МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

«МІКРОПРОЦЕСОРНА ТЕХНІКА»

КОНСПЕКТ ЛЕКЦІЙ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальністю 171 «Електроніка»,
спеціалізацією «Електронні компоненти і системи»*

Київ
КПІ ім. Ігоря Сікорського
2017

«Мікропроцесорна техніка»: конспект лекцій [Електронний ресурс]: навч. посіб. для студ. спеціальності 171 «Електроніка», спеціалізації «Електронні пристрої і системи» / КПІ ім. Ігоря Сікорського ; уклад.: Т. О. Терещенко, О.В. Хоменко – Електронні текстові данні (1 файл: 3880 кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2017. – 165с.

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № від р.)
за поданням Вченої ради факультету електроніки (протокол № 12/2017 від .12.2017 р.)

Електронне мережне навчальне видання

«МІКРОПРОЦЕСОРНА ТЕХНІКА»

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Укладачі: *Терещенко Тетяна Олександрівна,* докт. техн. наук.
Хоменко Олександр Васильович

Відповідальний редактор: *Ямненко Ю. С.,* завідувач кафедри промислової електроніки, д-р техн. наук, проф.

Рецензенти: *Михайлов С.Р.,* доцент кафедри електронних приладів та пристроїв, канд. техн. наук, доц.

Метою посібника «Мікропроцесорна техніка: конспект лекцій» є висвітлення основних принципів побудови мікропроцесорів та сприяння набуттю практичних навичок побудови апаратурної частини та програмного забезпечення мікропроцесорних систем та методів їх налагодження.

ЗМІСТ

ВСТУП

Курс “Мікропроцесорна техніка” – є складовою частиною дисциплін, які включені до переліку обов’язкових дисциплін бакалаврської підготовки з електронній техніці.

Під час вивчення курсу студенти набувають знань про архітектуру, принципи побудови, функціонування та програмування мікропроцесорів і мікроконтролерів.

Курс базується на наступних дисциплінах:

- Програмування;
- Інформаційні технології;
- Цифрові інформаційні системи;
- Пристрої цифрової електроніки;

Курс “ Мікропроцесорна техніка ” є базовим для.
наступних дисциплін:

- Мікропроцесорні пристрої (бакалаври)
- Пристрої перетворювальної техніки (бакалаври);
- Пристрої відображення та реєстрації інформації (магістри);
- Силові електронні системи (магістри);
- Системи електроживлення електронної апаратури (магістри);
- Мікропроцесорні системи (магістри);
- Електронні системи керування та регулювання (магістри).

Розділ 1. ПРИНЦИПИ ПОБУДОВИ МІКРОПРОЦЕСОРНИХ СИСТЕМ

ЛЕКЦІЯ 1

Основні поняття мікропроцесорної техніки

Мікропроцесор (МП) - це пристрій, який здійснює прийом, обробку і видачу інформації. Конструктивно МП містить одну чи кілька інтегральних схем та виконує дії за програмою, яка записана у пам'яті.

Мікропроцесорна система (МПС) - обчислювальна, контрольно-вимірювальна або керуюча система, в якій основним пристроєм обробки інформації є МП. Мікропроцесорна система будується з набору мікропроцесорних ВІС.

Мультимікропроцесорна (або мультипроцесорна) система - система, яка утворюється шляхом об'єднання деякої кількості універсальних або спеціалізованих МП, завдяки чому забезпечується паралельна обробка інформації і розподілене керування.

Мікропроцесорний комплект (МПК) - сукупність інтегральних схем, сумісних за електричними, інформаційними та конструктивними параметрами і призначених для побудови електронно-обчислювальної апаратури і мікропроцесорних систем керування. До типового складу МПК відносяться: 1) ВІС МП (один чи кілька корпусів інтегральних схем); 2) ВІС оперативних запам'ятовувальних пристроїв (ОЗП); 3) ВІС постійних запам'ятовувальних пристроїв (ПЗП); 4) інтерфейси або контролери зовнішніх пристроїв; 5) службові ВІС - тактовий генератор, регістри, шинні формувачі, контролери шин, арбітри шин.

Двійкова система числення

Двійкова система числення, або система з основою 2, використовує цифри 0 і 1. Ці цифри називаються ***бітами*** (**Bi**nary Digi**t**s). Фізично в цифрових електронних системах значення 0 відповідає напрузі низького рівня (L-рівня), а значення 1 - напрузі високого рівня (H-рівня).

