
18 Arhitectura microprocesorului Intel 386

Succesor al microprocesorului Intel 80286, a apărut în 1985 și este un procesor de 32 de biți. În versiunea de bază, pe 32 de biți, a fost numit 386 DX iar în versiunea apărută în 1988, de 16 biți, cu magistrala de adrese de 24 de biți, a fost numit 386 SX.

Intel 386 a marcat spectaculoase salturi calitative în evoluția microprocesoarelor, în special prin introducerea paginării memoriei și a memoriei caché.

Caracteristici de bază:

- ◆ realizat în tehnologie CMOS, 132 pini, tensiune de alimentare 5 V;
- ◆ magistrala de date de 32 de biți, cu posibilități de funcționare pe 8 sau 16 ;
- ◆ magistrala de adrese de 32 de biți, ceea ce permite adresarea unei memorii fizice de 4 GB și adresarea unei memorii virtuale de 64 TB (adrese de 46 biți);
- ◆ lungimea maximă a segmentelor de memorie se extinde pentru prima dată de la 64 kB la 4 GB;
- ◆ dimensiunea registrelor interne se extinde la 32 de biți;
- ◆ frecvența de tact: 16 MH, 20 MH, 25 MH, 33 MH;
- ◆ conține o unitate specializată în gestionarea memoriei (*Management Memory Unit - MMU*) cu funcțiile de segmentare, paginare și calculul adreselor reale și virtuale;
- ◆ fiind o extensie a familiei 80x86, setul de instrucțiuni conține toate instrucțiunile predecesorilor și conține instrucțiuni noi pentru transfer de date, operații aritmetice și pentru operații la nivel de bit;
- ◆ dispune de un coprocesor aritmetic propriu, specializat, 80387 DX, care va fi introdus în același circuit integrat cu 80386 DX CPU, rezultând *Intel 486*;

Microprocesorul Intel 80386 DX are o structură internă formată din 6 unități conectate la o magistrala internă, care funcționează în modul *pipeline*:

1. Unitatea de interfață cu magistralele - (BIU);
2. Unitatea extragere a instrucțiunilor din memorie (IU);
3. Unitatea de decodare a instrucțiunilor (IDU);
4. Unitatea de execuție (EU);
5. Unitatea de segmentare a memoriei (MMU);

6. Unitatea de paginare a memoriei (MMU);

Unitatea de gestionare a memoriei (MMU) conține unitatea de segmentare și unitatea de paginare. Cea de segmentare permite gestionarea eficientă a memoriei virtuale (realocarea de memorie pentru date și coduri) și implemetarea tehnicii *multitasking*. Unitatea de paginare funcționează în paralel cu cea de segmentare și permite gestionarea eficientă a memoriei fizice. Dacă se detectează inexistența adresei unui operand (segment inexistent sau pagină inexistentă) în memoria internă (RAM), se reia instrucțiunea curentă cu adresa în memoria externă (disc).

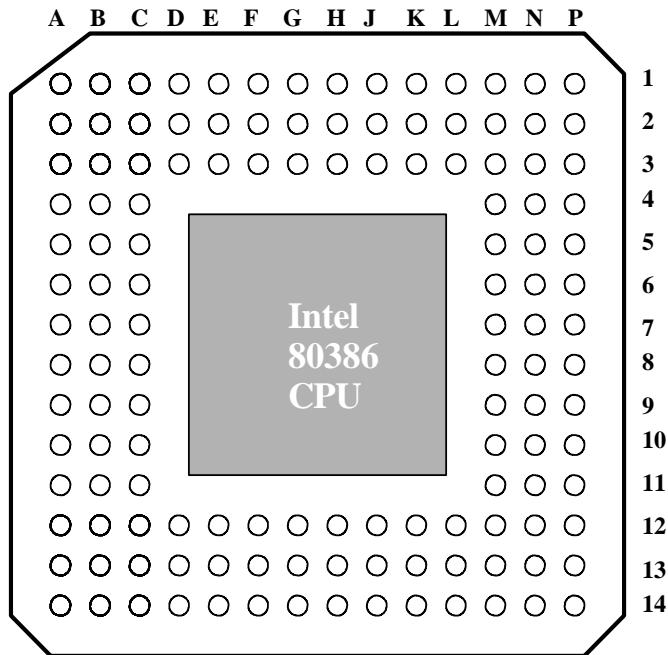


Fig.1 Intel 80386 - configurația pinilor (vedere dinspre pini)

18.1 Unitatea centrală de procesare

Conține unitățile de tratare a instrucțiunilor și unitatea de execuție. Unitățile de tratare sunt: unitatea de instrucțiuni și unitatea de decodare; prima extrage instrucțiuni din memorie și le plasează într-un fișier de așteptare (coadă de instrucțiuni) iar a doua are funcția de decodare (stabilește tipul operației, numărul de operanzi, modul de acces la operanzi, numărul de octeți din instrucțiune etc.). După decodare, instrucțiunile sunt transformate într-o succesiune de comenzi care vor fi transmise unității de execuție.

Unitatea de execuție conține cele 8 registre de uz general de 32 de biți (pentru adrese și operanzi) și un registru de deplasare de 64 de biți utilizat pentru deplasări, rotații, înmulțiri și împărțiri.

Algoritmul de înmulțire, de exemplu, realizează o înmulțire normală, pe 32 de biți, în cel mult o microsecundă.

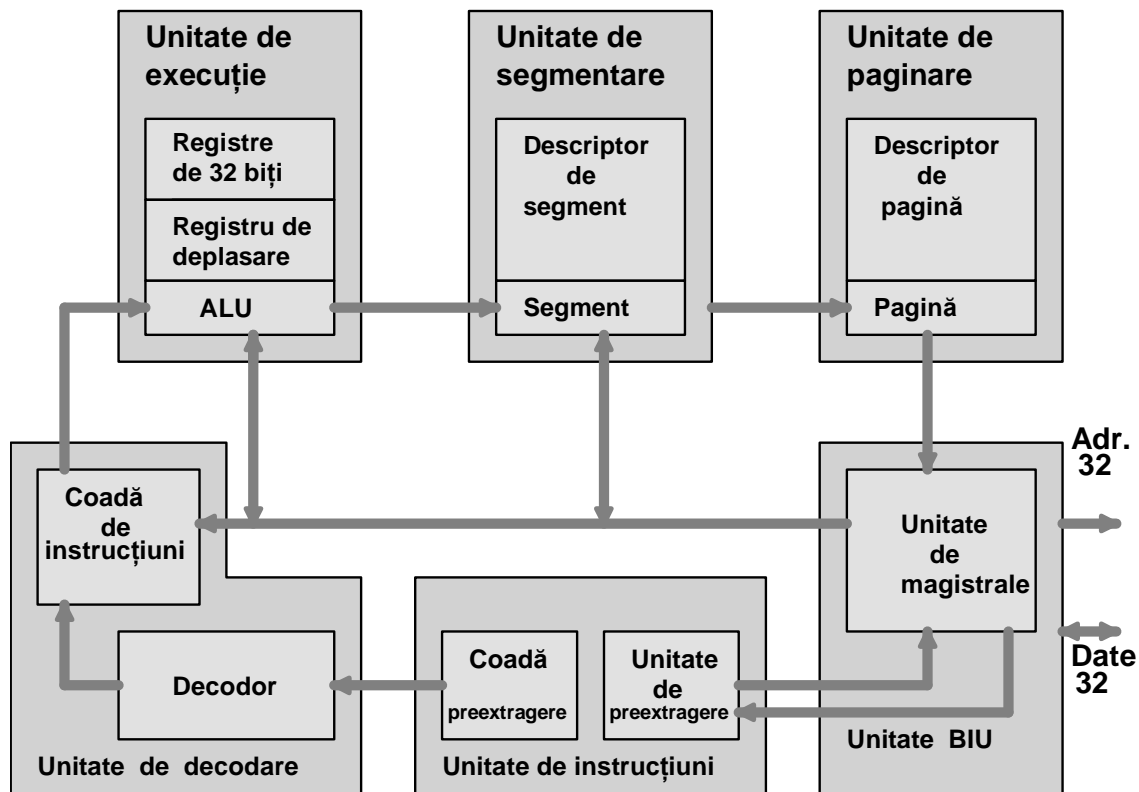


Fig. 2 Arhitectura internă a microprocesorului Intel 80386

18.2 Unitatea de gestionare a memoriei (MMU)

Conține unitatea de segmentare și unitatea de paginare. Unitatea de segmentare organizează memoria în segmente de lungime variabilă de maxim 4GB, fiecare segment având asociate atribute care precizează localizarea (adresa de început și lungimea în octeți), tipul (stivă, cod, sau date) și condițiile de acces.

Fiecare program (task) poate utiliza maxim 6 segmente simultan dar poate accesa 16 384 segmente de maxim 4GB fiecare. Pentru izolarea și protejarea reciprocă a aplicațiilor și sistemului de operare, unitatea de segmentare furnizează 4 niveluri de protecție (ca la 286).

Unitatea de paginare divide, din punct de vedere logic, memoria în unități de lungime fixă - 4 kB, în mod independent față de divizarea în segmente. Paginarea a fost impusă de conflictul între două deziderate: lungimea segmentelor să fie cât mai mare dar transferul unui segment din

memoria de masă în RAM (și invers) să fie cât mai rapid. Paginarea permite transferul rapid de date pentru blocuri de lungime fixă de 4 kB, indiferent de lungimea segmentului logic din care face parte blocul de date. În acest mod, pentru execuția unui task nu sunt necesare în memoria RAM toate paginile de cod și date simultan ci doar un număr redus dintre acestea. Dacă mai multe task - uri sunt executate "simultan", toate au paginile curente încărcate în memoria RAM, astfel că este posibilă comutarea rapidă a execuției de la un task la altul.

18.3 Unitatea de interfață cu magistralele

Generează semnalele electrice pentru citire/scriere memorie, citire/scriere porturi, stabilirea dimensiunii și sensului de transfer pentru magistrala de date, arbitrarea magistralelor (cedarea, preluarea), toate operațiile fiind coordonate cu activitatea unității centrale de procesare.

18.4 Registrele microprocesorului Intel 80386

Registrele de uz general sunt aceleași ca la Intel 8086/80286, dar de 32 de biți, cu posibilități de operare pe 8, 16, 32 și 64 de biți; ca registre de 32 de biți au denumirile EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP (*Extended AX, . . .*).

Față de setul de registre al lui *Intel 286*, apar în plus următoarele registre:

- ♦ FS și GS, registre pentru selectori de segmente de date curente (ca și DS, ES);
- ♦ CR2 - utilizat în mecanismul paginării, conține adresa liniară de 32 de biți a instrucțiunii care a cauzat eroarea de găsimă a paginii (*page fault*); informații suplimentare referitoare la acest eveniment se află în codul de eroare încărcat în stiva de tratare a erorilor de pagină;
- ♦ CR3 - conține baza adresei fizice a tabelii directorilor de pagină; această tabelă este întotdeauna aliniată la intervale de 4 kB = 2^{12} B, deci cei 12 biți mai puțin semnificativi trebuie să fie întotdeauna zero;
- ♦ TR - registrul segment al stării task - ului (*TSS - Task Status Segment*) conține selectorul descriptorului TSS împreună cu adresa liniară, limitele și atributele descriptorului respectiv și este utilizat în mecanismele de protecție, în modul de lucru protejat.

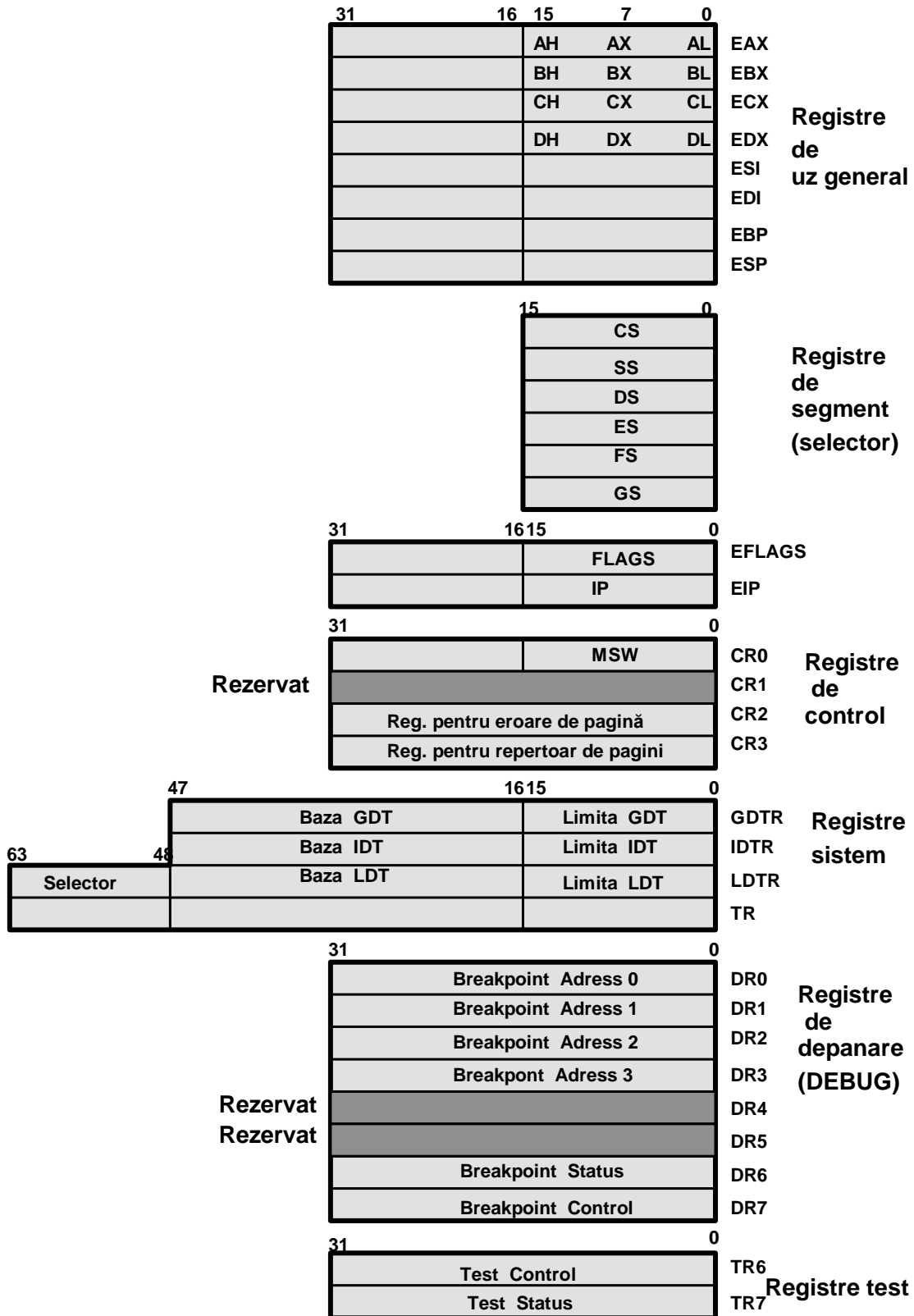


Fig.3 Cele 7 grupe de registre ale microprocesorului Intel 80386

- ♦ DR0 - DR7 - registre pentru depanarea programelor; primele 4 registre sunt destinate memorării adreselor liniare ale punctelor de oprire (*breakpoints*) inserate în program în procesul depanării.
- ♦ TR6, TR7 - registre de test folosite în testarea memoriei; TR6 este registrul comenzilor de test iar TR7 este folosit ca registru de date.

Din motive de compatibilitate cu procesoarele viitoare, anumite registre sunt rezervate (deși există fizic, ele nu pot fi utilizate). În figura 3 sunt marcate prin hașurare.

18.5 Indicatorii de condiții

Intel 80386 are în plus (față de 286) încă 2 indicatori :

VM - Modul virtual 8086 (*Virtual 8086 Mode*) ; prin setarea acestui indicator, când procesorul se află în modul de lucru protejat, se trece în modul virtual 8086, adică procesorul lucrează cu segmentele de memorie ca 8086 dar păstrează toate protecțiile de acces la memorie.

RF - Indicator de reluare (*Resume Flag*) - este utilizat pentru punctele de întrerupere (*breakpoints*) înregistrate în DR0 - DR3; indicatorul este testat înainte de procesarea oricărui breakpoint iar dacă este găsit în 1, va fi ignorată orice eroare generată de instrucțiunea următoare. Indicatorul este resetat automat la încheierea cu succes a tuturor instrucțiunilor, cu excepția instrucțiunilor IRET, POPF, JMP, CALL și INT, care produc o comutare a task - ului.

Indicatorii de condiții sunt reprezentați la pagina 333, fiind aceiași ca la Intel 486.

18.6 Semnalele electrice de intrare / ieșire

Microprocesorul 386 DX are carcasa de ceramică de tip PGA (*Pin Grid Array*) cu 132 de pini, dimensiune 36 x 36 mm și un consum de 2 W.

Semnalele electrice (fig. 4) formează 7 grupe funcționale (exclusiv semnalul de tact și conexiunile de alimentare).

1. Magistrala de date D0 - D31 (bidirecțională, linii cu 3 stări).

2. Magistrala de adrese A2 - A31, (30 de linii) împreună cu cele 4 linii de validare octet BE0 - BE3 (*Byte Enable*), active în 0, permite adresarea unei memorii de $1\text{GB} \times 4 = 4\text{GB}$.

Biții A0, A1 sunt decodificați intern, rezultând cele 4 combinații posibile BE0 - BE3, care pot fi utilizate în adresarea memoriei.

