**Arhitectura microprocesorului Intel 8086**

Scopul lucrării

a) Studiul arhitecturii interne a microprocesorului Intel 8086.

b) Studiul resurselor interne (registre) si externe (memoria principala) ale microprocesorului I 8086.

c) Programarea microprocesorului Intel 8086.

**1. Microprocesorul Intel 8086**

1.1 Sisteme cu microprocesor Intel 8086

Microprocesorul este o unitate centrala de prelucrare (UCP) realizata intr-un singur circuit integrat. Un sistem digital de prelucrare realizat cu ajutorul unui microprocesor este numit microcalculator. Microcalculatorul are trei functii principale: prelucrarea informatiilor in UCP, stocarea informatiilor in memorie si transferul informatiilor in interior si cu mediul exterior. Prin intermediul unor interfete, numite porturi de intrare/iesire (IO), microcalculatorul transfera informatii cu elementele mediului exterior, numite echipamentele periferice. Echipamente periferice sunt: tastatura, monitor, imprimanta, disc hard, discheta, cititor de CD-ROM, etc. Microprocesorul Intel 8086 este o unitate centrala de prelucrare care este formata din 2 componente:

1. Unitatea de executie (UE) decodifica instructiunile numerice, da comenzi interne pentru efectuarea calculelor si comenzi externe catre cea de-a doua unitate. UE contine 8 locatii de memorie interna numite registre de uz general. Registrele ofera o capacitate de memorare mica dar si un acces (citire sau scriere) foarte rapid. Pentru a stoca o cantitate mai mare de date (codurile numerice ale instructiunilor si variabilele programelor) este necesara conectarea microprocesorului cu o memorie de capacitate mare, numita memorie principala (MP). Desigur ca accesul la MP este mult mai lent.

2. Unitatea de interfata cu bus-urile (UI) calculeaza adresele MP si IO, transfera datele intre UE si MP sau intre UE si I/O, si transfera catre UE codurile numerice ale instructiunilor citite din MP. Pentru a identifica in mod unic fiecare dintre locatiile MP si IO, este necesara asocierea unor referinte numerice numite adrese. De aceea UI este responsabila cu generarea adreselor catre MP.

Microprocesorul Intel 8086 lucreaza cu date de 16 biti, numite cuvinte de date. Transferurile intre UE si UI sau intre microprocesor si MP sau IO se fac in general sub forma de cuvinte de date de 16 biti. De aceea bus-ul intern prin care comunica UE si UI al microprocesorului Intel 8086 este de 16 biti. Pentru compatibilitate cu microprocesoarele care lucrau cu date de 8 biti, si Intel 8086 poate transfera valori sub forma unor octeti (date de 8 biti). De aceea locatiile MP si ale IO sunt octeti. Microprocesorul Intel 8086 poate lucra cu maximum 1 M octeti de MP, adica poate genera cel mult 1 M de adrese distincte. Deoarece 1M = 216 , inseamna ca adresele la microprocesorul Intel 8086 sunt reprezentabile cu 20 de biti. In figura 1 este prezentata schema bloc a unui sistem cu microprocesor I8086. Microprocesorul Intel 8086 comunica cu exteriorul (MP si IO) prin 3 bus-uri sau magistrale:

- bus-ul de date (BD), care are 16 biti;

- bus-ul de adrese (BA), care are 20 biti;

- bus-ul de comenzi (BC).

