



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

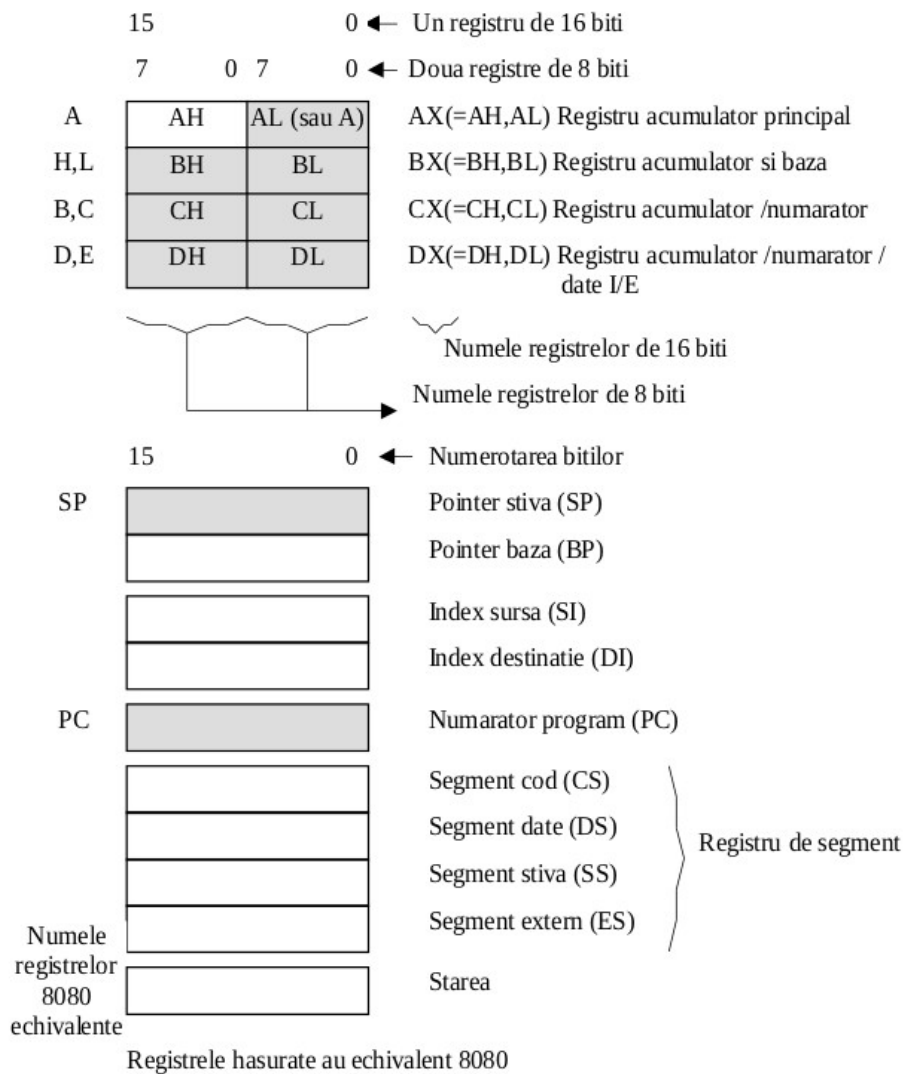
## Proiectarea cu Microprocesoare

### **5. Resursele interne**

Registrele programabile ale microprocesorului 8086 sunt înfățișate în figura 1.4. Prezintă în continuare registrele în legătură cu modurile de adresare ale lui 8086 deoarece multe din acestea sunt utilizate de logica de adresare a memoriei.

AX este acumulatorul principal utilizat, de exemplu, de instrucțiunile de I/O care transferă datele de obicei prin intermediul acestui registru. BX poate fi folosit atât ca acumulator cât și ca registru bază pentru calcularea adreselor de memorie. CX poate fi acumulator sau se utilizează ca numărător în execuția instrucțiunilor iterative. DX poate fi folosit ca acumulator sau ca registru numărător pentru adresarea zonelor de memorie în transferuri cu *port*-urile de intrare/ieșire.

Adresele de memorie pe 20 de biți, generate de 8086, se calculează prin însumarea conținutului unui *registru de segment* cu o *adresă efectivă*. Adresa efectivă se calculează printr-o varietate de moduri de adresare ca și la alte microprocesoare. Registrul de segment selectat este întâi deplasat la stânga cu patru poziții și apoi adunat cu adresa efectivă, generându-se o adresă de 20 de biți:



Conținutul registrului de segment: SSSS SSSS SSSS SSSS 0000 +  
Adresa efectivă de memorie: 0000 EEEE EEEE EEEE EEEE  
Adresa de memorie pe 20 de biți: SSSM MMMM MMMM MMMM MMMM

Registrele de segment sunt folosite deci ca registre de bază care pot puncta orice locație de memorie aflată la o adresă multiplu de 16. Fiecare registru de segment identifică începutul unei zone de memorie, al unui *segment*, de 64 kocteți. Deoarece 8086 are patru registre de segment în orice moment, din întreaga memorie vor fi selectate numai patru segmente de câte 64 kocteți. Nu se impune nici-o restricție asupra conținutului registrelor de segment, ceea înseamnă că aceste registre identifică numai *originea* segmentelor care pot fi plasate oriunde în spațiul de adresare de 1 Moctet al microprocesorului. Memoria de 1 Moctet nu este deci împărțită în pagini de 64 kocteți, ci adresată pe segmente care pot fi *separate* sau *suprapuse*.

