

## Тема 4. Архітектура мікропроцесорів серії x86

### 4.1. Архітектура МП intel 8086 (8088)

Свого часу (1978р.) поява 16-розрядного МП intel 8086(88) визначила цілий напрям у розвитку і застосуванні мікропроцесорних комплексів (МПК). У колишньому СРСР було також розроблено і виготовлено МПК K1810BM86, який був функціонально тотожним intel 8086 і був однокристальним (кремнієвий кристал) 16-розрядним МП зі швидкістю ~2,5 млн операцій за секунду, ємністю ОП до 1 Мбайта, та системою команд, яка містила їх 135.

У комплекті містився також математичний співпроцесор intel 8087 чи K1810BM87 — однокристальний 16-розрядний процесор для виконання команд арифметики з плаваючою комою.

Водночас з МП 8086 створено МП 8088 – однокристальний 16-розрядний МП з більшою швидкодією, ніж 8080 (~ у 5 разів), однак з 8-розрядною зовнішньою магістраллю для зв'язку з пристроями введення/виведення. Набір регістрів і система команд така ж, як і у 8086. Завдяки більшій розрядності 8086 працює на 20-60 % швидше, ніж 8088 з однаковою тактовою частотою.

Крім цього, у комплект входять: генератор тактових сигналів, системний контролер, арбітр шини та ін.

Головні властивості цих МП:

- розвинена регістрова структура, яка значно зменшує кількість звертань до пам'яті;
- конвеєрний принцип виконання команд з попереднім вибиранням, що забезпечує максимальну пропускну здатність системної магістралі;
- наявність мультиплексованої шини адреси / даних;

- багатофункціональне використання виводів МП;
- здатність до координації взаємодії кількох процесорів (зокрема співпроцесорів).

Характерною рисою цих процесорів, на відміну від 4- та 8-розрядних, є те, що вони зорієнтовані на паралельне виконання команд і вибирання з пам'яті. Тобто є дві частини, що працюють асинхронно: спряження з магістраллю і опрацювання даних чи команд.

На базі МП 8086 побудовані такі ЕОМ: ЕС1840, ЕС1841, "Искра 1030", Нейрон, мікроЕОМ СМ1810/СМ1814 та ін.

Структурна схема такого МП показана на рис. 4. 1.

Структурна схема мікропроцесора Intel 8086

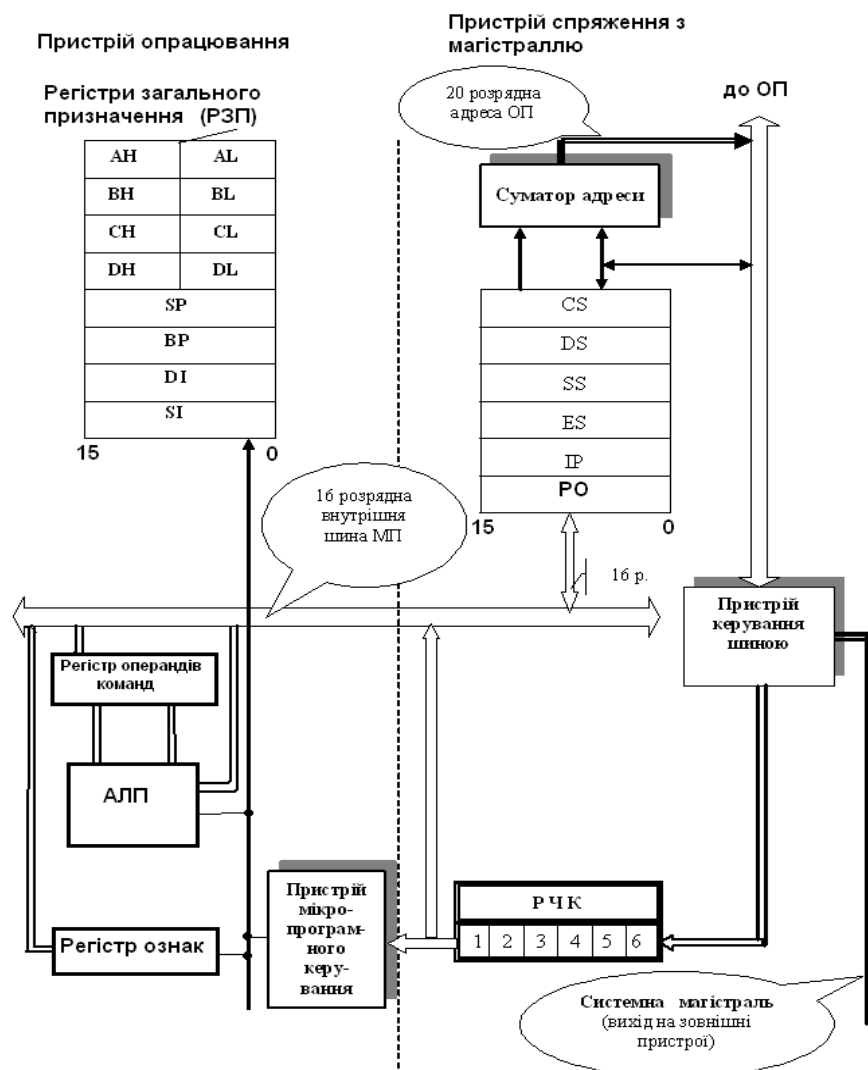


Рис.4.1. Структурна схема мікропроцесора intel8086.

**Пристрій спряження** з магістраллю (ПСМ) забезпечує формування 20-розрядної фізичної адреси пам'яті, вибирання команд і операндів з пам'яті, організацію черговості команд і запам'ятовування результатів виконання команд.

У складі ПСМ є:

шість 8-розрядних регістрів черги команд (РЧК);  
чотири 16-розрядні сегментні регістри (C,D,S,E);  
16-розрядний регістр адреси команд (IP);  
16-розрядний регістр обміну (OP);  
16-розрядний суматор адреси (CMA);  
пристрій керування шиною.

- Цикл вибирання з пам'яті слова розпочинається тоді, коли у черзі вивільняються принаймі 2 байти. Регістр черги команд влаштований так, що команди виконуються за правилом “перший прийшов — першим обслужили”. Виконання команд відбувається у логічній послідовності, яка задана програмою, оскільки в черзі є ті команди і в тій послідовності, як вони були у комірках пам'яті. Якщо трапляється команда передачі керування іншій комірці пам'яті, то пристрій спряження очищує регістр черги команд, вибирає команду за адресою переходу, передає її у пристрій керування і розпочинає нове заповнення регістрів. У разі повернення з програми (після CALL) чи внаслідок переривання відновлюється черга команд, адреси яких обчислюються за допомогою CMA. Якщо процесор виконує цикл читання або записування, то на час дії циклу вибирання команд припиняється.

**Зауваження:** у МП 8088 черга команд у РЧК скорочена до чотирьох.

- Сегментні регістри відповідають чотирьом сегментам пам'яті :

DS – даних;

SS – стеку (стек – спеціально організована область ОЗП, яку використовує ЕОМ для тимчасового зберігання даних або адрес. Число, записане в стек останнім, опрацьовується першим);

CS – коду;

ES – проміжних даних.

У кожному з цих регістрів зберігається 16 старших розрядів коду адреси відповідного сегмента пам'яті.

- Двадцятирозрядну фізичну адресу формує СМА шляхом зсуву базової адреси сегмента на чотири розряди вліво і додавання до нього 16-розрядної адреси зміщення. Адреса зміщення є в одному з регістрів вказівників або індексних і дає змогу адресувати 64 Кбайти вибраного сегмента; 20-розрядна адреса дає змогу працювати з пам'яттю 1 Мбайт.

Якщо отримане у СМА значення перевищує значення найстаршої адреси сегмента, то фізичну адресу знову відраховують від початку сегмента.

**Пристрій опрацювання (ПО)** призначений для виконання операцій з опрацювання даних, і, він складається з

МПК (мікропроцесорний пристрій керування);

16-розрядного арифметико-логічного пристрою (АЛП);

восьми 16-розрядних регістрів загального призначення;  
регістра ознак.

Команди, вибрані пристроєм спряження з пам'яті через РЧК, надходять у МПК. Цей пристрій містить пам'ять мікрокоманд, декодує команди і виробляє послідовність мікрокоманд, що керує процесом опрацювання.