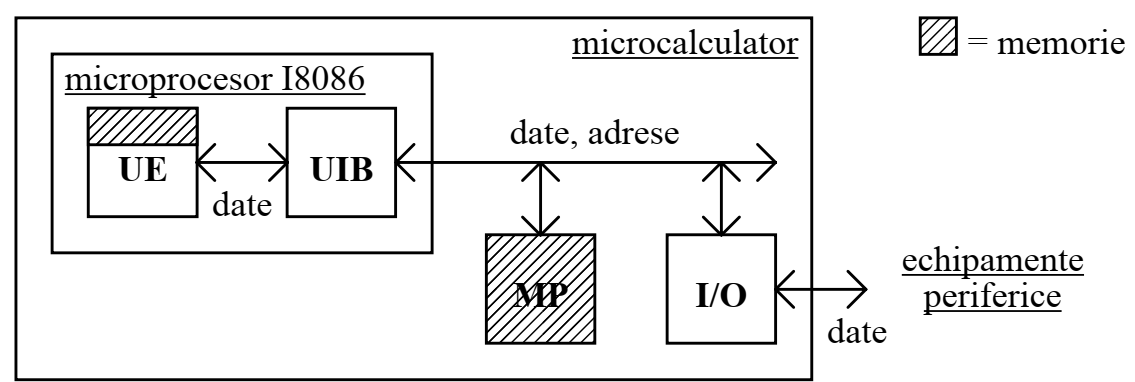


Fig.1. Schema bloc a unui sistem cu microprocesor I8086.

1.2. Arhitectura microprocesorului Intel 8086

Structura microprocesorului Intel 8086 este de tip "pipe-line" (prezinta paralelism temporal) permitand efectuarea in acelasi timp a doua operatii diferite de catre cele doua unitati diferite ale sale - UE decodifica instructiunile si efectueaza calculele, in timp ce UI calculeaza adresele si efectueaza transferurile. Astfel, cele doua unitati ce compun UCP efectueaza autonom secvente de operatii proprii, transferindu-si in acelasi timp informatii. Secventele de operatii efectuate de cele doua unitati ale microprocesorului pentru a executa instructiunile sunt numite cicluri de instructiune, pentru UE si cicluri masina de bus, pentru UI.