Cele patru registre de segment îndeplinesc funcții de adresare diferite: în funcție de tipul accesului la memorie, cod, date, stivă și extern, calculul adresei folosește registrul de segment asociat. Astfel la o extragere de instrucțiune, numărătorul de program, PC, este adunat cu conținutul registrului de segment de cod, CS. La orice instrucțiune de lucru cu stiva, de tip Push, Pop, Call sau Return, adresa de memorie se calculează adunând conținutul *pointer*-ului de stivă, SP, cu conținutul registrului de segment de stivă, SS. La instrucțiunile pe șiruri de date pentru calculul adreselor sursă și destinație sunt utilizate registrele index SI și DI împreună cu registrele de segment DS și ES, figura 1.5. Observăm că pentru aceste instrucțiuni este necesar ca șirurile sursă și destinație să fie plasate fiecare în cadrul unui singur spațiu de adresare de 64 kocteți, nu neapărat același. Instrucțiunile care accesează date calculează adresa prin adunarea adresei efective cu conținutul registrelor de segment DS sau SS.

Dăm în tabelul 1.1 o prezentare sintetică a modurilor de calculare a adresei de memorie.

Referința de memorie	Registrul de segment	Registrul bază	Registrul index	Deplasament pe 16 biți, fără semn	Deplasament pe 8 biți, semn extern	Fără deplasament
Date	DS (CS, SS, ES) <sup>1</sup>	–	SI	* <sup>2</sup>	*	*
			DI	*	*	*
	BX	–	SI	*	*	*
			DI	*	*	*
			–	*	*	*
	DS	–	–	*		
	SS (CS, DS, ES) <sup>1</sup>	BP	SI	*	*	*
			DI	*	*	*
			–	*	*	
Stivă	SS	SP	–			
Șiruri	DS	–	SI			
	ES	–	DI			
Extragere instrucțiune	CS	PC	–			
Salt	CS	PC	–		*	
Date I/O	DS	DX	–			

Tabelul 1.1. Calculul adreselor de memorie

Note: 1. Depășirea segmentului permite înlocuirea registrelor de segment DS sau SS cu unul din celelalte registre.

2. \* semnifică deplasamentele care pot fi utilizate la calculul adresei

Vom prezenta întâi opțiunile de adresare a datelor. Cea mai simplă adresare este cea *directă*. În acest caz registrul DS este adunat cu un *offset* sau deplasament pe 16 biți dat de doi octeți din codul obiect al instrucțiunii. Dacă se folosesc registrele index SI sau DI, prin adunarea conținutului unuia din acestea cu conținutul registrului DS se obține o adresare *implicită*. Prin adăugarea deplasamentului, de 16 sau 8 biți, obținem o adresare *directă indexată*. Dacă deplasamentul e pe 8 biți atunci cel mai semnificativ bit, semnul, se extinde pe încă 8 biți, mai semnificativi, pentru a forma un deplasament pe 16 biți, de exemplu:

```

Deplasament:    ...    1010 1101
Semn extins:   1111 1111 1010 1101    sau
Deplasament:    ...    0101 1010
Semn extins:   0000 0000 0101 1010

```

Adresa mai poate fi calculată și prin utilizarea unui registru-bază obținându-se o adresare *relativă*. Pentru adresarea relativă a datelor se pot utiliza două registre de bază, BX și BP, adresele

găsindu-se în segmentul DS, pentru date, și, respectiv, în segmentul SS, pentru stivă. În primul fel de adresare relativă cu bază a datelor, conținutul registrului BX este adunat la adresa efectivă calculată în unul din modurile exemplificate mai sus. Putem avea deci trei feluri de adresări relative: directe, implicite sau directe indexate. În al doilea fel de adresare, calculul adresei se face adunând la adresa efectivă conținutul registrului de segment SS.

În legătură cu utilizarea registrelor generale ca registre de bază atragem aici atenția asupra unui mod de adresare a dispozitivelor de I/O prin intermediul registrului DX. În acest mod de adresare conținutul registrului DX este plasat pe magistrala de adrese fără a fi adunat cu vreun registru de segment. Acesta este singurul mod de adresare în care conținutul unui registru este scos direct pe magistrală ca adresă fără a fi în prealabil prelucrat în logica de segmentare.

