit Line.

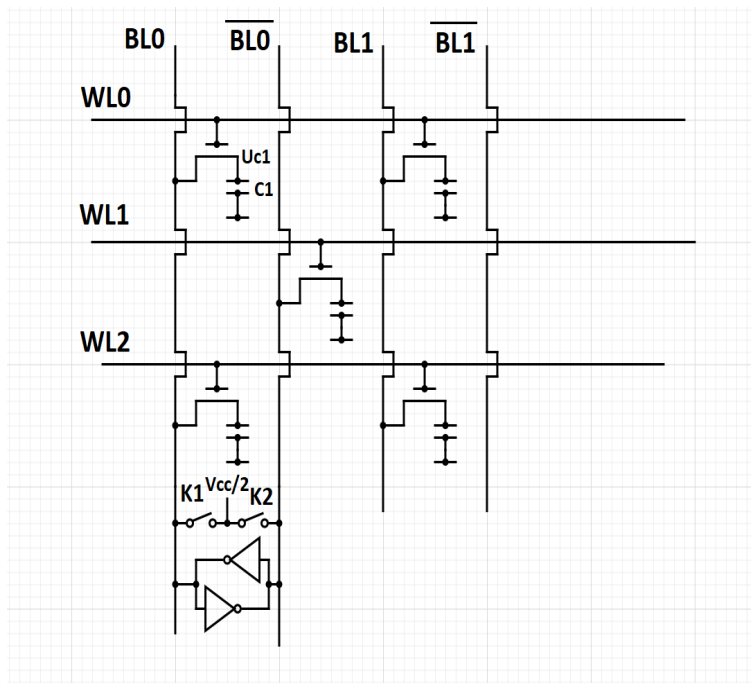
Pentru citire, cuplăm WL la 1 și măsurăm tensiunea de pe BL să vedem dacă e aproape de 1 sau 0 logic. Condensatorul e cât mai mic posibil, tensiunea variază în timp.

Dacă vreau să scriu informație în celula de memorie, activez WL corespunzător BL și punem pe BL octetul care dorim să fie scris.

Față de memoria statică, dimensiunea unui bit de celule s-a redus la un tranzistor și un condensator. Condensatorul în practică este tot un tranzistor în care se folosește doar zona de poartă. Deci celula de memorie de 8 tranzistoare s-a redus la 2.

Din punct de vedere practic, tensiunea de pe condensator variază în timp. Pentru citire folosim tot un bistabil, inițial la o stare intermediară, care va bascula în funcție de tensiunea de pe condensator.

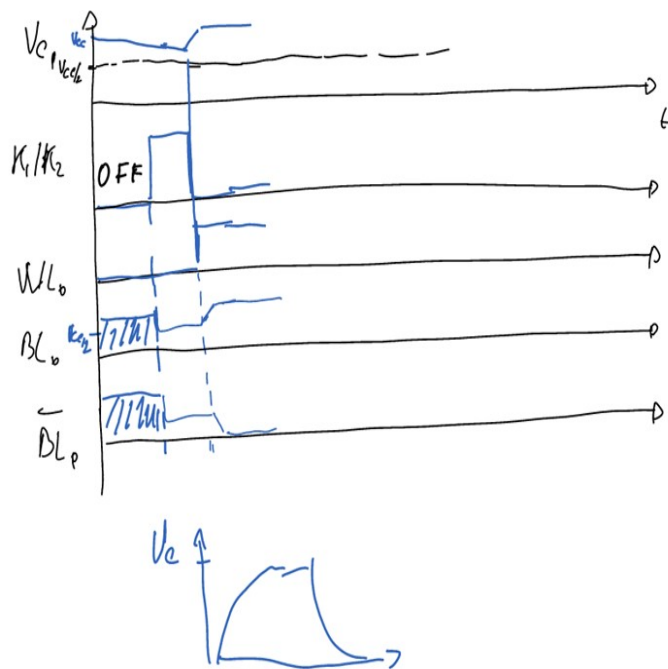
Zonarea alternativă care permite să folosim simultan două linii de date pentru citire



Înainte de a citi o celulă de memorie, comutatoarele K1 și K2 sunt închise temporar și BL0, BL0 negat sunt legate la  $V_{cc}/2$ . Avem un bistabil care are un punct fix la jumătatea tensiunii de alimentare. E o stare instabilă, orice perturbație va strica echilibrul, și pe asta ne bazăm. După aceea se deschid comutatoarele și este activat WL liniei pe care vrem să o citim.