У табл. 1.1 наведені значення ваг перших чотирьох двійкових позицій та показана відповідність між двійковим числом 1001_2 та його десятковим еквівалентом 9_{10}. Розряд, якому відповідає значення ваги позиції 1, називається ***молодшим бітом*** (МБ), а розряд, якому відповідає найбільше значення ваги позиції (в даному випадку 8), називається ***старшим бітом*** (СБ).

Таблиця 1.1. Значення позицій двійкових чисел

Степінь основи	3	2	1	0
Значення ваг позицій	8	4	2	1
	СБ			**МБ**
Двійкове число	1	0	0	1
Десяткове число	8 +	0 +	0 +	$1 = 9_{10}$

Перетворення двійкового числа на десятковий еквівалент. Під кожним бітом двійкового числа записуються десяткові значення кожної позиції. Десяткові числа підсумовуються. Приклад перетворення двійкового числа 1011 0110 на десятковий еквівалент наведений у табл. 1.2.

Таблиця 1.2. Перетворення двійкового числа на десятковий еквівалент

Степінь основи	7	6	5	4	3	2	1	0	
Значення ваг позицій	128	64	32	16	8	4	2	1	
Двійкове число	1	0	1	1	0	1	1	0	
Десяткове число	128	+ 0	+ 32	+ 16	+ 0	+ 4	+ 2	+ 0	$= 182_{10}$

Перетворення десяткового числа на двійковий еквівалент. Десяткове число ділиться на 2. Остача у вигляді 0 або 1 записується у молодший розряд двійкового числа. Частка від ділення знов ділиться на 2, остача (0 або 1) записується у наступний після молодшого розряд. Ці дії виконуються, поки частка від чергового ділення не стане рівною 1. Ця 1 записується у старший розряд двійкового числа. Приклад перетворення десяткового числа 155_{10} на двійковий еквівалент 10011011_2 наведений на рис. 1.1.

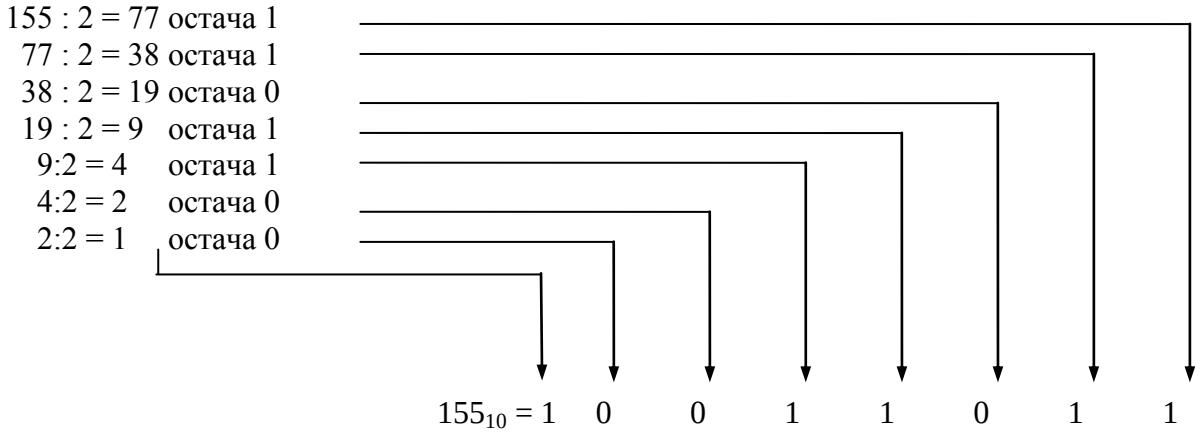


Рис.1.1. Перетворення десяткового числа 155_{10} на двійковий еквівалент

Приклад 1.1. *Перетворити десятковий дріб $0{,}366_{10}$ у двійкову систему числення з точністю 2^{-8}.*

Задане число 8 разів послідовно множимо на 2: (рис. 1.2)

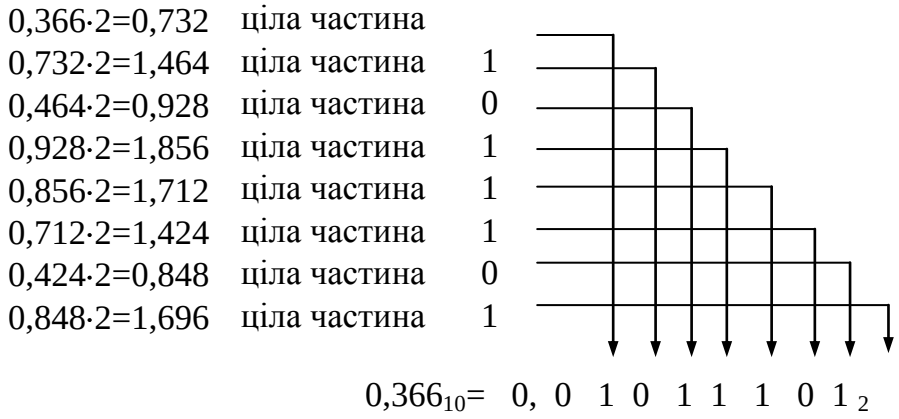


Рис.1.2. Перетворення десяткового числа $0{,}366_{10}$ на двійковий еквівалент з точністю 2^{-8}