В АЛП виконуються арифметичні і логічні операції над 8- і 16-розрядними числами з фіксованою комою. Для прискорення внутрішніх пересилань даних всі регістри і магістралі

мають 16 розрядів. Прямого зв'язку ПО із зовнішньою магістраллю немає, а обмін даними з ПСМ відбувається через реєстр обміну (РО).

**Програмно-доступними** функціональними частинами є **реєстри загального призначення** (для зберігання операндів та результатів виконання команд), **сегментні реєстри** (для зберігання базових адрес поточних сегментів пам'яті), **адреси команд і ознак**.

Реєстри загального призначення розбиті на дві групи по чотири у кожній:

- 1 даних;
- 2 індексні і вказівники.

Старші і молодші вісім розрядів групи реєстрів можуть бути адресовані окремо. В такому випадку вони утворюють набір з восьми 8-розрядних реєстрів. Реєстри даних можна без обмеження використовувати у більшості арифметичних і логічних операцій.

Інша група реєстрів — вказівники бази і стеку та два індексні реєстри (джерела і приймачі),— також може брати участь у більшості арифметичних і логічних операцій. Крім того, деякі команди використовують ці реєстри неявно у таких випадках :

- множення, ділення і введення–виведення слів (AX);
- множення, ділення і введення–виведення байтів, перекодування, десяткова арифметика (AL);
- множення, ділення байтів (AH);
- перекодування (BX);
- операції з рядками, цикли (CX);
- множення і ділення слів, неявне введення–виведення (DX);
- операції зі стеком (SP);
- операції з рядками (SI, DI).

У реєстрі ознак (див. рис. 4.2) використано тільки дев'ять розрядів. Шість із них слугують для реєстрації особливостей результату арифметичних і логічних операцій:

- ознака знака SF, якщо знаковий біт результату операції =1;
- ознака нуля ZF, якщо результат операції дорівнює нулю;
- ознака додаткового переносу за наявності перенесення з третього розряду AF (використовують команди десяткової арифметики);
- ознака парності PF, якщо результат містить парну кількість одиниць;
- ознака перенесення CF за наявності перенесення (у разі додавання), або позички (у разі віднімання) зі старшого розряду результату;
- ознака OVERFLOW (OF) відображає переповнення у випадку виконання операцій з числами зі знаком. (У Z80 цієї ознаки не було). Виробляє внутрішнє переривання.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				<b>O</b>	<b>D</b>	<b>I</b>	<b>T</b>	<b>S</b>	<b>Z</b>		<b>A</b>		<b>P</b>		<b>C</b>
				<b>F</b>	<b>F</b>	<b>F</b>	<b>F</b>	<b>F</b>	<b>F</b>		<b>F</b>		<b>F</b>		<b>F</b>

Рис. 4. 2. Склад регістра ознак МП 8086(88)

У регістрі ознак є ще три розряди, які можна використати для керування мікропроцесором:

- ознака DF = 1 – викликає автодекремент під час виконання операцій з рядками даних. Це означає, що опрацювання рядків відбувається від старших адрес до молодших (справа наліво). DF = 0 викликає автоінкремент, тобто опрацювання рядків зліва направо.
- ознака IF = 1 — дозвіл на переривання. Мікропроцесор реагує на зовнішні масковані запити переривань.

IF = 0 — забороняє переривання. У цьому разі IF не впливає на внутрішні переривання і немасковані зовнішні;

- ознака TF = 1 — переводить мікропроцесор у покроковий режим роботи. У цьому режимі після кожної команди мікропроцесор автоматично генерує переривання.

**Зауваження:** ознаку перенесення CF використовують команди додавання і віднімання багатобайтових чисел. Команди циклічного зсуву можуть ізолювати розряд, що зсувається у пам'яті чи регістрі, шляхом розміщення його у розряді ознаки перенесення.

#### **4.2. Про переривання роботи мікропроцесора**

*Переривання* — це тимчасове припинення виконання поточної програми, яке відбувається апаратно. Після цього повинна настати реакція на подію, тобто опрацювання переривання.

Використовують переривання з такою метою:

- 1) для збільшення ефективності роботи центрального процесора;
- 2) для доступу до апаратних засобів і програм, які контролює операційна система;
- 3) для виконання операцій введення – виведення незалежно від процесора (унаслідок різної швидкості виконання).

Розрізняють внутрішні і зовнішні переривання, залежно від природи їх появи, та масковані і немасковані залежно від способу реагування на їхню появу.

*Маска переривання* – засіб вибіркового подавлення переривання у тих випадках, коли їх опрацювання можна відкласти на пізніший час. Ось деякі асоціації з побутовими подіями.

*Приклад 1:* телефонна розмова і дзвінок у двері. Ваша реакція — закінчуєте розмову і відчиняєте двері. Це масковані переривання.

*Приклад 2:* черговий і сигнал пожежної тривоги. Черговий покидає всі справи негайно і реагує на сигнал. Це немасковані переривання.

Немасковані переривання МП мають найвищий пріоритет, а мікропроцесор негайно обслуговує їх. Такими перериваннями є, наприклад, у монопольному режимі персонального комп'ютера – помилка пам'яті чи збій живлення; у колективному – перевищення відведеного



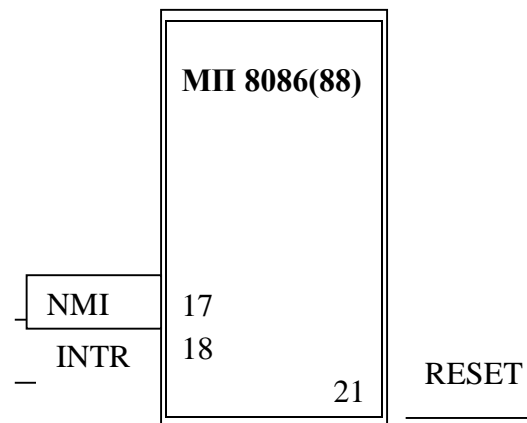
поля пам'яті, сигнал системного годинника на завершення роботи тощо.

Процесор 8086(88) автоматично перериває роботу у таких випадках (*внутрішні переривання*):

- у разі ділення на нуль;
- під час виконання програми, якщо ознака  $TF = 1$  (покроковий режим);
- у разі виконання команди з шістнадцятковим кодом  $CC$ . Це так зване переривання у заданій точці програми, що виникає під час налагодження програми.

*Зовнішні переривання*

Насамперед це переривання від пристроїв введення–виведення, наприклад клавіатури. Є три лінії переривань: RESET, NMI і INTR:



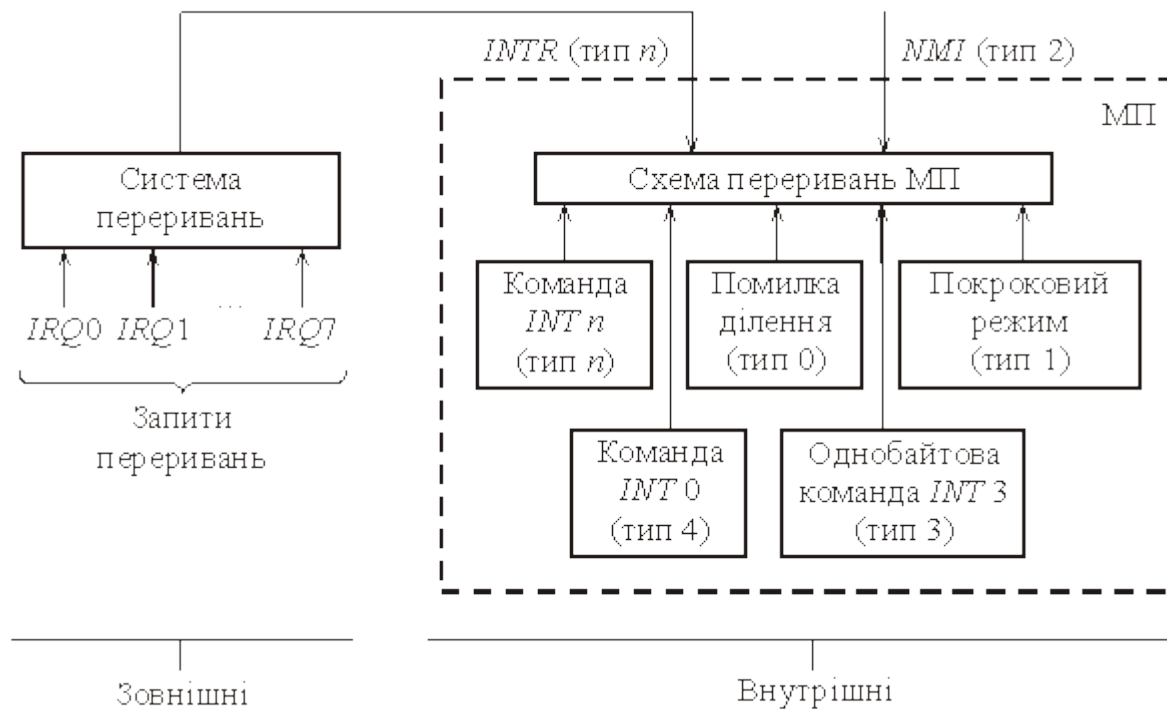


Схема опрацювання внутрішніх та зовнішніх переривань.  $IRQ$  (Interrupt ReQuest)- лінії апаратних переривань, напр., 1 – клавіатура, 3 - Com Port 1 (Мишка), 9 – звукова або мережева карта і т.д.

У разі появи сигналу в лінії RESET центральний процесор виконує такі дії:

- встановлює ознаку  $IF = 0$ . Це унеможливорює виконання маскованих і покрокових переривань;
- занулює вказівник команд  $IP$ ;
- засилає шістнадцятковий код FFFF у регістр сегмента команд.

Центральний процесор починає роботу з комірки FFFF0, де міститься команда JMP, яка передає керування процедурі ініціалізації, що і запускає в роботу комп'ютер.

Для ідентифікації процедур, необхідних для обслуговування вимог переривання, слугують **вектори переривань**. Кожній зовнішній вимозі може бути поставлений у відповідність код від 0 до 255. Таблиця векторів переривань (табл. 4. 1) займає 1024 молодші байти пам'яті, тобто комірки з фізичними адресами пам'яті від 0 до 03FFh. Кожний вхід таблиці є вказівником подвійного слова, яке містить початкову адресу процедури для обслуговування переривання цього типу. Старше 16-бітове слово містить базову адресу сегмента, а молодше — адресу процедури всередині сегмента.

Починаючи з процесора [80286](#), адреса у фізичній пам'яті й розмір таблиці переривань визначається 48-бітовим регістром IDTR.

Таблиця 4.1. Таблиця векторів переривань для МП І8086.

Адреса пам'яті	16-розрядний вказівник	Номер вектора переривання
033Fh	вказівник типу 255	255
03FCh	вказівник типу 254	254
.....	.....	.....
000Hh	вказівник типу 4	4 (переповнення)
000Ch	вказівник типу 3	3 (один байт містить CCh)
0008h	вказівник типу 2	2 (немасковане переривання)
0004h	вказівник типу 1	1 (покроковий режим)
0000h	вказівник типу 0	0 (помилка ділення)

*Масковані переривання* це зовнішні переривання, що надходять по лінії INTR. Команда завжди виконується до кінця і тільки після цього починається опрацювання запиту на переривання.

Якщо ознака переривання  $IF=0$  (переривання неможливе), то МП ігнорує запит. Якщо ознака дорівнює 1 (переривання дозволене), то МП підтверджує вимогу переривання і передає керування тій процедурі, яка повинна обслужити вимогу.

Для цього МП виконує вісім таких операцій:

- 1) генерує сигнал підтвердження зовнішнього переривання. Цей сигнал повідомляє зовнішній пристрій, що його вимога прийнята;
- 2) зчитує код переривання, який надходить на інформаційну шину від зовнішнього пристрою;
- 3) вміст регістра ознак записує в комірку пам'яті, адреса якої зберігається у поточній вершині стеку, яка визначається регістрами SS і SP;
- 4) занулює ознаку переривань (це запобігає появі нового переривання);
- 5) занулює ознаку пастки TF (це робить неможливим покроковий режим);
- 6) у стек завантажується вміст вказівника команд IP;
- 7) у вказівник команд IP засилається 16-бітове слово, яке міститься за такою адресою (фізичною):  
(тип переривання \*4) і (тип переривання \*4)+1;
- 8) у регістр сегмента команд CS засилається 16-бітове слово, яке міститься у комірках пам'яті з фізичними адресами:  
(тип переривання \*4)+2 і (тип переривання \*4)+3.

Після виконання двох останніх операцій керування передається процедурі обслуговування переривань, яка містить машинні команди, потрібні для задоволення маскованого переривання.

*Немасковані переривання.* Ці переривання надходять у МП по лінії NMI (від'єднання живлення, збій пам'яті та ін.). МП у відповідь на сигнал NMI виконує таку послідовність операцій:

- вміст регістра ознак записує в комірку пам'яті, адреса якої зберігається у поточній вершині стеку, яка визначається регістрами SS і SP;
- занулює ознаку переривань (це запобігає появі нового переривання);
- занулює ознаку пастки TF (це робить неможливим покроковий режим);
- у стек завантажується вміст регістра сегмента команд CS;
- у стек завантажується вміст вказівника команд IP;
- 16-бітове слово з комірки з фізичною адресою 00008h записує у регістр IP;
- 16-бітове слово з комірки з фізичною адресою 0000Ah записує у регістр CS.

Після цього, як і для маскованих переривань, виконується процедура обслуговування.

### **4.3. Організація пам'яті та адресування**

Як зазначено, 8086(88) має пам'ять 1 Мб. Для того, щоб виконати адресування для такої кількості комірок, необхідно мати 20-розрядний вказівник. Оскільки вказівник команд IP (як і інші регістри МП) має 16 розрядів, то МП не має прямого доступу відразу до всієї пам'яті. Це завдання вирішують за допомогою **сегментування пам'яті**, тобто поділу її на частини.

#### **4.3.1. Організація сегмента**

Поділ виконано так, що кожний сегмент містить до 64 Кбайт (65 536 байт). Початкова адреса може бути задана прикладною програмою і завжди повинна починатися з 16-байтових меж. Нагадаємо, що обмін інформацією між МП і пам'яттю відбувається за допомогою фізичних адрес.

Програма ж частіше використовує логічні адреси, що дає змогу записувати команди без попереднього визначення місця, де ця команда буде розташована у пам'яті. Логічна адреса має дві складові: *базу сегмента* і *значення зміщення* всередині сегмента. Як тільки блок спраження

звертається до пам'яті, СМА сформує фізичну адресу за правилом: значення бази сегмента зміщується на чотири розряди ліворуч і отримане 20-розрядне число додається до зміщення.

Базові адреси сегмента отримують діленням фізичної адреси початкової комірки сегмента на 16. Ці адреси містяться у сегментних регістрах (отже, МП 8086 може одночасно обслуговувати до чотирьох сегментів). На розташування сегмента накладається тільки одне обмеження: він повинен бути на межі шістнадцяти байт (тобто фізична адреса початкової комірки повинна ділитися на 16).

