

## Cursul nr. 11

## C11.1 Memorii RAM statice

Memoriile semiconductoare sunt referite adesea ca RAM ceea ce înseamnă *Random Access Memory*, sau memorii cu acces aleator. Aceasta implică faptul că orice cuvânt din memorie poate fi accesat în același timp.

O memorie este constituită dintr-o arie de dispozitive de memorare. Fiecare dispozitiv poate stoca un bit, un byte sau un cuvânt. Dimensiunile uzuale pentru lungimea cuvintelor sunt: 8, 16, 32 și 64. Un sistem digital dispune de o memorie care poate stoca un număr mare de cuvinte. Fiecare cuvânt poate fi adresat utilizând liniile de adrese ale memoriei. În plus, pe lângă liniile de date și liniile de adrese, o memorie mai dispune și de linii de validare a citirii sau scrierii cuvântului selectat. Acest tip de memorie este ușor de utilizat. O locație, sau o adresă, se selectează prin plasarea pe liniile de adrese a combinației binare ce reprezintă adresa locației. Dacă urmează să aibe loc o operație de citire, este activat semnalul de citire, iar datele stocate în locația pot fi preluate de pe liniile de date. Dacă urmează să aibe loc o operație de scriere, datele ce urmează să fie scrise vor trebui plasate pe liniile de date, apoi combinația binară a adresei locației de memorie pe liniile de adrese și se activează semnalul de scriere.

Descrierea unei memorii din punctul de vedere al unui utilizator reprezintă o viziune abstractă.

Implementarea memoriilor se bazează pe două tehnologii, memoriile RAM statice și memoriile RAM dinamice:

- SRAM
- DRAM

Este important de reținut că orice dispozitiv capabil să memoreze valori binare poate fi utilizat ca memorie și indiferent ce s-ar utiliza, viziunea abstractă a utilizatorului rămâne în mare aceeași.

Termenul **SRAM** indică faptul că datele depuse în memorie sunt stabile. Circuitele flip-flop sunt o variantă de circuite de memorare stabile.

Un dispozitiv simplu de memorare cu această proprietate este reprezentat în fig. C11.1.

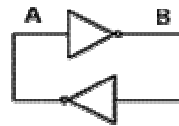


Fig. C11.1 Dispozitiv simplu de stocare

Dispozitivul de stocare este construit cu două porți NU; ieșirea fiecăreia este conectată la intrarea celeilalte, sau altfel spus, sunt conectate în cruce. Dacă punctul A este 1, atunci punctul B va fi 0; dacă punctul A este 0, atunci punctul B va fi 1. Dacă este posibil să fie stabilită o valoare inițială a unuia din punctele A sau B, acest dispozitiv va menține acea valoarea un timp nedefinit, presupunând că dispozitivul este alimentat tot acest timp. Considerând cele descrise, dispozitivul este un **dispozitiv static**.

Valoarea punctelor A și B poate fi inițializată utilizând circuitul reprezentat în fig. C11.2.

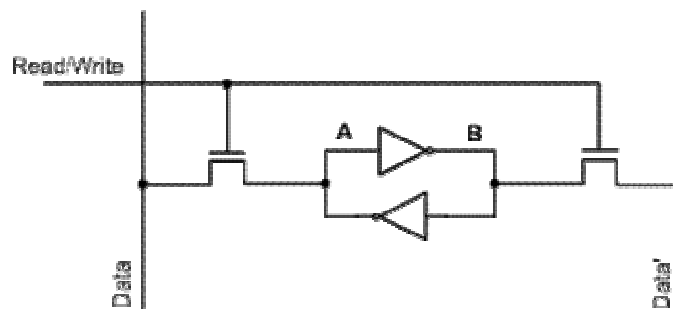


Fig. C11.2 Dispozitiv static de memorare cu circuit de inițializare

Tranzistoarele, numite și tranzistore de trecere, plasate pe partea dreaptă și pe partea stângă a circuitului de memorare, permit înscrierea unei stări în punctele A și B atunci când linia Read/Write este activă. Dacă linia Read/Write nu este activă, tranzistoarele de trecere nu conduc și liniile Data și Data' nu sunt conectate spre dispozitivul de memorare ceea ce face informația plasată pe liniile de date să nu poată ajunge la punctele A și B. Pentru ca o operație de scriere să aibă loc, se plasează valoarea dorită a fi înscrisă pe linia Data și complementul acesteia pe linia Data', după care se activează linia Read/Write. Tranzistoarele de trecere intră în conducție și valorile de pe liniile Data și Data' vor fi transferate către punctele A și B.

