
1 Memoria

*Memoria are rolul de a păstra informația utilizată de unitatea centrală sau dispozitivele periferice. Informația stocată este de două tipuri: programe și date. În funcție de capacitate, de tipul informației și de durata păstrării acesteia, memoria utilizată în sistemele de calcul este de trei categorii: memoria internă, memoria principală și memoria externă. O caracteristică fundamentală ce deosebește cele trei tipuri de memorie este **timpul de acces** : intervalul de timp din momentul adresării memoriei de către unitatea centrală și încărcarea datelor în/din memorie pe magistrala de date. În funcție de timpul de acces memoria poate fi **rapidă** (cu timp de acces redus) sau **lentă** (cu timp de acces mare).*

Memoria internă se află în interiorul microprocesorului și este formată din 10. . .100 de registre de 8, 16, 32, . . . biți fiecare, pentru memorarea temporară a operanzilor implicați în operațiile aritmetice și logice, a rezultatelor operațiilor și a adreselor curente. Este o memorie de capacitate mică dar foarte rapidă (timp de acces sub zece nanosecunde).

Registrele pot fi utilizate de programator prin intermediul instrucțiunilor. Referirea la un registru se face prin denumirea sa formată dintr-un grup de litere: A, B, C, D, E, H, L, AX, BX, CX, DX, BP, SP, ES, CS, EAX, EBX, etc. Microprocesoarele de mare complexitate (Pentium, Power PC) dispun, pe lângă setul de registre și de o memorie internă rapidă de 16 - 32 kB pentru instrucțiuni și date.

Memoria principală, numită și *memorie sistem* sau *memorie operativă*, este de capacitate medie (zeci de MB) și cu timp de acces mediu (sute de nanosecunde). Este conectată direct și permanent la magistralele sistemului, ceea ce permite accesul rapid la date. Unitatea de bază este celula de memorie de **un bit** , realizată cu dispozitive semiconductoare. Celula de memorie poate fi un circuit electronic cu două stări stabile, asociate celor două valori binare 0 și 1, sau un dispozitiv electronic (diodă, tranzistor) cu două stări: starea de conducție și starea de blocare.

Celulele de memorie sunt grupate în **locații** de 1, 2, 4, 8, 16, 32, 64 de biți. Fiecare locație are o **adresă** proprie și **unică** în sistem. La un moment dat o singură locație poate fi activă și conectată la magistrala de date și anume cea corespunzătoare adresei ce se află în acel moment pe magistrala de adrese.

Memoria externă, numită și *memorie de masă*, se află în dispozitivele periferice specializate în stocarea datelor: unități de discuri

magnetice (Hard Disk, Floppy Disk), unități de bandă magnetică, unități CD (Compact Disk).

Este memorie de capacitate practic nelimitată (zeci, sute GB), dar cu timp de acces mare (de ordinul zeci de milisecunde). Timpul de acces la discul magnetic, de exemplu, este compus din: timpul de căutare (se caută o anumită pistă), timpul de rotație (se caută un anumit sector) și timpul de transfer date, care depinde de viteza de transfer a unității de disc.

Memoria externă este utilizată pentru stocarea programelor și datelor pentru timp îndelungat (luni, ani).

1.1 Memoria principală

Este conectată direct la magistralele sistemului pentru a permite o viteză mare de transfer.

Din punct de vedere fizic, este formată din mai multe circuite integrate de memorie, care pot fi de tipuri și tehnologii diferite (fig.1).

Circuitele integrate de memorie au organizare matriceală: o locație de memorie poate fi o linie din matrice sau un element. În primul caz, adresa locației este numărul liniei iar în al doilea caz, adresa locației este formată din numărul liniei și al coloanei.

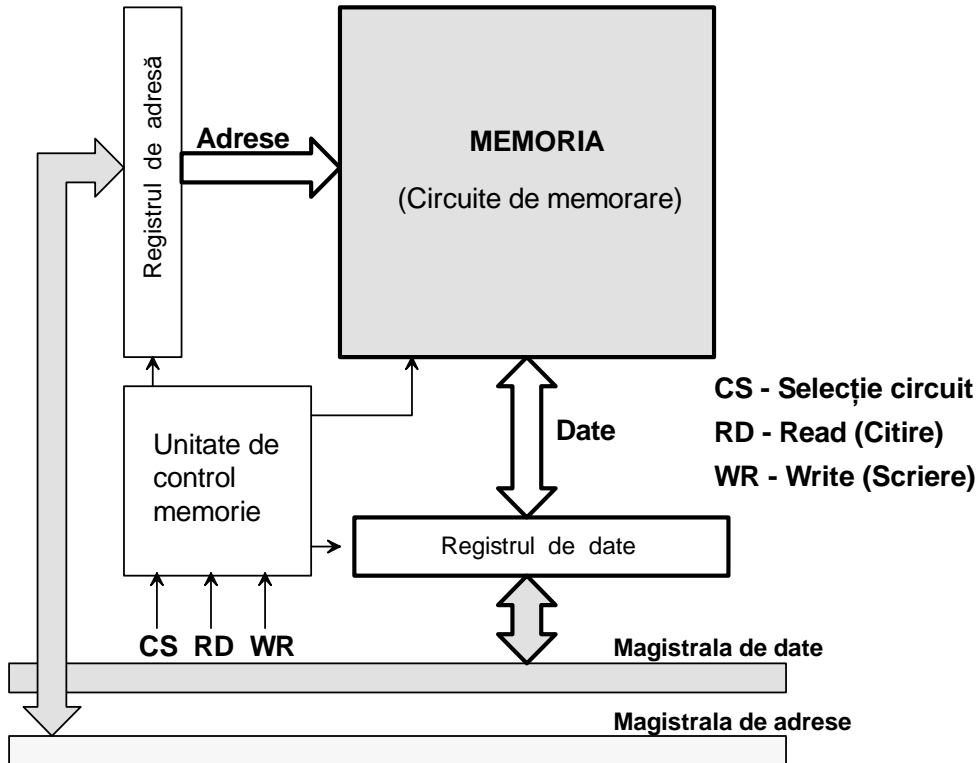


Fig.1 Schema bloc a unui circuit de memorie

Numărul total de locații determină capacitatea circuitului de memorie. O descriere completă a structurii unui circuit este dată de numărul de locații și numărul de biți / locație. De regulă numărul de locații și numărul de biți / locație sunt puteri ale lui 2.

Exemple: 1024 x 8 (capacitate 1kB);
1024 x 1024 x 8 (capacitate 1MB).

1.1.1. Clasificarea circuitelor de memorie

Un prim criteriu de clasificare este determinat de *durata păstrării informației în memorie*. Astfel, datele curente dintr-un program, care se modifică permanent pe durata execuției, programele aflate în execuție, imaginile de pe ecran, se memorează în circuite de memorie RAM (*Random Access Memory* - memorie cu acces aleatoriu). Memoria RAM poate fi citită și scrisă de către unitatea centrală (microprocesor) de un număr nelimitat de ori; se mai numește **memorie vie**. La întreruperea tensiunii de alimentare, informația se pierde "definitiv și iremediabil" - este memorie **volatilă**.

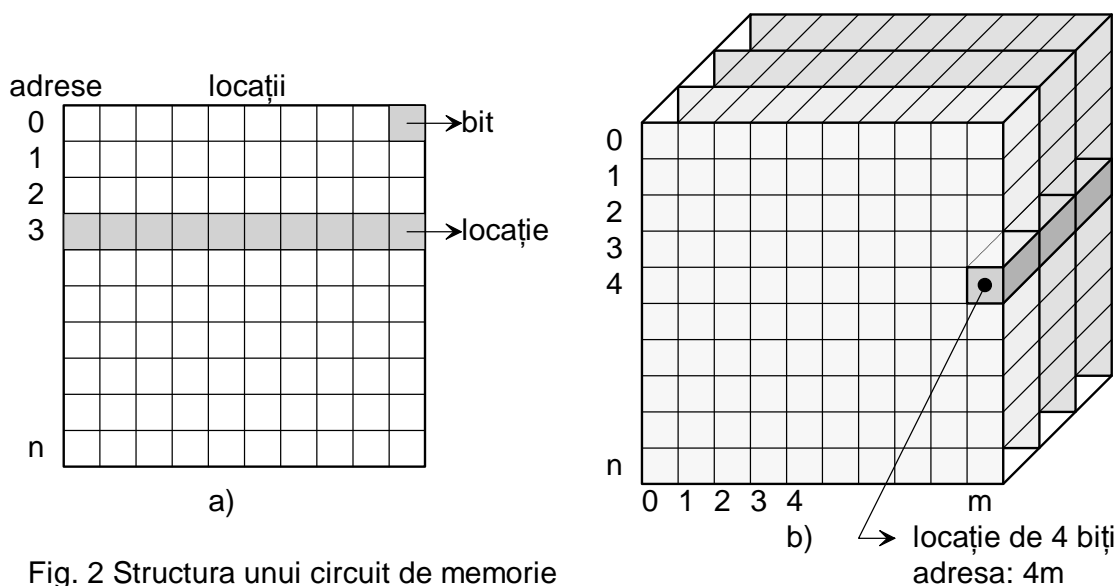


Fig. 2 Structura unui circuit de memorie
a) structură cu n - locații, b) structură cu $n \times m$ - locații

Alte informații, ca de exemplu tabele de conversie, programe fixe, anumite funcții de sistem, sunt necesare pe durata existenței sistemului de calcul sau pentru intervale lungi de timp (luni, ani). Acestea se memorează în circuite de memorie ROM (*Read Only Memory* - memorie care poate fi numai citită). Unitatea centrală poate citi de un număr nelimitat de ori memoria ROM, dar nu poate modifica informația stocată. Aceasta se păstrează un timp nelimitat chiar în absența tensiunii de alimentare iar scrierea informației se realizează cu dispozitive speciale fie, în procesul de fabricație, fie de către utilizator.

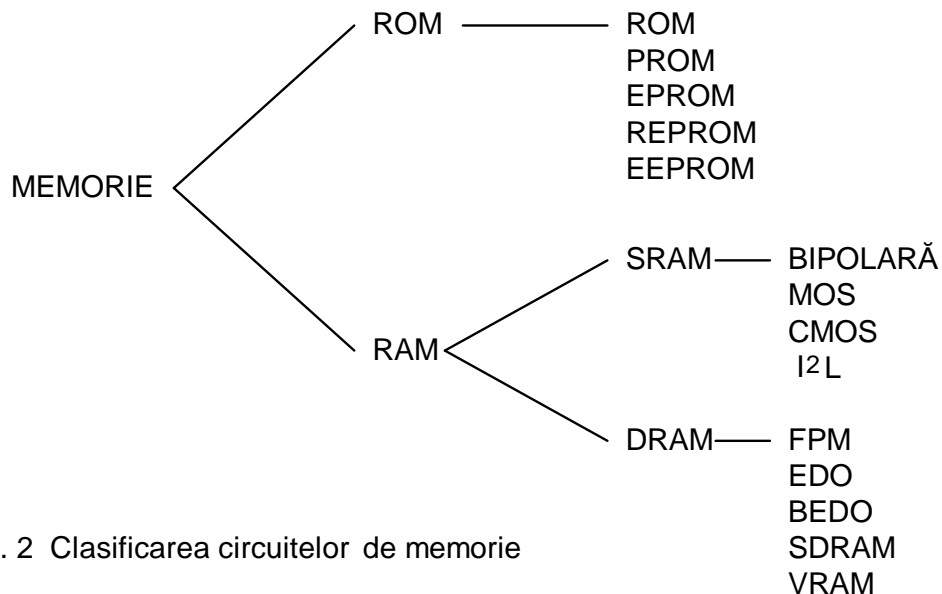


Fig. 2 Clasificarea circuitelor de memorie

Memoriile RAM și ROM sunt **cu acces aleatoriu**, adică timpul de acces la informație nu depinde de adresa locației în circuitul de memorie. Un alt tip de acces este cel **secvențial**, caz în care timpul de acces depinde de locul în care se află informația. Acest tip de acces este caracteristic memoriei externe (pe disc, pe bandă).