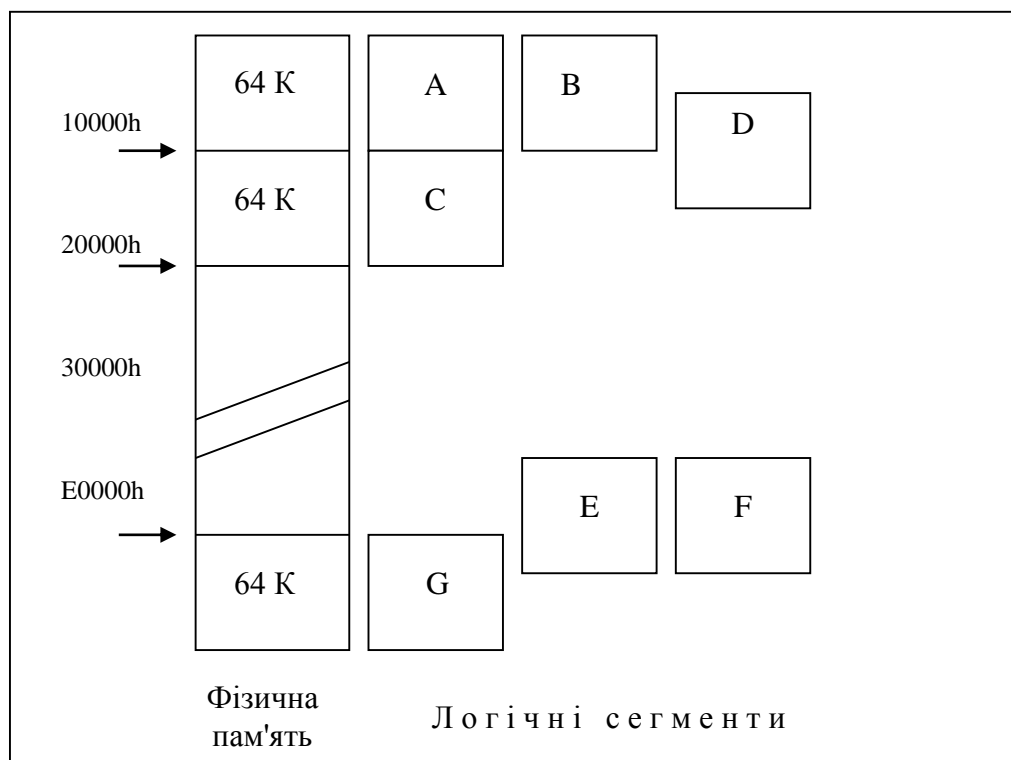


Рис. 4.3. Приклад розташування сегментів пам'яті.

Сегменти можуть бути суміжними (A,C), розділеними (C,E), перекриватися частково (B,D; E,G) або повністю (A,B чи E,F).

Якщо сегмент певним чином визначений, то регістри-вказівники можуть бути використані для адресного звертання до будь-якого байта чи слова в межах цього сегмента. Отже, МП має доступ тільки до 64 Кбайт пам'яті, а адресування до 1 Мбайта можливе тільки з застосуванням регістрів сегментів. Це один із недоліків МП сім'ї 8086 (88). Наприклад, МП Motorola MC 68000 (містить 32 розрядні регістри-вказівники) може безпосередньо адресувати до 16 Мбайт пам'яті.

Як зазначено вище, МП має чотири регістри сегментів. Отже, одночасно виділяють чотири адресовані сегменти :

командний	CS;
даних	DS;
додатковий	ES;
стековий	SS.

Командний сегмент містить машинні команди, які вибирає пристрій зв'язку, а виконує виконавчий пристрій.

Сегмент даних використовують для запам'ятовування програм, змінних і сталих.

Додатковий сегмент забезпечує додаткові 64 Кбайти робочої пам'яті.

Стековий сегмент містить стекову структуру ємністю до 64 Кбайти.

Якщо у всі регістри сегментів завантажити одну й ту ж базову адресу, то всі чотири сегменти відображатимуть одну й ту ж область фізичної пам'яті.

Сегментація ефективна для модульної побудови програмного забезпечення і структурного програмування, коли великі програми розбивають на дрібні модулі.

#### 4.3.2. Режими адресування

Пам'ять процесора можна уявити собі як послідовність байтів, наймолодший з яких має адресу 00000h, а найстарший – FFFFFh. Для адресування слова і подвійного слова діють такі домовленості:

- 1) 16-бітове слово уводиться у пам'ять так, що старший байт міститься у комірці з більшим номером;
- 2) 32-бітове подвійне слово уводиться складніше, а власне – з використанням його як вказівника для адресування байта чи слова у будь-якому сегменті. Молодше слово адреси означає адресу переміщення, а старше – початкову адресу сегмента. Два байти кожного слова уводяться за правилом запису 16-розрядного слова.

Процесор має досить гнучку систему адресування, що дає змогу ефективно використовувати різні форми задання операндів у командах програми. Система адресування забезпечує використання регістрових та безпосередніх операндів, пряме та непряме адресування, базово-індексне адресування зі зміщенням і без нього, адресування рядків даних, адресування портів введення-виведення.

*Регістрові операнди* – це приклад найефективнішого виконання команд процесора, тому що операції з операндами відбуваються без звертання до пам'яті. Джерелом, приймачем, або і тим, і іншим можуть бути регістри загального призначення. Сегментні регістри можуть бути або тільки джерелом, або тільки приймачем операндів.

Наприклад, команда (у дужках записано машинний код команди)

**AND AL, DL (20D0)**

виконає операцію логічного множення вмісту регістрів AL і DL, а результат помістить у регістр AL. Процесор виконає її за три такти.



*Безпосередні операнди* – визначають, як частину самої команди. Дані можуть бути 8- або 16-бітовими. Приймачем найліпше слугує регістр AL (AX). Наприклад, нам потрібно порівняти вміст регістра AL зі сталою 01. Машинна команда у цьому випадку має вигляд

**CMP AL, 01 (3C01),**

і процесор виконає її за чотири такти.

Перш ніж перейти до ознайомлення з іншими режимами адресування пам'яті процесора, з'ясуємо поняття виконавчої адреси.

**Виконавчою адресою** (ВА) операнда називають адресу всередині сегмента, яку процесор обчислює за певними правилами. ВА є переміщуваним 16-розрядним числом без знака, що забезпечує доступ до кожного байта сегмента. Щоб зрозуміти процес формування виконавчої адреси, необхідно розглянути структуру команди процесора:

Код операції	Mod	Reg	R/M	З м і щ е н н я	Безпосередній операнд
XX YY Y ZZZ					

Перед полем коду операції можлива наявність префіксного поля. Поле Mod (від *Mode* – режим) потрібне для того, щоб відрізнити, чи відповідають операнди команд регістрам ЦП або коміркам пам'яті. Величина XX має чотири можливі двійкові значення: 00, 01, 10, 11. Двійкове значення 11 у полі Mod свідчить про те, що операндами машинних команд є регістрові операнди. Двійкові значення 00, 01 і 10 задають типи адресації відповідно з нульовим, одно- або двобайтовим зміщенням. Зміщення може бути 8-бітовим числом зі знаком або 16-бітовим числом без знака, яке входить у машинну команду як її частина і яке використовують у разі обчислення виконавчої адреси.

Позначення YY Y відповідає трьом бітам, які утворюють поле Reg (від *Register* – регістр), його застосовують для визначення регістра, у якому міститься операнд. Поле Reg ідентифікує

8-бітовий регістр під час роботи зі словами. В деяких командах ці біти використовують також для ідентифікації групи, до якої належить машинна команда (наприклад, команда AND у групі логічних команд).

Три біти ZZZ формують поле Rm (від *Register/Memory* – регістр/пам'ять). Разом з полем Mod його застосовують для ідентифікації певного типу адресації. В команді допустима будь-яка комбінація в полях Mod і Rm, що забезпечує багатоманітність режимів адресації пам'яті в сім'ї мікропроцесорів 8086/8088.

*Пряме адресування* є найпростішим типом адресації – воно не використовує ніяких регістрів. Виконавчу адресу беруть безпосередньо з 16-бітового поля зміщення машинної команди. Ця пряма адреса однозначно визначає байт або слово пам'яті, які розміщені всередині сегмента.

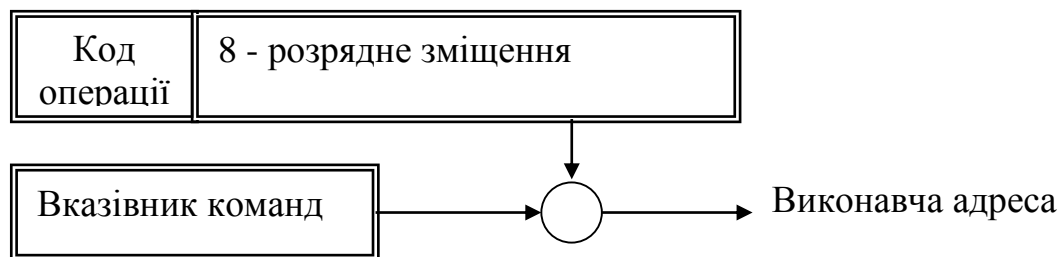


Пряму адресацію переважно використовують для роботи з простими змінними або сталими. В процесорі 8086(88) повинна виконуватись умова, згідно з якою пряма адреса не виходить за межі сегмента даних. Оскільки в програмах пряму адресу можна визначати стосовно будь-якого з чотирьох доступних сегментів, то для його модифікації можливе застосування префіксних команд. У разі використання прямої адресації в полі Mod міститься число 00, а в полі Rm – число 110.