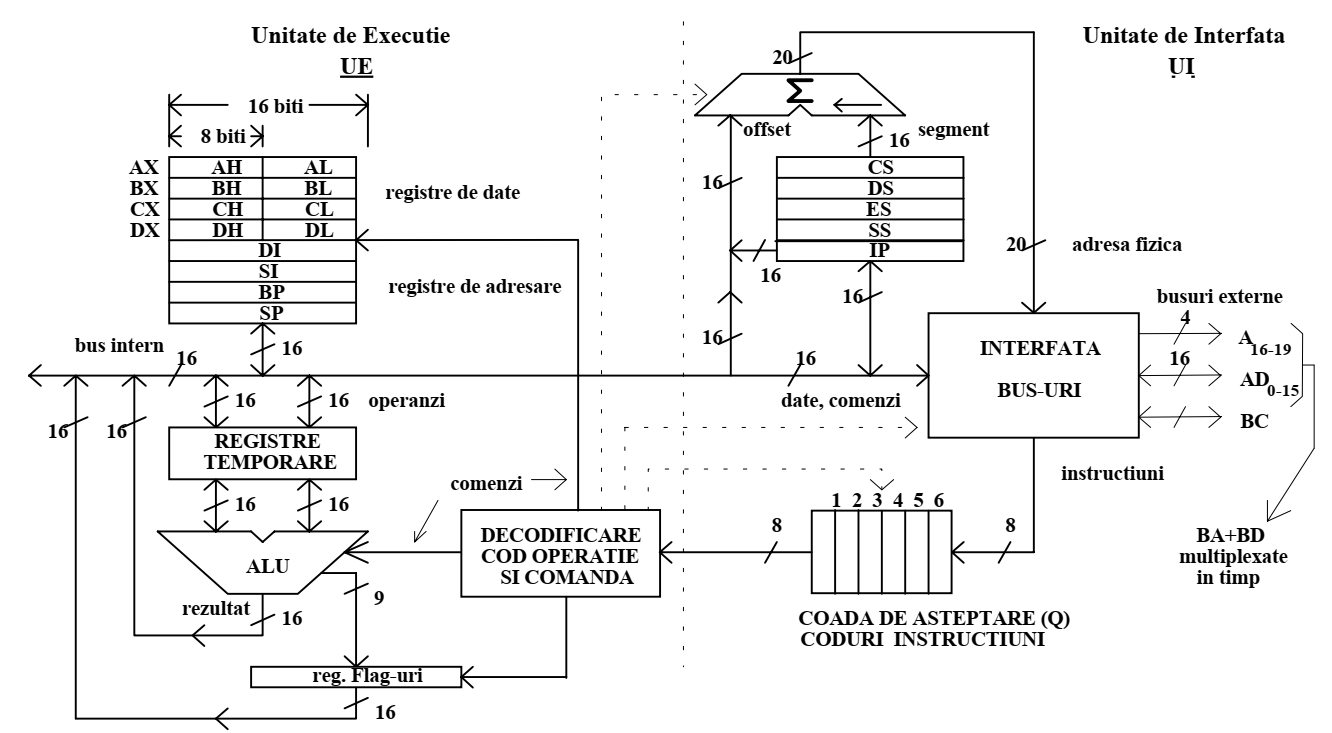


Fig.2 Arhitectura interna a microprocesorului Intel 8086

Unitatea de executie (UE) se compune din:

1. Unitatea aritmetica-logica (ALU), care executa operatii aritmetice, logice, deplasari si rotatii.

2. Registrele temporare (RT), care preiau operanzii de pe bus-ul intern si ii ofera unitatii ALU. Impreuna cu ALU formeaza un automat RALU.

3. Registrul de flag-uri F (biti indicatori de stare a ultimei operatii ALU), actualizat de catre ALU.

4. Blocul de comanda, care:

- decodifica codul instructiunii curente (preluat din coada de instructiuni Q);

- da comenzi interne catre celelalte blocuri ale UE pentru:

- calculul adreselor efective ale operanzilor din MP sau IO (daca este cazul),

- executia operatiei (ALU, transfer, etc.);

- da comenzi externe (informatii / cereri) catre UI pentru:

- calculul adreselor fizice a operandului din MP sau IO (daca este cazul),

- transferul operanzilor dinspre/catre MP sau IO (daca e cazul),

- transferul (citirea) operanzilor de tip imediat din coada de asteptare a UI (daca e cazul),

- calculul adresei urmatoarei instructiuni.

Se observa ca unitatea de executie este complet separata de exterior, toate sarcinile privind transferul cu exteriorul revenind unitatii de interfata.

Unitatea de interfata cu bus-urile (UI) se compune din:

1. Blocul de interfata intre bus-uri, care face toate transferurile la cererea UE:

- cicluri de scriere in MP sau IO (dinspre UCP catre exterior);

- cicluri de citire din MP sau IO (dinspre exterior catre UCP).

2. Coada de asteptare a codurilor de instructiune (Q), care:

- este incarcata cu coduri de instructiune (ciclu FETCH) de catre blocul de interfata atunci cind UE nu cere transferuri de date;

- este inactiva (UCP executa cicluri inactive de bus) daca este plina si nu se cer transferuri;

- este stearsa complet (resetata) daca instructiunea curenta este de salt.

3. Blocul de calcul al adreselor fizice incluzand:

- registrele segment, care contin componenta sement a adresei locatiei MP accesate;

- registrul indicator al instructiunii curente (IP), care contine componenta offset a adresei instructiunii curente;

- unitatea de deplasare-adunare pentru calculul adresei fizice din componentele segment si offset.

**2. Registrele microprocesorului Intel 8086**

Registrele de capacitate 16 biti ale microprocesorului Intel 8086 pot fi clasificate din punct de vedere al rolului pe care il au in executia instructiunilor in 4 grupuri:

- registrele generale: AX, BX, CX, DX, SP, BP, SI, DI;

- registrele segment: CS, DS, ES, SS;

- registrul indicator al adresei instructiunii curente: IP;

- registrul de flag-uri: F.

2.1. Registrele generale

Registrele generale se pot imparti in doua seturi de registre:

- registre de date: AX, BX, CX, DX;

- registre de adresare: SP, BP, SI, DI;

1. Registrele de date se deosebesc prin faptul ca jumatatile (de capacitate 8 biti) lor pot fi accesate (citite sau scrise) separat. Aceasta inseamna ca fiecare registru de date poate fi folosit ca un registru de 16 biti sau ca 2 registre de 8 biti :

- AX (acumulator) de 16 biti, poate fi accesat ca AH si AL, ambele de 8 biti;

- BX (baza in adresarea datelor) de 16 biti, poate fi accesat ca BH si BL, ambele de 8 biti;

- CX (contor) de 16 biti, poate fi accesat ca CH si CL, ambele de 8 biti;

- DX (date) de 16 biti, poate fi accesat ca DH si DL, ambele de 8 biti.