O denumire mai sugestivă a memoriei RAM ar fi RWM (*Read Write Memory*) dar se menține (de peste 30 de ani) denumirea tradițională.

Circuitele de memorie RAM și ROM se realizează în diverse tehnologii de fabricație prin care se urmărește: timp de acces cât mai redus, densitate mare de integrare (capacitate mare pe circuit), *preț de cost / bit* cât mai mic, putere consumată exprimată în $\mu w / bit$, cât mai scăzută. Aceste deziderate fiind contradictorii, anumite tehnologii optimizează numai anumite proprietăți. De exemplu, memoriile rapide sunt scumpe iar cele de mare capacitate au consum mare și timp de acces mediu.

1.1.2. Caracteristicile unui circuit de memorie

- ♦ Structura sau modul de organizare: numărul de celule de memorie dintr-o locație (biți / locație) și numărul total de locații. Operațiile de citire și scriere se fac numai cu locații.
- ♦ Capacitatea memoriei: numărul total de locații sau cuvinte de memorie.
- ♦ Timpul de acces: intervalul dintre momentul adresării memoriei de către unitatea centrală și momentul accesului efectiv la date.
- ♦ Puterea consumată, tensiunile de alimentare.
- ♦ Tipul memoriei: RAM, ROM.

- ♦ Tehnologia de realizare: bipolară (TTL, TTL Shottky, ECL), MOS, CMOS, I²L.

Circuitele bipolare sunt foarte rapide dar au densitate mică de integrare. Cele MOS au densitate mare de integrare, putere consumată mică dar sunt mai lente. Tehnologia I²L are performanțe între cele ale circuitelor bipolare și MOS.

1.2 Memorii ROM (*Read Only Memory*)

Celulele de memorie sunt organizate matriceal, locațiile fiind liniile matricei. Celulele de pe o coloană sunt conectate la aceeași linie de bit de date și sunt constituite din elemente semiconductoare: diode bipolare, tranzistoare bipolare, tranzistoare MOS etc.

Caracteristici generale:

- ♦ acces aleatoriu;
- ♦ nevolatile (informația conținută nu se pierde la întreruperea tensiunii de alimentare);
- ♦ pot fi doar citite de unitatea centrală (*Read Only*); scrierea se face cu dispozitive speciale;
- ♦ timp de acces 25 - 30 ns pentru tehnologia bipolară, 100 - 500 ns pentru MOS;
- ♦ densitate de integrare mare, consum mic, preț de cost scăzut pe bit.
- ♦ tehnologia de fabricație (bipolară, MOS, CMOS, etc.).

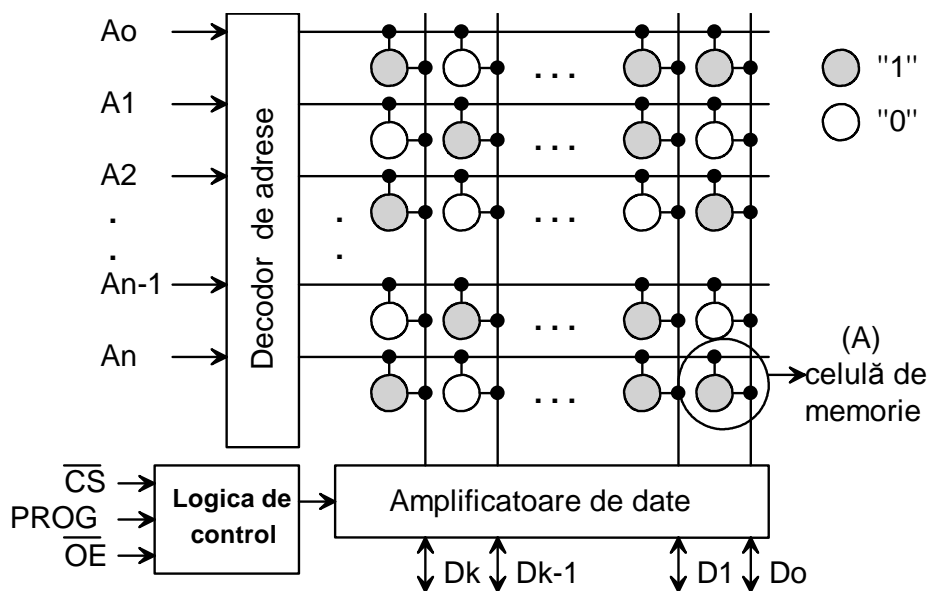


Fig. 3 Schema generală a unui circuit de memorie ROM

În schema generală a unui circuit de memorie ROM (fig.3), sunt utilizate notațiile obișnuite pentru semnalele electrice aferente:

- ♦ \overline{CS} - *Chip Select*, selecție circuit; semnalul în stare activă (0) activează circuitul, adică permite operații cu memoria;
- ♦ \overline{OE} - *Output Enable*, validare ieșire date;
- ♦ *PROG* - programare memorie (scriere date);
- ♦ A_0, A_1, \dots, A_n - linii ale magistralei de adrese (determină locația adresată);
- ♦ D_0, D_1, \dots, D_k - linii de date (se conectează la magistrala de date).

În figura 4 este prezentată diagrama semnalelor implicate în citirea memoriei.

Semnalul CS este generat de adresa circuitului de memorie iar semnalul OE (*Output Enable*) este generat de comanda de citire memorie. După intervalele de timp t_{CS} , respectiv t_{OE} , datele sunt disponibile și stabile la ieșirea circuitului (de la stabilirea adresei - timpul de acces). Timpul t_{OH} reprezintă intervalul în care datele se mențin la ieșire, după dispariția comenzii OE. Durata totală a operației de citire este egală cu durata menținerii adresei pe magistrală.

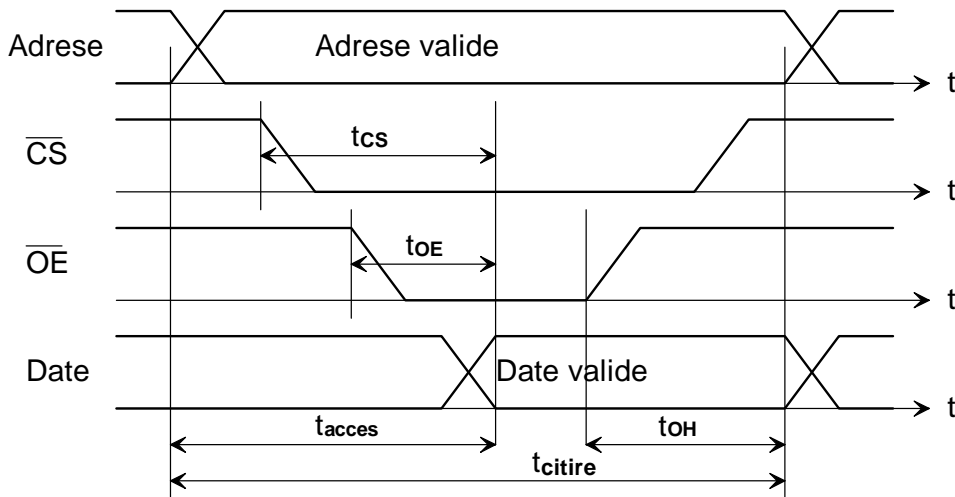


Fig.4 Diagrama semnalelor la citirea memoriei

1.2.1. Memoria ROM propriu-zisă

Este încărcată cu un anumit conținut încă din procesul de fabricație.

Informația înscrisă nu mai poate fi modificată. Avantaje : densitate foarte mare de integrare, preț de cost scăzut/bit, la serii mari. O largă utilizare au primit aceste memorii pentru tabele de date aritmetice, de conversie pentru coduri, programe fixe pentru sisteme automate, etc. Principalul dezavantaj este imposibilitatea modificării informației, ceea ce face inutile circuitele de memorie, când informația conținută nu mai prezintă interes.

Exemplu: Structura internă a unei **celule** de memorie MOS - ROM (fig.4). Celula de memorie este un tranzistor MOS. Informația este transmisă fabricantului de către utilizator, pentru crearea unor măști foto, prin intermediul cărora, pentru "1" logic, nu se finalizează tranzistorul din nod.

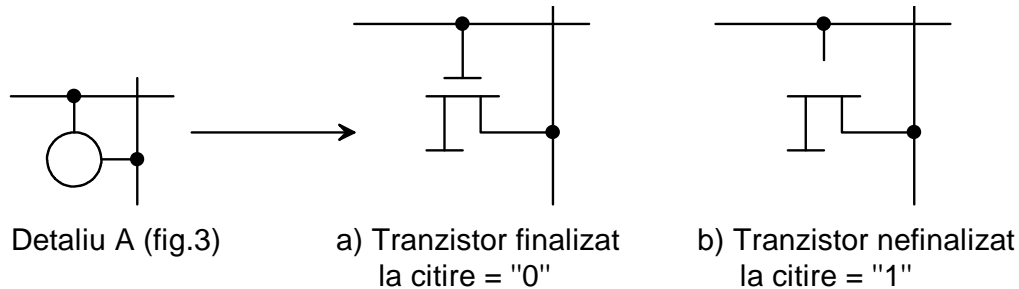


Fig. 4 Celula de memorie ROM de tip MOS

1.2.2. Memoria PROM (*Programmable ROM*)

Este o memorie de tip ROM - programabilă de către utilizator cu ajutorul unui echipament electronic special, conectat de regulă la un calculator.

În procesul de fabricație fiecare element de memorie (tranzistor) este conectat în serie cu un fuzibil din Al sau Ni-Cr, cu lungime de ordinul micronilor. În procesul de programare (scriere), fuzibilul poate fi distrus cu un tren de impulsuri de curent de 100-1000 mA. O conexiune intactă determină memorarea unui "0" logic iar o conexiune întreruptă determină memorarea unui "1" logic.

Ca și la memoria ROM, informația nu mai poate fi modificată decât prin întreruperea altor conexiuni (trecerea din 0 în 1). Structura internă a unei celule de memorie PROM cu tranzistoare bipolare este prezentată în figura 5.

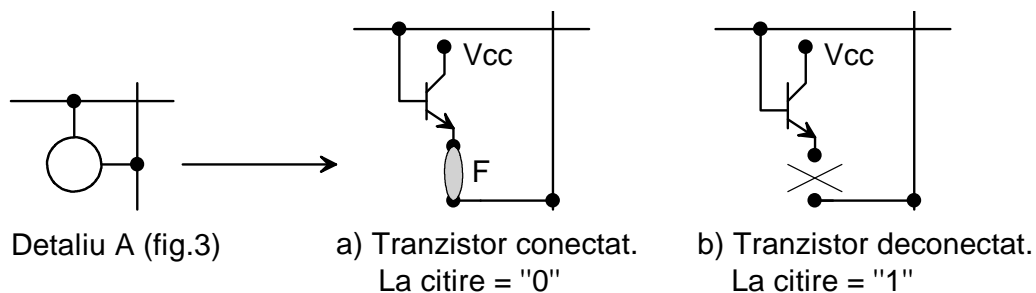


Fig. 5 Celula de memorie PROM cu tranzistoare bipolare.