Двійкова арифметика

Додавання, віднімання або множення двійкових чисел виконується за такими самими правилами, як і в арифметиці десяткових чисел. Правила додавання однорозрядних двійкових чисел наведені у табл. 1.3. Правила 1 та 2 очевидні; правило 3 ілюструє перенесення одиниці у старший розряд: 1+1=10. Правило 4 показує, що результатом додавання трьох 1 є число 11.

Таблиця 1.3. Правила двійкового додавання

Номер правила	1	2	3	4
1-й доданок	0	0	1	1 *Перенесення з молодшого розряду*
2-й доданок	+ 0	+ 1	+ 1	+ 1
Сума	0	1	1 0	1 1
			Перенесення 1 до старшого розряду	*Перенесення 1 до старшого розряду*

Приклад 1.2. *Додати два 8-розрядних двійкових числа: 10100010_2 і 01110101_2.*

Виконуємо додавання двох чисел, використавши правила двійкового додавання (табл. 1.7) для кожного з розрядів. У цьому прикладі результат додавання 8-розрядних чисел є 8-розрядним:

```
  1  111          - рядок перенесень
  10100011
+ 00110101
----------
  11011000
```

Правильність додавання перевіримо в десятковій системі (табл. 1.3).

Таблиця 1.3. Додавання у двійковій та десятковій системах

Значення ваг розрядів / Числа		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
1-й доданок	Двійкове	1	0	1	0	0	0	1	1	
	десяткове	128	+	32	+			2	+1	= 163_{10}
2-й доданок	двійкове	0	0	1	1	0	1	0	1	+
	десяткове			32	+16	+	4	+	1	= 53_{10}
Результат	двійкове	1	1	0	1	1	0	0	0	
	десяткове	128	+64	+	16	+8				= 216_{10}

Приклад 1.3. *Додати два двійкових числа: 10100011_2 та 01110101_2.*

Виконаємо додавання у двійковому вигляді. Результат є дев'ятирозрядним числом:

```
   111   11        – рядок перенесень
  10100011
+ 01110101
  ---------
 100011000
```

та у десятковому вигляді:

```
+  163
   117
  ----
   280
```

Якщо для позначення кожного двійкового числа відводиться вісім розрядів, то одержання дев'ятирозрядного результату додавання двох чисел призводить до некоректної роботи мікропроцесора. У прикладі 1.10 у восьми молодших розрядах результату додавання у двійковому вигляді знаходиться число $00011000_2 = 24_{10}$. При цьому відбувається перенесення 1 у дев'ятий розряд, вага якого $2^8=256$. Це перенесення можна врахувати програмно і скоректувати невірний результат, наприклад, розміщенням перенесення у додатковому регістрі. Тоді 2-байтове число буде містити 256+24=280, тобто правильний результат.

У табл. 1.4 наведені правила віднімання однорозрядних двійкових чисел. Правила 1,2,3 аналогічні правилам десяткового віднімання. Правило 4 вимагає позики зі старшого розряду, так що зменшуваним є число 10, від'ємником - 1, а різницею - 1.

Таблиця 1.4. Правила двійкового віднімання

Номер правила	1	2	3	4
Зменшуване	0	1	1	1 (0) Позика із старшого розряду
-	-	-	-	-
Від'ємник	0	0	1	1
Різниця	0	1	0	1

Приклад 1.4. *Відняти від двійкового числа 10100011_2 число 00110101_2.*

Виконуємо віднімання двох чисел, використавши правила двійкового віднімання (див. табл. 1.4):

	11111		- рядок позик
	10100011	=	163_{10}
-	00110101	=	53_{10}
	01101110	=	110_{10}

Відзначимо, що віднімання більшого числа від меншого приводить до некоректного результату внаслідок необхідності позики з дев'ятого розряду. Цей некоректний результат необхідно скоректувати програмним шляхом.