Для прямої адресації використовують два спеціальні види адресації: відносну та абсолютну.

У випадку відносної адресації поле зміщення відображене 8-бітовим числом зі знаком.

Виконавча адреса в цьому разі визначена сумуванням вмісту поля зміщення і регістра вказівника команд IP :



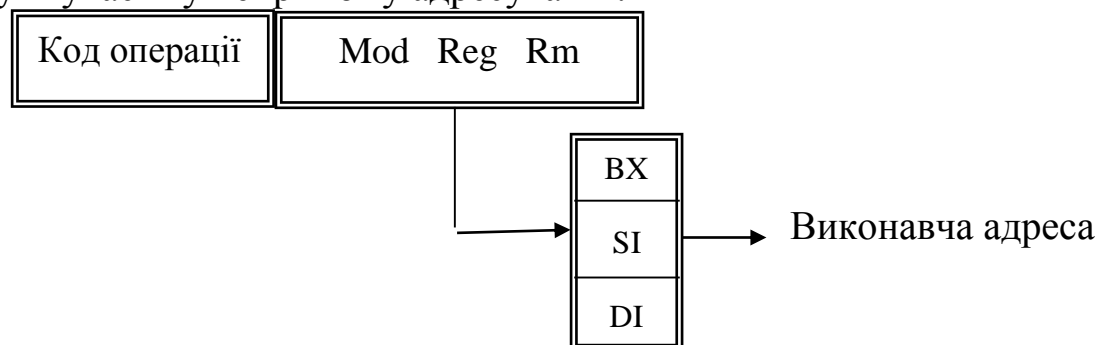
Відносну адресацію використовують для команд умовних переходів, таких як JE (перехід по рівності), JO (перехід по переповненню). У разі використання відносної адресації байт адресації (Mod/Reg/Rm) не потрібний.

У разі абсолютної адресації частина команди є 32-бітовим вказівником, який визначає фізичну адресу у пам'яті МП 8086(88). Молодше слово вказівника потрактоване в цьому випадку як переміщувана адреса сегмента, базовою адресою якого є старше слово вказівника:



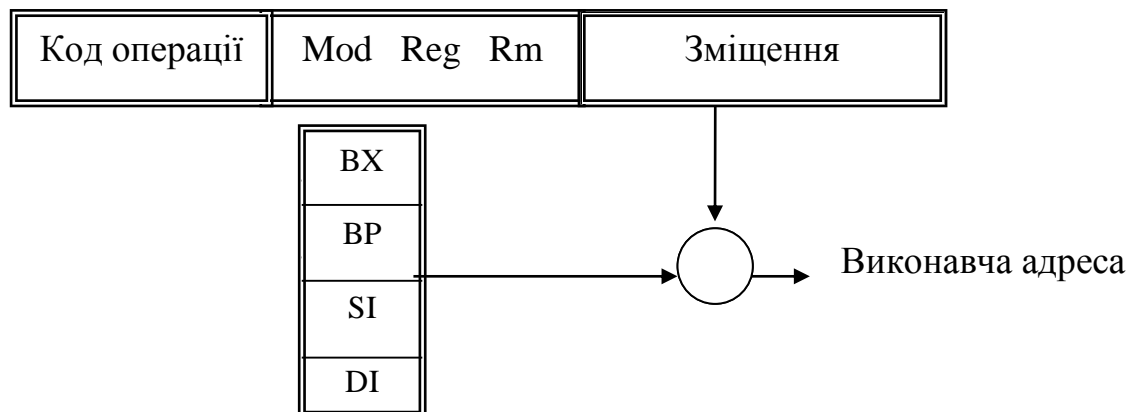
Для абсолютної адресації наявність байта адресації (Mod/Reg/Rm) також не потрібна.

*Непряме адресування.* За такої адресації виконавчою адресою є вміст базового або індексних регістрів (BX, SI або DI). Завдяки цьому одна і та ж команда може звертатися до кількох різних ділянок пам'яті через просту зміну вмісту базового або індексного регістрів, які беруть участь у непрямому адресуванні.



У разі непрямого адресування є один виняток: для команд JMP (перехід) і CALL (виклик процедур) виконавчою адресою може бути вміст будь-якого з 16-бітових регістрів загального призначення (AX, BX, CX, DX, SI, DI, BP, SP).

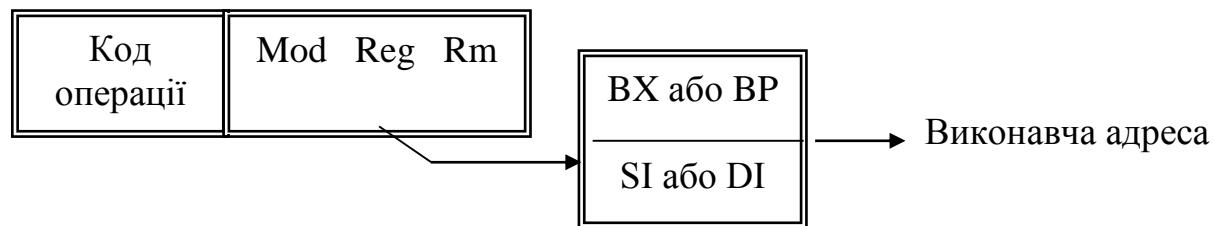
*Непряме адресування зі зміщенням.* У випадку такої адресації ВА визначена як сума зміщення і вмісту одного з регістрів – базового або індексного (BX, BP, SI, DI). Зміщення може бути 8-бітовим числом зі знаком або 16-бітовим без знака. У разі використання регістрів BX, SI за виконавчу адресу приймають переміщену адресу поточного сегмента даних, базова адреса якої визначена регістром сегмента даних DS. Якщо ж використовують регістр BP, то за виконавчу адресу приймають переміщену адресу поточного стекового сегмента, базова адреса якого міститься в регістрі сегмента стеку SS:



Оскільки в прикладних програмах можуть бути використані префіксні команди, то є змога переприсвоєння виконавчих адрес щодо інших сегментів пам'яті.

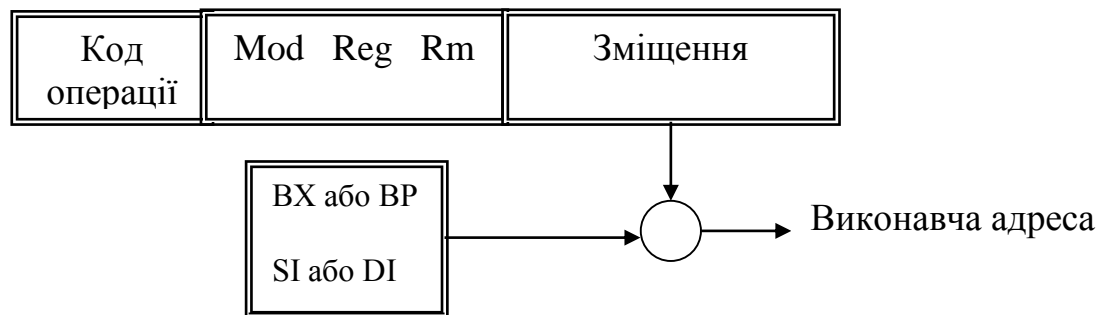
*Базово-індексне адресування.* За такого адресування виконавча адреса визначена як сума вмісту базового (BX або BP) та індексного реєстрів (SI або DI). Завдяки можливості зміни під час виконання програми вмісту базового та індексного реєстрів базово-індексне адресування є дуже гнучким засобом доступу до різних ділянок комп'ютерної пам'яті. У разі використання реєстра BX виконавча адреса визначена щодо сегмента даних з базовою адресою, яка міститься в реєстрі DS, а у випадку застосування реєстра BP – щодо сегмента стека, базова адреса якого міститься в реєстрі SS.