1.2.3. Memoria EPROM (*Erasable and Programmable ROM*)

Circuitele de memorie EPROM pot fi programate și șterse de un număr de ori (30 - 100), utilizând pentru scriere echipament electronic special, conectat la un calculator. Prin ștergeri și reprogramări repetate, crește timpul de ștergere (10 min - 60 min). Ștergerea se face cu raze ultraviolete, motiv pentru care "chip"-urile sunt prevăzute cu o fereastră transparentă de cuarț. Memoria EPROM de tip MOS stochează informația ca o sarcină definitivă într-o poartă a unui tranzistor MOSFET (tranzistor MOS cu efect de câmp), care are o poartă suplimentară (flotantă) izolată electric (tranzistor cu poartă flotantă și injecție în avalanșă).

Poarta flotantă poate fi încărcată (la programare) cu sarcină electrică printr-o injecție de electroni sub acțiunea unui câmp electric intens; sarcina negativă acumulată în poarta flotantă modifică tensiunea normală de prag, astfel că tranzistorul selectat cu U_G normal, prezintă la ieșire 1- logic dacă nu a fost injectat și 0 - logic dacă a fost anterior injectat; astfel, o memorie nescrisă ("albă") are toate celulele în 1 - logic, iar la programare unele celule trec în 0 - logic.

Poarta flotantă fiind izolată electric de restul componentelor semiconductoare, sarcina electrică se menține peste 10 ani, cu o pierdere de cel mult 30%, chiar la temperaturi ridicate.

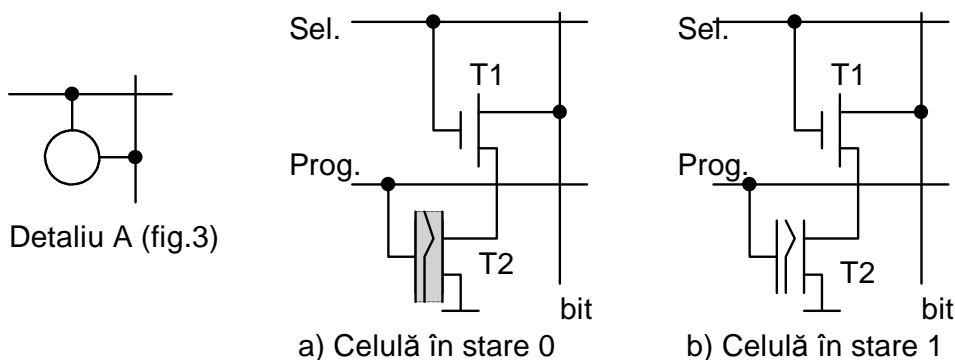


Fig. 5 Celula de memorie EPROM cu tranzistoare MOS.

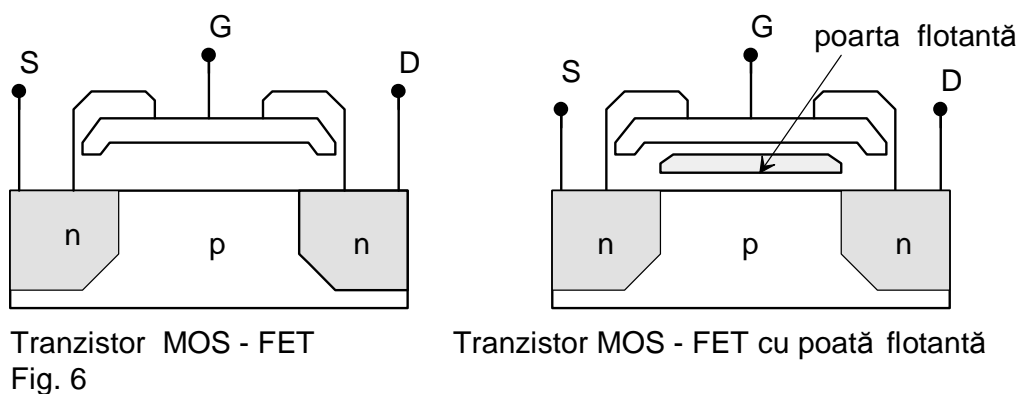


Fig. 6

Ștergerea memoriei se face prin descărcarea sarcinii electrice dintre poartă și substrat, prin iradiere cu raze ultraviolete sau raze X (UVEPROM), sau metode electrice (EEPROM). Programarea se face prin aplicarea unor impulsuri de tensiune ridicată (26 V), pe intrarea PROG, după selectarea locației.

Exemple de circuite de memorie EPROM:

- ◆ Intel 2708, 1kB (1024 x 8);
- ◆ Intel 2716 2716, 2kB (2048 x 8);
- ◆ Intel 27256, 32kB (32 x 1024 x 8);
- ◆ Intel 27512, 64kB (64 x 1024 x 8).

1.2.4. Memorii "flash"

Au intrat în producție de serie în 1988 (cele ROM și RAM în 1969). Principalul lor avantaj este că sunt programabile electric, nu sunt volatile, au timp de acces redus și o capacitate importantă (1MB - 40 MB !). La o asemenea capacitate, pot servi ca discuri, cu toate avantajele ce rezultă din soluția complet electronică față de discul electromecanic: dimensiuni reduse (tip carte de credit: PCMCIA - *Personal Computer Memory Card International Association*), viteză mare de lucru, consum redus de putere, masă redusă, fiabilitate ridicată.

Toate aceste avantaje incontestabile duc însă la un preț de cost mai ridicat decât al discurilor electromecanice, ceea ce limitează încă utilizarea lor.

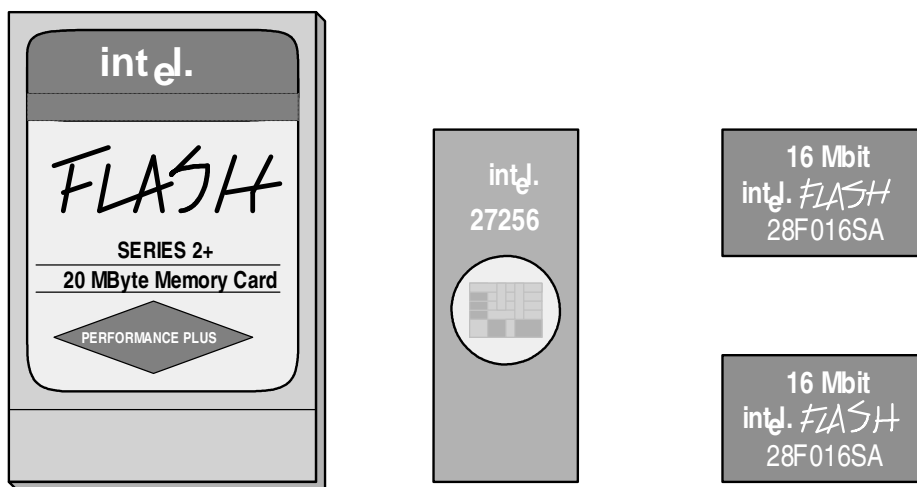


Fig. 7. Tipuri de memorii EPROM - Intel.

Tabel 1. Caracteristici ale memoriilor *flash* tip PCMCIA

Caracteristici	Serie 2	Serie 2+
Capacitate	2 - 20 MB	4 - 40 MB
Alimentare	5V și 12 V	3,3V sau 5V
Timp de acces la citire	200 ns	150 ns
Viteză de transfer	220 kB / sec.	850 kB / sec.
Număr de blocuri de 64 kB	320	640
Bocuri cu acces protejat	Nu	Da

1.3 Memorii RAM (Random Access Memory)

Denumirea arată că sunt "cu acces aleator", adică timpul de acces la informație nu depinde de adresa locației (ca și la ROM). Se păstrează organizarea matriceală, dar apar în plus circuite combinaționale necesare înscrierii informației. Principala lor caracteristică este că pot fi citite și scrise de unitatea centrală de un număr nelimitat de ori.

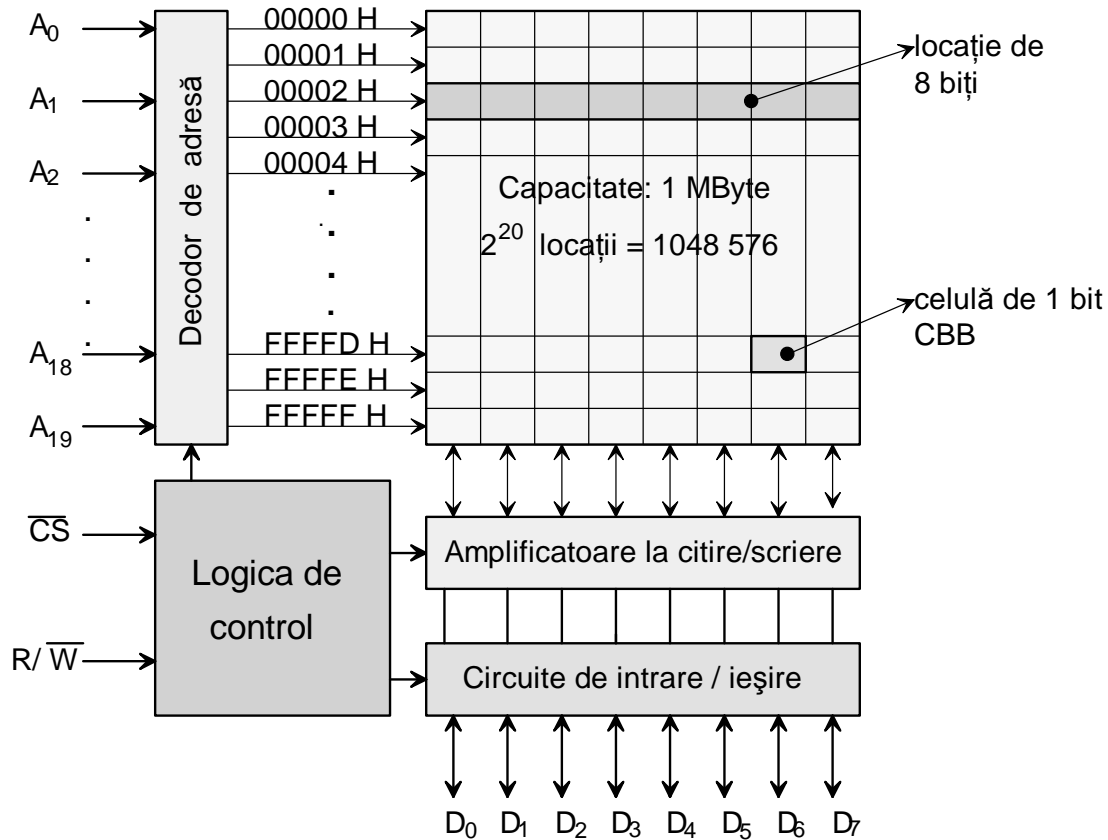


Fig. 8. Schema bloc internă a unui circuit de memorie SRAM

După timpul de păstrare a informației, memoriile RAM sunt de două tipuri:

- ♦ RAM static (SRAM) - informația se păstrează pe durata menținerii tensiunii de alimentare;
- ♦ RAM dinamic (DRAM) - informația se pierde dacă nu este regenerată periodic, chiar dacă se menține tensiunea de alimentare (perioada de regenerare este tipic de 2 ms).