У табл. 1.5 наведені правила множення однорозрядних двійкових чисел. У перших трьох випадках принаймні один множник дорівнює 0, тому добуток дорівнює 0. У четвертому випадку множниками є одиниці, тому добутком є 1.

Таблиця 1.5. Правила двійкового множення

Номер правила	1	2	3	4
Множник	0	1	0	1
	x	x	x	x
Множник	0	0	1	1
Добуток	0	0	0	1

Приклад 1.5. *Перемножити два двійкових числа: 1101_2 і 101_2.*

Знайдемо добуток, використавши правила (див. табл. 1.5):

	1101
x	101
	1101
+	0000
	1101
	1000001

Для перевірки представимо множники у десятковому вигляді: $1101_2=13_{10}$, $101_2=5_{10}$, результат множення $13x5 = 65_{10}=1000001_2$.

Відзначимо, що розрядність добутку дорівнює сумі розрядів множників.

Шістнадцяткова система числення є системою з основою 16 та містить 16 символів: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. У табл. 1.6 наведені двійкові і шістнадцяткові еквіваленти 16 перших десяткових чисел. Кожна шістнадцяткова цифра подається єдиною комбінацією чотирьох двійкових цифр. Так, шістнадцятковим еквівалентом двійкового числа 10011110_2 є число 9E16. Це означає, що старша тетрада (4 старших розряди) 1001 двійкового числа записується як 9_{16}, а молодша тетрада 1110 – як E_{16}.

Таблиця 1.6. Двійкові і шістнадцяткові еквіваленти десяткових чисел

Десяткове число	Шістнадцятковий еквівалент	Двійковий еквівалент			
Значення ваг позицій					
10^1 10^0	16^0	2^3	2^2	2^1	2^0
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
1 0	A	1	0	1	0
1 1	B	1	0	1	1
1 2	C	1	1	0	0
1 3	D	1	1	0	1
1 4	E	1	1	1	0
1 5	F	1	1	1	1

Приклад 1.6. *Перетворити двійкове число 111010_2 на шістнадцятковий еквівалент.*

Для перетворення двійкового числа на шістнадцятковий еквівалент необхідно розділити його на тетради, починаючи з наймолодшого розряду, а потім кожну

тетраду замінити еквівалентною шістнадцятковою цифрою. Значення молодшої тетради $1010_2 = A_{16}$, старшої $0011_2 = 3_{16}$. Отже, $111010_2 = 3A_{16}$.

Приклад 1.7. *Перетворити шістнадцяткове число $7F_{16}$ на двійковий еквівалент.*

Для перетворення шістнадцяткового числа на двійковий еквівалент кожну шістнадцяткову цифру необхідно замінити на двійковий еквівалент – тетраду (див. табл. 1.16). Еквівалентом шістнадцяткової цифри 7_{16} є двійкове число 0111_2, а цифри F_{16} – число 11112_2. Отже, $7F_{16} = 11110111_2$.

Приклад 1.8. *Перетворити шістнадцяткове число $2C6E_{16}$ на десятковий еквівалент.*

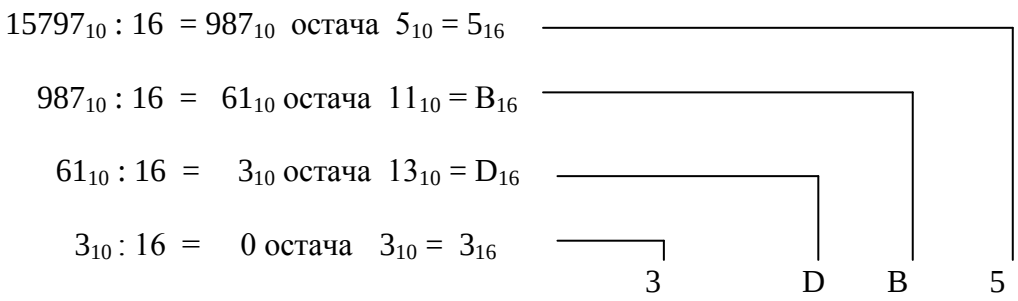
Процедура перетворення виконується згідно табл.1.17. Кожна цифра шістнадцяткового числа множиться на відповідну вагу позиції. Сума цих добутків дає десяткове число 11374_{10}.

