**Як працює адресація у реальному режимі**

Існує фізична та логічна адреса. Адресуватися можуть байти або слова(по 2 байти кожне). Фізична адреса передається по шині пам’яті в оперативну пам’ять, тому вона встановлює ліміт на адресацію пам’яті в реальному режимі. Розглянемо 20-розрядну шину, яка дає можливість адресуватися до 1 МБ пам’яті.

00000h <= [фізична адреса] <= FFFFFh.

0000h:0000h <= [логічна адреса] <= FFFFh:000Fh

Так як регістри 16 розрядні, то 20-розрядну фізичну адресу туди не покладеш. Для цього використовують логічну адресу, яка складається з сегментової частини та частини зміщення, кожна по 16 бітів. При обчисленні фізичної адреси, сегмент зсувається на 4 розряди вліво, а зміщення – на 4 вправо, потім компоненти додаються.

1 МБ адресованої пам’яті ділиться на сегменти по 64КБ кожен.

**Регістри**

**Сегментова компонента** може зберігатися в одному з наступних регістрів:

CS – Code Segment - вказує на початок блоку пам'яті об'ємом 64К, або сегмент коду, в якому знаходиться наступна інструкція.

DS – Data Segment - вказує на початок сегмента даних, який являє собою блок пам'яті об'ємом 64К, в якому знаходиться більшість розміщених в пам'яті операндів.

ES – Extra Segment - вказує на початок блоку пам'яті об'ємом 64К, який називається додатковим сегментом. Як зрозуміло з його назви, додатковий сегмент не служить для якоїсь конкретної мети, але доступний тоді, коли в ньому виникає необхідність. Іноді додатковий сегмент використовується для виділення додаткового блоку пам'яті об'ємом 64К для даних.

SS – Stack Segment - вказує на початок сегмента стека, який являє собою блок пам'яті об'ємом 64К, в якому знаходиться стек.

**Компонента зміщення** може знаходитися у одному з наступних регістрів:

IP - Instruction Pointer - зберігає адресу зміщення наступної команди, яку потрібно виконати. IP у поєднанні з регістром CS (як CS:IP) дає повну адресу поточної інструкції в сегменті коду.

SP – Stack Pointer – зберігає значення зміщення програмного стека. SP у поєднанні з регістром SS (SS:SP) означає поточну позицію даних або адреси в програмному стеку.

BP – Base Pointer – допомагає посилатися на змінні параметрів, передані підпрограмі. Адреса в регістрі SS поєднується зі зміщенням в BP, щоб отримати розташування параметра. BP також можна поєднувати з DI та SI як базовий регістр для спеціальної адресації.

SI (Source Index), DI (Destination Index) – індексні регістри, у яких зберігаються адреси операндів під час виконання програми.

BX – Base Pointer - базовий регістр в обчисленнях адреси, часто вказує на початкову адресу (званий базою) структури в пам'яті.

**Безпосередня адресація**

Операндом виступає саме значення, над яким одразу виконується операція. Процесор не звертається до пам’яті, тому цей тип адресації ще називають моментальним.

Приклад: ADD BYTE\_VALUE, 65

**Регістрова адресація**

Операнд знаходиться у регістрі.

Пряма – значення операнда знаходиться у регістрі.

Непряма – адреса на значення операнда знаходиться у регістрі.

Приклад: АDD EX, BX;

**Пряма адресація пам’яті**

Абсолютну адресу операнда задають у вигляді СЕГМЕНТ:ЗМІЩЕННЯ.

Приклад: MOV AX, SS:0037h

**Непряма адресація**

Адресу операнда знаходиться в:

Регістрова – адреса знаходиться у регістрах SI, DI, BX, BP.

Пам’яті – адреса знаходиться у комірці пам’яті.

Зрозуміло, що такий тип адресації є не ефективним.

Приклад:

MOV AX, [SI]

**Відносна адресація**

Адреса визначається як значення операнда + значення Instruction Pointer.

**Базова адресація**

Адреса визначається як сума значення базового регістра та зміщення.

Якщо базовий регістр – BP, то адреса операнда відраховується відносно сегмента, на який вказує SS. Якщо BX, то – DS.

Приклад:

MOV AX, [BX+2] ; береться слово, що знаходиться у сегменті, на який вказує DS, зі зміщенням BX + 2.

**Індексна адресація**

Адреса операнда обраховується як сума зміщення, яке задається в інструкції, та значення SI або DI.

Приклад:

MOV AX, [DI+05]

**Базово індексна адресація**

Зміщення операнда обраховується, як сума одного з базових регістрів (BP або BX) та індексних регістрів (SI або DI).

Приклад:

ADD AX, [BX+SI]

Якщо використовується базовий регістр – BP, то адреса операнда відраховується відносно сегмента, на який вказує SS. Якщо BX, то – DS.