Registrele de date sunt utilizate in majoritatea instructiunilor aritmetice si logice. Majoritatea instructiunilor aritmetice utilizeaza in acelasi mod toate registrele. Exista si instructiuni aritmetice pentru care anumite registre generale au intrebuintari speciale, prezentate in continuare:

|  |  |
| --- | --- |
| AX  AL  AH  BX  CX  CL  DX | - operatii de intrare/iesire pe 16 biti, implicit in inmultiri si impartiri pe 16 biti; - operatii de intrare/iesire pe 8 biti, implicit in translatii, aritmetica BCD, inmultiri si impartiri pe 8 biti;  - implicit in inmultiri si impartiri pe 8 biti; - operatii cu memoria - adresare indirecta, implicit in translatii; - implicit in operatii cu siruri sau bucle; - operatii de deplasare sau rotatie cu mai mult de 1 pozitie; - operatii de intrare/iesire - adresare indirecta, implicit in inmultiri si impartiri pe 16 biti. |

2. Registrele de adresare, de 16 biti, se impart la rindul lor in:

- registre indicatoare de adresa in stiva (pointer):

- SP (Stack Pointer) - contine adresa curenta a varfului stivei,

- BP (Base Pointer) - implicit contine adresa de baza pentru adresarea indirecta a stivei ;

- registre indicatoare de adresa pentru siruri (index):

- DI (Destination Index) - implicit contine adresa curenta pentru sirul destinatie,

- SI (Source Index) - implicit contine adresa curenta pentru sirul sursa.

Registrele de adresare pot fi utilizate si pentru date in anumite instructiuni aritmetice si logice. Registrele pointer contin componente offset ale adreselor din stiva (adresele relative in segmentul de stiva curent). Registrul BP poate fi utilizat si pentru adresarea in cadrul altor segmente. Registrele index contin componente offset ale adreselor variabilelor (adresele relative in segmentul de date curent). Ele sunt utilizate ca registre de adresare in instructiunile de transfer sau prelucrare de siruri de octeti (caractere). In acest ultim caz SI contine adresa relativa curenta a sirului destinatie in cadrul segmentului de date curent (DS), iar DI contine adresa relativa curenta a sirului sursa in cadrul segmentului de date suplimentar (ES).

2.2. Registrele segment

Spatiul de memorie de 1 Moctet este impartit in segmente logice de lungime 64K octeti. UCP Intel 8086 are acces la patru segmente deodata prin intermediul a patru registre segment. Cele 4 registre segment sunt:

- CS (Cod Segment) - contine componenta segment a adreselor codului (instructiunilor programului);

- DS (Data Segment) - contine componenta segment a adreselor variabilelor (segment date curent);

- ES (Extra Segment) - contine componenta segment a adreselor variabilelor (segment suplimentar);

- SS (Stack Segment) - contine componenta segment a adreselor datelor din segmentul stiva.

Instructiunea care urmeaza sa se execute se gaseste in segmentul a carui adresa se afla in registrul CS (Cod Segment), la adresa relativa continuta in registrul IP. Continutul registrului DS defineste segmentul de date curent. Toate referirile la datele din memorie, cu exceptia celor prin registrele BP si SP sau registrul DI in instructiunile pentru siruri, utilizeaza in mod implicit segmentul referit de registrul DS. Continutul registrului ES defineste segmentul de date suplimentar. Referirile la date in instructiunile pentru siruri utilizeaza in mod implicit segmentul referit de registrul ES. Continutul registrului SS defineste segmentul curent al stivei. Toate referirile la datele din memorie prin registrele BP si SP utilizeaza in mod implicit segmentul referit de registrul SS.