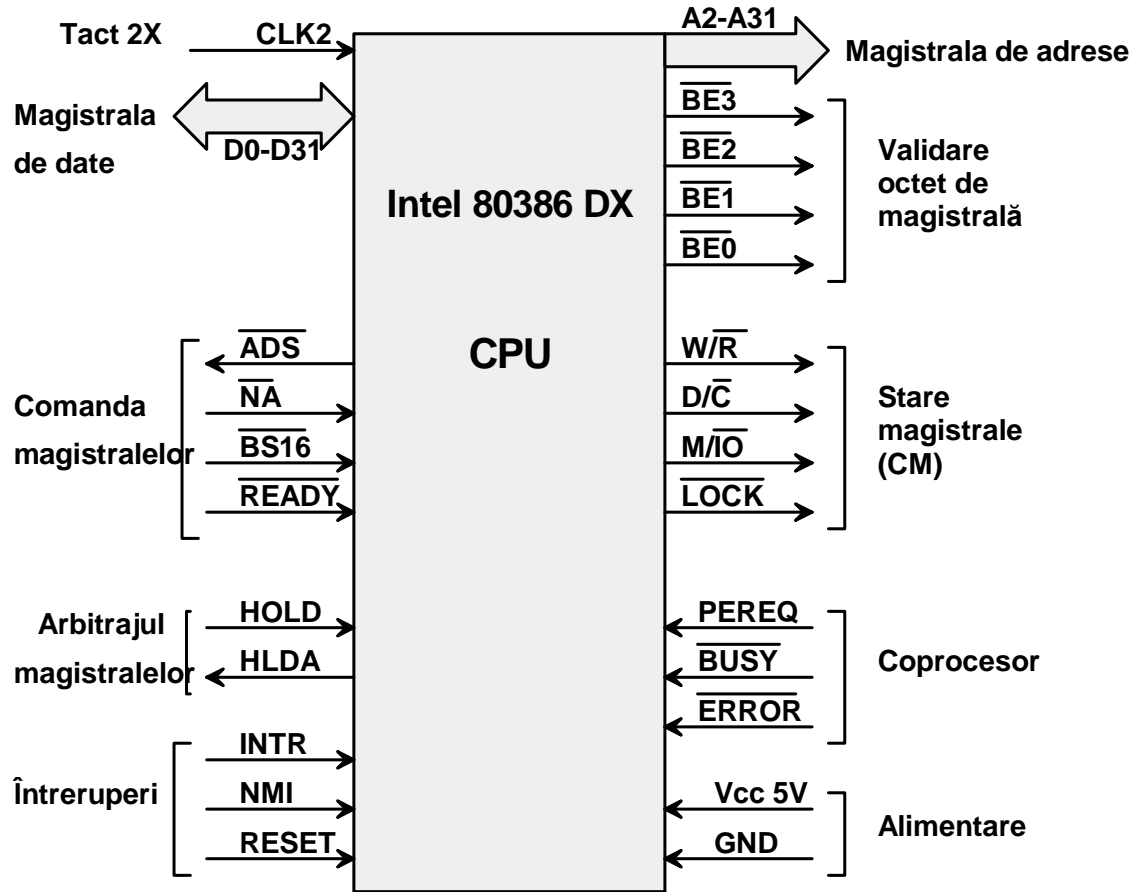


Fig.4 Distribuția funcțională a semnalelor la 80386

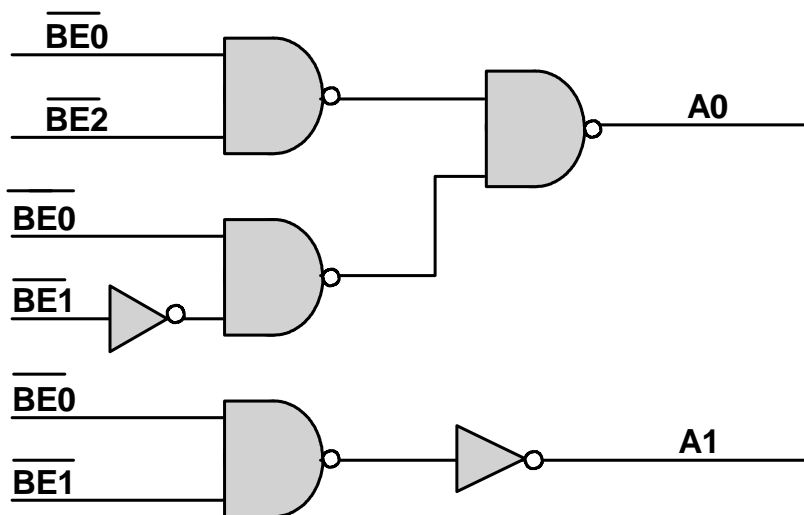


Fig. 5 Schema logică de generare a semnalelor A0, A1

În fig.5 este prezentată schema logică (inversă) de generare a liniilor de adresă A0, A1 din liniile BE0 - BE3.

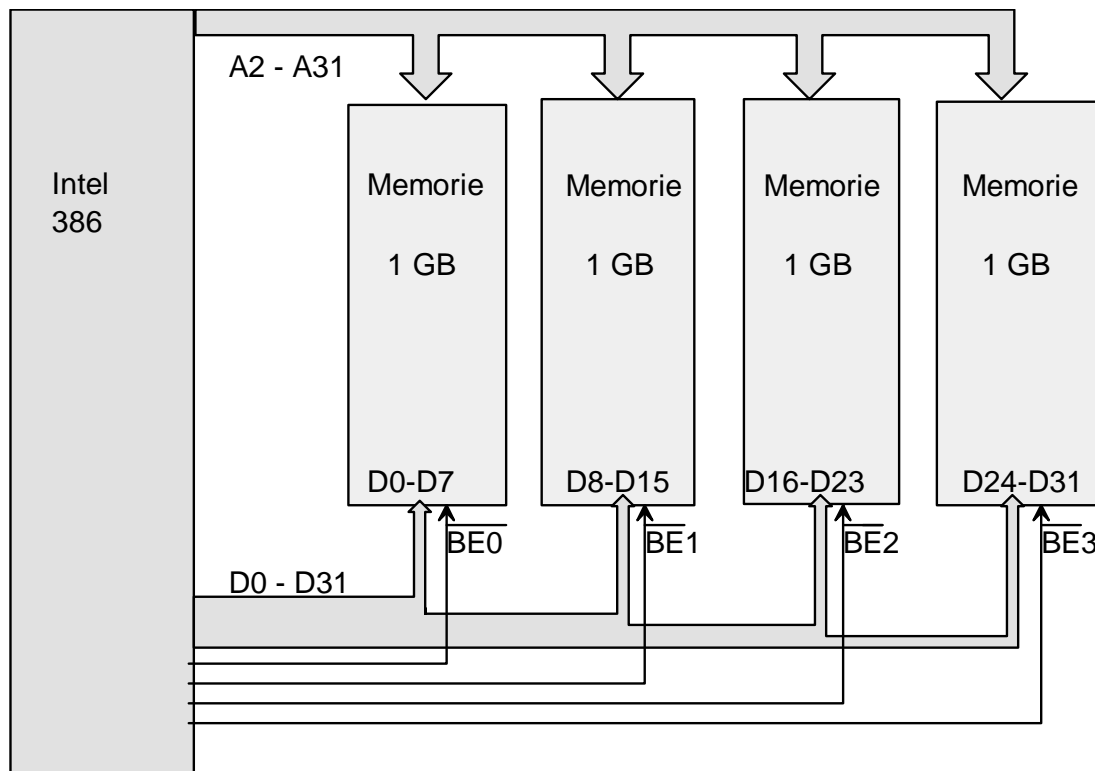


Fig.6 Adresarea a 4 blocuri de memorie de 1G x 8 biți fiecare prin utilizarea liniilor BE0-BE3

În figura 6 este prezentată schema de principiu a adresării memoriei folosind liniile de validare octet (*Byte Enable*): BE0 validează transferul pe octetul D0-D7, BE1 validează transferul pe octetul D8-D15, BE2 validează transferul pe octetul D16-D23 iar BE3 validează transferul pe octetul D24-D31.

Transferul de date se poate face în mod flexibil pe 8, 16, 24 sau 32 de biți, în funcție de liniile de validare (BE) active, memoria fiind organizată fizic în locații de 8 biți.

3. Semnale pentru controlul magistralelor (tipul de ciclu mașină):

- ♦ W/R comandă de scriere (1 logic) sau citire (0 logic);
- ♦ D/C magistrala de date conține operand (1 logic) sau instrucțiune (0 logic);
- ♦ M/IO operație cu memoria (1 logic) sau cu porturile (0 logic);
- ♦ LOCK magistralele nu pot fi cedate în ciclul mașină curent (0 logic).

4. Semnale pentru comanda magistralelor:

-
- ♦ ADS (*Adress Status*) când este activ (0), validează adresa de pe magistrală;
 - ♦ READY indică procesorului (în 0) că memoria răspunde cu date pe magistrală;
 - ♦ NA (*Next Adress*) cerere pentru adresa următoare;
 - ♦ BS16 (*Bus Size 16*) dimensiune magistrală de date = 16 biți (cerere externă);

5. Arbitrajul magistralelor:

- ♦ HOLD cerere de cedare a magistralelor;
- ♦ HLDA (*Hold Acknowledge*) răspuns privind autorizarea cedării magistralelor;

6. Întreruperi

- ♦ RESET inițializare, de maximă prioritate, trebuie să rămână activ 15 perioade de tact pentru a fi luat în considerare;
- ♦ NMI cerere de întrerupere nemascabilă cu prioritate medie;
- ♦ INTR cerere de întrerupere mascabilă, de prioritate joasă;
- ♦ INTA răspuns privind acceptarea unei cereri de întrerupere.

7. Semnale de interfață cu un coprocesor (protocol de colaborare):

- ♦ PEREQ (*Processor Request*) cerere de la un coprocesor aritmetic către 386 pentru acces la magistrale în vederea transferului de date;
- ♦ BUSY ocupat, coprocesorul este în stare "ocupat", 386 trebuie să aștepte;
- ♦ ERROR eroare: coprocesorul semnalizează eroare de calcul (depășire etc.).

18.7 Adresarea memoriei (segmentarea și paginarea)

Calculul adresei este mai subtil decât la 286, nu numai pentru că registrele sunt de dimensiune dublă dar se utilizează în plus paginarea.

1. Segmentarea

Partea vizibilă a registrelor segment este de 16 biți dar ele sunt de fapt de 80 de biți, deoarece se adaugă partea invizibilă de 64 de biți - descriptorul de segment - stocat în memorie, într-o tabelă de descriptori.

Adresa logică sau virtuală este o adresă pe 48 de biți, din care 16 biți constituie "selectorul" din registrul segment iar restul de 32 de biți constituie OFFSET - ul.

Adresa liniară pe 32 de biți, rezultă prin însumarea adresei de bază (extrase din descriptor) cu OFFSET - ul furnizat de modul de adresare.

Adresa liniară devine **adresă fizică** dacă nu este utilizată paginarea.

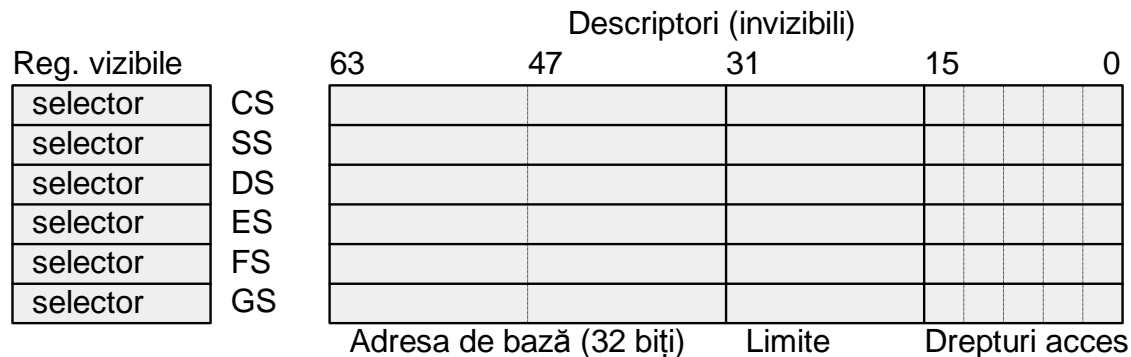


Fig.7 Registrele segment extinse la 80 de biți (selector + descriptor)

Câmpul **selector** din registrul segment este utilizat ca etichetă pentru extragerea descriptorului corespunzător din tabela de descriptori; după extragere, descriptorul furnizează toate informațiile necesare pentru adresarea memoriei.

Când un program se află în execuție, toate registrele segment sunt încărcate cu selectori, astfel că programul are acces la 6 segmente de memorie simultan, prin intermediul celor șase descriptori de segment corespunzători.

2. Paginarea

Unitatea de paginare transformă adresa liniară (32 de biți) în adresă fizică prin divizarea acesteia în 3 câmpuri: index, pagină și deplasament.

Dimensiunea unei pagini fiind fixată la 4kB, rezultă că un segment de 4GB conține 1 000 000 pagini, ceea ce impune constituirea unei tabele cu adresele de început de pagină care trebuie să conțină un milion de intrări în pagină (4MB memorie ocupată).

Pentru reducerea memoriei ocupate și creșterea vitezei de selecție, generarea adresei fizice se face în două etape, pornind de la adresa liniară.

Prima etapă constă în extragerea adresei de bază a tabelii de pagini din tabela directorilor de pagină (sau repertoarul de pagini); pentru acces

se utilizează conținutul registrului CR3 ca adresă de bază și câmpul *Index*

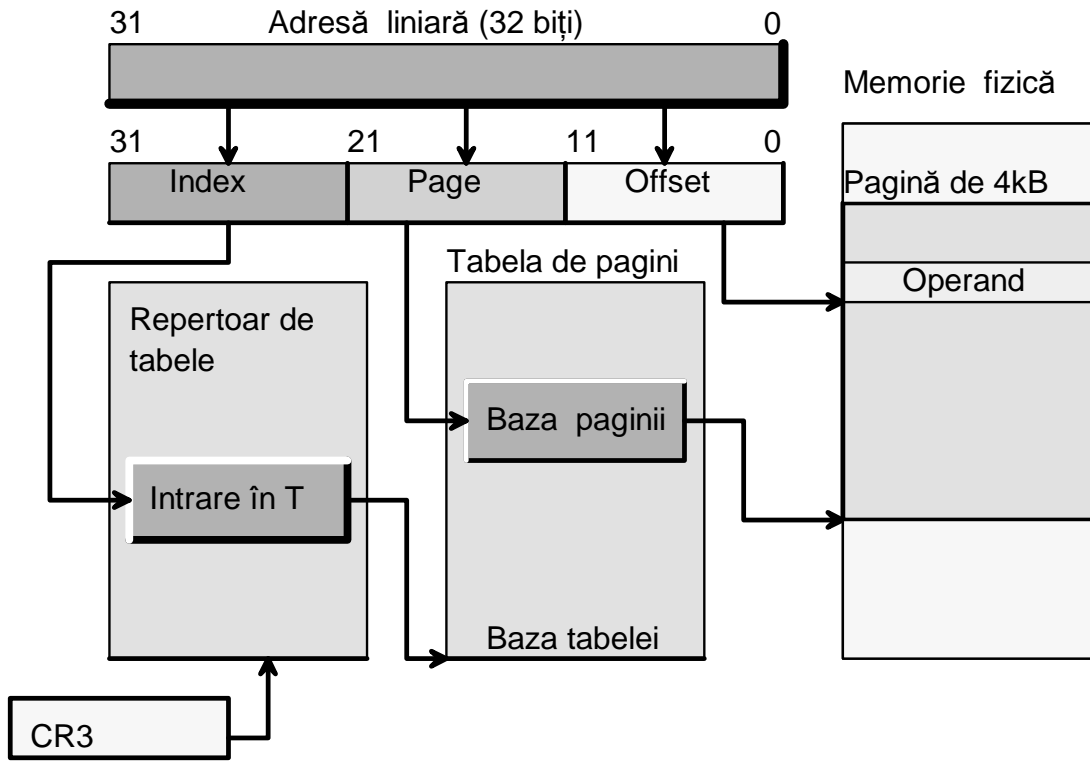


Fig. 8 Principiul adresării memoriei când se utilizează paginarea

(biții 22 - 31) al adresei liniare ca offset.

Se pot adresa $2^{10} = 1024$ baze de 4 octeți pentru tabelele de pagini.

O tabelă de pagini conține la rândul său 1024 de adrese de pagină (intrări în pagină).

Pentru extragerea unei intrări în pagină, se utilizează câmpul *Page* al adresei liniare, de 10 biți, ca Offset față de baza tabelii. O intrare în pagină este o adresă fizică de 32 de biți care este de fapt adresa fizică a primului octet al paginii.

Pentru obținerea operandului adresat, se utilizează adresa de bază a paginii extrase din tabelă și ca deplasament în pagină, câmpul *Offset* de 12 biți (0 - 11).

Cu cei 12 biți se pot adresa 4096 octeți / pagină = 4kB / pagină.

3. Adresarea memoriei

Microprocesoarele Intel 386 / 486 au 11 moduri de adresare a memoriei pentru specificarea operanzilor.

Aceste moduri sunt optimizate pentru a permite execuția eficientă a programelor scrise în limbaje de nivel înalt (Pascal, C), acoperind toate modurile de referire la date, necesare în aceste limbaje de programare, fiind avantajate modurile de adresare utilizate frecvent.