(Condensatorul între W0 și W1) Condensatorul este cuplat la una dintre BL. Inițial amândouă sunt  $V_{cc}/2$ , condensatorul are o tensiune de dorit mai mare sau mai mică, dacă este stocat 1 sau 0 logic, și tensiunea este cuplată la BL corespunzător. Va pica acea perturbație, pozitivă sau negativă, care va fi amplificată de circuitul cu comutatoare și bistabil, de reacție pozitivă. Inițial cele 2 linii sunt la  $V_{cc}/2$ , aplicăm perturbația, dacă e mai mare ca  $V_{cc}/2$  spre 1 logic crește potențialul pe BL0, dacă este mai mică decât  $V_{cc}/2$  va scăde. Cele 2 BL se vor stabiliza în 0 sau 1. Un efect al momentului de citire: dezechilibrul se va propaga și în condensator. Dacă nu are tensiunea aproape de  $V_{cc}$ , după ce se va activa WL corespunzător, el va ajunge înapoi la  $V_{cc}$  din cauza circuitului cu reacție pozitivă.

### Diagrama în timp



Vc1- tensiunea scade lent la  $V_{cc}/2$ , nu este stabilă în timp, există tensiuni parazite.

Inițial K1, K2 au starea OFF, și ajung la ON la un moment dat pentru a preîncărca liniile de date.

Inițial nu știm cum sunt Bit Line-urile. Se vor stabiliza la  $V_{cc}/2$  la un moment dat cand comutatoarele ajung pe ON.

Apoi când dezactivăm K1 K2 se activează WL. În momentul în care am activat WL, tensiunea de pe condensator este transferată la BL0, și fiind un pic mai mare de  $V_{cc}/2$  va trage potențialul în sus la  $V_{cc}$ , mutând potentialul de la BL0 negat către 0. Când se activează WL, și condensatorul este încărcat înapoi la  $V_{cc}$ .

În momentul citirii se și rescrie informația pe condensator. Am citit și am și reîmprospătat informația pe condensator. Informația este garantată de producător să rămână o durată de timp de ordinul milisecundelor (64ms standard). Odată la 64ms trebuie citite toate celulele de memorie din circuitul de memorie, fie că e sau nu nevoie de ele. În general există un controller interbus între CPU și memorie care face o citire periodică a circuitului de memorie, astfel încât să fie continuu reîmprospătată informația.

Dacă vrem să scriem în DRAM, setăm BL la valoarea pe care vrem să o scriem (daca vrem să scriem 0 punem BL0 la 0 si BL0 negat la 1), informația e ținută de bistabil, și activăm WL, informația e apoi transferată în condensator. Condensatorul poate avea o informație anterioară, dar nu e suficient de puternic pentru a schimba starea acestor circuite, ținând cont și de capacitatea parazită de pe BL.

Scrierea = încărcarea sau descărcarea unui condensator. Avem o încărcare exponențială, o descărcare exponențială, nu este instantanee. Cu cât este mai mic tranzistorul, cu cât rezistența este mai mare, cu atât constanta de timp e mai mare, încărcarea se face mai lent. Ne trebuie o capacitate importantă cât să ne fie garantată stocarea pe cele câteva zeci de ms.

Din timpul total al memoriei, o bună parte din el este dedicat reîmprospătării. Pe perioada reîmprospătării nu putem citi decât linia de memorie care este în acel moment reîmprospătată. Dacă avem de făcut altă citire trebuie să întrerupem procesul de reîmprospătare și apoi să revenim la el. Pot introduce o latență/intârziere, care este în general limitată dacă punem mai multe astfel de blocuri în paralel. Lucrurile se pot paraleliza la nivel de siliciu, și pot reduce la un minim efectul latenței. Însă, vor rămâne procesele fizice de scriere și citire care au o limită. Nu putem pune un condensator prea mic, să îl încărcăm suficient de repede pentru că s-ar descărca la fel de repede și ar necesita o reîmprospătare mai deasă.

Memoria DRAM are densitate mai mare decât SRAM, deci preț mai mic (avantaje), dar latență mai mare și viteză mai mică de citire și scriere (dezavantaje). Consumul de energie: circuitul de reîmprospătare funcționează continuu (pentru fiecare BL), fie că citim sau scriem sau nu facem nimic, pentru a păstra informația. Memoria DRAM consumă energie continuu, fie că este folosită sau nu. De aceea, microcontrollerele precum Arduino folosesc SRAM.

Deși garanția producătorului este în jur de 60ms că memoria își va păstra informația, pentru a obține garanția la toți biții, circuitul este proiectat pentru o stocare pe termen mult mai îndelungată. Dacă oprim tensiunea de alimentare a memoriei, condensatoarele rămân izolate, tranzistoarele sunt oprite. Întrucât timpul real de stocare este de ordinul secundelor sau minutelor, memoria DRAM își păstrează informația o vreme după oprirea tensiunii. La pornire își inițializează cu 0 zona de memorie, dar poate pune probleme de securitate în anumite sisteme. Și de asemenea, dacă memoria nu e inițializată în program pot rămâne informații de la o pornire la cealaltă.

De aceea când resetezi ceva se precizează la unele proceduri că îl deconectezi de la tensiunea de alimentare și îl lași o perioadă de 30 de sec oprit. Condensatoarele de filtrare de pe sursele de alimentare pot păstra o vreme tensiunea de alimentare astfel încât circuitul să nu fie oprit decât dacă a fost scos din priză suficient timp.