Таблиця 1.7. Перетворення шістнадцяткового числа на десяткове

Значення позицій	4096	256	16	1	
Значення ваг позицій	16^3=4096	162=256	161=16	160=1	
Шістнадцяткове число	2 x 4096	3 x 256	6 x 16	E x 1	
Десяткове число	8192 +	3072 +	96 +	14	$=11374_{10}$

Приклад 1.9. *Перетворити десяткове число 15797 на шістнадцятковий еквівалент.*

При перетворенні десяткове число 15797_{10} ділиться на 16, що дає частку 987_{10} і остачу $5_{10}=5_{16}$. Таким чином, молодший розряд шістнадцяткового числа має значення 5. Частка 987_{10} стає діленим і знову ділиться на 16, що дає частку 61_{10} і остачу $11_{10}=B_{16}$., яка стає значенням другого розряду шістнадцяткового числа. Ділення 61_{10} на 16 дає частку 3_{10} і остачу $13_{10}=D_{16}$. Ділення 3_{10} на 16 дає частку 0 і остачу $3_{10}=3_{16}$. Оскільки остання частка дорівнює 0, то цифра 3_{16} стає значенням старшого розряду шістнадцяткового числа:



Отже, 15797_{10}=3DB5$_{16}$.

Двійково-десяткова система числення є системою, в якій кожна десяткова цифра від 0 до 9 подається 4-розрядним двійковим еквівалентом. Двійково-десяткова система числення дозволяє скоротити програмні та апаратні витрати при перетворенні двійкових чисел, які використовуються при обробці інформації у процесорі, на десяткові, які виводяться на пристрої відображення. Ця система є ефективною і при перетворенні десяткових чисел на двійкові. Двійково-десяткові числа записують з індексом 2-10 або ДДК (двійково-десятковий код), наприклад $01001001_{2\text{-}10}$, $0100_{ДДК}$.

Приклад 1.10. *Перетворити десяткове число 3691_{10} на двійково-десятковий код (2-10).*

При перетворенні кожна цифра десяткового числа перетворюється на двійковий 4-розрядний еквівалент:

Десяткове число	3	6	9	1
Двійково-десяткове число	0011	0110	1001	0001

Отже, 3691_{10}=0011 0110 1001 $0001_{2\text{-}10}$:

Приклад 1.11. *Перетворити двійково-десяткове число $1000\ 0000\ 0111\ 0010_{2\text{-}10}$ на десятковий еквівалент.*

Кожна тетрада двійково-десяткового числа перетворюється на десятковий еквівалент:

Двійково-десяткове число	1000	0000	0111	0010
Десяткове число	8	0	7	2

Отже, 1000 0000 0111 0010 $_{2\text{-}10}$=8072_{10}.

Подання чисел у мікропроцесорах

Інформація розміщується у регістрах або комірках пам'яті мікропроцесора у вигляді двійкових чисел, причому для кожного розряду числа приділяється окрема комірка, що зберігає один біт інформації. Сукупність комірок, призначених для розміщення одного двійкового числа, називають ***розрядною сіткою***. Кількість комірок у розрядній сітці обмежена і залежить від конструктивних особливостей мікропроцесора.

Натуральним кодом називається подання числа як *цілого беззнакового числа* у двійковій системі числення. Діапазон чисел у натуральному коді, які можуть бути розміщені у *n*-розрядній сітці, складає від 0 до $2^n - 1$, тобто для 8-розрядної сітки діапазон чисел у натуральному коді складає від 0 до 255. Наприклад, натуральний код числа 53_{10} у 8-розрядній сітці наведено на рис 1.3.

Вага розрядів	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
53_{10}	0	0	1	1	0	1	0	1

Рис.1.3. Натуральний код числа 53_{10} у 8-розрядній сітці

Для подання *цілих знакових чисел* використовуються наступні коди: 1) ***прямий код***; 2) ***обернений код***; 3) ***додатковий код***. Старший розряд сітки є знаковим. Значення цього розряду дорівнює нулю для позитивних чисел і одиниці – для негативних. В інших розрядах розміщується модуль числа. *Позитивні числа* подаються однаково у всіх трьох кодах. Так, позитивне число $+53_{10}$ має вигляд, поданий на рис.1.3, однак внаслідок того, що старший розряд є знаковим, діапазон позитивних чисел складає від 0 до $2^{n-1}-1$. Наприклад, для 8-розрядної сітки діапазон позитивних чисел складає від 0 до +127. Подання *негативних чисел* залежить від використання того чи іншого коду.

Подання негативного числа у ***прямому коді*** здійснюється наступним чином: у старшому, знаковому, розряді розміщується 1, а в інших розрядах – модуль числа. Діапазон негативних чисел у прямому коді складає від 0 до– $(2^{n-1}-1)$. Для прикладу на рис.1.4 наведено прямий код числа -53_{10} у 8-розрядній сітці, для якої діапазон негативних чисел складає від 0 до -127=11111111.