2.3. Alte register

Registrul indicator al adresei instructiunii curente IP (Instruction Pointer) este un registru de 16 biti care contine componenta offset a adresei instructiunii in segmentul de cod curent. Programele nu au acces direct la IP, dar exista instructiuni care il modifica si il incarca sau il descarca prin stiva. Registrul de flag-uri F cuprinde bitii indicatori de stare si control numiti si biti indicatori de conditii, fanioane sau flag-uri. Indicatorii de conditii sunt utilizati pentru a memora informatii referitoare la rezultatul unor operatii aritmetice si logice (OF,SF,ZF,AF,PF,CF) si pentru memorarea unor informatii de control al microprocesorului (TF,DF,IF). Cei 9 biti mentionati sunt plasati in registrul F ca in figura 3.

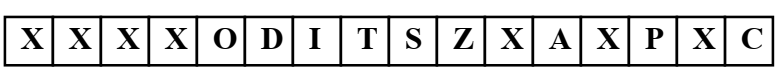


Fig. 3. Continutul registrului indicatorilor de stare si control

Semnificatiile flag-urilor sint urmatoarele:

- CF (Carry Flag) - indicator de transport - reflecta transportul in exterior al bitului cel mai semnificativ al rezultatului operatiilor aritmetice. Astfel, acest indicator poate fi folosit in cazul operatiilor in dubla precizie. Valoarea CF = 1 semnifica fie transport la adunare fie imprumut la scadere. De asemenea, indicatorul CF este modificat si de instructiunile de deplasare si rotatie.

- PF (Parity Flag) - indicator de paritate - este 1 daca rezultatul are paritate para (contine un numar par de biti 1). Acest indicator este folosit de instructiunile de aritmetica zecimala.

- AF (Auxiliary Carry Flag) - indicator de transport auxiliar - este 1 daca a fost transport de la jumatatea de octet inferioara la jumatatea de octate superioara (de la bitul 3 la bitul 4). Acest indicator este folosit de instructiunile de aritmetica zecimala.

- ZF (Zero Flag) - indicatorul de zero - este 1 daca rezultatul operatiei a fost zero.

- SF (Sign Flag) - indicatorul de semn - este 1 daca cel mai semnificativ bit al rezultatului (MSb) este 1, adica in reprezentarea numerelor in complement fata de 2 (C2) rezultatul este negative (are semn -).

- OF (Overflow Flag) - indicatorul de depasire aritmetica (a gamei de valori posibil de reprezentat) - este 1 daca dimensiunea rezultatului depaseste capacitatea locatiei de destinatie si a fost pierdut un bit (indica la valorile cu semn faptul ca se "altereaza" semnul).

- IF (Interrupt Flag) - indicatorul de validare a intreruperilor - prin valoarea lui 1 permite UCP sa recunoasca cererile de intrerupere externe mascabile. Prin valoarea 0 intreruperile externe mascabile vor fi invalidate. Acest indicator nu afecteaza intreruperile interne sau pe cele externe nemascabile.

**3. Memoria principala a unui sistem cu microprocessor**

Microprocesorul INTEL 8086 poate adresa un spatiu de memorie principala (MP) de 1Moctet. Instructiunile si datele, pentru a oferi o utilizare eficienta a memoriei, pot fi stocate in memorie fara sa conteze alinierea lor. Conform conventiei INTEL, datele formate din mai multi octeti (multioctet) sunt intotdeauna memorate cu octetul cel mai semnificativ (MSB) la locatia de adresa cea mai mare. Asadar octetul cel mai putin semnificativ (LSB) este memorat la adresa cea mai mica din grupul de octeti ai unei date multioctet.