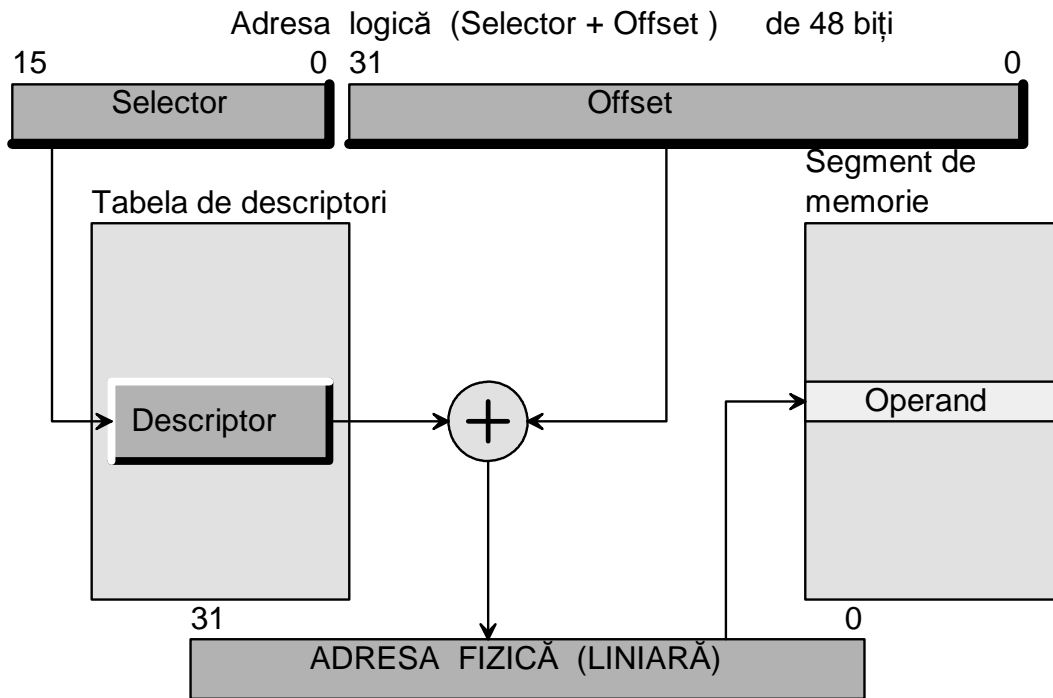


Fig. 9 Principiul adresării memoriei când se utilizează numai segmentarea

Tabel 1. Parametrii utilizați în adresarea memoriei

Parametrul	Semnificația
Deplasament	Valoare imediată pe 1, 2 sau 4 octeți, specificată în instrucțiune.
Adresă efectivă	Adresă pe 32 de biți, calculată în funcție de modul de adresare, specificat în instrucțiune.
Offset	Sinonim cu Adresă efectivă (decalaj)
Adresă liniară	Suma dintre adresa de bază a segmentului și Offset
Adresă fizică	Adresa încărcată pe magistrală pentru selecția unui operand din memorie.

Bază	Conținutul unui registru de uz general (fără ESP)
Index	Conținutul unui registru de uz general (fără ESP)
Factor de scală	Constantă care multiplică un index (1, 2, 4, 8).

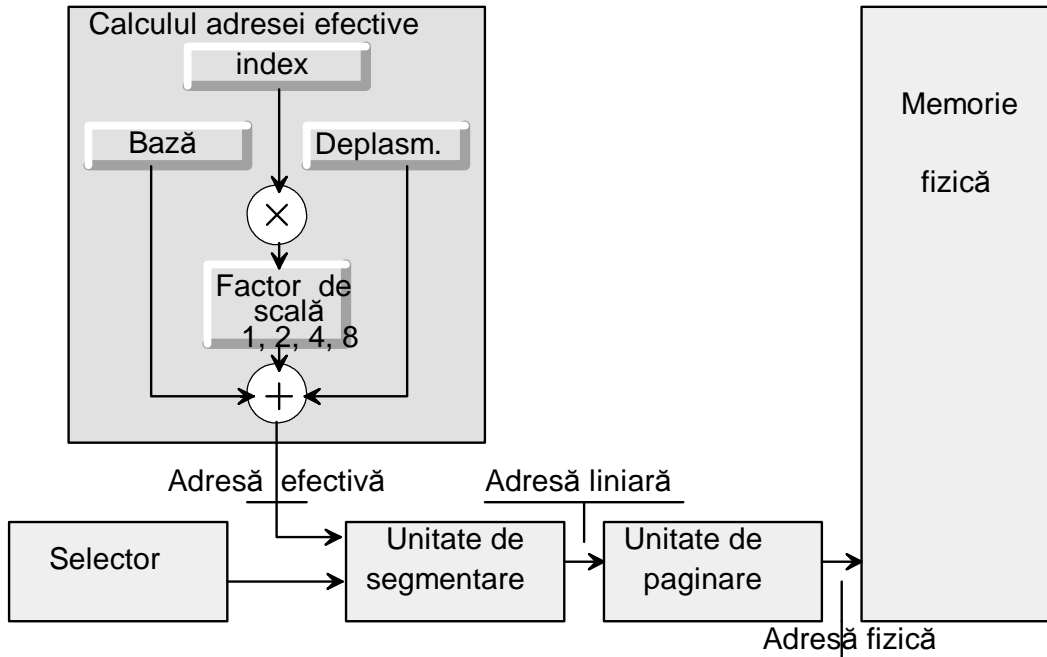


Fig. 10 Schema generală de adresare. Modul de calcul al adresei efective (inclusiv factorul de scală) se specifică în instrucțiune.

4. Descriptorii de segment

Păstrează compatibilitatea cu descriptorii lui Intel 80286, primii 6 octeți fiind identici iar octetul drepturilor de acces (5) are aceeași structură și semnificație a biților. Au apărut în plus octeții 6 și 7, biții octetului 6 având următoarele semnificații:

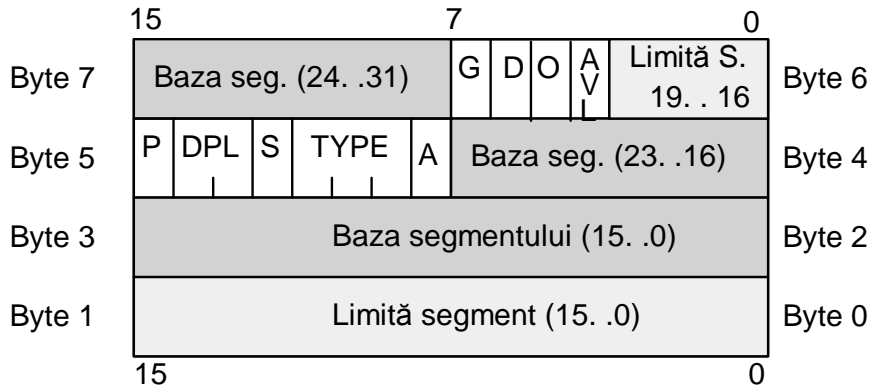


Fig. 11 Structura descriptorilor de segment la Intel 386/486.

- ♦ **G** - *Granularity* - bit de divizare a memoriei: dacă este 1 logic, unitatea de transfer în cadrul segmentului este pagina (4kB) iar dacă este 0 logic, unitatea de transfer este octetul.
- ♦ **D** - *Dimension* - dimensiunea implicită a operanzilor: dacă este 1 logic operanzii sunt implicit de 32 de biți iar dacă este 0 logic operanzii sunt de 16 biți și în plus, segmentele au dimensiunea maximă de 64 kB.
- ♦ **AVL** - bit la dispoziția utilizatorului sau a sistemului de operare.

Baza segmentului este de 32 de biți iar limita de 20 de biți. Rezultă o dimensiune maximă de 2^{20} unități de transfer/segment, care pot fi pagini de 2^{12} octeți sau octeți ($G=0$), caz în care dimensiunea maximă a segmentului devine 1MB.

19 Arhitectura microprocesorului Intel 486

Apărut în 1989, constituie un salt important în tehnica microprocesoarelor. Acest salt a fost posibil datorită progresului realizat în domeniul tehnologiilor de integrare, astfel că *i 486* conține 1.000.000 de tranzistoare integrate pe o pastilă de siliciu cu suprafața de cca. 300 milimetri pătrați. El conține într-un singur "*chip*" un microprocesor *i 386*, coprocesorul său aritmetic *i 387*, o unitate de gestiune a memoriei (MMU) și o memorie SRAM de 8 ko (cache).

Ca și predecesorii săi, *i 486* a apărut în două variante: 486 DX și 486 SX; de data aceasta însă, varianta SX dispune de aceleași magistrale de 32 de biți și în mod riguros de aceeași arhitectură dar are coprocesorul aritmetic inhibat, ceea ce a permis firmei Intel o mai bună penetrare a pieței prin prețul semnificativ mai mic.

Ca noutate, Intel a inaugurat prin 486 o nouă tehnică, "*Over Drive*" de multiplicare a frecvenței de tact ce se aplică microprocesorului, astfel că prin înlocuirea *chip* - ului cu o variantă mai rapidă poate crește viteza de calcul fără nici o altă modificare hardware.

19.1 Caracteristici hard și soft

Microprocesor de performanță pentru unități centrale cu sisteme de operare evaluate UNIX, WINDOWS.

- ◆ Realizat în tehnologie CHMOS, 168 de pini dispuși matricial, alimentare 5Vcc;
- ◆ Semnal de tact extern, 20 - 100 MHz, cu multiplicare internă;
- ◆ Compatibilitate cu toate circuitele din familia Intel;
- ◆ Magistrala de date de 32 de biți (D0 - D31), magistrala de adrese de 32 de biți (A2 - A31, și BE0 - BE3), memorie fizică direct adresabilă de 4 Go, memorie virtuală adresabilă 64 To;
- ◆ Conține coprocesor aritmetic integrat (FPU - *Floating Point Unit*) care operează cu formate de 32, 64 și 80 biți / operand;
- ◆ Dispune de 8 ko de memorie SRAM internă - memorie cache (ascunsă) invizibilă pentru utilizator;

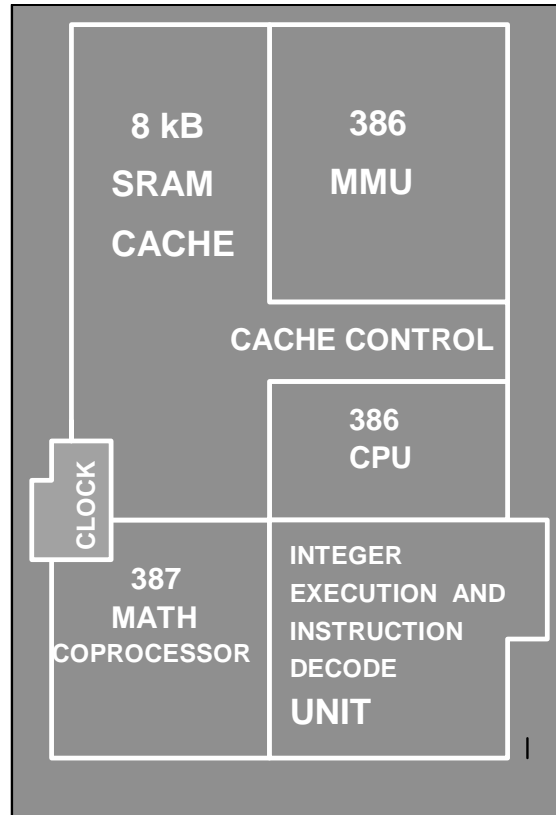


Fig. 1 Structura internă a microprocesorului i486 realizată după copie foto la scara 1:30.

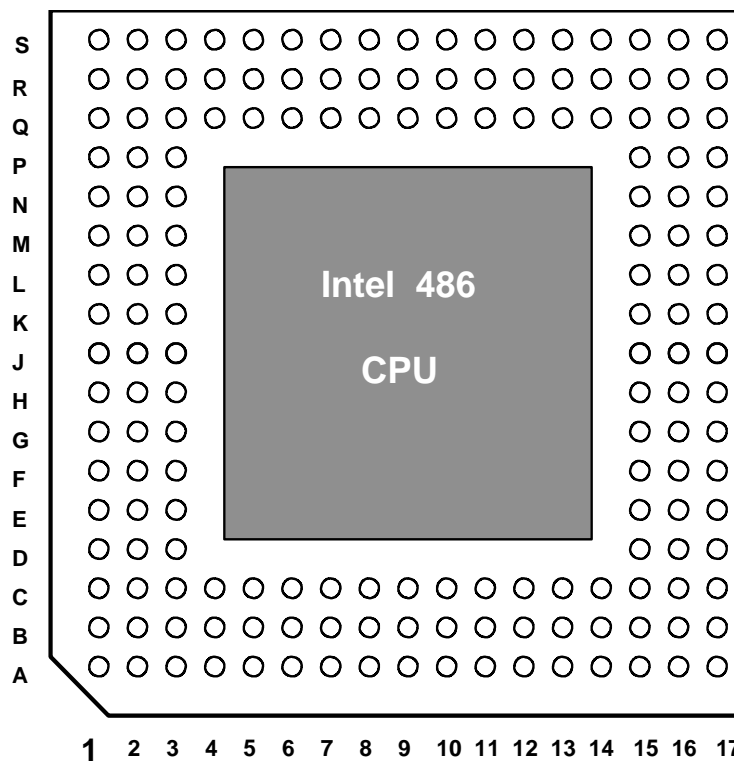


Fig. 2 Configurația pinilor la i486 - vedere dinspre pini.

- ♦ Magistrala de control permite funcționarea în sisteme multiprocesor (mai multe procesoare "slave" și un "master");
- ♦ Este asigurată compatibilitatea la nivel de cod mașină cu *i 386* (programele pentru 386 sunt executabile fără modificări);
- ♦ Setul de registre este identic cu cel de la *i 386* ca structură și dimensiuni;
- ♦ Segmentarea și paginarea memoriei ca la *i 386*; memoria este organizată în locații de 8 biți, cu adresă proprie în domeniu 0 - 4 Go; un program poate utiliza 16 384 segmente cu dimensiunea maximă de 4 Go (dar numai 6 simultan),

accesul la un segment se face prin intermediul descriptorului de segment, cu aceeași structură ca la *i 386*; paginarea permite fragmentarea memoriei în pagini de 4ko/pag.

19.2 Structura internă

Microprocesorul Intel 486 conține 8 unități interne *pipe-line* (care pot funcționa în principiu, simultan):

1. Unitatea de interfață cu magistralele (*Bus Interface*);
2. Unitatea de preextragere, care conține un fișier de 32 de octeți;
3. Unitatea de decodare a instrucțiunilor;
4. Unitatea de control și protecție (*Control Unit*);
5. Unitatea de calcul aritmetic în virgulă mobilă (FPU);
6. Unitatea UAL de calcul aritmetic și logic pentru operanzi întregi;
7. Unitatea de gestiune a memoriei (MMU);
8. Memorie SRAM de 8ko (cache) pentru instrucțiuni și operanzi.

19.3 Moduri de lucru

1. Modul real - execută instrucțiuni 8086 și altele în plus; este modul care se instalează la inițializare;

2. Modul protejat - utilizare la maxim a resurselor hardware și software;

3. Modul virtual protejat - se obține din modul protejat; permite executarea programelor scrise pentru 8086 cu păstrarea protecției privind accesul la date și trecerea de la magistrala de 16 biți la 32 și invers.

Mecanismul "salvă" (*Burst*) de magistrală permite încărcarea memoriei cache din memoria externă cu viteză sporită. În modul "salvă",

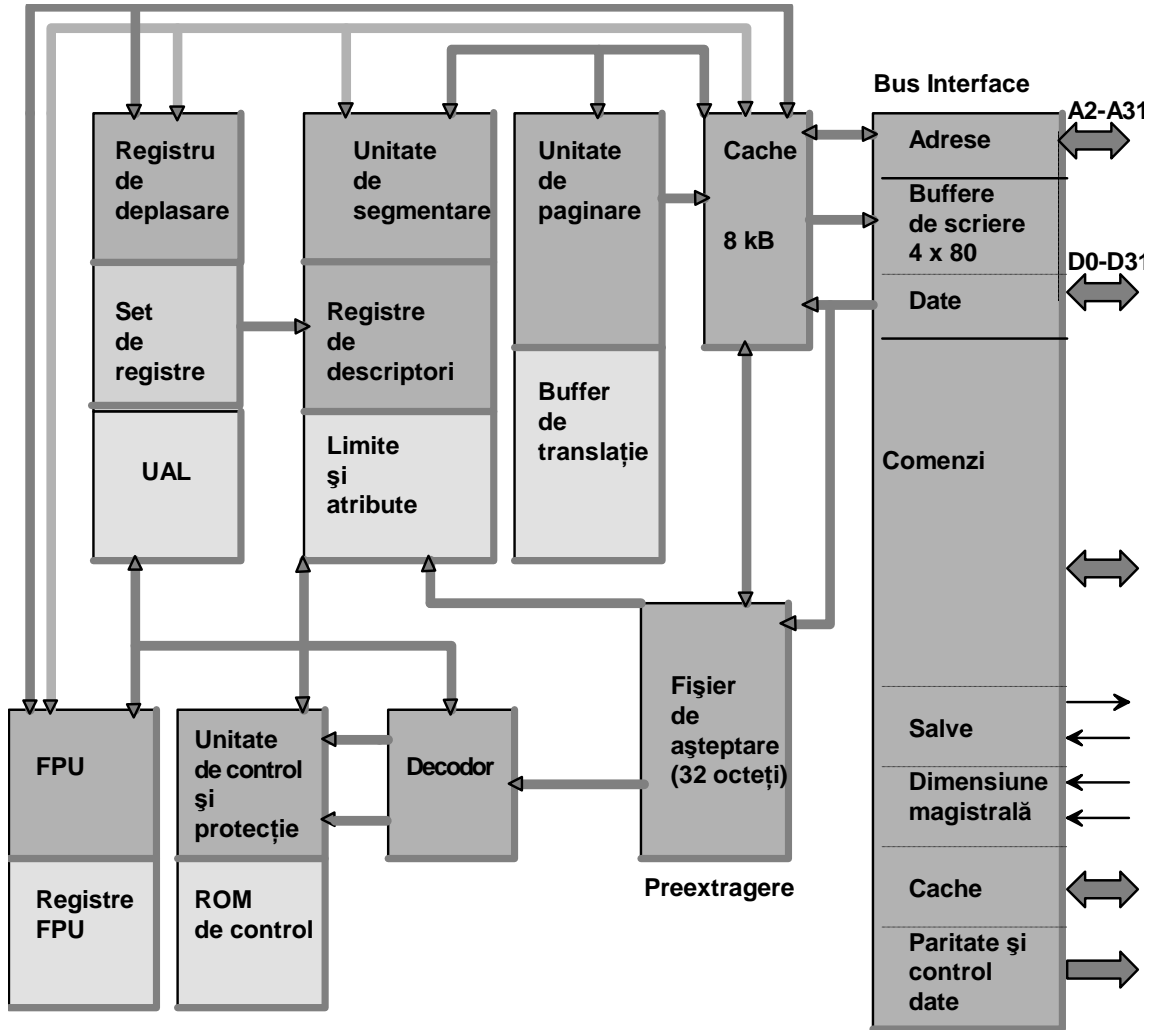


Fig.2 Intel 486 - arhitectura internă

datele pot fi validate la o rată de transfer de 1 - 16 octeți / perioadă de tact și se aplică în toate cazurile de transfer pentru blocuri de date.