Значення ваг позицій	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Знак числа	Модуль числа						
1	0	1	1	0	1	0	1

Рис. 1.4. Прямий код числа -53_{10} у 8-розрядній сітці

До переваг прямого коду відносяться простота здійснення арифметичних операцій та однаковий діапазон значень позитивних і негативних чисел. Недоліками прямого коду є те, що операції додавання і віднімання чисел з різними знаками потребують додаткових операцій визначення більшого за модулем числа та визначення знаку результату. Крім того, наявність знаку при поданні числа 0 (+0=000000 і -0=1000000) веде до необхідності виконання зайвих операцій.

Подання негативного числа у ***оберненому коді*** здійснюється обчисленням числа, яке доповнює позитивне число з тим самим модулем до найбільшого беззнакового числа, яке може бути розташоване в даній розрядній сітці. Одержання оберненого коду негативного числа зводиться до порозрядного інвертування розрядів позитивного числа, включаючи знаковий розряд. Діапазон негативних чисел у оберненому коді складає від 0 до $-(2^{n-1}-1)$. Для прикладу на рис. 1.5 наведено обернений код числа -53_{10} у 8-розрядній сітці, для якої діапазон негативних чисел складає від 0 до -127_{10}, при цьому оберненим кодом найменшого числа -127_{10} є число $1000\ 0000_2$.

Значення ваг позицій	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Знак числа	Модуль числа						
1	1	0	0	1	0	1	0

Рис.1.5. Обернений код числа -53_{10} у 8-розрядній сітці

До переваг оберненого коду відносяться простота операцій одержання та додавання чисел із різними знаками. Недоліками є те, що нуль має два подання: +0=00000000 і -0=111111111, та потрібна апаратна реалізація коректування результату.

Подання негативного числа у ***додатковому коді*** здійснюється обчисленням числа, яке доповнює позитивне число з тим самим модулем до найбільшого беззнакового числа, з додаванням одиниці. Інакше кажучи, додатковий код отримується шляхом додавання 1 до оберненого коду.

Додатковий код можна одержати за наступним формальним правилом: цифри прямого коду позитивного числа необхідно інвертувати послідовно зліва направо до останньої одиниці, не включаючи її. Останню праву одиницю і наступні за нею (праворуч) нулі необхідно залишити без зміни. Діапазон негативних чисел у додатковому коді складає від 0 до -2^{n-1}. Для прикладу на рис. 1.7 наведено додатковий код числа -53_{10} у 8-розрядній сітці, для якої діапазон негативних чисел складає від 0 до -128_{10}, при цьому додатковим кодом найменшого числа -128_{10} є число $1000\ 0000_2$.

Значення ваг позицій	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Знак числа	Модуль числа						
1	1	0	0	1	0	1	1

Рис.1.6. Додатковий код числа -53_{10} у 8-розрядній сітці

До переваг додаткового коду відносяться простота операцій одержання та додавання чисел із різними знаками, а також те, що нуль має єдине подання +0=00000000. Завдяки цим перевагам додатковий код використовується найбільш часто.

ЛЕКЦІЯ 2

Класифікація мікропроцесорних комплектів

Мікропроцесори та мікропроцесорні комплекти класифікують за наступними ознаками: 1) за призначенням; 2) за кількістю ВІС; 3) за способом керування; 4) за типом архітектури; 5) за типом системи команд.

За призначенням мікропроцесори поділяють на універсальні та спеціалізовані.

Універсальними мікропроцесорами є мікропроцесорами загального призначення, які вирішують широкий клас задач обчислення, обробки та керування.

Спеціалізовані мікропроцесори призначені для виконання задач лише певного класу. Вирішення таких задач є найбільш швидкодіючим і ефективним. До спеціалізованих відносяться: 1) сигнальні; 2) медійні і мультимедійні; 3) трансп'ютери.

Сигнальні процесори призначені для вирішення задач цифрової обробки сигналів у реальному масштабі часу, наприклад: фільтрація сигналів; обчислення згортки; обчислення кореляційної функції; підсилення, обмеження і трансформація сигналу; пряме та зворотне перетворення Фур'є. До сигнальних процесорів відносяться процесори фірм *Texas Instruments* - TMS320C80, *Analog Devices* -ADSP2106x, *Motorola* - DSP560xx DSP9600x.