3.1. Gestiunea memoriei principale. Segmentarea

Microprocesorul Intel 8086 vede memoria organizata ca un grup de segmente. Un segment este un bloc de memorie, constituit din locatii de memorie contigue, de dimensiune 64 Kocteti. Fiecare segment poate fi accesat (poate fi citit sau scris) in mod independent. Pentru aceasta este necesar ca fiecare segment sa poata fi adresat independent. De aceea un segment are asociata o adresa de baza, care este adresa locatiei la care incepe segmentul. Aceasta adresa este multiplu de 16. Segmentele pot fi adiacente, disjuncte, partial suprapuse sau suprapuse complet (ca in figura 4). O locatie fizica poate sa apartina unuia sau mai multor segmente. Fiecare program (aplicatie) defineste si utilizeaza segmentele diferit. Spatiul de memorie accesibil la un moment dat este de 64 K octeti pentru cod (prin intermediul CS), 64 K octeti pentru stiva (prin SS), 128 K octeti pentru date (prin DS si ES) (vezi figura 4).

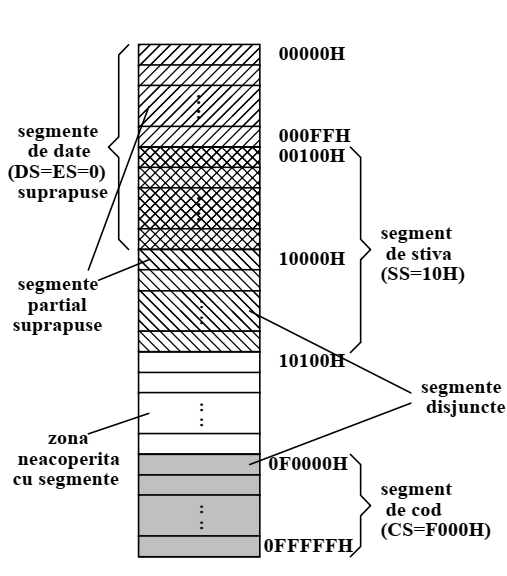


Fig. 4. Exemplu de organizare a memoriei in segmente

3.2. Generarea adresei fizice

Fiecare locatie de memorie are o adresa propriu-zisa, numita adresa fizica, pe care microprocesorul Intel 8086 o calculeaza din cele doua componente, segment si offset, ale adresei logice. Adresa fizica este o valoare de 20 biti care identifica unic o locatie din spatiul de adresare. Adresa fizica poate fi in domeniul 0H ... 0FFFFFH. Pentru ca programele sa fie relocabile, microprocesorul foloseste insa o adresa logica pentru a calcula adresa fizica. Adresa logica consta dintr-o componenta segment de 16 biti si o componenta offset de 16 biti. Din componenta segment microprocesorul poate calcula adresa de baza (de inceput) a segmentului prin inmultirea cu 10 H = 10000 B.

Adresa fizica se calculeaza adunand la adresa de baza a segmentului componenta offset. De aceea componenta offset se mai numeste si deplasare, adresa relativa sau adresa efectiva, iar adresa fizica se mai numeste si adresa absoluta. Notatia consacrata pentru adresa logica este urmatoarea: segment : offset

Calculul adresei fizice se face cu formula:

segment \* 10 H + offset

unde: segment \* 10 H este adresa de baza a segmentului, offset este adresa relativa la baza segmentului: Deoarece inmultirea cu 10 H = 10000 B este echivalenta cu deplasarea cu o cifra hexazecimala la stanga, respectiv cu deplasarea cu 4 cifre binare (biti) la stanga, rezulta ca adresa fizica se calculeaza prin deplasarea cu 4 biti la stanga a segmentului si adunarea cu offset-ul, asa cum este ilustrat in exemplul din figura 5.

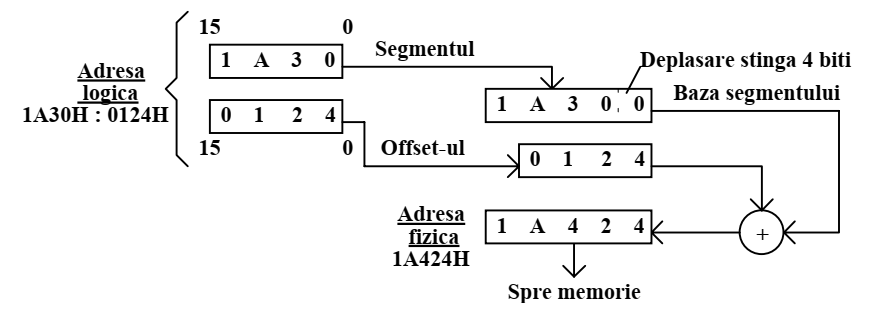


Fig. 5. Calculul adresei fizice din adresa logica

Pentru orice locatie de memorie, adresa de baza a segmentului este adresa primului octet al segmentului care contine locatia, iar adresa relativa este distanta in octeti de la inceputul segmentului pina la locatia respectiva (vezi figura 6).