Caracteristici generale:

- ♦ acces aleator;
- ♦ volatile (informația se pierde la decuplarea tens. de alimentare);
- ♦ pot fi citite și scrise de unitatea centrală;
- ♦ timp de acces 25 - 30 ns pentru tehnologia bipolară, 70 - 400 ns, pentru MOS;
- ♦ densitate de integrare mare, consum mediu, preț de cost ridicat pe bit.
- ♦ tehnologia de fabricație (bipolară, MOS, CMOS, etc.).

1.3.1. Memoria RAM de tip static

Celula elementară de memorare este un circuit basculant bistabil (CBB cu două stări stabile electric, una pentru "0" logic și alta pentru "1" logic). Structura circuitului bistabil depinde de tehnologia de fabricație. În figura 8, este prezentat un circuit de memorie SRAM cu 20 de linii de adresă, ceea ce corespunde unui număr de 1 048 576 locații de 8 biți / locație (1MB). Selecția circuitului pentru citire sau scriere se face cu semnalul **CS** (*Chip Select*). Pentru citire / scriere, semnalul de mod, **R/W** (*Read / Write*) trebuie să fie "1" logic la citire și "0" logic la scriere.

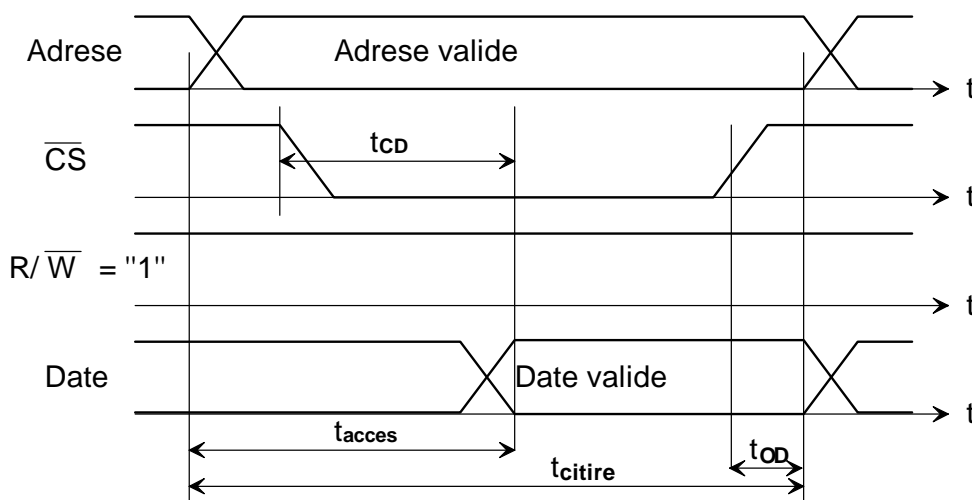


Fig.9 Diagrama semnalelor la citirea memoriei

La citire, datele sunt disponibile după un timp t_{CD} față de frontul anterior al semnalului CS și se mențin stabile la ieșire după un timp t_{OD} după ce semnalul CS devine inactiv (fig. 9). La scriere (fig. 10), semnalul de comandă R/W este activ în "0" și trebuie să aibă o durată prevăzută în

catalog. În general, scrierea are loc pe frontul crescător al semnalului R/W; ca urmare, datele trebuie să fie stabile la intrare mai devreme cu t_{DA} față de acest front și să fie menținute un timp după front

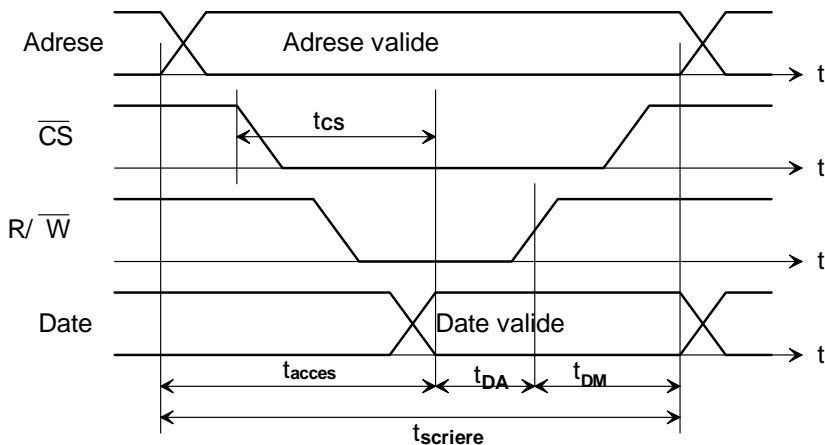


Fig.10 Diagrama semnalelor la scrierea memoriei

1.3.2. Memorie SRAM cu tranzistoare bipolare

Dacă celula de memorie nu este selectată, linia de selecție este la 0,3 V ("0" logic) iar emitorul (b) la "0".

Pentru citire : linia de selecție trece la $>3V$ ("1" logic) și informația memorată se transferă în linia de bit (de ieșire). Dacă T_1 este în conducție , atunci $V_b > 3V$, adică "1" logic și se citește "1". Dacă T_1 este blocat, prin nici unul din emitorii a sau b nu circulă curent și pe linia de bit se detectează "0". După citire, linia de selecție coboară la 0,3 V și se menține starea logică anterioară a celulei de memorie.

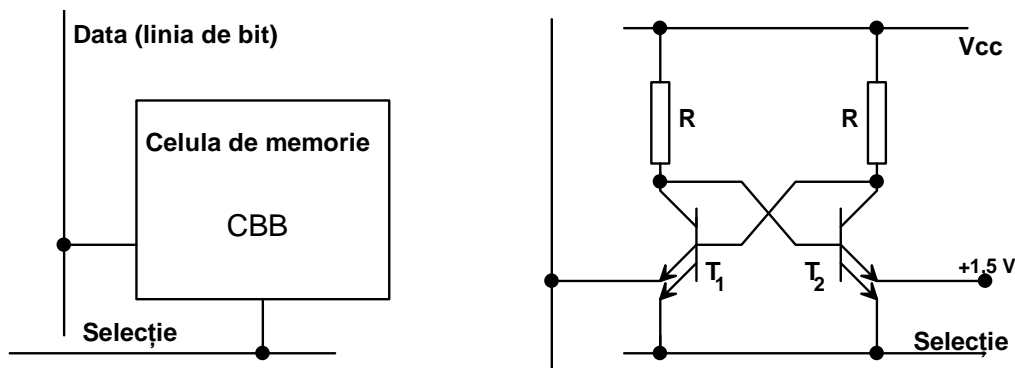


Fig. 11. Celula de memorie SRAM bipolară

Pentru scriere : se aplică tensiune pe linia de bit și apoi se trece în "1" (3 V) linia de selecție . Dacă $b=0$, T_1 trece în conducție iar T_2 se blochează . Dacă $b=1$, T_1 se blochează iar T_2 trece în conducție . Circuitul își păstrează starea după trecerea liniei de selecție în "0" logic .

Memoriile bipolare au cea mai ridicată viteză de lucru, timpul de acces fiind de ordinul 10 ns dar fiecare celulă necesită o suprafață mare pe pastila de siliciu și consum relativ ridicat de putere. Rezultă o densitate de integrare scăzută și preț de cost /bit, ridicat.

1.3.3. Memoria SRAM de tip MOS

Prezintă avantajul unui preț de cost / bit mai redus, densitate ridicată de integrare, dar sunt mai lente, timpul de acces fiind cuprins în domeniul 100 - 500 ns.

Celulele de memorie SRAM - MOS sunt circuite basculante bistabile cu funcționare asincronă, fără impuls de tact.

În varianta cu 6 tranzistoare (fig. 12), T_1 și T_2 formează un circuit bistabil de tip RS, T_3 și T_4 au rol rezistiv iar T_5 și T_6 sunt utilizate pentru conectarea celulei la liniile complementare de date, D, \bar{D} .

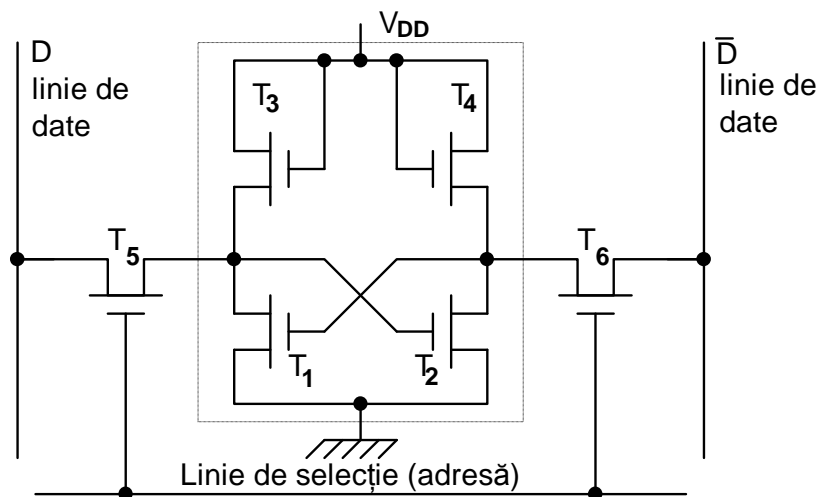


Fig. 12 Celulă de memorie SRAM - MOS

La citire, T_5 și T_6 sunt deblocate cu "1" pe linia de selecție (sau de adresă) și astfel informația (de 1 bit) conținută în circuitul bistabil este copiată în liniile de date. Dacă T_1 era blocat iar T_2 în conducție, de exemplu (celula în stare "1"), în linia de date D se transmite un nivel ridicat de tensiune (+5V) = "1" logic iar în linia de date \bar{D} se transmite nivelul (0 V) = "0" logic.

La scriere, după selectarea celulei cu "1" logic pe linia de selecție, se aplică pe liniile de date valorile logice corespunzătoare bitului ce trebuie înscris: pentru înscrisul unui "1" logic, $D = 1, \bar{D} = 0$, iar pentru înscrisul unui "0" logic, $D = 0, \bar{D} = 1$.

1.3.4. Memoria RAM de tip dinamic (DRAM - Dynamic RAM)

Pentru realizarea unor circuite de memorie de mare capacitate, se simplifică structura celulei de bit, până la un tranzistor / bit. Se obține o densitate de integrare mai mare și consum de putere mai redus față de memoria SRAM. Ca urmare, scade prețul de cost pe bit.

Celula de memorie DRAM utilizează ca element de memorie capacitatea poartă-substrat a unui tranzistor MOS, care există prin construcție, fiind o capacitate parazită pentru aplicațiile obișnuite. Prin construcție, capacitatea este adusă la o valoare corespunzătoare stocării informației sub formă de sarcină electrică.

Păstrarea informației este temporară, datorită curenților reziduali de descărcare a condensatorului în câteva milisecunde. Simplificarea celulei de bit aduce așadar un dezavantaj important - **celula de bit pierde informația în timp.**

Pentru menținerea informației, aceasta trebuie regenerată periodic prin citire și reînscrisere la intervale de 2 ms (tipic).

Schema bloc a unui circuit de memorie DRAM (fig. 13) diferă de cea a unui circuit de memorie SRAM. Deși organizarea este tot matriceală, locația de memorie este de 1 bit. Adresa unui bit este fragmentată în adresă de linie și adresă de coloană. Cele două tipuri de adrese sunt validate de semnale diferite: **RAS** (*Row Address Strob*), pentru adresa de linie și **CAS** (*Column Address Strob*), pentru adresa de coloană.