Медійні і мультимедійні процесори призначені для обробки аудіосигналів, графічної інформації, відеозображень, а також для розв'язування ряду задач у мультимедіакомп'ютерах, іграшкових приставках, побутовій техніці. До медійних і мультимедійних процесорів відносяться процесори фірм *MicroUnity* – Mediaprocessor, *Philips* – Trimedia, *Cromatic Reserch* - Mpact Media Engine, *Nvidia* - NV1, *Cyrix* – MediaGX.

Трансп'ютери призначені для масово-паралельних обчислень і роботи у мультипроцесорних системах. Для них характерним є наявність внутрішньої

пам'яті і вбудованого міжпроцесорного інтерфейсу, тобто каналів зв'язку з іншими ВІС МП. До трансп'ютерів відносяться процесори фірм *Inmos* - Т-2, Т-4, Т-8, Т9000.

За кількістю ВІС у МПК розрізняють багатокристальні МПК і однокристальні мікроконтролери (ОМК). До багатокристальних комплектів відносять: МПК з однокристальними МП і секційними МП.

Однокристальний мікропроцесор є конструктивно і завершеним виробом у вигляді однієї ВІС. Інша назва однокристальних мікропроцесорів - мікропроцесори ***з фіксованою розрядністю*** даних. До цього типу відносяться процесори фірм *Intel* -Pentium (P5, P6, P7), *AMD* -K5, K6, *Cyrix* - 6x86, M1, M2, *Diqital Equipment* - Alpha 21064, 21164A, *Silicon Graphics* - MIPS R10000, *Motorola* - Power PC 603, 604, 620, *Hewlett Packard* - PA-8000, *Sun Microsystems* - Ultra SPARC II.

У ***секційних*** мікропроцесорах в одній ВІС реалізується лише деяка функціональна частина (секція) процесора. Інша назва секційних мікропроцесорів - ***розрядно-модульні*** мікропроцесори або мікропроцесори ***з нарощенням розрядності.*** Секційність ВІС МП обумовлює значну гнучкість мікропроцесорних систем, можливість нарощення розрядності даних, створення специфічних технологічних команд із набору мікрокоманд. До секційних відносяться мікропроцесори серій К589, К1804.

Однокристальний мікроконтролер являє собою пристрій, який виконаний конструктивно в одному корпусі ВІС і містить основні складові частини МПК. До однокристальних мікроконтролерів відносяться ОМК фірм *Intel* - MCS-196/296, *MicroChip* - PIC17C4x PIC17C75x, *Mitsubishi Electric* - M3820, *Motorola* - MC33035, MC33039.

За способом керування розрізняють МП зі схемним та МП з мікропрограмним керуванням. Мікропроцесори ***зі схемним керуванням*** мають фіксований набір команд, розроблений фірмою-виробником, який не може змінюватися споживачем. У мікропроцесорах ***з мікропрограмним***

керуванням система команд розробляється при проектуванні конкретного МПК на базі набору найпростіших мікрокоманд з урахуванням класу задач, для вирішення яких призначений МПК.

За типом архітектури, або принципом побудови, розрізняють МП з фоннеймановською архітектурою і МП з гарвардською архітектурою.

За типом системи команд розрізняють CISC (Complete Instruction Set Command) процесори з повним набором команд, і на RISC (Reduced Instruction Set Command) процесори зі зменшеним набором команд.

Слід зазначити, що багато мікропроцесорних комплектів потрапляють під різні класифікаційні ознаки, оскільки здатні вирішувати задачі різних класів. Так, існують універсальні мікропроцесори з мультимедійним розширенням наборів команд, наприклад, Pentium MMX, Pentium П, Cyrix 6x86MX, AMD K6, Ultra SPARC. У CISC-процесорах Pentium PRO реалізовано ядро з RISC-архітектурою.

ЛЕКЦІЯ 3

Організація шин

Шина - це інформаційний канал, який об’єднує всі функціональні блоки мікропроцесорної системи та забезпечує обмін ***даними*** у вигляді двійкових чисел. Конструктивно шина являють собою n провідників, та один спільний провідник (земля). Дані по шині передаються у вигляді ***слів***, що є групою бітів. У ***паралельній шині*** n бітів передаються по окремих лініях одночасно, у ***послідовній шині*** біти передаються по єдиній лінії послідовно у часі. Паралельні шини виконуються у вигляді плоского кабелю, а послідовні – у вигляді коаксіального або волоконно-оптичного кабелю. Коаксіальний кабель використовується при передачі даних на відстань до 100 метрів, причому треба узгоджувати передавальні і приймальні каскади з хвильовим опором лінії. Волоконно-оптичний кабель використовується для передачі на більші відстані.