Microprocesorul Intel 486 prezintă particularitatea de cedare a magistralei de adrese în plus față de cedarea tuturor magistralelor. În acest caz special, numai magistrala de adrese trece în starea de înaltă impedanță, magistralele de date și de control fiind active.

Această situație este folosită pentru rescrierea memoriei cache interne.

Un ciclu de magistrală (transfer de date) are cel puțin două perioade de tact; începe cu activarea lui ADS și se încheie prin activarea semnalului RDY sau BRY; un ciclu de magistrală poate fi compus din unul sau mai multe cicluri de date.

Grup funcțional	Simbol	Intr. I Ieșire E	Comentariu privind funcția semnalelor
Semnal de tact	CLK	I	Semnal de tact generat extern
Magistrala de adrese	A4 - A31 A2, A3	I / E E	Linii de adresă ; A0 și A1 se obțin cu BE0 - BE3. Se folosesc ca intrări pentru scriere în cache.
	BE0 - BE3	E	Byte Enable: specifică octetul autorizat pentru citire / scriere; BE0 pentru D0-D7; BE1 pentru D8-D15; BE2 pentru D16-D23; BE3 pentru D24-D31.
Magistrala de date	D0 - D31	I / E	Linii de date cu 3 stări; 0, 1, înaltă impedanță.
Control de paritate	DP0 - DP3	I / E	Biți de paritate pe octet
	PCHK	E	Parity Check - în stare 0 la detectarea erorii de paritate
Ciclu de magistrală	M/IO, D/C W/R	E	Cele 3 semnale definesc tipul de ciclu mașină; semnalele sunt validate de ADS.
	LOCK	E	Indică imposibilitatea cedării magistralelor în CM curent
	PLOCK	E	Pseudo-Lock : blocarea magistralelor încă un CM
Comenzi de magistrală	ADS	E	Address status: Adresa de pe magistrală este validă
	RDY	I	Ciclul de magistrală gata, transfer efectuat
Comenzi pentru salvă	BRDY	I	Burst Ready: RDY, dar pentru transfer de tip salvă
	BLAST	E	Burst Last: Următorul BRDY activ va marca sfârșitul transferului
Întreruperi	RESET	I	Inițializare
	INTR	I	Cereri de întrerupere mascabile
	NMI	I	Cereri de întrerupere nemascabile

Arbitraj de magistrale	BREQ	E	Cerere de acces la magistrale
	HOLD	I	Cerere externă de preluare a magistralelor
	HLDA	E	Hold Acknowledge: acceptare cerere de magistrale
	BOFF	I	Backoff: comandă de trecere în stare SIR a magistralelor pe durata perioadei de tact următoare.
Invalidare memorie cache	AHOLD	I	Address Hold: este acceptată o adresă pe liniile A4-A31
	EADS	I	O adresă externă este validată pentru CM de invalidare cache.
Controlul memoriei cache	KEN	I	Activare memorie cache
	FLUSH	I	Reîncărcarea întregii memorii cache
Paginare cache	PWT	E	Page Write Through (starea bitului PWT)
	PCD	E	Page Cache Disable (starea bitului PCD)
Erori numerice	FERR	E	Floating Point Error: eroare de calcul FPU
	IGNNE	I	Ignore Numeric Error: continuarea calculelor
Dimensiunea magistralei de date	BS16	I	Stabilire dimensiune la 16 biți
	BS8	I	Stabilire dimensiune la 8 biți
Masca adresei	A20M	I	Pentru modul real: linia de adr. A20 este suspendată pentru a simula un 8086.

19.4 Memoria cache

O noutate în arhitectura microprocesoarelor, memoria internă (cache) de tip RAM este utilizată pentru instrucțiuni și date (la Pentium sunt două memorii cache distincte).

Memoria cache este structurată în 4 blocuri de câte 2ko fiecare. Adresa fizică este divizată în câmpul *index* de 7 biți (A4 - A10) și câmpul *tag* de 21 de biți (A11-A31). Fiecare bloc este format din 128 de linii de 16 octeți fiecare.

- ◆ Selecția unui octet din cei 16 se face cu biții inferiori de adresă A0-A3;
- ◆ Selecția unei linii din cele 128 se face cu câmpul index (7 biți);
- ◆ Selecția blocului de 128 linii se face cu câmpul tag (21 de biți);

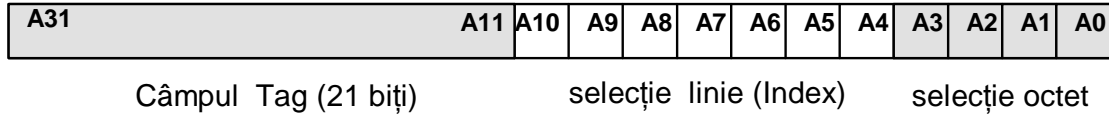


Fig. 3 Adresarea memoriei cache

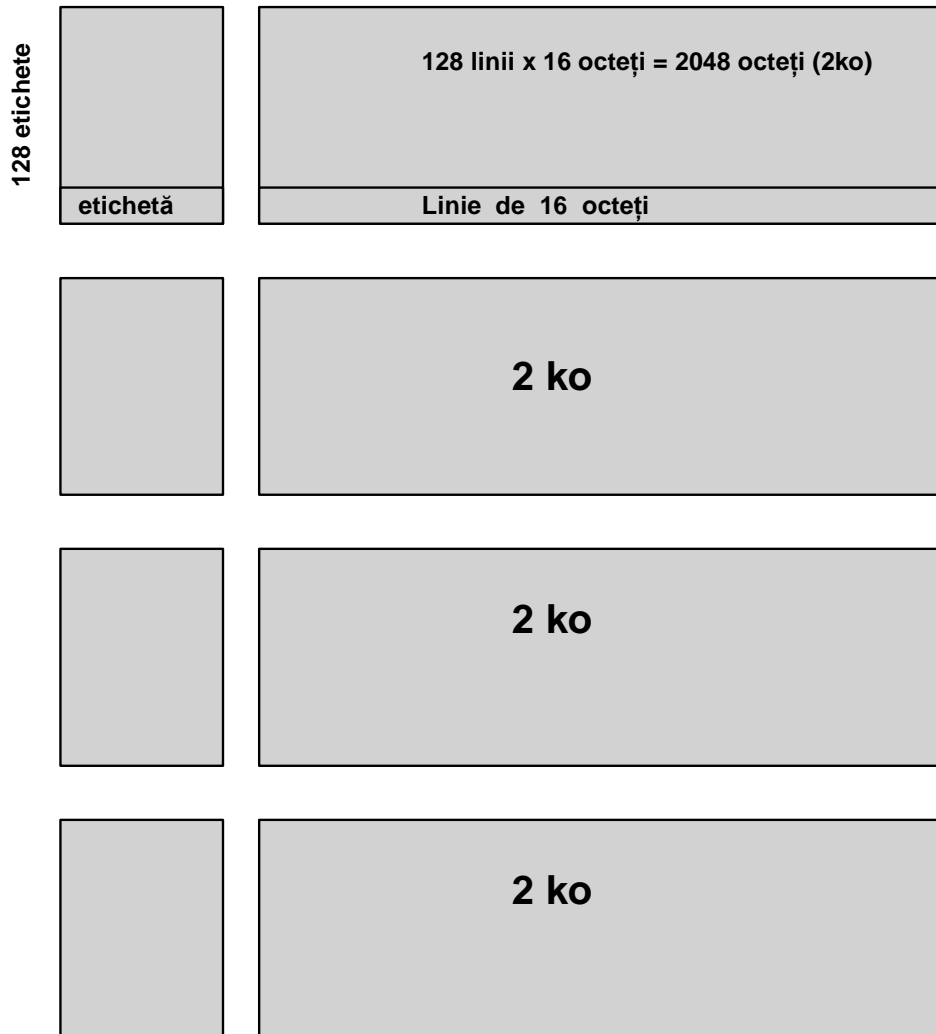


Fig. 4 Organizarea fizică a memoriei cache

Validarea unui transfer către sau de la memoria cache se face pentru o linie întregă de 16 octeți.

Modul de scriere în cache este numai "Write through" astfel că la orice scriere în cache se face și scriere în memoria externă. Un ciclu de magistrală specific memoriei externe (de tip *miss*) nu afectează memoria cache.

19.5 Indicatorii de condiții

Intel 486 are un singur indicator în plus față de 386:

AC (*Alignment Check*) - controlul alinierii datelor.

Testează alinierea datelor în memorie și declanșază semnal de eroare dacă apare eroare de aliniere; de exemplu în cazul adresării unui cuvânt de 2 octeți pornind de la o adresă impară.

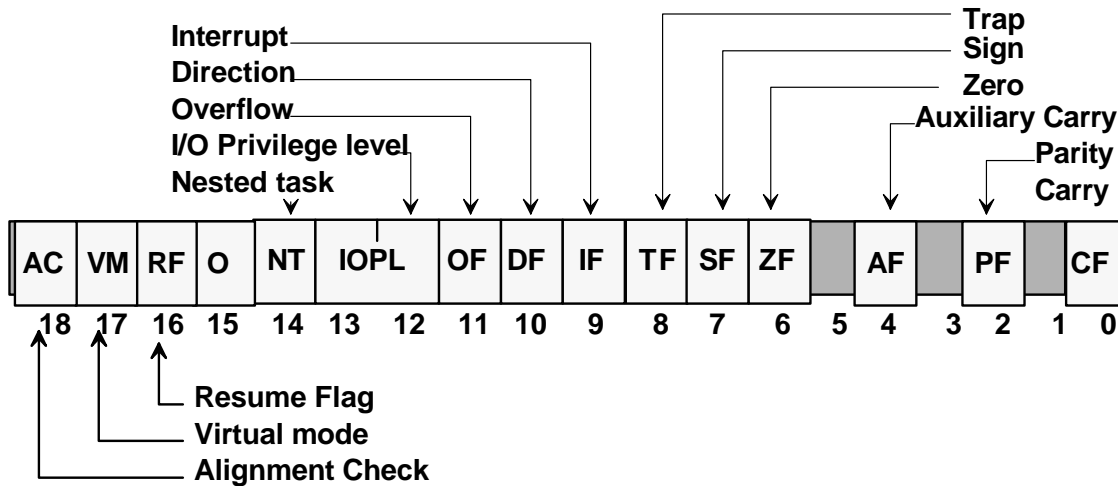


Fig. 7 Registrul indicatorilor de condiții la Intel 486

20 Microprocesorul PENTIUM (direcția RISC)

Arhitectura microprocesorului Pentium este marcată de tehnicile aplicate la sistemele de tip RISC (Reduced Instruction Set Computer). De exemplu, are instrucțiuni simple, cablate, arhitectură superscalară, se aplică conceptul extragerii predictive, apelează memorii "caches" specializate pentru instrucțiuni și date etc. El rămâne însă un microprocesor cu arhitectură de tip CISC (Complex Instruction Set Computer) în privința caracteristicilor de bază.

La origine, trebuia să se numească *i-586*, dar din considerente de protecție comercială (la data lansării pe piață apăruse deja un microprocesor cu numele 586) s-a preferat denumirea Pentium ("al cincilea").

Este realizat în tehnologie mixtă, CMOS și bipolară. CMOS pentru consum redus și densitate mare de integrare și bipolară pentru putere și rapiditate.

Are peste 3,1 milioane de tranzistoare în structură (1 milion la 486) și trasee interne de 0,35 microni (1 micron la 486). A apărut deja în câteva variante dintre care se remarcă cele cu alimentare tradițională de 5V și cele cu alimentare de 3,3 V (variantele diferă esențial în ceea ce privește numărul de pini (273 - 296) și dimensiunile carcasei).

Variantele sunt marcate pe *chip* prin litere: STD, VR, VRE/MD, MD, MMX etc.

Consumul, în funcție de variantă, poate atinge 15 W, ceea ce a determinat montarea unui microventilator fixat direct pe radiatorul aflat în contact cu capsula de ceramică (temperatura în centrul capsulei nu trebuie să depășească 85°C).

Frecvența de tact internă de 60 MHz la prima variantă (din 1995) a crescut permanent la variantele ulterioare: 66, 100, 120, 133, 180, 200, 280 MHz, 500 MHz (la Pentium II - 1999), 800 MHz (la Pentium III).

Principalele aspecte privind creșterea performanțelor, în comparație cu *i-486* :

- ♦ arhitectura superscalară permite creșterea vitezei de execuție la mai mult de o instrucțiune pe o perioadă de tact;
- ♦ memorii interne "caches" distincte pentru instrucțiuni (8 kB) și date (8 kB);
- ♦ utilizarea extragerii predictive dinamice a instrucțiunilor și datelor;

- ♦ o unitate de calcul în virgulă mobilă ce poate lucra în tehnica "pipe line" ;
- ♦ magistrala de date externă de 64 biți permite o viteză de transfer de 528 MB/s la o frecvență de tact de 66 MHz;
- ♦ magistrala de date internă de 256 biți (128 biți la 486);
- ♦ suport hard și soft pentru sisteme multiprocesor;
- ♦ extinderea dimensiunii paginii de memorie; Pentium permite opțiunea de 4 MB / pagină față de 4 kB / pagină (tradițional);
- ♦ detecția erorilor și redondanță funcțională;
- ♦ capacitate de extensie prin tehnica OverDrive, prin multiplicarea frecvenței de tact extern chiar cu coeficienți fracționari de multiplicare.

20.1 Arhitectura internă

Schema din figura 1 pune în evidență arhitectura superscalară a Pentium - ului, cu mai multe unități de execuție ce pot funcționa simultan:

- ♦ unitatea de instrucțiuni (unitate de preextragere, memorie cache de instrucțiuni, unitate de extragere predictivă);
- ♦ două unități aritmetico-logice pentru operanzi întregi (UAL);
- ♦ unitatea de calcul în virgulă mobilă (FPU), care dispune de registre, blocuri specifice de adunare, înmulțire, împărțire, toate pentru operanzi de 80 de biți.

Aceste unități pot prelucra fiecare câte o instrucțiune pe perioadă de tact și ca urmare, în mod teoretic, pot fi prelucrate două instrucțiuni pentru operanzi întregi și o instrucțiune pentru FPU într-o singură perioadă de tact.

Se utilizează instrucțiuni simple cablate (de tip MOV, STORE) sub formă de "*firmware*", ceea ce reprezintă o caracteristică RISC. În cazul instrucțiunilor complexe se conservă tehnica microcodării (CISC).

Se utilizează două unități distincte de memorie cache, pentru date și pentru instrucțiuni. Memoria cache de date prezintă două particularități:

1. Scrierea se face cu comandă specifică;
2. Se aplică protocolul MESI (" *Modified, Exclusive, Shared, Invalid* ").

Acest protocol garantează că datele transferate sunt identice cu cele ale sursei de date (se face verificarea scrierii corecte prin comparare cu datele din altă zonă cache, eventual din memorie). Memoria cache ocupă o suprafață importantă pe pastila de siliciu, ceea ce reprezintă principala

cauză de limitare a capacității sale (16 kB).

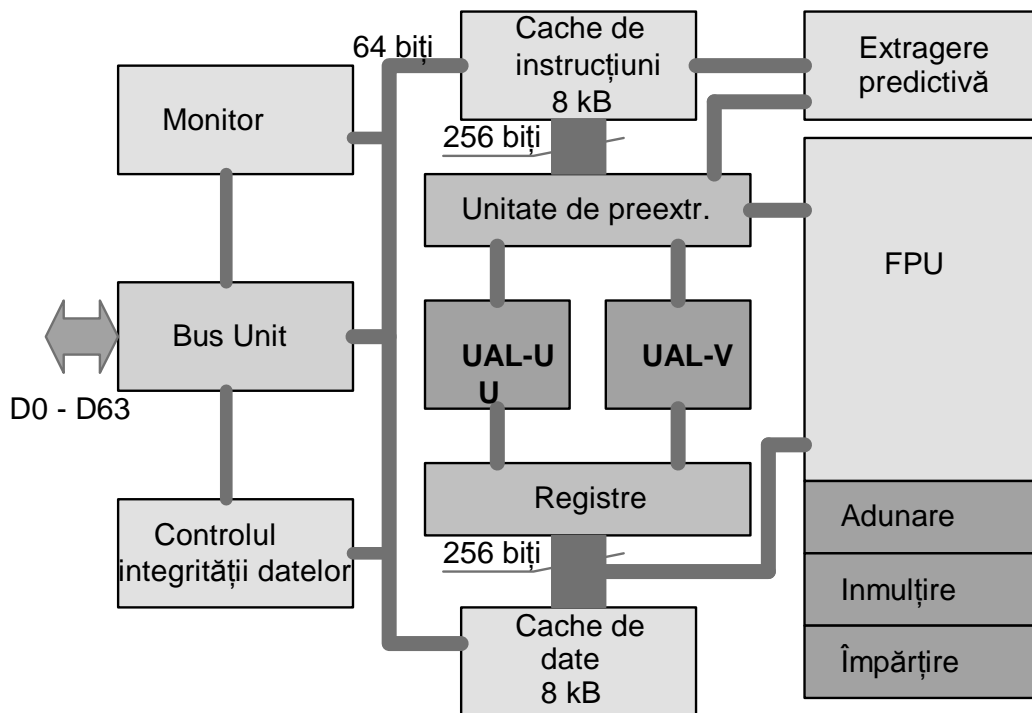


Fig. 1 Arhitectura internă simplificată a microprocesorului Pentium

Unitatea pentru controlul magistralelor (*Bus Unit*) realizează gestiunea magistralelor de date (64 biți), de adrese (32 biți) și de control.