Circuitele DRAM nu au semnalul CS, selecția circuitului realizându-se numai prin perechea de semnale RAS, CAS. Regenerarea se realizează simultan pentru toate celulele de pe o linie din matrice. Ca urmare, la regenerare este activat numai semnalul RAS, nu și CAS. De exemplu, la o capacitate de 16 kb, matricea are 128 linii și 128 coloane. Adresele de linie și de coloană au fiecare 7 biți. Sunt necesare 128 de cicluri de regenerare (*refresh*), care trebuie să se desfășoare în 2 ms, ceea ce duce la 15 μ s / linie. În sistemele de calcul, regenerarea se face simultan pentru toate circuitele de memorie DRAM, oricare ar fi capacitatea lor, astfel că durata totală de regenerare este tot 2 ms.

La circuitele cu capacitate mai mare de 16 kb, matricea de celule are tot 128 de linii dar un număr mai mare de coloane; regenerarea se face simultan pentru toate celulele unei linii.

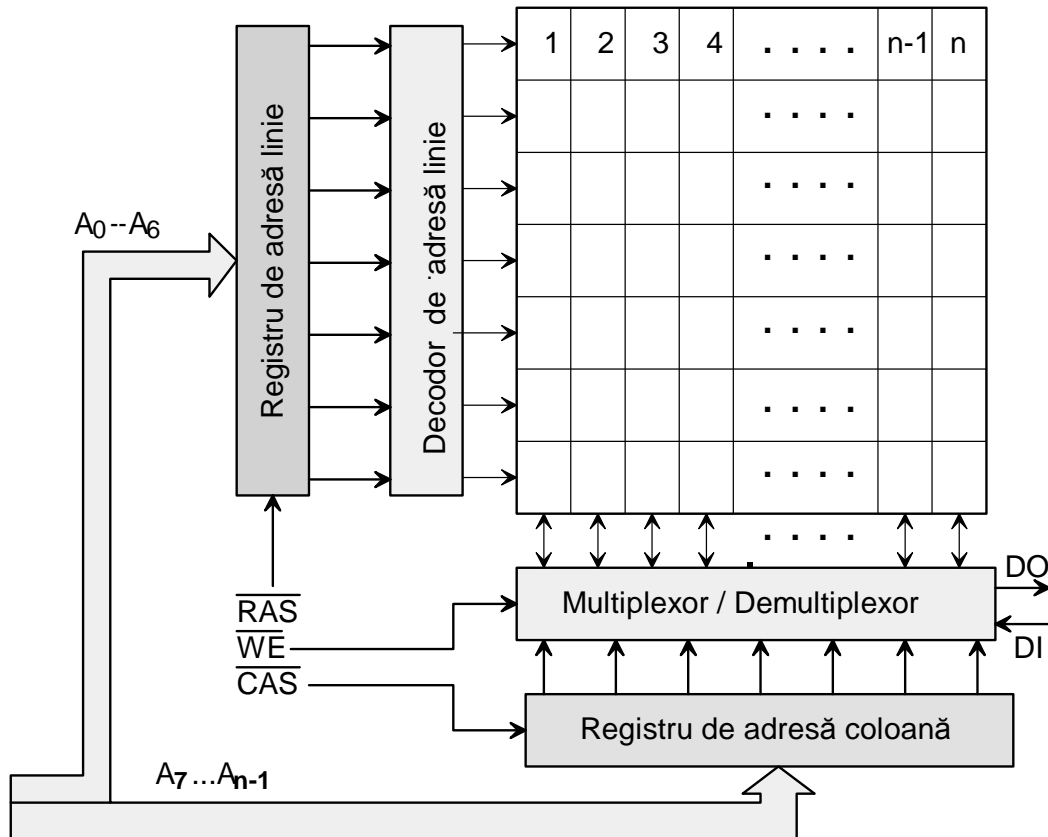


Fig. 13. Schema bloc internă a unui circuit de memorie DRAM

1.3.5. Memorie DRAM - MOS

Varianta 1 Nu există tensiune de alimentare ! (fig. 14a). Tranzistoarele T_3 și T_4 realizează conectarea celulei la liniile de bit (selecție), T_1 și T_2 alcătuiesc circuitul bistabil, CBB iar C_1 și C_2 sunt elementele de memorare.

La citire, se aplică "1" pe linia de selecție; T_3 și T_4 intră în conducție și conectează celula la liniile de date. Dacă C_1 este încărcat, T_1 este în conducție iar T_2 este blocat și în liniile de date se citește $D=0, \overline{D}=1$. În timpul citirii, C_1 se reîncarcă din linia de selecție iar C_2 se menține descărcat din cauza lui T_1 care este în conducție, T_2 fiind blocat. La citire se face deci și regenerarea valorii logice.

La înscriere, după selecție, informația aplicată liniilor de date determină comutarea celulei în starea corespunzătoare.

Varianta 2 Nu există tensiune de alimentare ! (fig. 14b). Elementul de memorie este condensatorul C . Inversorul cu T_2 și T_3 este utilizat pentru citirea informației. La citire, linia de date va conține complementul informației din C .

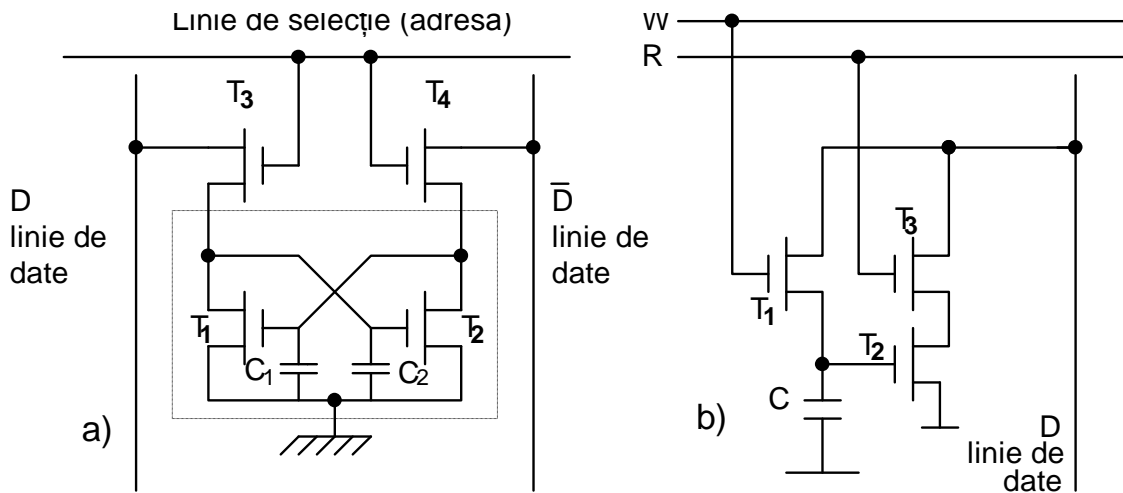


Fig. 14 Celule de memorie DRAM - MOS

Considerăm că inițial capacitatea C este încărcată la aproximativ V_{DD} . La citire, linia de selecție R (*Read*) fiind "1" logic, inversorul $T_2 - T_3$ conduce și pe linia de date se citește "0" logic. În cazul în care capacitatea C este descărcată, la selecție, inversorul nu conduce și pe linia de date se citește "1" logic.

La înscriere, pe linia de selecție W (*Write*) se aplică "1" logic și ca urmare T_1 intră în conducție, conectând capacitatea C la linia de date. Valoarea logică din linia de date se transferă în C .

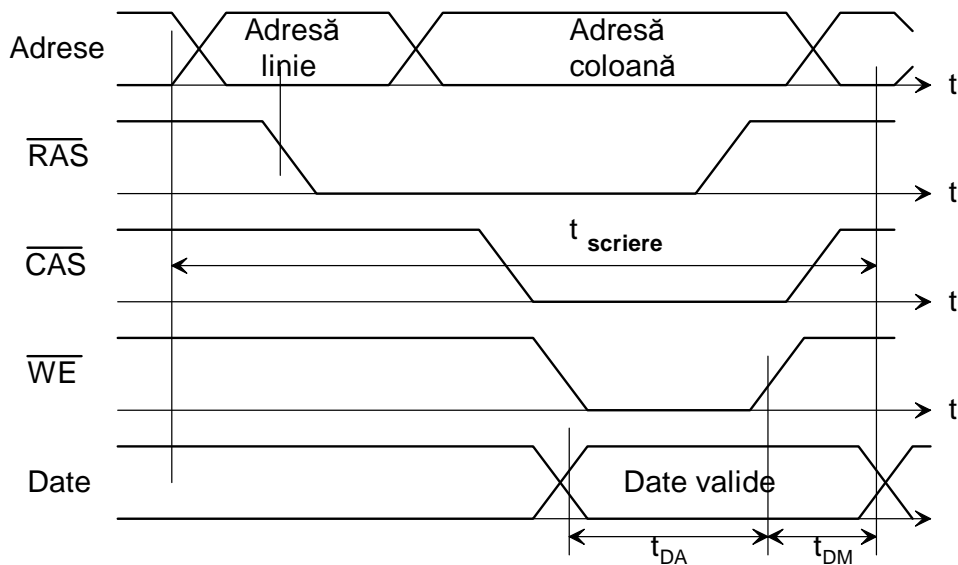


Fig.15 Diagrama semnalelor la scrierea memoriei DRAM

Reînprospătarea se face prin comanda de citire. Capacitatea C se încarcă la tensiunea maximă din linia de citire R , prin inversorul $T_2 - T_3$, dacă are inițial un nivel suficient de tensiune. Dacă inițial capacitatea C este descărcată, rămâne în aceeași stare, deoarece inversorul nu conduce, T_2 fiind blocat.

Circuitele de memorie DRAM permit realizarea unei densități ridicate de integrare, preț de cost scăzut / bit și consum redus de putere. Se utilizează pentru blocuri de mare capacitate (zeci de MB).

1.3.6. Tehnologii noi pentru memorii DRAM

Creșterea frecvenței de lucru și a vitezei de calcul la noile generații de procesoare a impus găsirea unor noi soluții tehnologice pentru reducerea timpului de acces al memoriilor de mare capacitate. Există o limită fizică pentru timpul de propagare a semnalelor electrice prin blocurile unui circuit de memorie (decodificator, celula de memorie, amplificatoare finale de ieșire). De aceea, soluția reducerii timpului de acces constă în optimizarea structurii interne, eliminarea anumitor etape ale procesului de transfer și accesul paralel.

1.FPM DRAM (*Fast Page Mode DRAM*) memorie cu acces rapid pe pagină

Acest tip de memorie asigură un acces mai rapid la datele conținute de o linie a matricei de memorie (pagină), atunci când transferul este secvențial, de la adrese consecutive. La citirea paginată, adresa de linie se specifică o singură dată, la începutul ciclului de transfer și apoi se generează secvențial numai adresa de coloană în limita numărului de coloane. În acest fel timpul mediu de acces, calculat pe o locație de memorie, scade cu cca. 50%.