O poartă NU poate fi construită cu două tranzistoare MOS, astfel o celulă RAM statică, ce este capabilă să memoreze un bit de informație, poate să fie construită cu 6 tranzistoare MOS. În fig. C11.3 este ilustrat un model abstract al unei celule RAM statică, model ce ascunde detaliile constructive.

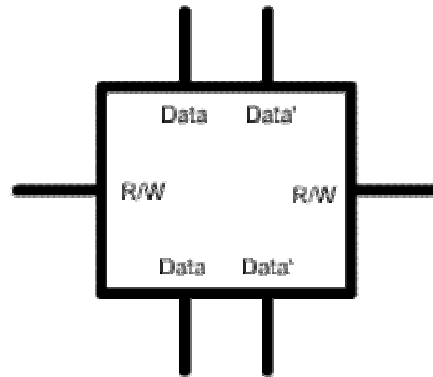


Fig. C11.3 Reprezentare abstractă a unei celule RAM statică

Pentru a forma o memorie pentru un cuvânt, se plasează într-un același rând atâtea celule de memorare cât este lungimea cuvântului. În fig. C11.4 se arată cum este construit un dispozitiv pentru memorarea unui cuvânt de patru biți.

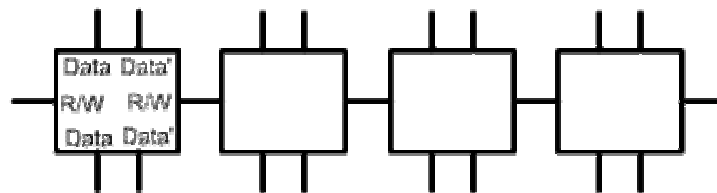


Fig. C11.4 Dispozitiv pentru memorarea unui cuvânt de 4 biți

Observați că prin activarea liniei R/W din stânga, toate cele patru celule sunt conectate la liniile Data și Data'.

O memorie ce corespunde necesităților reale trebuie să fie capabilă să stocheze un număr mare de cuvinte. Aceasta se realizează prin stivuirea mai multor celule de memorare a unui cuvânt, așa cum ilustrează fig. C11.5.

Atunci când se activează o linie R/W, valorile interne din celulele de memorare de pe rândul respectiv sunt transferate pe liniile Data și Data' sau valorile plasate pe liniile Data și Data' sunt transferate în celulele de memorare ale aceluși rând. La un moment dat, este permis unui singur semnal R/W să fie activ. Pentru a ne asigura că un singur semnal R/W este activ, se adaugă structurii, pe partea stângă a ariei de celule, un decodor 2:4.

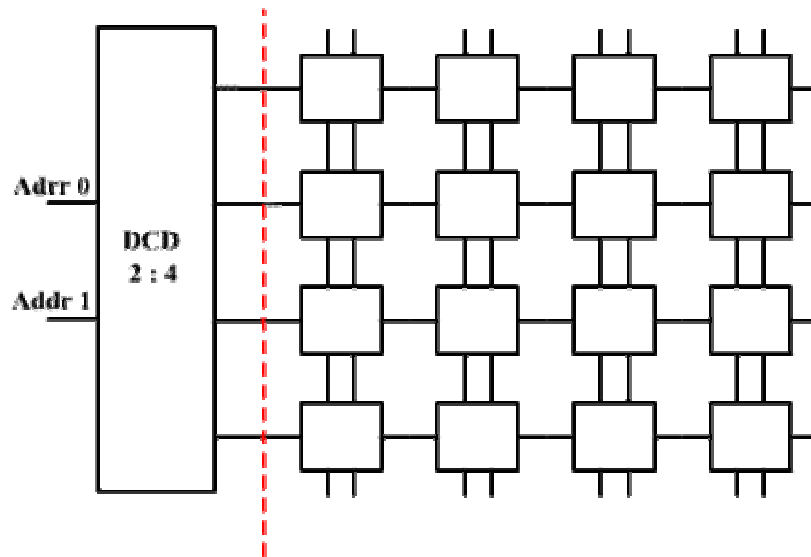


Fig. C11.5 Memorie pentru mai multe cuvinte de 4 biți

În urma acestei modificări, primul cuvânt este selectat când la intrarea decodorului se aplică 00; valoarea aplicată la intrarea decodorului este numită **adresă de memorie**. Următorul cuvânt se află la adresa 01, iar cel mai de jos cuvânt se află la adresa 11.