Прикладні програми можуть мати префіксні команди, які дають змогу переприсвоювати виконавчу адресу щодо інших сегментів пам'яті комп'ютера.



У випадку базово індексного адресування поле Mod визначене числом 00, поле Rm – двійковим числом 000, 001, 010 чи 011, відповідно до того, яке адресування використано: [BX+SI], [BX+DI], [BP+SI] чи [BP+DI].

*Базово-індексне адресування зі зміщенням.* У разі такого адресування виконавча адреса визначена як сума вмісту базового реєстра (BX чи BP), індексного реєстра (SI чи DI) і розміру зміщення. Як зазначено, розмір зміщення є частиною команди і може бути 8-бітовим числом зі знаком чи 16-бітовим без знака. За виконавчу адресу приймаємо переміщувану адресу поточного сегмента даних чи стекового сегмента залежно від того, який з реєстрів використовуємо – BX чи BP:



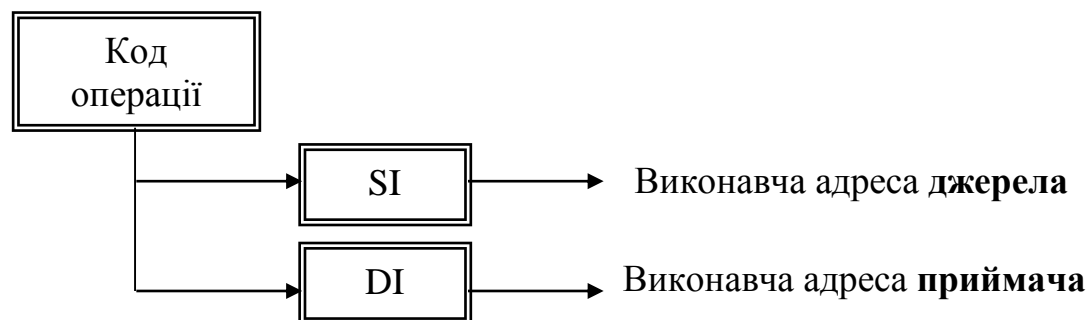
Прикладні програми можуть мати префіксні команди, завдяки яким можливо переприсвоювати виконавчі адреси щодо інших сегментів пам'яті.

У випадку базово-індексного адресування зі зміщенням поле Mod містить двійкове число 01 чи 10 залежно від того, як визначено зміщення: 8- чи 16-бітовим числом. Поле Rm таке саме, як і в разі базово-індексного адресування.

*Адресування рядків даних.* Рядкові команди, такі як MOVS (переслати рядок), CMPS

(порівняти рядок), LODS (завантажити рядок), STOS (записати рядок) і SCAS (сканувати рядок) не використовують жодного з розглянутих типів адресації для вибору своїх операндів. Вміст індексних реєстрів (SI і/або DI) використовують для безпосереднього визначення потрібної ділянки пам'яті. Регістр SI завжди використовують як вказівник першого байта чи слова рядка-джерела.

Регістр DI застосовують як вказівник першого байта чи слова рядка-отримувача. Команда LODS пропонує використання реєстра SI як вказівника джерела. Команди STOS і SCAS використовують реєстр DI як вказівник отримувача. Команди MOVS і CMPS застосовують обидва реєстри. Переміщувана адреса, яка є в реєстрі DI, завжди стосується поточного додаткового сегмента. Для команд робіт з рядками даних можна визначити і префіксні команди, які дають змогу переприсвоювати сегмент даних (для команд LODS, MOVS і CMPS) чи

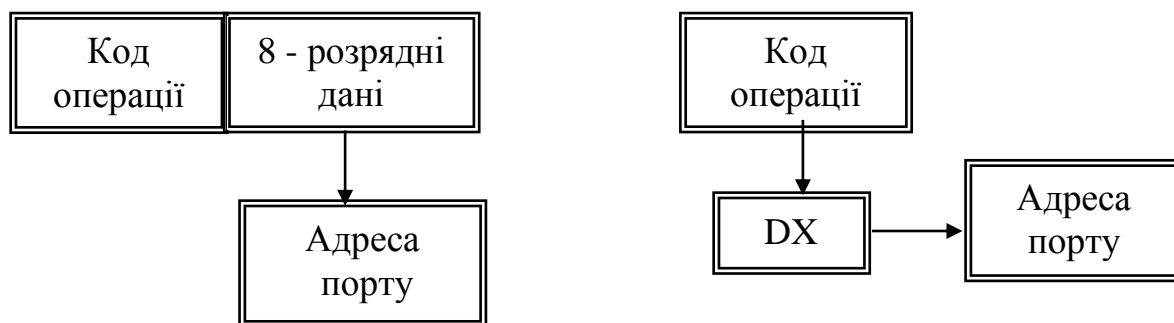


додатковий сегмент (для команди SCAS) іншому сегменту, який міститься в будь-якій частині пам'яті 8086(88):

Під час виконання рядкової команди збільшення або зменшення вмісту реєстрів SI чи DI визначене станом (нульовий чи одиничний) ознаки напряму. Залежно від того, з чим працює команда – з байтом чи з словом, – вміст індексних реєстрів збільшується/зменшується, відповідно, на 1 чи 2.

*Адресування портів введення-виведення.* Цей тип адресування забезпечує доступ до пристроїв введення-виведення (У/В), які приєднані до У/В області пам'яті 8086(88). Область У/В, як доповнення до мегабайтової пам'яті, містить ще 65 Кбайт. Ця область не є сегментованою і її можна використовувати як окремо, так і разом з усією пам'яттю 8086(88). В останньому випадку зменшується мегабайтова пам'ять на 64К, однак забезпечена додаткова гнучкість під час програмування, оскільки всі розглянуті раніше типи адресування можуть забезпечити доступ до цієї області.

Кожний байт області У/В 8086(88) може бути призначеним як адреса однозначно визначеного порта У/В. Отже область У/В може обслуговувати 64 К ( 65536) 8-бітових портів У/В. Дані можуть бути передані між накопичувачем і будь-яким портом У/В, який адресується всередині 64-кілобайтної області У/В. Регістр AL використовують для передавання байта, а регістр AX – для передавання слова. Для адресування порту У/В є команди IN (ввести з порту) і OUT (вивести в порт). Якщо номер порту перебуває в межах 0-255 (десятичних чисел) то його можна визначити частиною машинної команди; якщо номер порту перевищує 255, то для його зберігання використовують регістр DX:

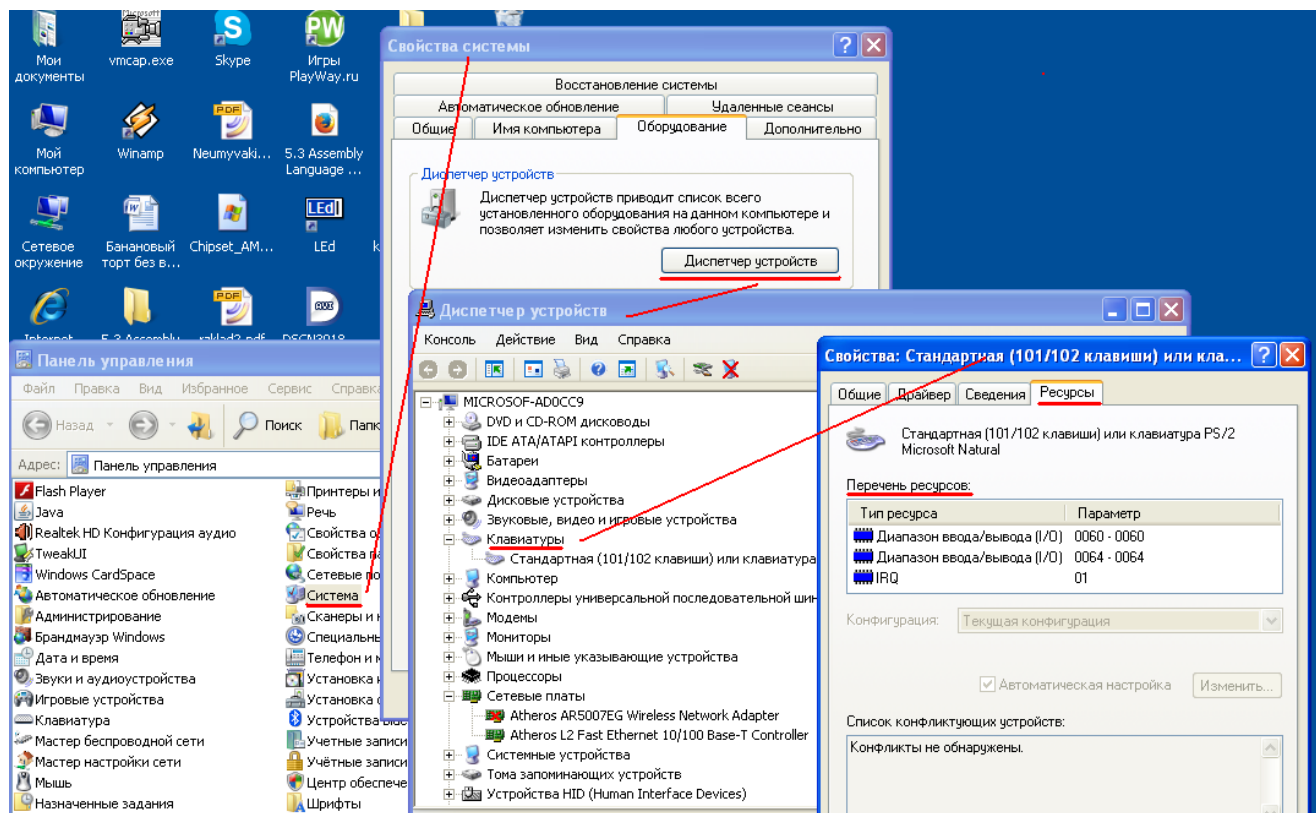




Приклад: нехай порт з адресою 3D5h є портом уведення у буферну пам'ять пристрою друку.  
Виведення у буфер коду ASCII відбуватиметься за допомогою такої послідовності команд:

MOV DX, 3D5h	завантажити адресу порту у DX
MOV AL, 1Bh	код 1B у регістр AL
OUT AL, DX	вивести код у буферну пам'ять.

Приклад відображення адрес портів уведення-виведення для клавіатури ПК:

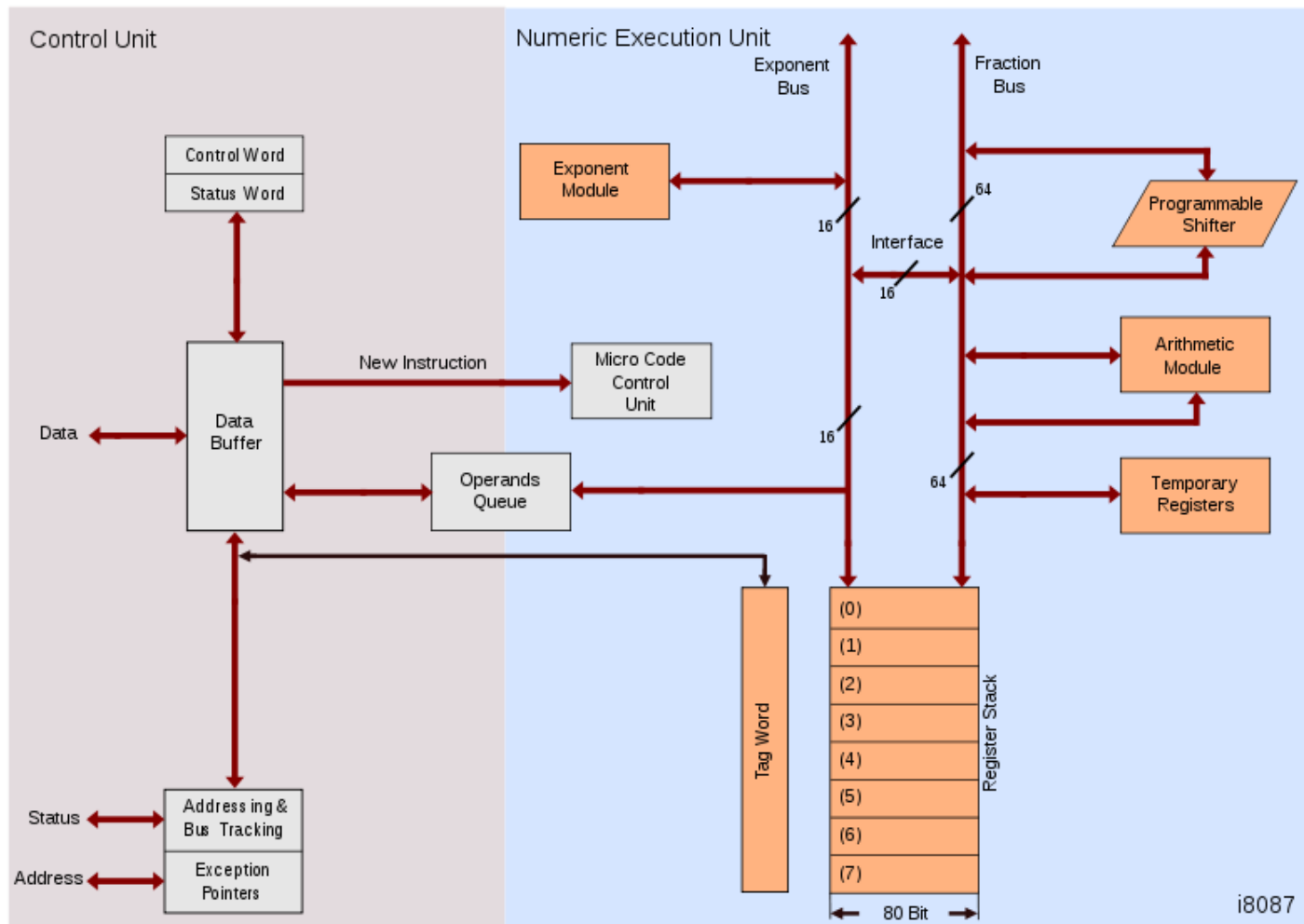


*Стекова пам'ять.* **Стек** – це область пам'яті, спеціально виділена для тимчасового зберігання параметрів чи програмних даних, які необхідні для інформаційного зв'язку програм і процедур. Запис і читання даних у стеку організовані за принципом "першим прийшов – останнім пішов". У стек може бути завантажено в певній послідовності низку даних, які потім вибирають (зчитують) зі стеку, проте вже в зворотній послідовності. Кількість стеків у пам'яті ЕОМ обмежена виділеною областю пам'яті. Максимальна ємність стекової структури становить 32К (32768) 16-бітових слів. Отже стек може повністю розміститися в одному сегменті пам'яті. (У захищеному режимі можливий розмір стека – 4Гб).

Оскільки є тільки один регістр сегмента стеку SS, який зберігає початкову (базову) адресу стеку, то в будь-який момент часу можна звертатися тільки до одного стеку. Вказівник стеку SP використовують для зберігання адреси останнього члена послідовності (вершини стеку), який записано в стек.

#### **4.4. Математичний співпроцесор 8087**

Офіційна назва (фірма Intel) - NPX (*Numeric Processor extension*). Призначення: розширення обчислювальних можливостей МП 8086/8088, 80186/80188. Системою команд 8086 додатково передбачено 68 мнемонічних команд (арифметичних, тригонометричних, експоненціальних, логарифмічних) і підтримка шести типів даних з плаваючою комою і 18-розрядні числа у двійково-десятковому форматі. NPX виконує операції апаратно, причому продуктивність більше ніж у сто разів перевищує програмну емуляцію команд ЦП.



Співпроцесор (СП) містить свій блок регістрів даних, регістр керування, групу регістрів стану і вказівників. Регістри даних (R0-R7) мають розрядність 80 біт і організовані у стек. Внутрішня шина має 84 розряди.

Іноді співпроцесор позначають як FPU (*Floating Point Unit*), що не охоплює інші типи даних, або NPU (*Numeric Processing Unit*), що не відображає його принципову різницю від ЦП.

Одноименні виводи ЦП і СП сполучені між собою. СП, відстежуючи сигнали стану ЦП, переглядає і декодує інструкції. Адресування виконує ЦП. Якщо команда передбачає читати більше ніж одне слово, то СП викликає керування локальною шиною і продовжує читання операндів, збільшивши адресу.