2. EDO DRAM (*Extended Data Output DRAM*) memorie cu validarea separată a amplificatorului de ieșire

La memoriile DRAM clasice circuitul de memorie este selectat prin activarea intrărilor RAS și CAS. La transferul pe pagină, la fiecare activare și dezactivare a semnalului CAS, ieșirile circuitului comută din starea de înaltă impedanță în starea selectată și invers, comutări ce măresc de fapt durata transferului. La circuitele de tip EDO s-a introdus o intrare separată de selecție care permite validarea ieșirii circuitului pe toată durata unui transfer de bloc, ceea ce reduce durata transferului. Reducerea este valabilă numai pentru operațiile de citire memorie. Statistic, numărul de citiri este de aproximativ de 10 ori mai mare decât numărul de scrieri, ceea ce conferă o eficiență ridicată procedurii.

3. BEDO DRAM (*Burst Extended Data Output DRAM*) memorie EDO cu regim de lucru în avalanșă

Acest tip de memorie cu funcționare în regim de transfer paginat, generează automat adresele de coloană, în interiorul circuitului. După

înregistrarea unei adrese de coloană, se generează alte trei adrese consecutive. În acest fel se poate realiza accesul la 4 locații, cu o singură adresă de coloană. Regimul de lucru al acestei memorii este 4-1-1-1, adică pentru citirea primei locații sunt necesare 4 unități de timp iar pentru citirea următoarelor trei câte o singură unitate de timp.

Memoriile BEDO DRAM au cunoscut o mai mică utilizare deoarece noile procesoare Intel au oferit suport hard pentru utilizarea memoriilor de tip SDRAM.

4. SDRAM (*Synchronous DRAM*) memorii DRAM sincrone

Această clasă de memorii conține mai multe variante constructive, toate având în comun regimul de lucru sincron, controlat de un semnal de tact. Sunt adaptate noului standard de magistrală sincronă introdus de microprocesoarele familiei Intel P6 (Pentium Pro, Pentium II și III).

Principalele caracteristici funcționale sunt:

- ♦ funcționare sincronă cu semnalul de tact al procesorului;
- ♦ memoria este organizată pe blocuri cu funcționare paralelă;
- ♦ favorizează accesul în regim de avalanșă (*burst*);
- ♦ se utilizează o structură DRAM dar cu celule de memorie mai rapide;
- ♦ adresarea se poate face în regim "*pipe line*" (bloc după bloc);
- ♦ este permisă lansarea unui nou acces înainte de terminarea celui anterior;
- ♦ timpul de acces mediu este de 4 ori mai mic în comparație cu o memorie DRAM clasică.

În fig.16 este prezentată structura unei memorii organizate pe 4 blocuri, care sunt accesibile în mod concurent. Structura favorizează operațiile în regim "salvă". Se poate programa numărul de locații dintr-o salvă la 1, 2, 4, 8 octeți sau o pagină de memorie.

Accesul începe la o adresă specificată și continuă cu adresele succesive până la numărul de locații programate.

Circuitul de memorie acceptă mai multe tipuri de comenzi, fiecare comandă fiind specificată printr-o combinație a semnalelor de intrare.

Aceste comenzi sunt:

- ♦ *Command Inhibit* - inhibarea comenzilor următoare, fără afectarea celor curente;
- ♦ *No Operation (NOP)* - comandă neoperantă (nu se face nici citire, nici scriere), pentru prevenirea înscrierii de noi comenzi când circuitul este selectat;
- ♦ *Load Mode Register* - încărcarea registrului de mod prin intrările A0 - A11. Prin conținutul registrului de mod se programează regimul de lucru;

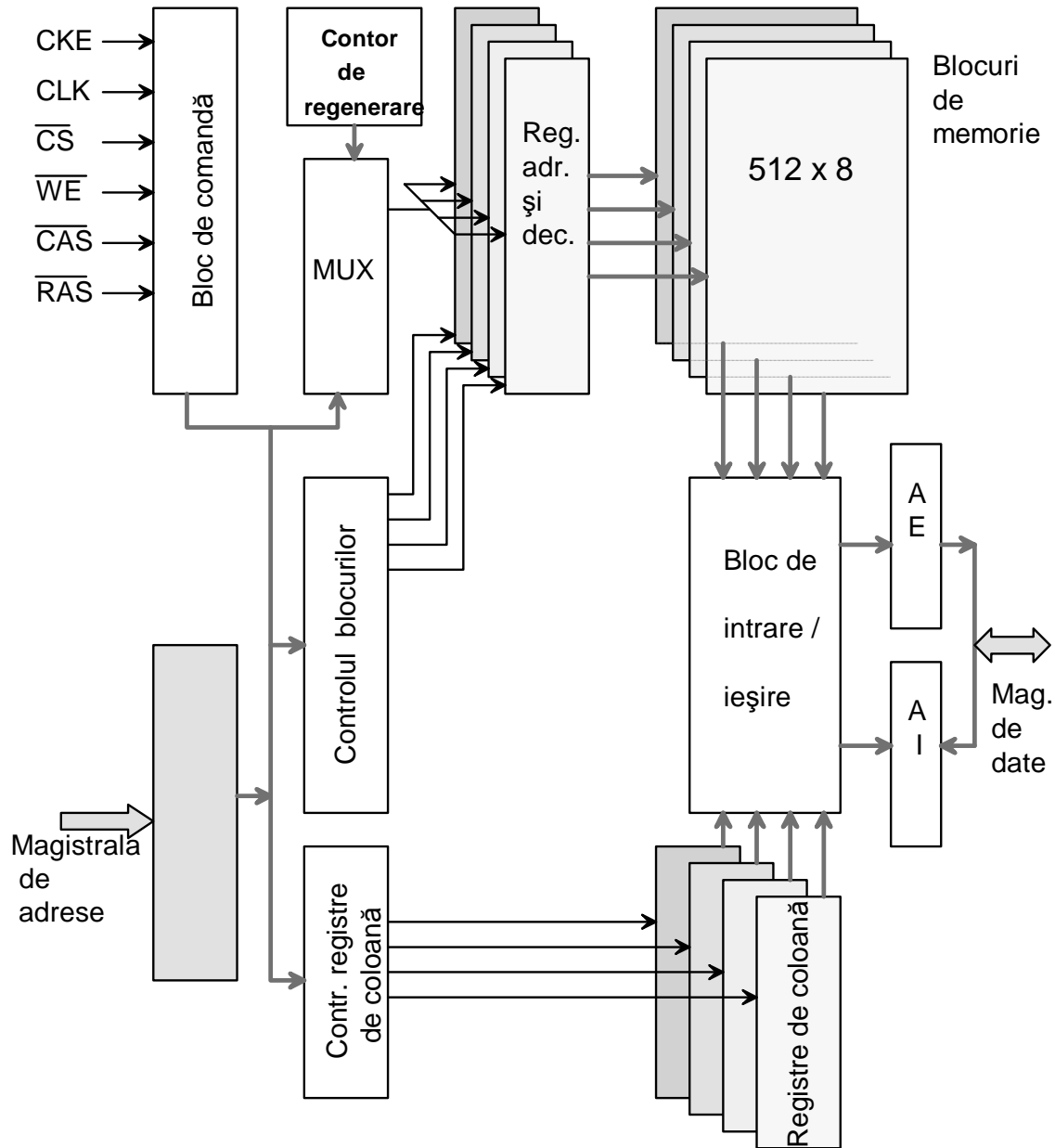


Fig. 16 Structura internă a unei memorii SDRAM (8 MB)

- ◆ *Active* - precede o comandă de citire sau scriere, deschizând o linie de memorie în blocul adresat;
- ◆ *Read* - comandă de citire în regim salvă, începând cu linia activată;
- ◆ *Write* - comandă de scriere în regim salvă, începând cu linia activată;
- ◆ *Precharge* - comandă de dezactivare a liniei selectate într-un bloc sau în toate;
- ◆ *Auto Charge* - efectuează operația anterioară în mod automat la încheierea transferului în salvă;
- ◆ *Burst Terminate* - comandă de trunchiere a unui transfer de tip salvă;

- ♦ *Auto Refresh* - comandă de regenerare a informației;
- ♦ *Self Refresh* - comandă de regenerare a informațiilor din SDRAM chiar dacă sistemul de calcul se află în regim de consum redus (*power down*).

În tabelul de mai jos sunt prezentați comparativ principalii parametri de timp corespunzători tipurilor specificate de memorie DRAM.

Tabel 2

Parametrul	FPM DRAM	EDO DRAM	BEDO DRAM	SDRAM
<i>Timp de acces (ns)</i>	50	50	52	50
<i>Durata operației (ns)</i>	30	20	15	10
<i>Frecvența magistralei</i>	16 - 66 MHz	33 - 75 MHz	60 - 100 MHz	60 - 100+MHz

DRDRAM (Direct Rambus DRAM) memorie realizată în noua tehnologie de magistrală Rambus

Tehnologie dezvoltată de firma Rambus pentru creșterea vitezei de lucru și adaptarea la noile generații de procesoare Intel. Magistrala asigură o viteză de transfer de 1,6 GB/sec., la o frecvență de tact de 800 MHz.

JEDEC SDRAM (Joint Electronic Device Engineering Council), standard industrial pentru circuite DRAM sincrone

Memoria este organizată pe două blocuri și poate funcționa în mai multe regimuri de transfer în avalanșă (*burst*), la frecvențele de 83 MHz și 100 MHz. Alte standarde similare definite de firma Intel sunt PC66 SDRAM (pentru o magistrală sistem de 66 MHz) și PC100 SDRAM (pentru 100 MHz).

ESDRAM (Enhanced SDRAM) memorie SDRAM îmbunătățită

Circuitul de memorie include o memorie RAM statică de capacitate redusă, pe lângă memoria SDRAM de mare capacitate. Transferul de date dintre memoria statică și cea dinamică se face printr-o magistrală internă de mare viteză, cu număr mare de linii de date. Este una dintre cele mai performante memorii actuale.

VRAM (Video RAM) memorie RAM video

Sub numele VRAM sunt grupate circuite de memorie utilizate în interfețele video pentru păstrarea informației afișate pe ecranul monitorului. Sunt construite așa încât să permită accesul concurrent la date din partea procesorului și din partea unității de reîmprospătare a imaginii. Cu cât timpul de acces este mai redus, se poate realiza o rezoluție (număr de pixeli) mai bună a imaginii. Circuitele de memorie video sunt atât de tip SRAM cât și de tip DRAM. Dintre cele mai performante sunt circuitele WRAM care pot fi utilizate până la o rezoluție de 1600 x 1200 pixeli.

1.4 Memorii intermediare (caches)

Au apărut odată cu procesoarele evolute, de înaltă performanță. O memorie *cache* este o memorie situată între microprocesor și memoria centrală (sau principală). Principalul avantaj al ei este rapiditatea (timp de acces foarte redus) iar principalul dezavantaj este prețul de cost.

Memoriile *caches* au adus noțiuni noi: *caches* primare și secundare, transparență la scriere/citire, *caches* de date, de instrucțiuni.

La utilizarea memoriilor *caches* este necesară asigurarea că ele conțin aceleași date ca și memoria centrală și că aceleași date se află la aceleași adrese. Intervine un protocol de verificare în acest sens: funcția de supraveghere este asigurată de o procedură *snooping* iar actualizarea se face cu procedura *MESI*.

1.4.1. Rolul memoriei intermediare

Memoria intermediară este un bloc de memorie RAM foarte rapidă dar de capacitate limitată, din cauza prețului de cost ridicat.