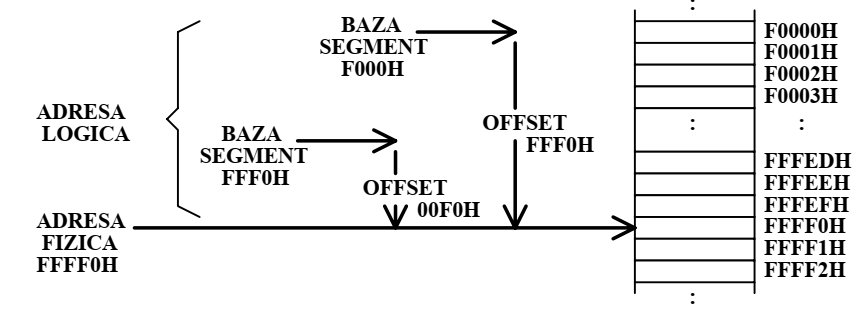


Fig. 6. Adresa logica (16 biti + 16 biti) si adresa fizica (20 biti)

Aceeasi locatie poate fi accesata (citita sau scrisa) prin diferite adrese logice. De exemplu, din adresa logica F000H : FFF0H se calculeaza adresa fizica F000H \* 10H + FFF0H = FFFF0H, iar din adresa logica FFF0H : 00F0H se calculeaza adresa fizica FFF0H \*10H + F0H = FFFF0H. Unitatea de interfata cu magistrala (UI) obtine adresa logica a unei locatii de memorie diferit in functie de modul de adresare a memoriei. Structura pe segmente a memoriei UCP Intel 8086 face posibila scrierea unor programe care sunt independente de pozitia lor in memorie, adica sint relocabile dinamic. Pentru ca un program sa fie relocabil dinamic trebuie sa fie scris astfel incat sa nu altereze registrele sale segment si sa nu faca transferuri directe la o locatie in afara segmentului de cod. Aceasta permite programului sa fie mutat oriunde in memoria disponibila, atita timp cit registrele segment sunt actualizate cu noua adresa de baza.

4. Desfăşurarea lucrării  
1. Se studiaza arhitectura microprocesorului Intel 8086.  
2. Din programa de emulare emu8086 se alege un program exemplu se studiază programul și instrucțiunile utilizate

5. Teme si exercitii  
1. Sa se calculeze adresele fizice corespunzatoare urmatoarelor adrese logice:  
a) 1205H : 709H,  
b) ABCDH : 89ABH,  
c) FFF0H : 0FFH,  
d) 3333H : 4444H,  
e) 8000H : 8000H.  
2. Sa se calculeze componentele offset corespunzatoare urmatoarelor adrese fizice (se  
cunoaste componenta segment: 2000H):  
a) 20002H,  
b) 20010H,  
c) 20300H,  
d) 24000H,  
e) 2FFFFH.  
3. Sa se calculeze componentele segment corespunzatoare urmatoarelor adrese fizice (se  
cunoaste componenta offset: 400H):  
a) 10400H,  
b) B0400H,  
c) 30800H,  
d) CDE00H,  
e) FFFF0H.  
4. Care dintre urmatoarele adrese fizice apartin segmentului care are componenta segment  
2400H: a) 33FFFH,  
b) 23000H,  
c) 27890H,  
d) 33000H,  
e) 34000H.

Raportul trebuie sa conțină:

1. Codul unui program simplu cu comentariile la fiecare instrucțiune utilizată
2. Rezultatul îndeplinirii programului
3. Concluzia