#### **4.5. Інші 16-розрядні процесори**

Наступні після 8086(88) процесори i80186/80188 та їхні модифікації 80C186/80C188 не є новим поколінням архітектури. Вони містять умонтовані периферійні контролери переривань, прямого доступу до пам'яті, триканальний таймер і генератор синхронізації. Вмонтована периферія має програмний інтерфейс, який не сумісний з IBM PC-специфікаціями. Ці процесори використовують у вмонтовуваних контролерах і комп'ютерах, які не потребують 100% IBM сумісності. Сумісність на рівні MS DOS забезпечена.

**Процесор Intel 80286.** Випущений у 1982 р., друге покоління 16-розрядних процесорів. Найсуттєвіша відмінність від 8086 – це наявність нового механізму керування адресуванням пам'яті. Має спеціальні засоби для підтримання механізму перемикання задач. Розширена система команд. МП працює у двох режимах:

- *8086 Real Address Mode* – режим реального адресування, повністю сумісний з 8086 (1 Мбайт);
- *Protected Virtual Address Mode* – захищений режим віртуального адресування (16 Мбайт). Використання сторінкового механізму адресування дає змогу заадресувати 1 Гбайт віртуальної пам'яті кожної задачі.

Призначення регістрів головно збігаються з 8086. Різниця є у регістрі ознак: біти з 0 до 11

збігаються з 8086, а додано у 80286: (12,13) IOPL (Input Output Privilege Level) - рівень привілейованості уведення/виведення, (14) NT (Nested Task Flag) – ознака вкладення задач.

Збільшено тактову частоту.

Під керуванням MS DOS 80286 переважно використовують у реальному режимі, тобто він працює як швидкий 8086. Захищений режим використовує ОС типу XENIX, UNIX, OS/2, Netware 286 і MS Windows.

З'являються поняття сторінки, селектора та дескриптора сегментів. Усі ці поняття властиві і 32-розрядним процесорам.

#### **4.6. Особливості архітектури x86**

**x86** є [CISC](#) архітектурою із змінною довжиною команд. Сумісність нових моделей процесорів із попередніми була ключовою у розвитку архітектури x86. Архітектура двічі розширювалась до більшого розміру [машинного слова](#). Перші x86 процесори були 16-бітними. У [1985 Intel](#) випустила 32-бітний процесор [80386](#), його архітектуру назвали x86-32 або [IA-32](#) (Intel Architecture, 32 bits). У 2003р. [AMD](#) випустила [Athlon 64](#), в якому реалізовано розширення архітектури до 64 біт, яке називають AMD64 (AMD), EM64T (Intel), чи x64 ([Microsoft](#)). Нова архітектура [x64](#) замінила **x86** і починаючи з 2003 року стала новим індустріальним стандартом при виробництві усіх нових процесорів (за винятком процесорів типу [ARM](#), що зараз широко використовуються в [мобільних телефонах](#), [смартфонах](#) та [нетбуках](#)).

Дозволено невіривняний по машинному слову доступ до пам'яті. Порядок байтів у слові — [little-endian](#) (найменш значущий байт має меншу адресу). Сучасні x86 процесори перекладають набір команд x86 на [RISC](#)-подібні мікрокоманди для ефективного їх виконання.

В таблиці нижче наведені бренди процесорів, які згруповані за поколіннями і є важливими в історії x86.

Покоління	Вперше представлений	Відомі бренди процесорів	Лінійна/фізична адреси простору
1	1978	<a href="#">Intel 8086</a> , <a href="#">Intel 8088</a> та клони <a href="#">Intel 80186</a> , <a href="#">Intel 80188</a> та клони, <a href="#">NEC V20/V30</a>	Перші x86 мікропроцесори
2	1982	<a href="#">Intel 80286</a> та клони	<b>16-bit</b> (30-bit віртуальні) / 24-bit (сегментовані)
3 ( <a href="#">IA-32</a> )	1985	<a href="#">Intel 80386</a> та клони, <a href="#">AMD Am386</a>	
4 ( <a href="#">FPU</a> )	1989	<a href="#">Intel486</a> та клони, <a href="#">AMD Am486/Am5x86</a>	
4/5	1997	<a href="#">IDT/Centaur-C6</a> , <a href="#">Cyrix III</a> -Samuel, <a href="#">VIA C3</a> -Samuel2 / <a href="#">VIA C3-Ezra</a> (2001), <a href="#">VIA C7</a> (2005)	<b>32-bit</b> (46-bit віртуальні) / 32-bit
5	1993	<a href="#">Pentium</a> , <a href="#">Pentium MMX</a> , <a href="#">Cyrix 5x86</a> , <a href="#">Rise mP6</a>	
5/6	1996	<a href="#">AMD K5</a> , <a href="#">Nx586</a> (1994)	
	1995	<a href="#">Pentium Pro</a> , <a href="#">Cyrix 6x86</a> , <a href="#">Cyrix MII</a> , <a href="#">Cyrix III</a> -Joshua (2000)	
6	1997	<a href="#">AMD K6/-2/3</a> , <a href="#">Pentium II/III</a>	
	2003	<a href="#">Pentium M</a> , <a href="#">Intel Core</a> (2006)	<b>36-bit</b> фізична ( <a href="#">PAE</a> )
7	1999	<a href="#">Athlon</a> , <a href="#">Athlon XP</a>	
	2000	<a href="#">Pentium 4</a>	
	2000	<a href="#">Transmeta Crusoe</a> , <a href="#">Efficeon</a>	
7/8	2004	Pentium 4 <a href="#">Prescott</a>	
	2006	<a href="#">Intel Core 2</a>	
	2008	<a href="#">VIA Nano</a>	
8 ( <a href="#">x86-64</a> )	2003	<a href="#">Athlon 64</a> , <a href="#">Opteron</a>	
8/9	2007	AMD <a href="#">Phenom</a>	48-bit фізична для AMD Phenom

	2008	<a href="#">Intel Core i3/i5/i7</a> , <a href="#">AMD Phenom II</a> <a href="#">Intel Atom</a>
	2011	AMD <a href="#">Bobcat</a> , <a href="#">Llano</a>
9 ( <a href="#">GPU</a> )	2011	Intel <a href="#">Sandy Bridge/Ivy Bridge</a> , AMD <a href="#">Bulldozer and Trinity</a>
	2013	Intel <a href="#">Larrabee</a>
	2014	Intel <a href="#">Haswell</a>

### Список літератури до теми

1. Дао Л. Программирование микропроцессора 8088/ Пер. с англ. – М.,1988.
2. Мячев А.А., Степанов В.Н. Персональные ЭВМ и микроЭВМ. Основы организации: Справочник. – М., 1991.
3. Каган Б.М. Электронные вычислительные машины и системы: Учеб. пособие для вузов. – М., 1991.
4. Григорьев В.Л. Микропроцессор i486. Архитектура и программирование. Кн. 1. – М., 1993.
5. Гук М. Процессоры Intel: от 8086 до PentiumII. – СПб., 1997.
6. Злобін Г.Г, Рикалюк Р.Є. Архітектура та апаратне забезпечення ПЕОМ: Навч.посіб. –К., 2006., -К.,2012.

### **Контрольні запитання до теми**

1. Які головні функції пристрою спряження з магістраллю?
2. З яких компонент складається пристрій опрацювання і які функції він виконує?
3. Що таке переривання і з якою метою їх застосовують?
4. Яка відмінність між перериваннями внутрішніми і зовнішніми?
5. Які переривання називають маскованими, а які немаскованими?
6. Який алгоритм опрацювання переривань?
7. Що називаємо виконавчою адресою?
8. Скільки і які компоненти використовують для формування виконавчої адреси?
9. Яка довжина сегмента у 16-розрядному процесорі?
10. Назвіть чотири типи сегментів.
11. Скільки режимів адресування є у 16-розрядному процесорі?
12. Що називаємо портом введення-виведення?
13. Скільки 8-бітових портів може мати 16-розрядний процесор?
14. Яку функцію виконує співпроцесор?
15. Які відмінності між процесорами 8086 і 80286?