În fig. C11.6 este arătată reprezentarea abstractă a unei memorii de 4 cuvinte a câte 4 biți.

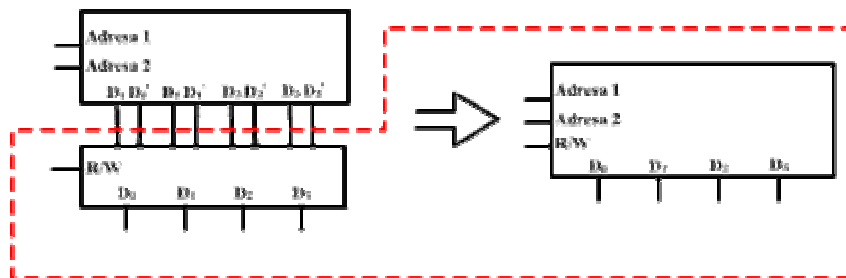


Fig. C11.6 Reprezentarea unei memorii de 4 cuvinte a câte 4 biți

Modulul de memorie este funcțional, dar este dificil de utilizat din cauza celor patru linii de date și a complementelor acestora (sunt prea multe). Utilizatorul ar trebui să poată să scrie și să citească informație fără a mai considera și liniile negațe. Cu cât memoria este mai mare pe dimensiunea verticală (crește numărul de cuvinte), inversoarele celulei de memorare vor comanda greu liniile lungi ce conectează toate celulele unei coloane. Pentru a crește viteza de lucru la memoriile mari, fiecărei coloane din structură i se adaugă un amplificator. Totodată, amplificatorul sesizează diferența dintre liniile Data și Data'. Pe lângă amplificatoare, se mai completează structura cu circuite tampon (*buffer*).

Dispozitivul ilustrat în fig. C11.6 dispune de o singură linie I/O (*Input/Output* - Intrare-Ieșire) pentru fiecare bit al cuvântului; bitul este citit sau scris pe aceeași linie. Există memorii care au linii separate pe a accesa un bit în mod scriere, respectiv în mod citire.

Observați că atunci când linia R/W este activă, dispozitivul este în mod citire, iar atunci când nu este activă, dispozitivul este în mod citire; nu există linii diferite pentru a selecta modurile scriere, citire.

## C11.2 Memorii RAM dinamice

Memoria RAM dinamică este asemănătoare cu memoria RAM statică, cu excepția faptului că în loc să stocheze informație un timp nedefinit, aceasta pierde cu timpul informația. Deși pare ridicol, memoria RAM dinamică prezintă totuși o serie de avantaje; problema pierderii informației este destul de ușor rezolvată. O celulă de memorie RAM dinamică, capabilă să stocheze un bit de informație, este construită cu un singur tranzistor și un condensator, așa cum se arată în fig. C11.7.

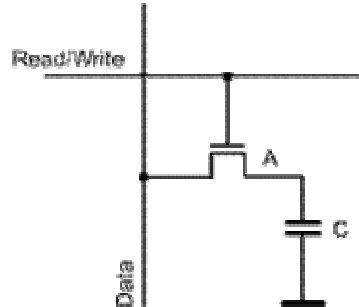


Fig. C11.7 Celulă DRAM

Condensatorul este elementul care memorează starea; într-un univers ideal, acesta ar stoca informația un timp nedeterminat. În universul real, sarcina electrică stocată în condensator se scurge încet și aceasta trebuie înlocuită cu regularitate. Atunci când se activează linia Read/Write, valoarea stocată în condensator este plasată pe liniile de ieșire date; dacă are loc o operație de scriere, valorile plasate pe liniile de date sunt stocate în condensator.