Față de microprocesoarele precedente ale seriei *Intel*, Pentium este dotat cu două blocuri inedite: blocul *Monitor*, destinat analizării performanțelor și blocul pentru *Controlul integrității datelor*.

Aceste funcții erau rezervate până la Pentium doar calculatoarelor de mare putere.

Detecția erorilor interne se realizează prin biți de paritate plasați pe octet atât pentru octeții de cod cât și pentru datele și instrucțiunile plasate în memoriile caches și în registrele temporare.

Controlul redundanței funcționale se realizează în sisteme multiprocesor, în care, un procesor "*master*" lucrează în paralel cu unul "*slave*"; un dezacord între cele două produce o întrerupere de un anumit tip (alertă).

Analiza performanțelor permite programatorilor și proiectanților de sisteme să optimizeze produsele lor prin identificarea procedurilor (din programe) și blocurilor (din sistem), neeficiente.

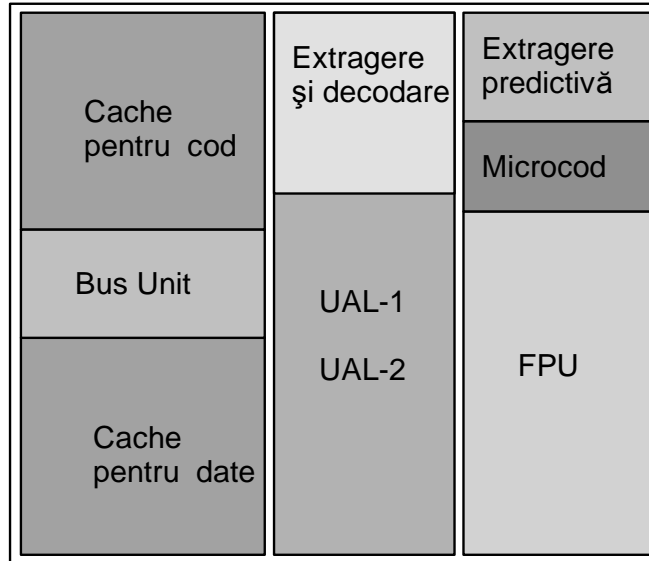


Fig. 2. Pentium - Structura internă la scară, mărită

În figura 2 este prezentată structura reală a microprocesorului Pentium, mărită de circa 50 de ori.

Se poate observa suprafața importantă ocupată de memoriile caches și raportul suprafețelor diferitelor blocuri componente.

20.2 Adresarea memoriei

Microprocesorul Pentium poate adresa (cu cei 32 biți de adresă) o memorie fizică de maxim 4 GB. Memoria poate avea locații de un octet, doi, patru sau opt octeți (64 biți). Dacă într-un sistem coexistă toate tipurile de locații enumerate mai sus, adresarea memoriei se poate realiza ca în schema din figura 3.

Blocul de memorie cu locații de 64 biți, la capacitate maximă, poate fi format din 8 secțiuni de 0.5 GB, fiecare secțiune fiind activată de o linie BE (Byte Enable). Rezultă o capacitate de

$$0.5G \times 8 \times 64 = 4G \times 8 \text{ Byte} = 32 \text{ GB}.$$

În acest caz nu mai pot fi adresate blocuri de 32, 16 sau 8 biți pe locație.

Celelalte blocuri pot avea maxim 16 GB, 8GB respectiv 4GB în cazul locațiilor de 8 biți, dacă se utilizează un singur tip de locație.

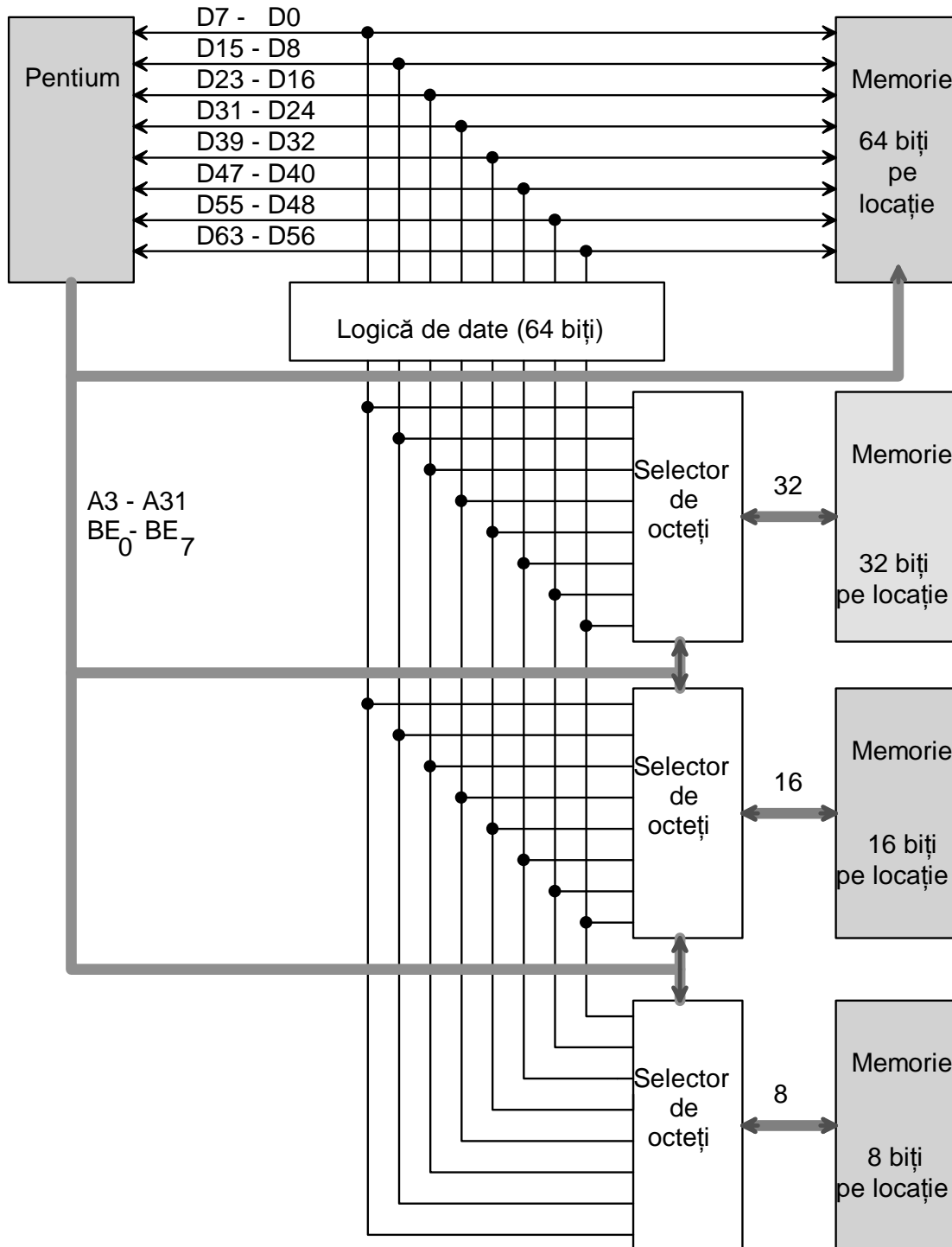


Fig. 3 Adresarea memoriei în funcție de dimensiunea locațiilor

Ramificarea predictivă. Este o tehnică de creștere a vitezei de execuție a buclor repetitive prin executare simultană a mai multor instrucțiuni ce pot fi cuplate; se reduce de circa 3 ori timpul de execuție față de I - 486, la aceeași frecvență de tact.

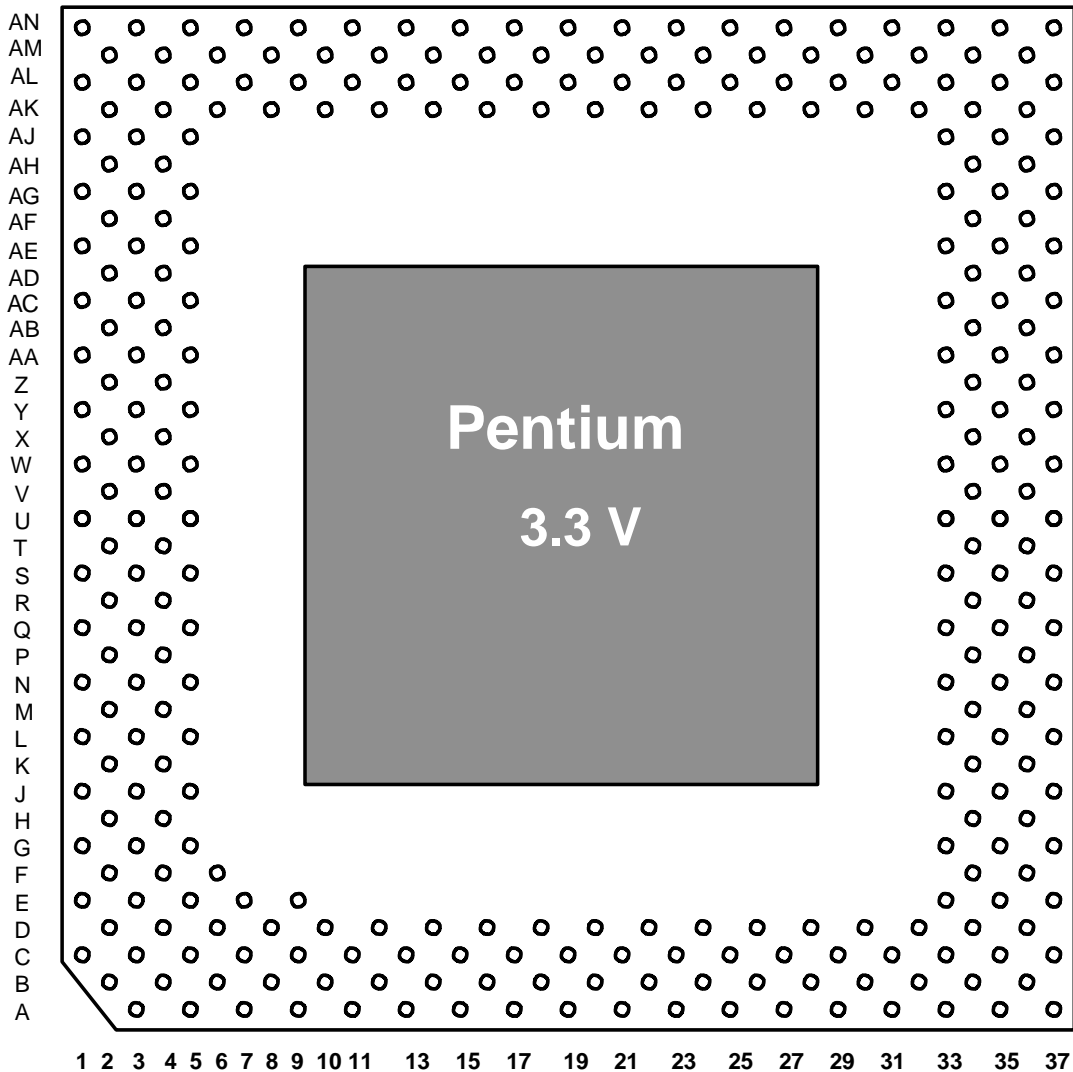


Fig. 4. Distribuția pinilor la Pentium - varianta 3,3 V
(296 pini, dimensiuni 55 mm x 55 mm)

Pe baza acestei tehnici, unitatea centrală anticipează instrucțiunea din program la care se va face următoarea ramificație, ceea ce este posibil prin identificarea instrucțiunilor de salt condiționat, de salt necondiționat, de apel și revenire la/din proceduri. Se utilizează registre speciale de ramificație, BTB (*Branch Target Buffer*).

Intel ilustrează avantajele ramificării predictive printr-un exemplu de fragment de program care determină numerele prime:

```
FOR (K=I+PRIME ; K<=SIZE ; K +=PRIME)
  FLAGS(K) = FALSE ;
```

Un compilator obișnuit distribuie variabilele în registre astfel:

- PRIME este adresat cu **ecx** ;
- K este încărcat în **edx** ;
- a1 conține valoarea FALSE ;

În limbaj de asamblare, secvența de program devine:

```
inner_loop:
    mov byte ptr flags[edx], a1
    add edx, ecx
    comp edx, SIZE
    jle inner_loop
```

Fiecare iterație din buclă necesită 6 perioade de tact cu un i486; cu Pentium, instrucțiunea MOV este asociată cu ADD și COMP la instrucțiunea JLE. Utilizând ramificația predictivă, fiecare tur de buclă nu necesită mai mult de două perioade de tact.

20.3 Controler de întreruperi integrat

Versiunea PENTIUM de 3.3V, dispune de un controler de întreruperi integrat APIC (*Advanced Programmable Interrupt Controller*).

- Într-un sistem monoprosesor, el poate înlocui total un controler extern, tipic *Intel 8259* - controler programabil pentru 8 niveluri de întreruperi externe;

- Într-un sistem biprosesor, trebuie să fie prezent un controler de întreruperi extern conform schemei de principiu din figura 5.

20.4 Pentium (varianta 5V) - semnificația semnalelor

Din punct de vedere funcțional, se pot pune în evidență 25 de grupe de semnale:

1. Semnal de tact generat extern CLK , \overline{STPCLK} (*Stop Clock*), cerere de oprire a semnalului de tact pentru reducerea consumului;
2. Inițializare $RESET$, $INIT$;
3. Magistrala de adrese $A_3; -A_{31}$, $\overline{BE}_0; -\overline{BE}_7$;
4. Mască de adrese $\overline{A}_{20}M$;
5. Bit de paritate pentru adrese AP și controlul parității, \overline{APCHK} ;
6. Magistrala de date $D_0 - D_{63}$;
7. Paritate/octet $DP_0 - DP_7$, \overline{PCHK} (*Parity Check*), \overline{PEN} (*Parity Enable*);

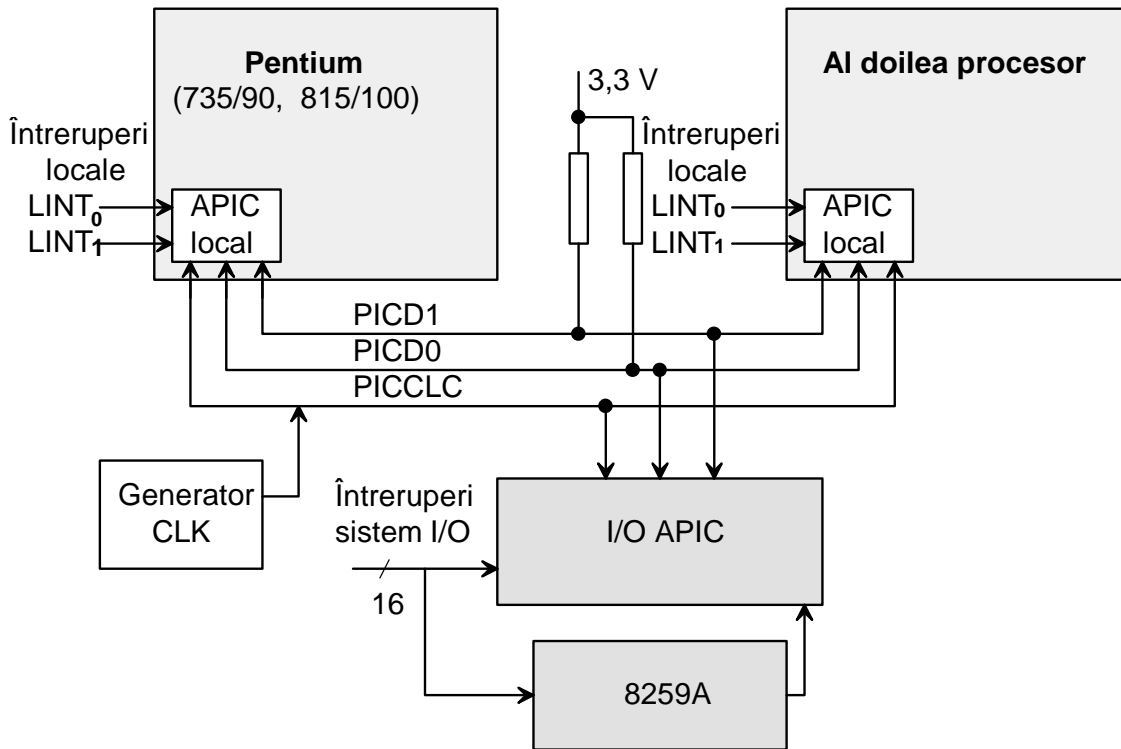


Fig.5 Controlul întreruperilor interne și externe într-un sistem biprocesor cu Pentium 3,3V (master).

