

11.11.1 Проаналізувати використання режимів адресації даних в програмах на мові Асемблера ПЕОМ.

Переважає більшість команд процесорів 80x86 мають адресну частину, яка у загальному випадку містить байти `modr/m`, `sib` та зміщення в команді. На основі цих даних процесор формує зміщення в сегменті, яке в даному випадку назвали ефективною адресою. У загальному випадку ефективна адреса є сумою трьох компонент – зміщення в команді, бази та індексу. База та індекс містяться в регістрах загального призначення, які використовуються як адресні регістри. Індекс може мати множник 2, 4 або 8, який показує, на яку величину необхідно помножити вміст 32-розрядного індексного регістра перед формуванням ефективного адреси (для 16-розрядних регістрів множник не задається). Будь-яка з компонент в адресному виразі може бути відсутня, що визначає наступні режими адресації (табл. 3.2):

Таблиця 3.2 - Режими адресації даних

Зміщення в команді	База	Індекс	Режим адресації	Приклад
-	-	+	Посередня регістрова	[si], [eax] [esp]
-	+	+	Базова індексна	[bx+si], [ecx+edx] [ebx+esi*4]
-	+	-	Посередня регістрова	[bx], [ecx]
+	-	+	Індексна	Dat1[si] Dat1[ecx] Dat1[edi*8]
+	-	-	Пряма	Dat1
+	+	+	Базова індексна зі зміщенням	Dat1[bx+di] Dat1[ebx][edx] Dat1[edx+esi*2]
+	+	-	Базова	[bp+4], [bp-6] [ecx+7]

Примітка 1. Для формування ефективного адреси можуть використовуватись лише регістри BP, BX, SI і DI, а також лише наступні їх пари: BX+SI, BX+DI, BP+SI та BP+DI. Для 32-розрядних регістрів загального призначення таке обмеження на їх використання відсутнє, за виключенням регістра ESP – він не може задаватись із множником.

Примітка 2. Мовою Асемблера можна задати посередню регістрову адресацію із множником, наприклад, `Add eax,[edx*4]`, але в процесорі такі команди відсутні. Асемблер сформує машинну команду, в якій зміщення в команді буде мати нульове значення.

При програмуванні мовою Асемблера розглянуті вище режими адресації доцільно використовувати наступним чином:

- пряму – для адресації простих (скалярних) даних, адреси яких при виконанні програми не змінюються;
- посередню регістрову - для адресації скалярних даних, адреси яких змінюються

при виконанні програми;

- **індексну**– для адресації елементів масивів. Особливо ефективно можна використовувати індексну адресацію для доступу до елементів одновимірних масивів байтів, масивів слів, подвійних та квадро слів;
- **базову** – для адресації елементів структур, відносні адреси яких при виконанні програми не змінюються;
- **базову індексну** – для адресації елементів структур, відносні адреси яких змінюються при виконанні програми;
- **базову індексну зі зміщенням** – для адресації елементів масивів структур або для адресації елементів багатовимірних масивів.