O memorie RAM obișnuită are un timp de acces de ordinul 70 ns. Dacă procesorul lucrează la o frecvență de tact mai mare de 100 MHz, perioada de tact este mai mică de 10 ns. Rezultă din aceste date că o operație cu memoria RAM (scriere sau citire) durează mai multe perioade de tact, ceea ce este inacceptabil pentru procesoarele evolute, proiectate să lucreze la 1 GHz !

Pentru a se justifica prețul de cost al procesoarelor rapide, este necesar ca memoria să răspundă (citire/scriere) într-o singură perioadă de tact; acest deziderat este realizabil dar foarte scump mai ales în condițiile în care aplicațiile solicită capacități enorme de memorie. Iată motivul principal pentru care s-au introdus memoriile intermediare, foarte rapide dar de capacitate mult mai mică în comparație cu memoria centrală. Ele mai sunt numite "antememorii" sau memorii "tampon".

Memoria *cache* stochează o mică parte din datele și instrucțiunile ce se află în memoria centrală. Acestea sunt încărcate anticipat în memoria cache, unde procesorul le poate accesa rapid; rezultatele unor calcule se stochează de asemenea în *cache* de unde vor fi transferate în memoria centrală.

Dacă procesorul are nevoie de un anumit operand, sunt două cazuri:

- ♦ operandul se află în memoria *cache* - el poate fi accesat rapid;
- ♦ operandul se află în memoria centrală (lentă) - accesul se face în câteva perioade de tact (se pierde mult timp).

Statistic, se constată că probabilitatea ca operandul să se afle în memoria cache este de până la 70% dacă aceasta are cel puțin 8 ko și ajunge la 90% la o capacitate de memorie cache de peste 64 ko.

Se explică, astfel, de ce capacitatea minimă de memorie *cache* este de 8 ko (memoria *cache* integrată pe chip este 8 ko la Intel 486).

1.4.2. Caches materiale și logice

Conceptul de *cache* se aplică memoriei centrale dar și memoriei de masă (disc). Nu este vorba însă de aceeași memorie *cache*. Un disc actual este accesibil în 10 - 40 milisecunde; se poate pune o memorie *cache* între disc și procesor dar aceasta va fi instalată în memoria centrală de data aceasta, fără introducerea unor circuite speciale.

Avem atunci două tipuri de *cache*: *cache RAM* și *cache disc*.

Memoria *cache RAM* este **materială**, deoarece este implementată cu circuite de memorie speciale (rapide); memoria *cache disc* este în general **logică**, fiind o parte a memoriei centrale și implementarea ei presupune existența unui program special de gestiune.

Memoria *cache disc* înregistrează ultimele date citite de pe disc sau stochează anticipat date citite, apoi le furnizează procesorului, la cerere. Sistemul de operare Windows dispune de program special pentru cache și anume *SmartDrive*.

1.4.3. Cache transparent și cache de scriere

Datele citite din memoria centrală sunt încărcate în memoria *cache* prin multiple strategii. De exemplu, procesorul poate încărca în *cache* un operand și toate datele următoare până la umplerea spațiului disponibil. Când este necesar un operand, procesorul îl caută cu prioritate în *cache*. De aceea se numește *cache de citire*.

Se poate utiliza în mod asemănător memoria *cache* pentru scriere. În loc să trimită datele direct în memoria centrală, procesorul expediază datele în memoria *cache*, mult mai rapidă. Așadar:

- ♦ memoria cache poate transfera date în memoria centrală, la intervale regulate de timp;
- ♦ memoria cache transferă date în memoria centrală numai după ce a acumulat o anumită cantitate de date;
- ♦ memoria cache profită de disponibilitatea magistralelor pentru a executa transferul.

În toate cele trei cazuri de mai sus, apare o perioadă critică, în care memoria centrală nu este actualizată ("la zi"). Este motivul pentru care nu se utilizează (decât excepțional) memoria cache în modul descris mai sus. Scrierea în cache se execută dar aceasta se comportă ca un pasaj, lăsând datele să curgă către memoria centrală. De aceea este numită memorie ***cache transparentă la scriere*** sau ***cache numai pentru citire***.

Memoria cache, în termeni consacrați, poate funcționa:

- ♦ *Write-back*, în mod scriere / citire;
- ♦ *Write-through*, transparent la scriere; informațiile traversează liber, sub formă de pagină sau bloc.

Un incident clasic de pierdere de date la PC-uri sub Windows 3.0, a avut la origine memoria cache. A fost vorba de transfer de date către *hard*

disk (printr-un canal "logic"). Windows utilizează o memorie intermediară creată prin comanda SmartDrive, pentru *hard disk*. Memoria lucrează tipic prin scriere (cache de scriere). La ieșirea din Windows, pentru încheierea sesiunii de lucru, acțiunea următoare, după întoarcerea în DOS, consta adesea în oprirea calculatorului.

În intervalul dintre ieșirea din Windows și oprire, programul *SmartDrive* nu avea timp să transfere blocul de date din memoria cache, pe disc. Numeroși utilizatori au fost surprinși să constate pierderi de date și chiar de fișiere întregi. Ca urmare, începând cu Windows 3.1, *Microsoft* a remediat această situație prin două măsuri:

- ♦ Prompterul DOS să nu apară pe ecran, la ieșirea din Windows, decât după ce s-a încheiat transferul de date către *hard disk*;
- ♦ Memoria cache să fie, prin definiție, transparentă la scriere; utilizarea memoriei cache pentru scriere să fie opțiune expresă a utilizatorului.

1.4.4. Memorie cache primară și secundară

Memoriile caches pot exista la mai multe niveluri:

1. Cache primară, plantată în microprocesor; este foarte eficientă dar de capacitate limitată, între 8ko și 64 ko.

2. Cache secundară, în exteriorul microprocesorului, care poate fi de aceea de capacitate mai mare; depășește adesea 128 ko.

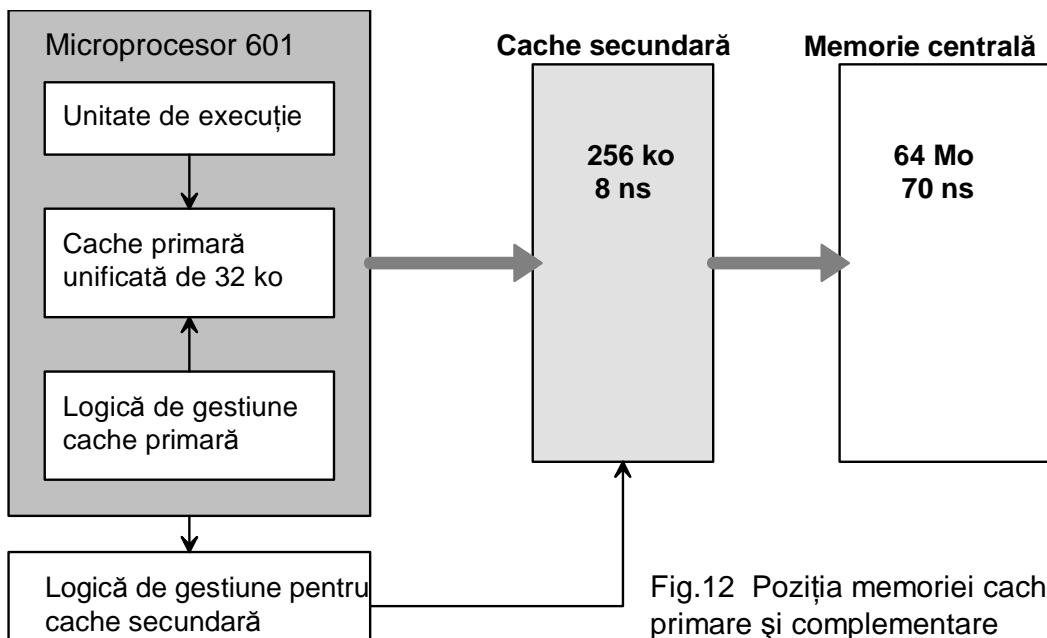


Fig.12 Poziția memoriei cache primare și complementare

Figura 12 ilustrează amplasarea memoriei cache primare în interiorul microprocesorului (Power PC) iar a celei secundare între microprocesor și memoria centrală iar figura 13 arată mai precis poziția memoriei cache primare în structura internă a lui Power PC 601. Memoria este partajată în cache de date și cache de instrucțiuni.

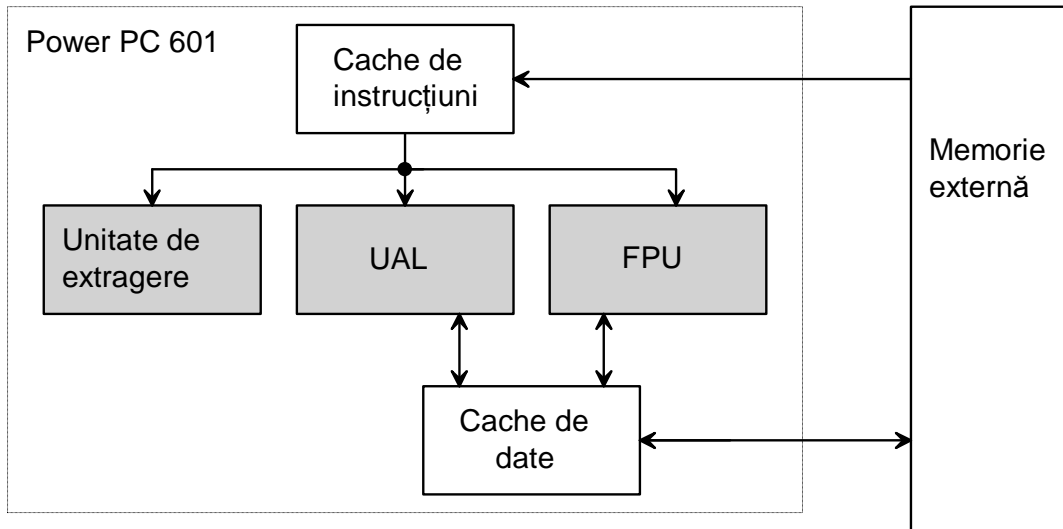


Fig. 13 Relația între memoria cache primară și celelalte unități interne

1.4.5. Memorii caches unificate, memoriile cache separate

Informația stocată în memoria intermediară poate fi de două tipuri: instrucțiuni (programe) și numere (operanzi). Ca urmare, pot fi imaginat două tipuri de arhitecturi:

- ♦ Cu o singură memorie cache, în care sunt stocate atât instrucțiuni cât și date; această memorie este numită **cache unificată**. Este mai simplu de implementat, dar nu este formula optimă.
- ♦ Cu două memoriile cache distincte, una destinată instrucțiunilor iar alta pentru date. Realizarea este mai dificilă dar este formula cea mai eficientă. Se înlătură o cauză de conflict la accesarea memoriei, care se manifestă la cele unificate. Se utilizează denumirea de **cache separate** iar structura este numită **arhitectură Harvard**.

1.4.6. Proceduri de supraveghere a memoriei cache

Informația din memoriile caches este o copie a celei din memoria centrală. Acest fapt este verificat prin intermediul unor proceduri speciale și este numit *coherență*. Verificarea se face sub controlul procesorului. Cea mai uzuală strategie este protocolul MESI (*Modified, Exclusive, Shared, Invalid*), cu cele 4 stări ale sale, înregistrate în cei 4 indicatori specifici pentru fiecare linie de date din cache.

- ♦ *Modified* (Modificare): linia în cache a fost modificată în raport cu conținutul memoriei, altfel spus, data din cache este validă dar cea corespunzătoare din memoria centrală nu este actualizată.