8. Erori interne de paritate \overline{IERR} (Internal Error);
9. Erori de sistem $BUSCHK$ (Bus Check);
10. Definirea ciclului mașină \overline{MIO} , $\overline{D/C}$, $\overline{W/R}$, \overline{CACHE} , \overline{SCYC} , \overline{LOCK}
11. Controlul magistrelor \overline{ADS} , \overline{BRDY} , \overline{NA} (Next Adress);
12. Posibilitatea paginării memoriei cache PCD (Page Cache Disable), PWT (Page Write Through);
13. Comenzi pentru cache \overline{KEN} (Cache Enable), $\overline{WB/WT}$ (Write Back /Write Through);
14. Supravegherea memoriei cache \overline{AHOLD} , \overline{EADS} , \overline{HIT} , \overline{HITM} , \overline{INV} ;
15. Golirea memoriei cache \overline{FLUSH} ;
16. Ordin de citire \overline{EWBE} (External Write Buffer Empty);
17. Arbitrajul magistrelor \overline{BOFF} (Back OFF), \overline{BREQ} (Bus Request - Pentium cere acces la magistrale), \overline{HOLD} , \overline{HLDA} ;
18. Cereri de întrerupere externe \overline{INTR} , \overline{NMI} ;
19. Raport de eroare de la FPU: \overline{FERR} , \overline{IGNNE} (Ignor Numeric Error);
20. Modul de gestiune a sistemului \overline{SMI} (Sistem Management Interrupt), \overline{SMIACT} (Sistem Management Active);

21. Test de redundanță funcțională \overline{FRMC} , (*Functional Redundancy Checking Master /Checker* - testează dacă Pentium e în modul master), \overline{IERR} (*Internal Error* - eroare internă de paritate sau de redundanță) ;

22. Test pentru porturi TCK (*Testability Clock* - funcție a orologiului secundar), TMS (*Test Mode Select*), TDI (*Test Data Input*), TDO (*Test Data Output*), \overline{TRST} (*Test Reset* - reinițializare a controlorului de test) ;

23. Puncte de oprire, supraveghere performanțe:

PM_0/BP_0 , PM_1/BP_1 , BP_2 , BP_3 (PM - *Performance Monitor*, BP - *Break Point*) ;

24. Traseul execuției BT_0 , BT_1 , BT_2 , BT_3 , (*Branch Trace*), IU , (*Instruction Unit U*), IV (*Instruction Unit V*), IBT (*Instruction Branch Taken* - salt acceptat);

25. Modul test R/\overline{S} (*Restart/Stop*), $PRDY$ (*Probe Ready*).

Pentru alimentare se utilizează 37 de pini pentru $V_{cc}=+5V$ și 49 de pini pentru $V_{ss}=0 V$ (masă), pentru asigurarea curentului absorbit de cca. 3 A.

21 Tehnici de accelerare a execuției:

(pipeline, paralelism și anticipare)

Aceste tehnici au apărut odată cu microprocesoarele avansate atât la cele de tip CISC cât și RISC, din necesitatea de creștere a vitezei de execuție altfel decât prin mărirea frecvenței de tact.

Prin introducerea unui fișier de așteptare pentru instrucțiuni, procesorul extrage instrucțiunea următoare din fișier, fiind astfel eliminat timpul suplimentar de așteptare pentru citirea memoriei externe; câștigul de timp este apreciabil.

*Conform concepției **pipeline** o instrucțiune este executată prin trecere succesivă de la un post de prelucrare la altul, ca "pe bandă rulantă".*

Paralelismul presupune creșterea numărului de unități de execuție ale procesorului astfel că prin funcționarea lor simultană se obține o creștere importantă a vitezei de execuție.

Alte concepte sunt mai subtile și se aplică în funcție de structura programului; în particular este vorba de predicția salturilor în buclele repetitive și alinierea datelor.

21.1 Fișier de așteptare și preextragere

Unitatea de instrucțiuni a procesorului citește instrucțiunile din memoria externă, din segmentul de cod, înainte ca acestea să fie necesare în execuție. Acestea se stochează, în ordinea în care au fost extrase, într-un fișier format din câteva registre care se află în interiorul procesorului; pot fi stocate două până la șase instrucțiuni.

Fișierul de așteptare este completat automat de unitatea de instrucțiuni pe măsură ce procesorul "consumă" instrucțiunile. Încărcarea fișierului de așteptare se face din memoria externă, când magistralele sunt libere, astfel că operația nu interferă cu celelalte operații ce se efectuează cu magistralele.

21.2 Tehnica pipeline și structura superscalară.

Un important criteriu de performanță pentru microprocesoare este viteza de execuție, exprimată de numărul de operații / secundă. Viteza depinde în mod direct de frecvența de tact, astfel că la o frecvență de 200 MHz se pot efectua 200 000 000 operații / secundă, dacă se efectuează o operație / perioadă de tact. Frecvența de lucru maximă este însă limitată de tehnologia de fabricație, în special de capacitățile parazite dintre traseele conductoare ale circuitului integrat, dar și de capacitățile interne ale dispozitivelor semiconductoare (diode, tranzistoare).

Frecvența de tact fiind limitată, pentru creșterea vitezei la aceeași frecvență singura cale este creșterea numărului de operații / perioadă de tact, care se obține prin funcționarea simultană a mai multor unități de execuție (dacă funcționează simultan 3 unități, se efectuează 3 operații / perioadă de tact, deci viteza crește de 3 ori, la aceeași frecvență). O arhitectură cu mai multe unități ce funcționează în paralel este **superscalară**.

Termenul *superscalar* desemnează tehnica prin care un grup de instrucțiuni succesive sunt distribuite către mai multe unități de execuție, într-un singur ciclu mașină (modul *scalar* presupune existența unei singure unități de execuție, astfel că o instrucțiune trebuie să aștepte ca precedenta să fie executată pentru a intra în unitatea de execuție). În figura 6 se pot vedea cele 4 unități de execuție cu funcționare paralelă, la Power PC 603, microprocesor cu arhitectură RISC.

Există 4 unități de execuție independente:

- Unitatea aritmetică și logică pentru calcule cu operanzi întregi, UAL;
- Unitatea de calcul în virgulă mobilă, FPU (coprocesor aritmetic);
- Unitatea de gestiune a memoriei, specializată în operații de tip Load și Store;
- Unitatea de gestiune sistem, specializată în ramificații și salturi.

Viteza de execuție crește doar teoretic de 4 ori; practic dacă într-un șir de instrucțiuni, nici una nu utilizează FPU (de exemplu), această unitate "șomează". În orice caz, viteza crește de peste două ori.

În figura 7 este pusă în evidență funcționarea paralelă a celor 4 unități în situația favorabilă, în care toate unitățile sunt active. Se observă că în anumite perioade de tact se efectuează 4 operații simultan iar în altele numai una, două sau trei operații.

Execuția completă a unei instrucțiuni se face în 4-6 etape care se succed într-o anumită ordine: extragerea din memorie, decodarea,

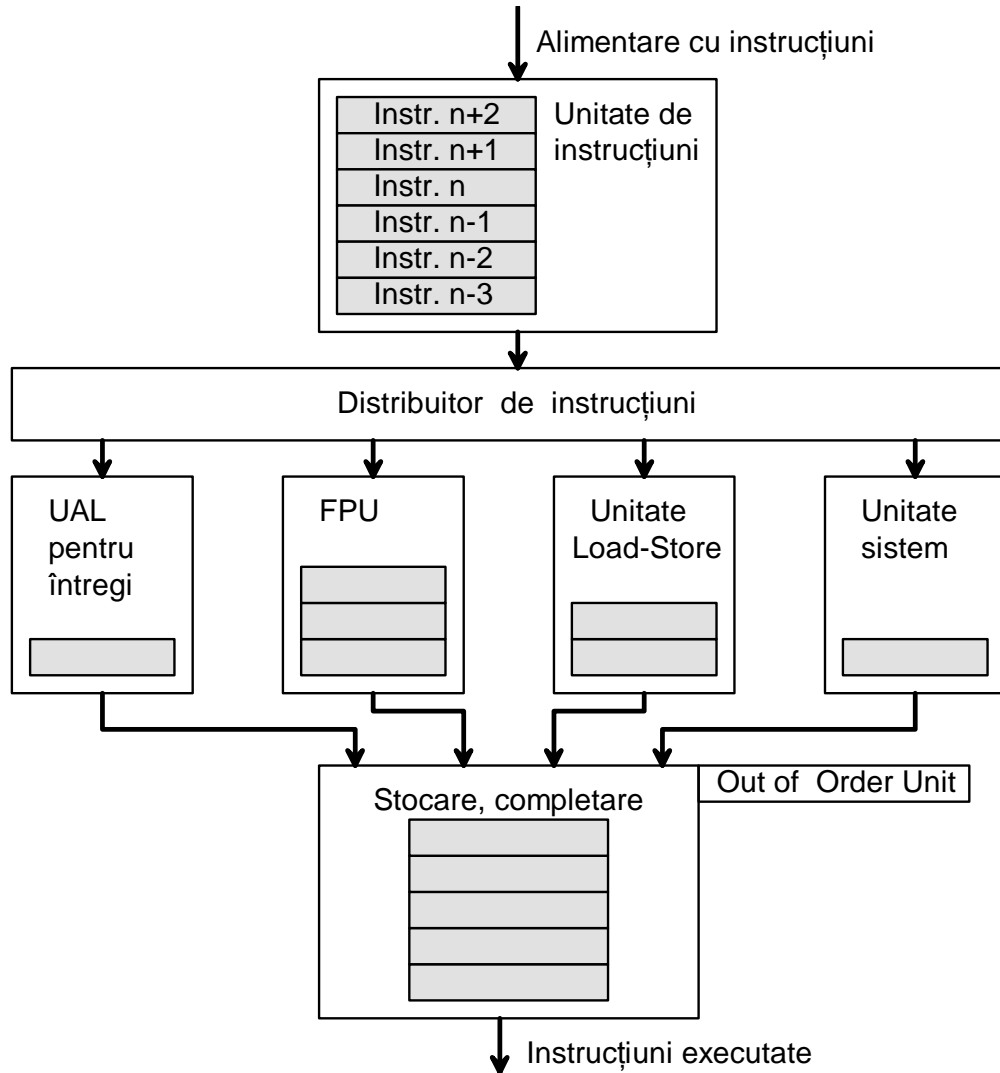


Fig.6 Distribuirea instrucțiunilor la procesorul RISC Power PC 603

extragerea operanzilor, efectuarea operației de bază, stocarea rezultatelor. Aplicând un principiu de succes din industrie, principiul "pe bandă rulantă", proiectanții de microprocesoare au adaptat structura internă astfel încât instrucțiunile să fie transferate de la o unitate de prelucrare la alta, fiecare unitate realizând o etapă specifică de execuție. Aceasta este tehnica **pipeline**, care a sporit considerabil viteza de execuție la aceeași frecvență de tact.

Execuția unei instrucțiuni se descompune în operații elementare efectuate în serie (fig.7). Când o operație a fost terminată și instrucțiunea trece la operația următoare, aceeași operație se aplică instrucțiunii următoare; acest mod de lucru este posibil datorită existenței unor subunități independente specializate în operații elementare.

Dacă presupunem că se pot efectua 4 operații elementare (încărcare, decodare, execuție, scrierea rezultatului, atunci 4 instrucțiuni diferite se

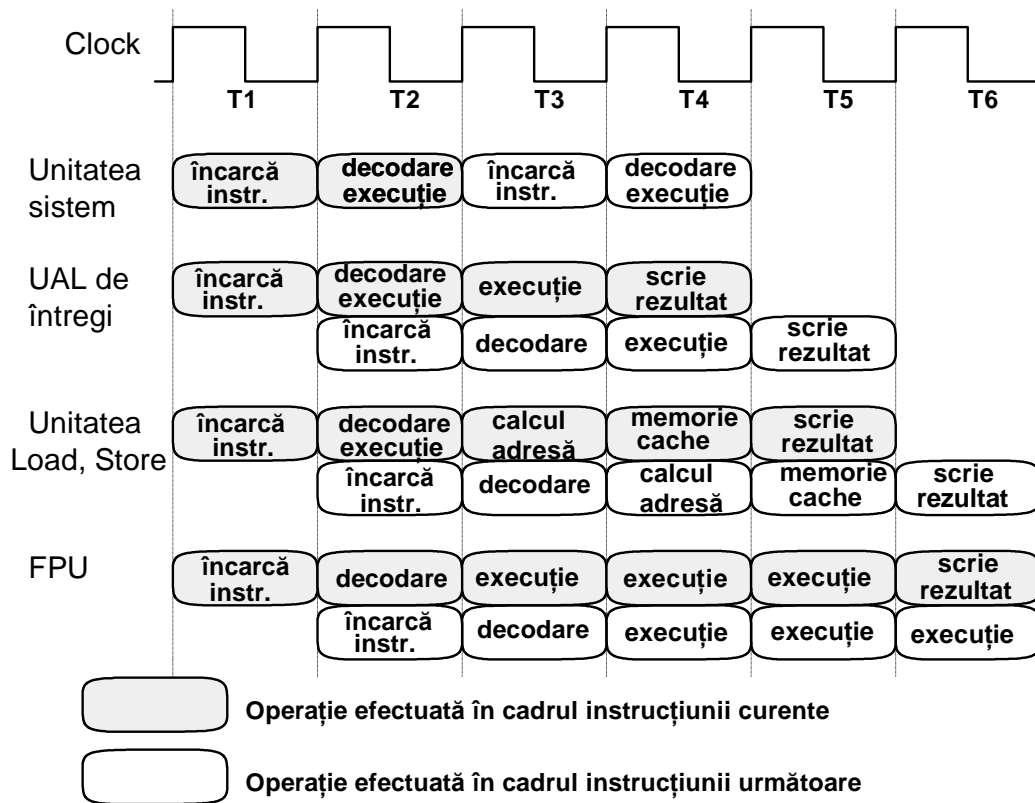


Fig.7 Funcționarea paralelă a unităților la Power PC 603. Fiecare unitate lucrează în modul "pipeline"

pot afla în diferite stadii într-o unitate de execuție (fig.8) și viteza globală crește teoretic de 4 ori.

În figura 8 este prezentat principiul *pipeline* pentru o unitate de execuție care are 4 subunități independente; instrucțiunea 1 se află în faza finală, instrucțiunea 2 ste în curs de execuție, instrucțiunea 3 este în etapa de decodare în timp ce instrucțiunea 4 este în curs de încărcare în unitate.

21.3 Reordonarea instrucțiunilor

Funcționarea superscalară prezintă pe lângă avantaje incontestabile și un dezavantaj major: posibilitatea schimbării ordinii logice a instrucțiunilor.

Acestea sunt executate de unități de execuție diferite în intervale de timp (număr de perioade de tact) diferite; de exemplu o instrucțiune ce necesită calcule în virgulă mobilă conține un număr de operații elementare mai mare decât o instrucțiune de calcul cu întregi.

Apare posibilitatea ca o instrucțiune să fie terminată înaintea instrucțiunii următoare din program și este necesară reordonarea instrucțiunilor după execuție.

Unitate de execuție				
subunitate	1	2	3	4
Instrucțiune 1				rezultat
Instrucțiune 2			execuție	
Instrucțiune 3		decodare		
Instrucțiune 4	încărcare			

Fig.8. Principiul pipeline: fiecare subunitate efectuează o operație specifică asupra unei instrucțiuni

Unitatea de reordonare este numită *Out of Order Unit* (fig.6, unitatea de stocare, completare) și conține un fișier de așteptare și o logică de ordonare prin care se implementează mecanismul de punere a efectelor instrucțiunilor în ordinea logică ce rezultă din program.

21.4 Anticiparea ramificațiilor condiționate

Este încă un mod de accelerare a execuției pentru programele care conțin numeroase bucle repetitive (este cazul programelor de mare complexitate). O buclă repetitivă conține cel puțin un salt condiționat de reluare a execuției pe baza testării unei condiții logice.

Statistic s-a constatat că în medie, o instrucțiune din 7 conține o ramificație de tip salt condiționat, ceea ce justifică implementarea unei tehnici de anticipare și optimizare a execuției buclelor. Principiul constă în anticiparea adresei de salt la care se transferă controlul după executarea unui tur complet de buclă și este cunoscut sub denumirile de *salt predictiv*, *salt anticipat* sau *salt speculativ*. Printre metodele de predicție a ramificațiilor, cele mai importante sunt:

- salturi predictive statice;
- salturi predictive dinamice.

Cele mai frecvente sunt cele **statice**; o buclă se repetă *de mii de ori* iar ieșirea din buclă se face *o singură dată*. Este deci rezonabil să se considere că la termiarea execuției unei bucle, aceasta va reluată, deci se va executa prima instrucțiune din buclă.

Să examinăm cazul procesorului Power PC 601 (clasa RISC); unitatea centrală de procesare (*Central Processing Unit*) anticipează ordinea în care vor fi executate instrucțiunile din fiecare secvență.

În general există două cazuri distincte: saltul necondiționat și saltul condiționat.

În primul caz, saltul este obligatoriu și trebuie executat: unitatea de ramificații din CPU indică unității de instrucțiuni să înceapă extragerea instrucțiunilor din memorie de la noua adresă, dată de instrucțiunea de salt, ceea ce se face fără penalizare în timp.

În cazul salturilor condiționate, saltul se realizează numai dacă este îndeplinită condiția specificată în instrucțiunea de salt. Sunt posibile două cazuri:

1. Condiția poate fi evaluată înainte de începerea extragerii instrucțiunilor, situație asemănătoare cu cea de la saltul necondiționat sau cu absența saltului.

2. Condiția nu poate fi evaluată; în acest caz se aplică tehnica ramificării anticipate. Se consideră condiția de salt îndeplinită și se încarcă instrucțiunile de la adresa de salt. Pentru programele scrise în limbaj de nivel înalt, compilatorul este cel care optimizează situațiile generate de ramificații.

Când procesorul anticipează corect salturile, nu se consumă timp suplimentar; dacă anticiparea este incorectă, se consumă timp suplimentar pentru reîncărcarea fișierului de așteptare cu instrucțiuni de la noua adresă.

În cazul metodei predictive **dinamice**, se utilizează o tabelă de salturi reușite cu 4 stări care se actualizează permanent. Tabela - BHT (*Branch History Table*) se află în subunitatea de ramificații; ea conține, de exemplu, 512 intrări cu 4 stări de predicție. Se utilizează doi biți / intrare, care furnizează 4 niveluri de predicție dinamică a ramificațiilor, ca în tabelul de mai jos.