Abstracția prezentată pentru memoriile RAM statice poate fi folosită și pentru memoriile dinamice, pentru a ascunde detaliile constructive. În această abstractizare, cele două tipuri de memorie se comportă aproximativ la fel, diferența constă doar în problema pierderii informației, problemă caracteristică memoriilor RAM dinamice.

Atunci când se stochează informație într-o memorie RAM dinamică, datorită imperfecțiunilor condensatorului, aceasta se pierde cu timpul. În medie, o celulă RAM dinamică pierde informația în 32 de milisecunde. Ori de câte ori este citită informația dintr-o celulă de memorie, condensatorul este descărcat și informația nu mai poate fi citită a doua oară, aceasta se pierde. Atunci când se face o operație de scriere în memorie, condensatorul este încărcat, prin tranzistor, de către un agent extern.

Ambele probleme de pierdere a informației, în timp și la citire, pot fi depășite. Problema pierderii în timp a informației este rezolvată prin declanșarea unui proces cunoscut ca **împrospătare** (*refresh*). Procesul poate fi înțeles mult mai ușor dacă se descrie pentru început cum este rezolvată problema pierderii informației în urma unei citiri.

În urma citirii unei celule, condensatorul este descărcat; sarcina acumulată este folosită pentru a genera curent pe liniile de date pe care este citită informația. Circuitele logice prin care s-a efectuat citirea au recepționat valoarea cerută, dar totodată au descărcat condensatorul. O soluție simplă pentru a nu pierde informația înscrisă în celulă, este aceea de a rescrie imediat informația după ce aceasta a fost citită. Acest lucru poate fi realizat ușor utilizând circuite suplimentare. Cu alte cuvinte, o citire va fi de fapt o citire urmată imediat de o scriere înapoi a valorii citite. În acest fel circuitul de citire va primi valoarea cerută și condensatorul va fi reîncărcat.

Dacă celula de memorie nu este accesată în nici un fel, nici pentru citire, nici pentru scriere, mai mult de 32 de milisecunde, informația din aceasta se va pierde. O soluție o reprezintă citirea fiecărei celule de memorie din aria de celule la fiecare 32 milisecunde. Așa cum a fost precizat în paragraful anterior, citirea va fi asociată cu o scriere și astfel datele vor fi împrospătate. Logica atașată la liniile de date va ignora starea liniilor de date în timpul ciclurilor de împrospătare.

**EXEMPLUL 11.1**

O memorie cu capacitatea de 32 Mbyte conține 268,435,456 ( $32 \cdot 1024 \cdot 1024 \cdot 8$ ) de celule de memorare. Dacă fiecare celulă necesită împrospătare la fiecare 32 milisecunde, ar fi necesar un sistem mai rapid decât oricare existent în prezent, doar pentru a îndeplini sarcina de împrospătare a memoriei.

Când se adresează o memorie, este citit un rând întreg de celule. Dacă memoria este structurată în cuvinte de 32 de biți, sunt necesare 8,388,608 citiri în 3.725 nano-secunde la o frecvență de 262 MHz. În continuare, frecvența necesară împrospătării este foarte mare.

Memoriile sunt construite de fapt în arii aproape pătrate; memoriile sunt modelate ca arii liniare de cuvinte doar pentru o înțelegere mai ușoară a funcționării acestora. Structura fizică a memoriei nu respectă modelul funcțional.

Pentru exemplul considerat, memoria de 8,388,688 de cuvinte este organizată într-o matrice de 4096 coloane și 2048 rânduri. De fiecare dată când este citit un rând, toate celulele din acel rând sunt împrospătate. Dacă se citește fiecare rând la fiecare 32 milisecunde, operația de împrospătare se va face cu frecvența 64 KHz, ceea ce este acceptabil. În acest fel este depășit neajunsul ce privește frecvența de împrospătare.

Compararea memoriilor SRAM și DRAM se face considerând câțiva parametri: consumul de putere, preț, viteză și densitatea de integrare.

Memoria SRAM este mult mai rapidă decât memoria DRAM. Consumul memoriei SRAM este mai mare, densitatea de integrare este mai mică și prețul este mai mare. Din cauza acestor însușiri, memoria SRAM este utilizată doar acolo unde este necesară o memorie de capacitate mică, dar care trebuie să funcționeze la viteză mare.