- ♦ *Exclusive* (Exclusiv): linia din cache conține date identice cu cele din memorie, la aceleași adrese. Nici o altă memorie cache nu dispune de aceleași date.
- ♦ *Shared* (Partajare): linia din cache conține date identice cu cele din memorie, la aceleași adrese. O altă memorie cache dispune de aceleași date.
- ♦ *Invalide* : linia din cache nu conține date valide.

Ținând seamă de valoarea indicatorilor (0 sau 1), procesorul poate verifica linie cu linie conformitatea și poate opera corecțiile necesare.

Protocolul MESI se poate aplica și sub o formă redusă.

Verificarea coerenței memoriei cache în raport cu conținutul memoriei centrale sau conținutul altei memorii intermediare este garantată de logica *snooping* (supraveghere), care funcționează în corelație cu protocolul MESI.

Coherența este necesară, în particular, atunci când un procesor *master* cere date memoriei centrale; el solicită procesoarelor *slave* examinarea memoriilor lor cache înainte de a ști dacă datele au fost deja copiate în memoria centrală.

De regulă, fiecare linie de memorie cache dispune de o etichetă de acces la un port dedicat tranzacțiilor cu magistralele prin *snooping*; de aceea, traficul produs de *snooping* nu interferă cu accesul procesorului la memoria cache.

1.4.7. Arhitectura memoriilor caches

Există mai multe tipuri de arhitecturi pentru memorii caches. Dintre acestea, cele mai folosite sunt:

1. Caches cu adresare directă;
2. Caches asociative;
3. Caches asociative multigrup sau cu căi multiple.

Memoriile caches cu adresare directă (*Direct Mapped Memory*) sunt copii fidele ale memoriei centrale, adică în memoria cache se reproduce locație cu locație (adresă cu adresă) tot ceea ce se petrece în memoria centrală. În acest caz, memoria cache citește adresa generată de procesor și furnizează informația pe care o posedă. Metoda este simplă și rapidă dar nu are eficiență maximă.

Memoriile caches asociative. Informațiile extrase din memoria RAM centrală sunt plasate (cu adresă cu tot) în cache la alte adrese, în funcție de spațiul liber disponibil (nu contează unde). Extragerea unei date se face prin analogie: controlerul de cache primește adresa operandului și verifică dacă aceasta se găsește în memoria cache, parcurgând adresă cu adresă. Memoria cache poate înregistra rapid cele mai recente informații

dar căutarea unui operand necesită logică suplimentară și durează mai mult.

Memoriile caches asociative multigrup. Acest tip de arhitectură asociază elemente din cele două tipuri anterioare. Memoria cache este divizată în două, patru, sau opt grupe, care înregistrează fiecare câte un bloc de date.

- ♦ *Un bloc de date din memoria centrală se poate găsi în oricare din grupele memoriei cache.*

Într-o memorie asociativă cu două grupe (căi), există două poziții posibile pentru un operand; dacă sunt 4 grupe, există 4 poziții posibile etc.

Adresa este divizată în două părți:

1. Partea superioară sevește ca etichetă (*Tag*) și permite selectarea grupei în care se află operandul căutat;

2. Partea inferioară localizează operandul în grupa selectată.

Utilizând divizarea adresei, controlerul de cache determină mai întâi grupa în care se află operandul, pe baza etichetei (*Tag*) și apoi compară partea inferioară a adresei cu adresele din grupă. Localizarea este mai rapidă deoarece căutarea în grupe se face în paralel.

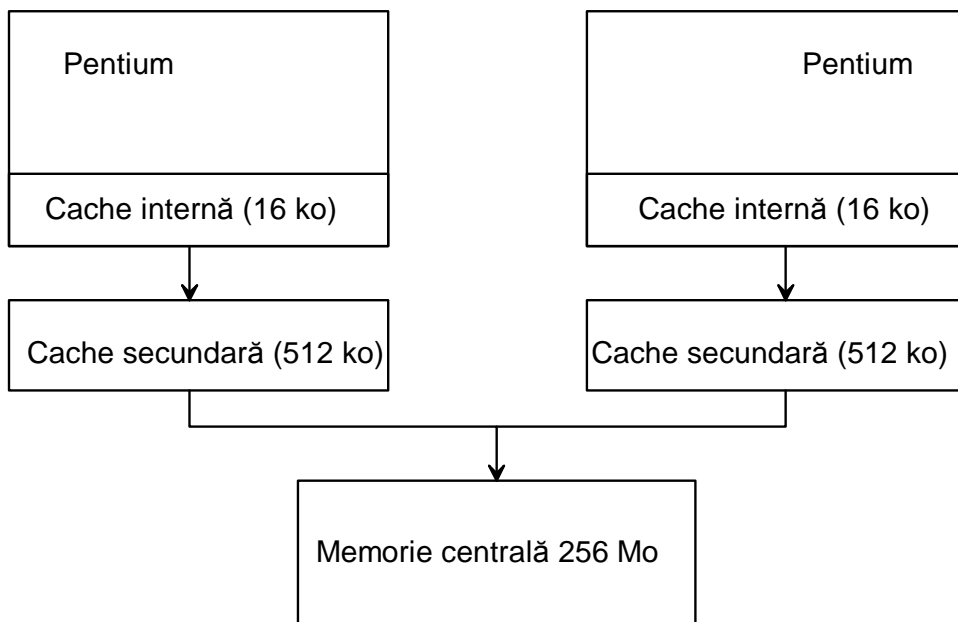


Fig. 14 Sistem multiprocesor simetric cu două memorii caches secundare distincte

1.4.8. Algoritmii LRU și MRU

În mod clasic, memoriile intermediare dispun de un algoritm de plasare a datelor, fondat pe următoarele noțiuni:

- ♦ LRU (*Least Recently Used*) referitor la date "*mai puțin recent utilizate*", adică date neutilizate de mai mult timp, date "*vechi*".

Conform acestui algoritm, datele vechi sunt înlocuite în cache cu cele mai recent utilizate, până la umplerea completă a memoriei.

- ♦ MRU (*Most Recently Used*), referitor la datele cele "mai recent utilizate", adică cele mai proaspete. Variabilele din program mai frecvent utilizate, sunt transferate în cache cu prioritate, indiferent unde se află în memoria centrală. Algoritmul este folosit în mod curent în modul de lucru *predictiv*, caracteristic procesoarelor de înaltă performanță.

1.4.9. Sisteme multiprocesor

Creșterea continuă a performanțelor microprocesoarelor permite producerea de microcalculatoare cu performanțe comparabile cu cele ale calculatoarelor de categorie superioară. În aplicațiile de tip client-server, sistemele complexe de calcul pot fi înlocuite cu microsisteme cu mai multe procesoare (multiprocesor). În aceste cazuri este imperativă utilizarea memoriilor caches. Se utilizează două strategii de bază:

- ♦ Montaj simetric - SMP (*Symmetric Multi-Processing*), format din două sau patru microprocesoare identice funcțional și care pot fi schimbate între ele; nu contează care dintre ele execută un anumit program.
- ♦ Montaj asimetric AMP (*Asymmetric Multi-Processing*) în care procesoarele sunt specializate, nu sunt de același tip și nu execută

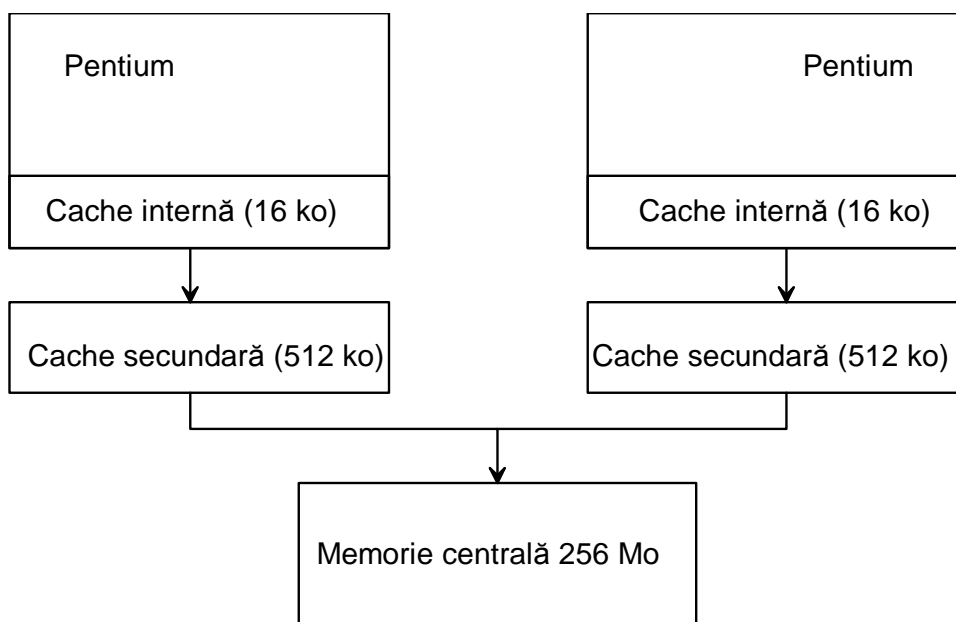


Fig. 14 Sistem multiprocesor simetric cu două memorii caches secundare distincte aceleași programe.

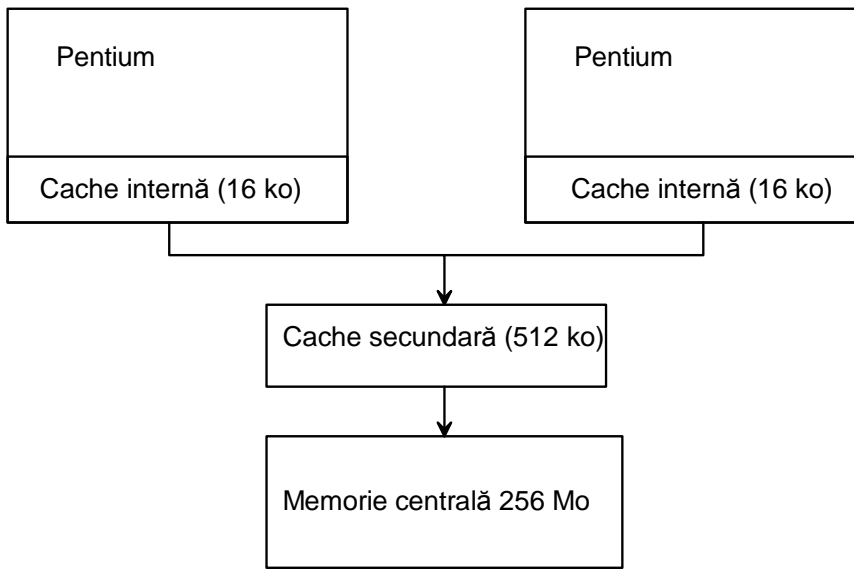


Fig. 15 Sistem multiprocesor simetric cu memorie cache comună

De fapt, montajul fizic cu mai multe procesoare constituie numai un aspect al problemei. Celelalte două sunt:

- ♦ Partajarea timpului de funcționare pentru fiecare procesor;
- ♦ Programele, care trebuie să exploateze sistemul multiprocesor.

O memorie cache unică pentru două procesoare (fig.15) este o soluție economică dar mai puțin eficientă în comparație cu soluția bazată pe memorii cache externe separate (fig.14). Problema coerenței între memoria cache și memoria centrală rămâne una de importanță deosebită.