Tabel 1. Cele patru niveluri de predicție dinamică a salturilor

Valoare	Nivel	Interpretare
OO	Strongly not - taken	Total ignorat
O1	Not - taken	Ignorat
O2	Taken	Utilizat
O3	Strongly taken	Frcvent utilizat

La întâlnirea unui salt executabil, valoarea corespunzătoare din tabelă este incrementată până la valoarea maximă 03. Când saltul nu se efectuează, valoarea se decrementează până la 0.

De exemplu, dacă valoarea curentă este 2, semnificând *salt utilizat*, iar noul salt este executat la aceeași adresă, valoarea crește la 3, semnificând *salt frecvent utilizat*. Dacă viitorul salt nu mai indică aceeași adresă, valoarea din tabelă redevine 2.

Primul procesor Power PC care aplică acest concept este 604.

21.5 Suprimarea salturilor inutile prin *Branch Folding*

Tehnica *branch folding* aplicată la seria Power PC, permite suprimarea din fișierul de așteptare a unei instrucțiuni de salt (condiționat, cu condiția evaluabilă sau necondiționat), ceea ce are ca efect creșterea vitezei de execuție.

Unitatea care tratează salturile, BPU (*Branch Processing Unit*), validează saltul predictiv dar calculează și adresa de continuare în cazul neefectuării saltului.

De fapt BPU emite două adrese:

- adresa de salt pentru ramificație;
- adresa de recuperare dacă predicția de salt este eronată.

Sunt luate astfel în calcul ambele cazuri și execuția se realizează fără consum suplimentar de timp.

Se utilizează 3 registre specializate în realizarea ramificațiilor în program:

1. Un registru pentru condiție CR (*Condition Register*) de 32 de biți, care conține rezultatul operațiilor de evaluare a condiției de salt.

2. Un registru de legătură LR (*Link Register*), de 64 de biți, care conține adresa de retur necesară la reluarea execuției buclelor.

3. Un registru de numărare CTR (*Count Register*), de 64 de biți, care servește drept contor de repetare a buclelor sau poate conține adresa de revenire din buclă.

21.6 Execuție speculativă

Câteva procesoare din familia Power PC aplică pentru salturi o tehnică mai subtilă, numită "execuție speculativă".

Obiectivul constă în anticiparea unor operații privind evaluarea condițiilor de salt care nu pot fi evaluate; rezultatul nu va fi definitiv decât când evaluarea va putea fi realizată.

Practic, execuția speculativă este un fel de pariu privind operațiile viitoare în care programul are toate șansele de partea sa și așa ca la Casino, iese totdeauna câștigător.

Execuția speculativă completează mecanismul de anticipare a salturilor care depind de o condiție ce poate fi evaluată. La Power PC 604 pot fi tratate până la 4 instrucțiuni de salt condiționat cu condiție evaluabilă și până la două instrucțiuni de salt cu condiție ce nu poate fi evaluată. Mecanismul este destul de complex.

Rezultatele salturilor nerezolvate dar executate (speculativ) sunt conservate în mod provizoriu în *registre tampon suplimentare*, care conțin o parte din stările viitoare ale procesorului și vor fi definitiv adoptate după confirmarea lor prin evaluarea condițiilor de salt. În cazul predicției eronate, registrele tampon vor fi utilizate pentru restabilirea situației anterioare executării saltului.

Aceste registre, la Power PC 604, se numesc *registre fantomă*.

Metoda execuției speculative are așadar următoarele consecințe:

- dacă predicția (speculativă) a fost corectă se câștigă timp;
- dacă predicția a fost incorectă, registrele fantomă sunt restaurate și execuția se reia din punctul de ramificație fără penalizare de timp.

La Power PC 604, unitatea specializată în *încărcare - stocare*, LSU (*Load Store Unit*), rezolvă toate dependențele determinate de stocarea în memorie a datelor la aceeași adresă de la care au fost citite (încărcate). În caz de dependență, LSU va corecta datele eronate.

21.7 Alinierea datelor

Pentru creșterea eficienței în operațiile de transfer de date cu memoria, datele sunt alinate în memorie la adrese ce sunt multipli ale lungimii lor. Astfel, când procesorul adresează memoria, citește imediat data al cărui prim bit se află în prima poziție a locației adresate.

Tabelul de mai jos arată modul de adresare aplicat la Power PC pornind de la operanzi pe un octet.

Tabel 2. Principiul alinierii datelor

Operand	Lungime operand	Adresa (ultimii 4 biti)
Octet	1 octet	x x x x
Cuvânt	2 octeti	x x x 0

Dublu cuvânt	4 octeti	x x 0 0
Quadruplu cuvânt	8 octeti	x 0 0 0
Hexa cuvânt	16 octeti	0 0 0 0

Prin convenție, o locație de memorie standard este de un octet; dacă data memorată are mai mulți octeți, numărul acestora este deductibil prin examinarea ultimilor 4 biți de adresă.

Concluzii:

- operanzii pe un octet pot fi stocați la orice adresă;
- operanzii pe 2 octeți sunt stocați la adrese pare (care se termină cu 0);
- operanzii pe 4 octeți sunt stocați la adrese multiplu de 4 (se termină cu 00)
- operanzii pe 8 octeți sunt stocați la adrese multiplu de 8 (se termină cu 000);
- operanzii pe 16 octeți sunt stocați la adrese multiplu de 16 (se termină cu 0000);

22 Microprocesorul PowerPC 601

(arhitectură RISC)

Apple, IBM și Motorola au inițiat în 1994 un proiect comun pentru realizarea unei familii de procesoare cu arhitectură RISC (Reduced Instruction Set Computer), de mare eficiență. Primul procesor al familiei a fost PowerPC 601 care a fost utilizat ca unitate centrală în primul microcalculator Power - Macintosh Apple.

Microprocesoarele familiei PowerPC s-au dovedit a fi puternice, economice și au fost utilizate în microcalculatoarele profesionale Apple-Macintosh.

Principalele caracteristici

Este un microprocesor de 64 de biți, cu arhitectură RISC, fiind capabil să execute până la trei instrucțiuni pe perioadă de tact.

- A fost realizat în tehnologie CMOS cu trasee conductoare de 0,6 microni și 4 straturi de metalizare la primele serii, apoi cu trasee de 0,5 microni.
- Suprafața pastilei de siliciu este de $10,95 \times 10,95 \text{ mm}^2$, adică cca. $1,2 \text{ cm}^2$ și conține 2 800 000 tranzistoare.
- Tensiunea de alimentare este $3,6 \text{ V} \pm 5\%$, intrările și ieșirile sunt compatibile TTL și absoarbe o putere de 7 W la frecvența de tact de 66 MHz.
- Funcționează la o temperatură ambiantă în gama $0 - 40^\circ \text{C}$.
- Frecvența de tact: 50, 66, 80 și 100 MHz (pentru diferite variante).
- Carcasă ceramică patrată, plată, QFP (*Quadruple Flat Pack*) cu 304 pini.
- Setul de instrucțiuni conține peste 220 instrucțiuni; instrucțiuni ca *Branch*, *Link*, *Load Multiple*, *Load String*, *Store Multiple*, provin direct de la puternicele calculatoare IBM.

22.1 Arhitectura internă

Procesor superscalar, PPC 601 conține trei unități de execuție cu funcționare independentă, conectate la o unică magistrală internă de date:

1. Unitatea de calcul pentru întregi IU (*Integer Unit*), numită FXU de IBM pentru a evita confuzia cu unitatea de instrucțiuni.
2. Unitatea pentru procesarea ramificațiilor BPU (*Branch Processing Unit*).

3. Unitatea de calcul în virgulă mobilă FPU (*Floating Point Unit*).

Majoritatea instrucțiunilor pentru operanzi întregi sunt executate într-o singură perioadă de tact.

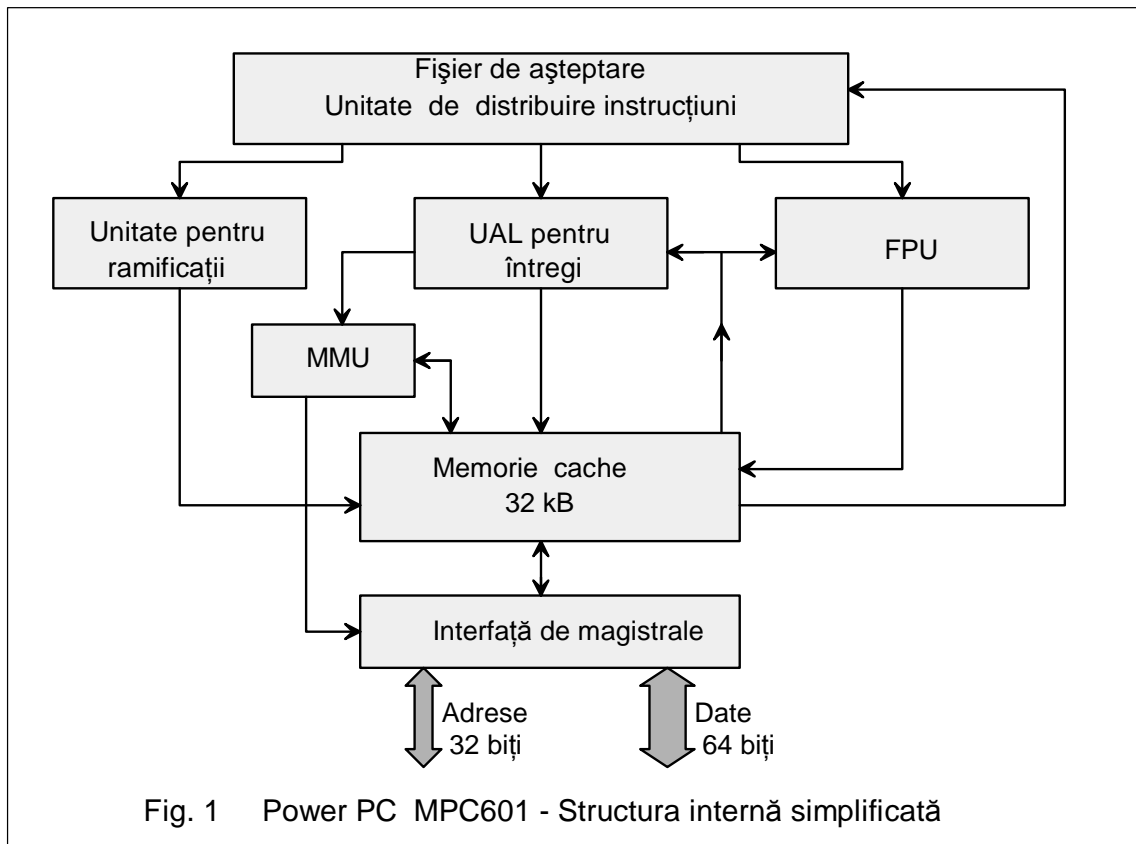
Structura internă mai conține:

- O memorie cache asociativă de înaltă performanță, de 32 kB;
- O unitate de gestiune a memoriei, MMU (*Memory Management Unit*), care conține un buffer de translatare *look-aside* asociativ, cu 256 de intrări.

Protocolul de interfațare permite mai multor procesoare "master" de a prelua controlul asupra magistralelor prin intermediul unui arbitru central extern.

PPC 601 mai dispune de :

- două moduri de adresare a memoriei;
- transferuri simple sau de tip salvă cu memoria;
- adresare cartografică sau nu pentru porturi (*I/O, In/Out*);
- două niveluri de privilegii: supervizor și utilizator.



22.2 Unitățile funcționale

Figura 1 prezintă în variantă simplificată structura internă a microprocesorului Power PC - MCP601. Sunt puse în evidență următoarele unități:

- **Fișierul de instrucțiuni și distribuitorul**

Unitatea dispune de o magistrală de 256 de biți pentru conectare cu memoria cache și poate distribui trei instrucțiuni către cele trei unități de execuție; fișierul de instrucțiuni poate stoca 8 instrucțiuni și este încărcat într-o singură perioadă de tact (dimensiunea magistralei permite transferul a 8 x 32 biți simultan) din zona de stocare a instrucțiunilor din cache.

▪ Unitatea de ramificații și extragere anticipată BPU

Este conectată la fișierul de instrucțiuni astfel încât primele patru instrucțiuni pot fi transferate în bufferul fantomă (*shadow buffer*), numit ITLB (*Instruction Translation Lookaside Buffer*). Unitatea permite anticiparea statică a salturilor prin utilizarea registrului de evaluare a condițiilor care autorizează saltul. Un sumator dedicat calculează adresa instrucțiunii următoare. BPU caută în ultimele 4 instrucțiuni din fișierul de așteptare instrucțiunea de salt și aplică tehnica predicției statice pentru ramificație.

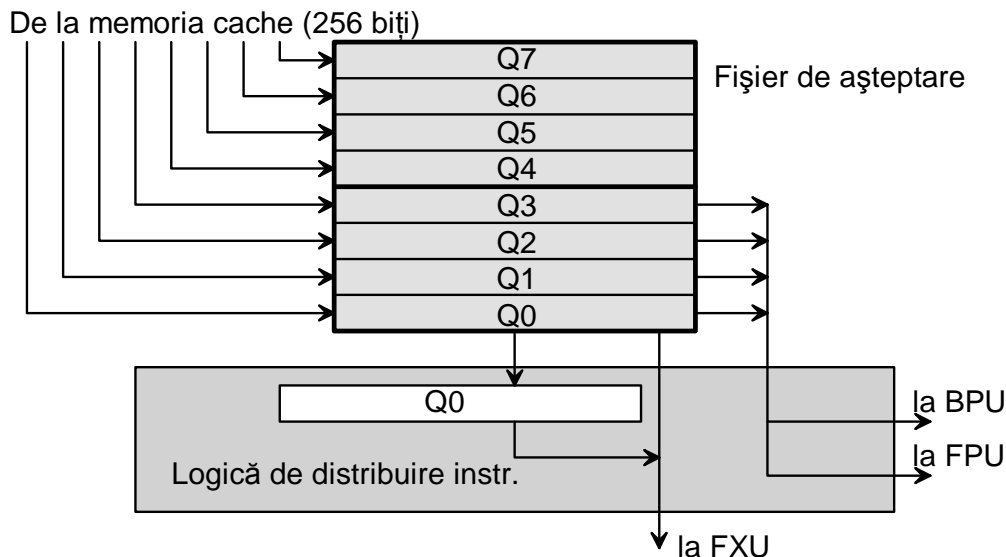


Fig. 2 Fișierul de instrucțiuni și distribuitorul

Instrucțiunile de salt sau de calcul în virgulă mobilă pot fi extrase din oricare din primele 4 poziții ale fișierului de instrucțiuni dar cele de calcul cu operanzi întregi, în mod obligatoriu sunt extrase numai din prima poziție (Q_0).

După execuție, instrucțiunile sunt repuse în ordinea din program.

22.3 Unitatea de calcul cu operanzi întregi - IU

Unitatea de calcul în virgulă fixă (după denumirea IBM) dispune de 32 de registre de uz general de 32 biți, GRP (*General Purpose Registers*) și execută cea mai mare parte din instrucțiuni într-o singură perioadă de

tact; toate instrucțiunile de înmulțire și împărțire sunt cablate (scheme combinaționale cu porți logice).

Unitatea dispune de următoarele elemente:

- O unitate clasică ALU (*Arithmetic and Logic Unit*);
- Un multiplicator și un divizor, realizate cu porți logice;
- Un registru special pentru excepții, XER (*Exception Register*);
- Registre de uz general.

22.4 Unitatea de calcul în virgulă mobilă - FPU

Conform normelor IEEE-754 lucrează în simplă precizie (64 de biți) sau în dublă precizie (80 de biți) și acceptă toate tipurile de date utilizate în limbajele de programare de nivel înalt. Sunt disponibile 32 de registre de 64 de biți. FPU conține:

- O rețea cablată de multiplicatoare - sumatoare pentru efectuarea înmulțirilor, împărțirilor, adunărilor și operațiilor combinate de înmulțire - adunare;
- Două niveluri de execuție, în funcție de complexitatea operațiilor;
- Un registru de stări și comenzi, FPSCR (*Floating Point Status and Control Register*), de 32 de biți;
- 32 de registre de 64 de biți, notate FPR0 . . . FPR31;
- Două fișiere de așteptare pentru instrucțiuni, unul pentru intrare și unul pentru ieșire, pentru evitarea blocajelor rezultate din nesincronizarea cu celelalte unități;

Instrucțiunile pentru FPU provin din locațiile Q_0, Q_1, Q_2, Q_3 ale fișierului de așteptare din Unitatea de instrucțiuni.

22.5 Memoria cache de instrucțiuni și date

Este o memorie RAM de 32 kB, având o structură de 512 rânduri x 64 octeți. O memorie unificată, pentru date și instrucțiuni are avantajul simplității realizării tehnologice dar și dezavantajul accesului nesimultan la date și instrucțiuni; acest dezavantaj a dus la separarea celor două memorii la Power PC 603 (ca la Pentium !).

- Memoria cache este formată din 8 blocuri identice, fiecare cu 64 de rânduri;
- Fiecare rând are 3 zone: zona de memorare = două sectoare de 32 octeți fiecare, zona de stare (4 biți, din care 2 pentru sector și 2 pentru control) și o etichetă de adresare.

