

Arhitectura sistemelor de calcul

Curs 5 - calculul de adresă

Adresa este o valoare care reprezintă numărul de octeți consecutivi dintre începutul memoriei RAM și începutul unei locații. Aceasta se putea manipula de către programator doar pe programarea sub 16 biți. Pe programarea pe 32 de biți calculul de adresă este făcut de componenta **ADR** din BIU sub o manieră necunoscută.

O succesiune de adrese (locații de memorie) care deserve scopuri similare formează un **segment de memorie**. De exemplu: code segment și data segment sunt subdiviziuni logice a memoriei (a unor succesiuni de adrese).

Un segment de memorie este caracterizat de o adresă de bază, dimensiunea segmentului de memorie (se mai numește și limită) și tipul care poate fi code sau data. Dimensiunea maximă a unui segment pe 32 de biți este $4\text{ GB} = 2^{32}$.

Se numește **offset** sau **deplasament** numărul de octeți de la începutul segmentului de memorie și locația în cauză.

Pentru a reprezenta începuturile de segmente sunt suficienți 16 biți (știm asta deoarece este același sistem de adresare pe 16 biți care a fost extins doar la nivel de offset).

Selectorul de segment este o valoare numerică furnizată de sistemul de operare reprezentată pe 16 biți care identifică în mod unic segmentul de memorie accesat și caracteristicile acestuia.

Programatorii nu pot lucra cu adrese fizice efective, ele sunt calculate și gestionate doar de componenta ADR din BIU. Programatorii pot lucra doar cu **specificări de adrese** care sunt perechi formate din selectorul de segment și un offset. O scriere hexazecimală a unei adrese este

$S_3 S_2 S_1 S_0 : 0_7 0_6 0_5 0_4 0_3 0_2 0_1 0_0$
selector de segment offset

Barza și limita unui segment de memorie sunt determinate de procesor în urma aplicării mecanismului de segmentare, adică selectorul $S_3 S_2 S_1 S_0$ indică accesarea unui

segment a cărui adresă de bază este de forma $b_7b_6b_5b_4b_3b_2b_1b_0$ și are o limită $l_7l_6l_5l_4l_3l_2l_1l_0$ care sunt furnizate de către sistemul de operare către procesor doar dacă selectorul de segment pentru care dorim să realizăm adresarea înregistrului de segment este valid.

Pentru a fi permis accesul către locația specifică trebuie:

$$\underbrace{00000000}_{\text{offset}} \leq \underbrace{l_7l_6l_5l_4l_3l_2l_1l_0}_{\text{limita (size) segment de memorie}}$$

Adresa de segmentare se face printr-un calcul de ADR: $a_7a_6a_5a_4a_3a_2a_1a_0 = \underbrace{b_7b_6b_5b_4b_3b_2b_1b_0}_{\text{adresa de intrare al selectorului de segment}} + \underbrace{00000000}_{\text{offset}}$

O specificare de adresă poartă numele de **FAR** dacă este de forma selector de segment + offset și poartă numele de **NEAR** dacă este de forma offset. Ambele modele de adresare sunt corecte, iar în cazul folosirii unui model de adresare NEAR, microprocesorul păstrează automat selectorul de segment curent și folosește doar offsetul oferit.

pentru calculul adresei de segmentare.

Moduri de adresare a operanzilor:

Modul registru, dacă pe post de operand se găsește un registru `mov eax, 15`

Modul imediat, dacă pe post de operand se găsește o valoare numerică `mov eax, 15`

Modul adresare la memorie, dacă operandul este undeva în memorie `mov eax, [v]`

Calculul offsetului unui operand este:

$$\text{offset} = [\text{bază}] + [\text{index} \cdot \text{scală}] + \text{constantă}$$

baza poate să fie EAX, EBX, ECX, EDX, EBP, ESP, ESI, EDI

index poate să fie EAX, EBX, ECX, EDX, EBP, ESP, EDI fără

ESP deoarece acesta este rezervat lucrului cu stiva

scala poate să fie 1, 2, 4 sau 8 (ex: avem un sir array[10]

unde fiecare element este reprezentat pe 4 bytes, dacă dorim

al 3-lea element atunci `ebx = array`, `ecx = 3`, atunci `array[3]` va

corespunde cu `[ebx + ecx * 4]`

Constanta poate să fie reprezentată pe 32 biți

Moduri de adresare la memorie:

Directă atunci când în formula de calcul a offsetului apare numai constantă.

Bază atunci când în formula de calcul a offsetului apare numai bază.

Scală-indexată atunci când în formula de calcul a offsetului apare unul dintre registri index.

Cele trei moduri de adresare a memoriei pot fi combinate, de exemplu poate să apară adresare directă bazată.

Relativă cîm avem o instrucțiune de salt.

saltul este în save
JMP [0084]↓
offset de la
instrucțiunea JMP

Indirectă este cea care nu este directă și este specificat cel puțin un registru