Toate instrucțiunile de salt condiționat utilizează adresarea relativă la programe, cu registru de bază PC, permițându-se astfel generarea unui cod relocabil. Deplasamentul pe 8 biți reprezintă un număr binar cu semn cuprins în gama ([-128, +127]).

Pentru instrucțiunile de salt necondiționat și de apel sunt folosite trei moduri de adresare: adresare relativă la program, registru de bază PC, cu deplasament pe 8 sau 16 biți; adresare directă; adresare *indirectă*. În adresarea indirectă se utilizează una din formele de adresare descrise mai sus pentru a citi din memorie o nouă adresă de memorie. Există două opțiuni de adresare indirectă: una în care cuvântul de 16 biți citit din memorie e încărcat în PC și instrucțiunea de salt sau apel adresează o locație de memorie din cadrul segmentului CS curent; cealaltă în care se citesc două cuvinte de 16 biți pentru a fi încărcate în PC, respectiv CS și care permite adresarea oricărei locații din spațiul de memorie al microprocesorului.

În figura 1.6 dăm structura registrului de stare al microprocesorului 8086 conținând indicatorii de condiții.

Indicatorul *Transport*, C sau CF, este inversat indicând de fapt un împrumut: după o scădere, realizată de procesor în complement față de 2, acest bit va fi pus pe "1" dacă nu a existat transport la bitul cel mai semnificativ și pus pe "0" dacă a existat un transport.

Indicatorul *Paritate*, P sau PF, este pus pe "1" dacă rezultatul conține un număr par de "1" și pe "0" dacă rezultatul operației are un număr impar de "1"-uri.

Indicatorul *Zero*, Z sau ZF, este standard fiind pus pe "1" dacă rezultatul operației e zero și pe "0" dacă rezultatul nu este zero.

Indicatorul *Semn*, S sau SF, este pus pe "1" dacă rezultatul unei operații este negativ având bitul cel mai semnificativ "1" și pus pe "0" dacă rezultatul este pozitiv.

Indicatorul *Direcție*, D sau DF, determină dacă operațiile pe șiruri vor incrementa sau decrementa registrele index. Dacă *Direcție* este "1" atunci conținuturile registrelor SI și DI vor fi decrementate, șirurile fiind accesate începând cu adresele cele mai mari. Dacă indicatorul este "0" conținutul registrelor SI și DI va fi incrementat, șirurile fiind accesate începând cu adresele cele mai mici.

Indicatorul *Validare/Invalidare Întrerupere*, I sau IF, trebuie să fie "1" pentru a valida întreruperile și "0" pentru a le invalida (cu excepția NMI).

Indicatorul *Derută*, T sau TF, face ca microprocesorul 8086 să intre în modul de lucru "pas-cu-pas" și este strâns legat de sistemul de întreruperi (vezi §1.4.5.1 și §2.3.6).

Indicatorul *Depășire*, O sau OF este, de asemenea, legat de sistemul de întreruperi și de execuția instrucțiunii INTO.

Indicatorul *Transport Auxiliar*, A sau AF indică, după o operație, un transport din bitul 3 în 4.

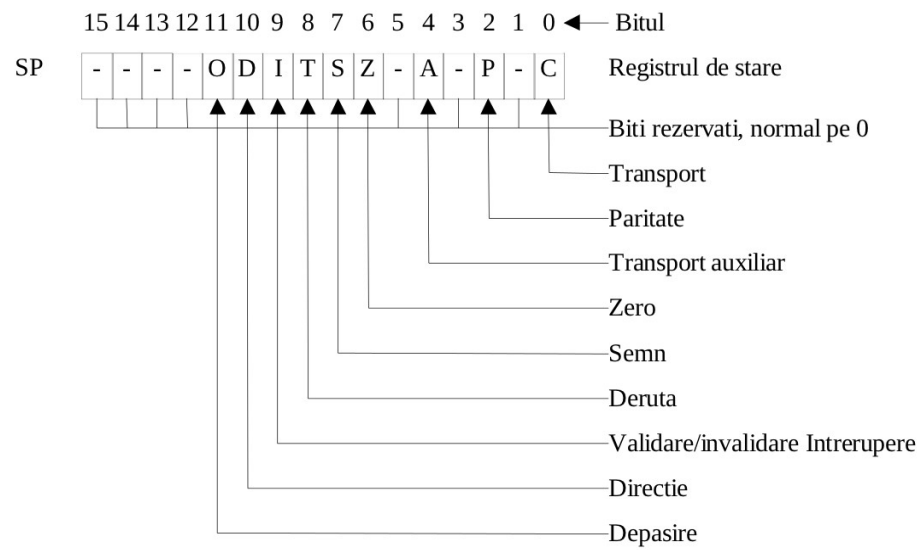


Figura 1.6. Registrul de stare