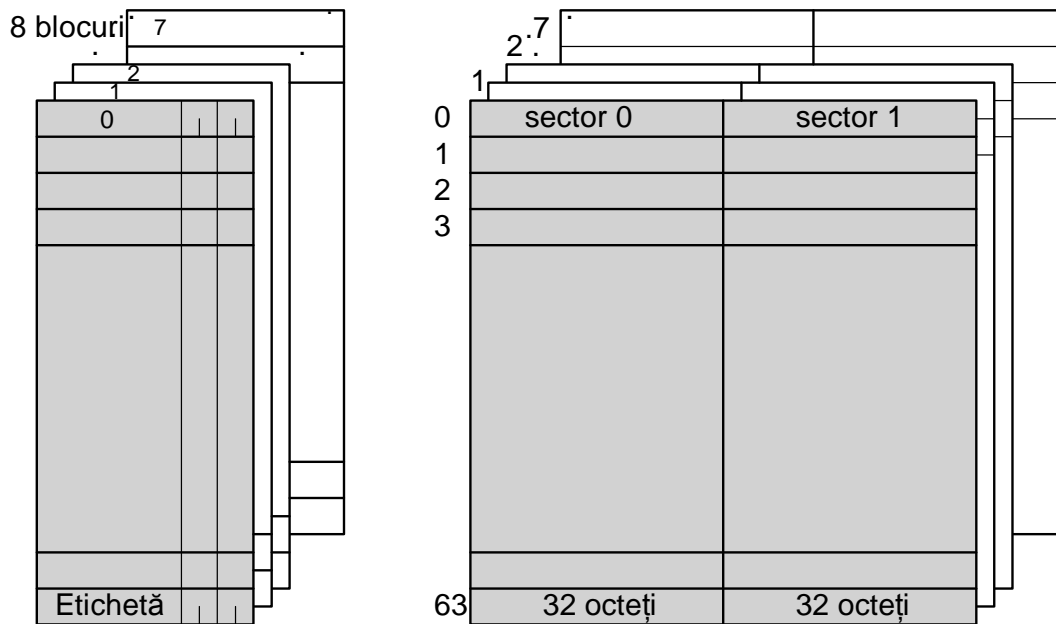


Fig. 3 Organizarea memoriei cache la Power PC 601

Toate operațiile cu memoria cache se execută pe sectoare, cu blocuri de 32 de octeți. Fiecare rând are o adresă (index) dată de câmpul etichetă și doi biți care indică sectorul. Adresa este una *fizică*, deci pentru operații de citire / scriere nu este necesar să se efectueze translatarea adresă virtuală - adresă fizică și accesul este mai rapid.

Memoria cache se încarcă din memoria externă cu date și instrucțiuni până se umple la capacitate.

Pentru a nu fi ocupată mult timp cu date inutile, se aplică un algoritm de înlocuire a datelor neutilizate - LRU (*Least Recently Used*).

Pentru facilitarea accesului la cache, fiecare etichetă de rând dispune de un port de intrare - ieșire care conectează rândul cu magistrala internă de date. Traficul de date prin aceste porturi, nu interferă cu accesul unității centrale la cache, acces separat și dedicat UC.

Posibilele coliziuni între cele două moduri de acces sunt rezolvate de un algoritm de supraveghere.

22.6 Unitatea de gestiune a memoriei MMU (*Memory Management Unit*)

Are funcțiile de translatare a adreselor virtuale în adrese fizice (reale) pe 32 de biți, segmentare și paginare a memoriei externe. Adresa

virtuală este de 52 de biți, ceea ce permite utilizarea unei memorii virtuale de 4 000 000 Go = 4000 To.

Spațiul adresabil poate fi divizat în fragmente de:

- 256 Mo - segmente de memorie;
- 128 ko - 8 Mo, blocuri de memorie;
- 4 ko - pagini.

Utilizarea blocurilor este caracteristică prelucrării imaginilor grafice.

Pagina de 4 ko este unitatea de bază pentru transferul de date de la disc la memoria centrală și invers.

Sunt disponibile două niveluri de acces la memorie (priorități):

- Modul supervisor pentru sistemul de operare (acces total);
- Modul utilizator, pentru programele de aplicație (acces limitat).

22.7 Interfața cu magistralele

Magistrala de date de 64 de biți și cea de adrese de 32 de biți sunt separate și supuse controlului de paritate pe octet de către unitatea centrală, ceea ce permite depistarea erorilor de transfer (în număr impar) pe fiecare magistrală. Magistrala de control are 52 de linii de stare și de informare.

Transferurile pe magistrala de date sunt tipic de 64 de biți dar pot fi de dimensiune mai mică în cazul transferurilor de tip salvă (*burst*).

Interfața permite extensii pentru operații cu porturile și dispune de semnale pentru funcționare în sistem multiprocesor.

22.8 Funcționarea în paralel a unităților (superscalară)

Cu trei unități de execuție independente, Power PC 601 este un procesor de tip superscalar. Fiecare unitate, la rândul său, dispune de subunități specializate în tratarea instrucțiunilor în modul *pipeline*.

Unitățile de execuție primesc instrucțiunile de la unitatea de distribuție, din fișierul de așteptare, IQ (*Instruction Queue*) (fig.4).

- Instrucțiunile pentru întregi, sunt extrase numai din prima locație IQ0, unde sunt în general în stare decodată.
- Instrucțiunile FPU pot fi extrase din oricare din primele 4 locații; ele pot fi executate în altă ordine decât cea din program și de aceea sunt marcate cu etichete ce permit punerea lor în ordine după execuție.

- Când o instrucțiune pentru întregi are nevoie de rezultatul unei instrucțiuni precedente, acesta poate fi accesat imediat într-un registru GPR.

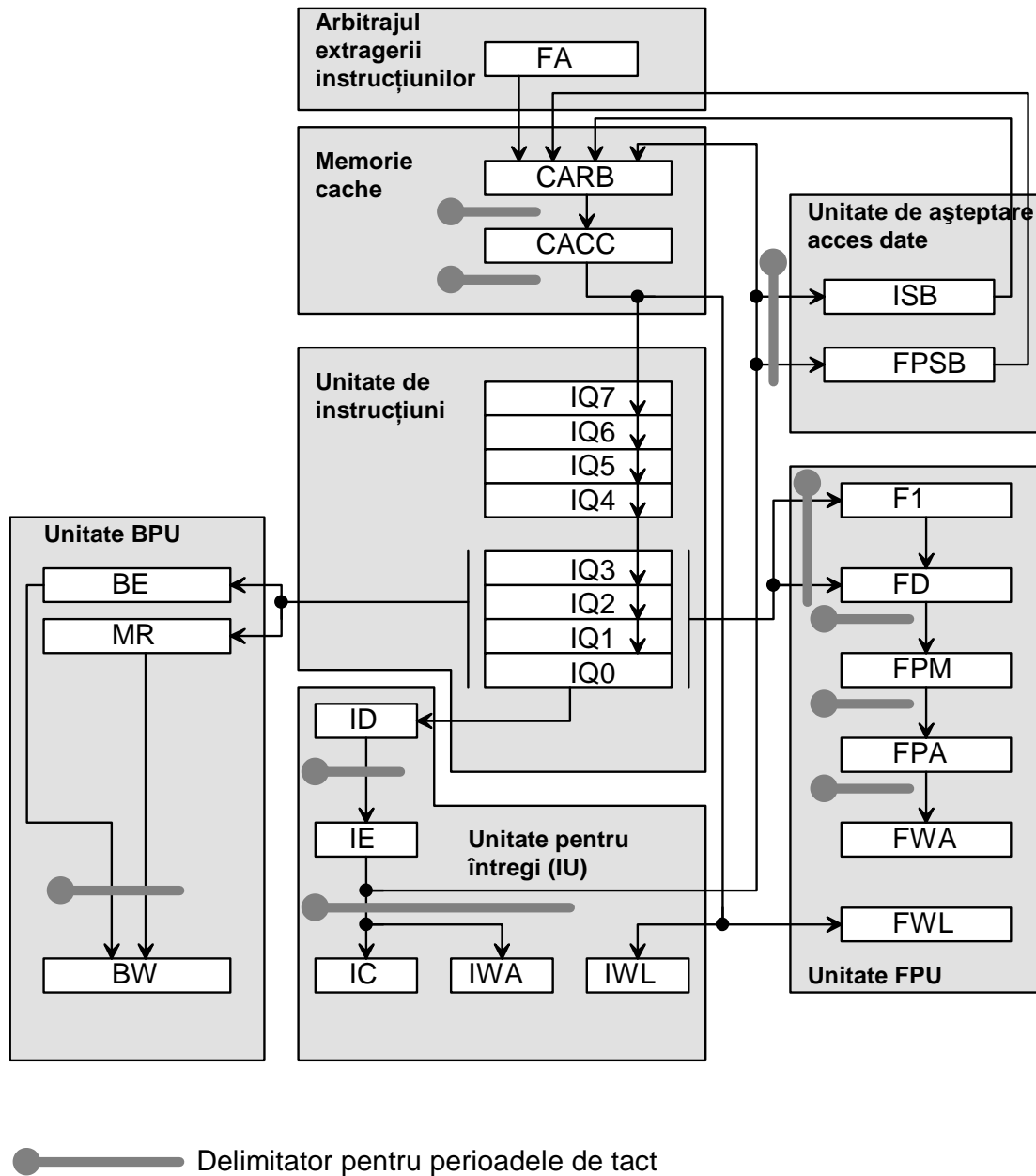


Fig.4. Traseul instrucțiunilor în unitățile de execuție. Modul de lucru pipeline în interiorul unităților .

22.9 Setul de registre

Cea mai mare parte a instrucțiunilor de calcul se execută între registre. Operanzii sursă sunt fie în registre, fie sunt conținuți în instrucțiune ca valori imediate. Microprocesorul Power PC 601 are un set

bogat de registre (peste 100), format din grupe specializate în anumite operații.

- 32 registre GPR (*General Purpose Registers*) de uz general, de 32 de biți, care servesc la adresarea operanzilor pentru calcule cu întregi (în virgulă fixă).
- 32 registre FPR (*Floating Point Registers*) de uz general, de 64 de biți, care servesc operanzii pentru calcule în virgulă mobilă.
- 32 registre speciale SPR (*Special Purpose Registers*), de 32 de biți, utilizate pentru comenzi, stări, configurare procesor, operații diverse. Din acestea, 6 sunt destinate modului *utilizator* iar 26 sunt destinate exclusiv modului *supervizor*.

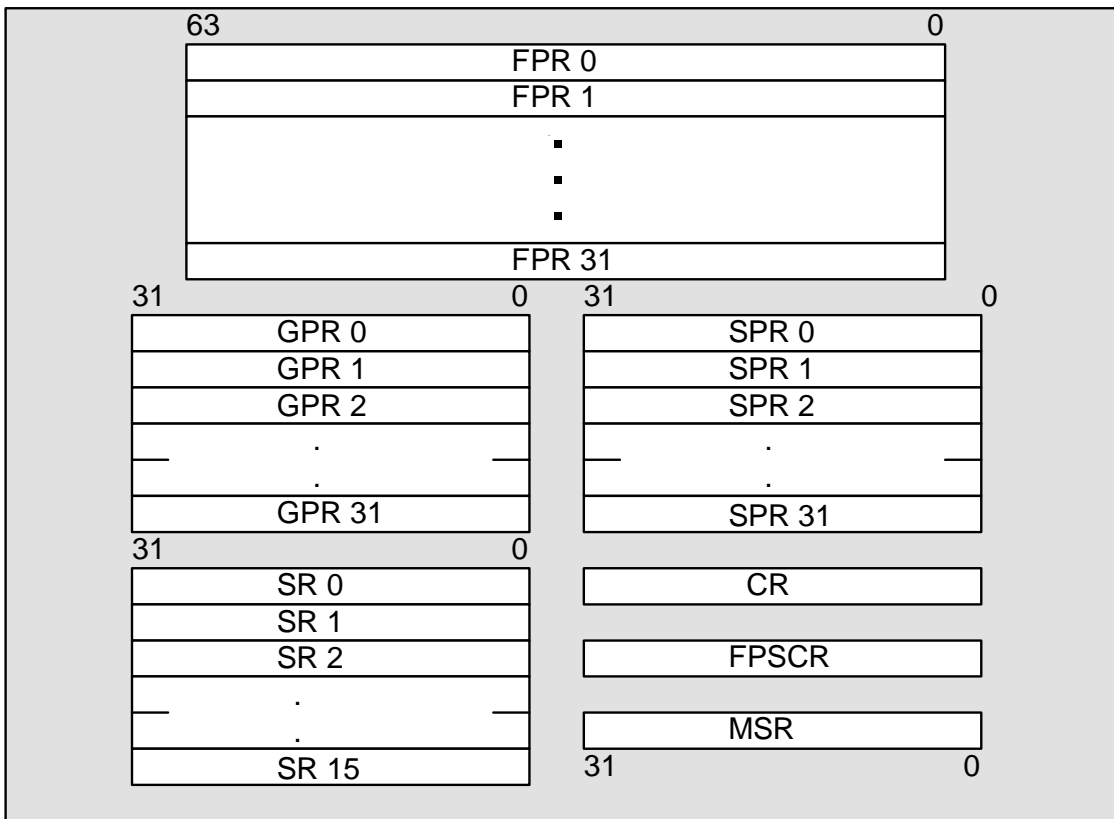


Fig.5 Power PC 601 - Setul de registre

- Registrul de condiții, CR (*Condition Register*) de 32 de biți, are 8 câmpuri de 4 biți fiecare și corespunde registrului indicatorilor de condiții (EF) de la Pentium.
- Registrul de control și stare în virgulă mobilă, FPSCR (*Floating Point Status and Control Register*), conține informații privind unitatea FPU.
- Registrul de stare mașină, MSR (*Machine Status Word*) pentru nivelul supervizor, de 32 de biți, conține câmpuri de biți care definesc starea procesorului.
- 16 registre segment, SR (*Segment Register*), de 32 de biți, pentru adresarea

memoriei; conținutul unui registru segment este interpretat în funcție de valoarea bitului SR_0

Adresarea indirectă: Registru+Deplasament **Adresarea indexată: Registru + Registr**

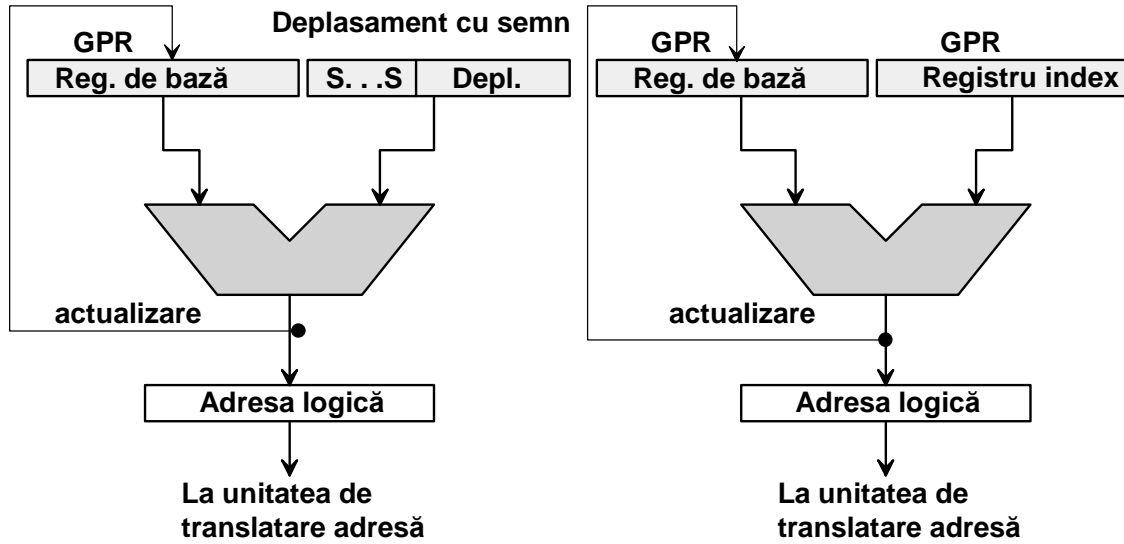


Fig.6 Cele două moduri de adresare a memoriei

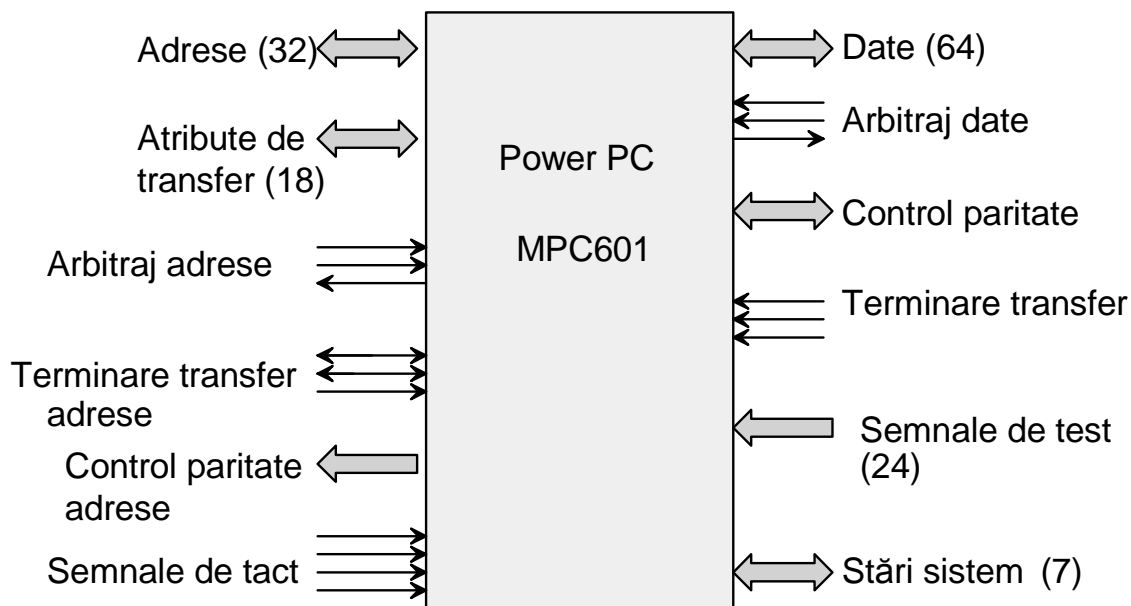


Fig. 7. Grupele de semnale: 64 linii de date, 32 linii de adrese
52 linii de stare, control, și informare