Memoria cache este o memorie mică ce are menirea să stocheze date și instrucțiuni des utilizate.

Memoriile DRAM sunt folosite pentru a construi memorii mari. Densitatea mare de integrare, costul scăzut și consumul mic de putere face ca memoriile DRAM să fie foarte potrivite pentru a construi blocuri de memorie cu capacitatea de ordinul Gbyte.

**C11.3 Memorii LIFO și FIFO**

RAM este acronimul de la *Random Access Memory*, dar RAM a devenit în timp un substantiv cu înțelesul: memorie volatilă. La început, când RAM era un acronim, se referea la metoda folosită pentru a localiza anume date într-un sistem de stocare a informației. În acest sens, informația poate fi accesată prin una din următoarele metode:

- acces aleator (*random access*);
- acces secvențial;
- primul intrat - primul ieșit (*First-In-First-Out* - FIFO);
- ultimul intrat - primul ieșit (*Last-In-First-Out* - LIFO).

**Acces aleator** înseamnă că orice locație poate fi selectată și accesată în pasul următor. Cuvintele sunt accesate aleator. Toate celelalte metode de acces nu sunt aleatoare.

Accesul aleator este specific memoriilor semiconductoare, totodată, aceasta nu este singura metodă de accesare a datelor utilizată în memoriile semiconductoare. Există memorii semiconductoare cu acces nealeator.

În memoriile cu arhitecturi cu **acces secvențial**, informația este stocată într-un șir lung. Când se dorește accesare unei părți din șir, este necesar să "călătorești" prin toată informația stocată în șir de la locația curentă până la locația dorită.

Benzile din casetele audio și video constituie un exemplu familiar pentru acest mod de acces.

O altă metodă de acces este **FIFO**. Această arhitectură de stocare poate fi privită ca un rând de așteptare în care informația este disponibilă în ordinea în care a fost înscrisă în memorie. Privind un rând format din oameni care așteaptă să plătească bunurile pe care le-au pus în coș spre a fi cumpărate, se poate observa că primul așezat la rând va fi primul care va plăti și va pleca cu bunurile cumpărate, al doilea așezat va pleca al doilea, și așa mai departe.

Metoda FIFO este utilizată în produse semiconductoare, la blocurile de stocare a informației, de exemplu - registre de deplasare, sau în comunicații.

Pentru a înțelege metoda de accesare **LIFO**, imaginați-vă o cutie tubulară cu mingi de tenis în care introduceți mingile pe rând; ultima pe care ați introdus-o în cutie este prima pe care o puteți scoate, penultima introdusă este a doua pe care o puteți scoate și la sfârșit veți scoate din cutie prima minge pe care ați introdus-o. Ordinea în care aveți acces la mingi este cea inversă celei în care ați introdus mingile în cutie.

Accesul LIFO este utilizat în anumite blocuri de memorie în microprocesoare și microcontrolere. Funcția acestor blocuri de memorie este de a stoca adresa unei instrucțiuni dintr-o succesiune de instrucțiuni atunci când execuția seriei de instrucțiuni este întreruptă deoarece sistemul este solicitat să îndeplinească o altă sarcină mai importantă (sarcina 1). Se stochează în memoria LIFO adresa instrucțiunii care urma să fie executată atunci când a fost întrerupt șirul execuției. Să presupunem că în timp ce este executat șirul de instrucțiuni din rutina ce deservește sarcina 1 sistemul este din nou solicitat să îndeplinească o sarcină și mai urgentă decât prima (sarcina 2). Din nou se stochează în memoria LIFO adresa instrucțiunii care urma să fie executată. Memoria LIFO seamănă cu o stivă; aici sunt depuse una peste alta adresele de instrucțiuni; memoria LIFO este numită și stivă. După ce se termină servirea sarcinii 2, sistemul apelează stiva pentru a extrage adresa instrucțiunii ce trebuie executată pentru a continua execuția rutinei ce servea sarcina 1. Sistemul va găsi adresa necesară în partea superioară a stivei și va extrage ultima adresă ce a fost depusă aici (ultima depusă este prima extrasă). Când se termină și execuția rutinei ce deservea sarcina 1, sistemul apelează din nou stiva de unde ia adresa instrucțiunii ce urmează a fi executată și continuă execuția programului principal.