Усі основні блоки мікропроцесорного комплекту під'єднуються до єдиної паралельної шини, яка називається ***системною шиною SB*** (System Bus). Системна шина містить три шини: адреси, даних і керування.

Шина адреси AB (Address Bus) є однонапрямленою і призначена для передавання адреси комірки пам'яті або пристрою введення/виведення. Напрямок передавання по шині адреси – від мікропроцесора до зовнішніх пристроїв. Варіанти умовних позначень однонапрямленої паралельної шини наведені на рис.1.7, на якому стрілка вказує напрямок передавання.

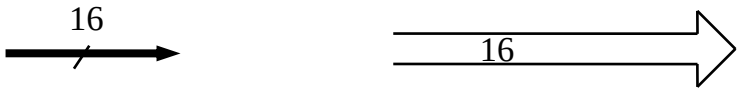


Рис. 1.7. Варіанти умовних позначень однонапрямленої паралельної 16-розрядної шини

Число 16 на рис 1.7 позначає розрядність шини. Зазначимо, що допускається позначення шин і без наведення розрядності.

Шина даних DB (Data Bus) є двонапрямленою і призначена для передавання даних між блоками мікропроцесорної системи. Інформація по одних і тих самих лініях DB може передаватися у двох напрямках - як до мікропроцесора, так і від нього. Варіанти умовних позначень двонапрямленої шини наведені на рис.1.8.

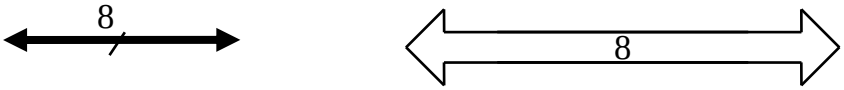


Рис. 1.8. Варіанти умовних позначень двонапрямленої паралельної 8-розрядної шини

Шина керування CB (Control Bus) призначена для передавання керуючих сигналів. Хоча напрямок керуючих сигналів може бути різним, однак шина керування не є двонапрямленою, бо для сигналів різного напрямку використовуються окремі лінії.

Принципи побудови мікропроцесорних систем

У основу побудови мікропроцесорних систем покладено три принципи: 1) магістральності; 2) модульності; 3) мікропрограмного керування.

Принцип магістральності визначає характер зв'язків між функціональними блоками мікропроцесорної системи – всі блоки під'єднуються до єдиної системної шини.

Принцип модульності полягає в тому, що система будується на основі обмеженої кількості типів модулів, які є конструктивно і функціонально завершеними. Кожний модуль мікропроцесорної системи має вхід керування третім (високоімпедансним) станом. Цей вхід називається $\overline{CS}$ (Chip Select) – вибір кристала або $\overline{OE}$ (Output Enable) – дозвіл виходу. Дія сигналу $\overline{CS}$ для тригера показана на рис. 1.9.

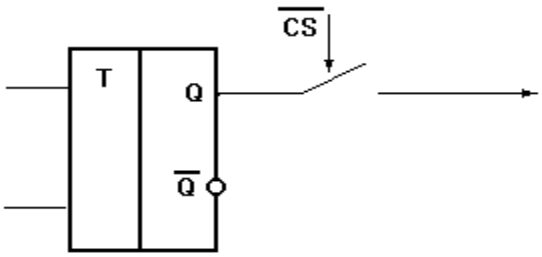


Рис. 1.9. Дія сигналу $\overline{CS}$ для тригера

Вихідний сигнал тригера Q з'явиться на виводі лише при активному (у даному випадку – нульовому) рівні сигналу $\overline{CS}$. При $\overline{CS}$ =1 тригер буде переведений у високоімпедансний стан. Вихід тригера є ***тристабільним***, тобто може знаходитися у одному з трьох станів: логічна одиниця, логічний нуль або високоімпедансний стан. У кожний момент часу до системної шини мікропроцесорної системи під'єднані лише два модулі - той, що приймає, і той, що передає інформацію. Інші знаходяться у високоімпедансному стані. Принципи магістральності і модульності дозволяють нарощувати керуючі і обчислювальні можливості мікропроцесорної системи шляхом під'єднання інших модулів.

Принцип мікропрограмного керування полягає у можливості здійснення елементарних операцій - мікрокоманд (зсув, пересилка інформації, логічні операції). Шляхом певної комбінації мікрокоманд можна утворити набір команд, який максимально відповідає призначенню системи, тобто створити *технологічну мову*. У секційних процесорах мікрокоманди можуть бути замінені спеціальною мікросхемою пам'яті мікрокоманд.

Узагальнена структурна схема мікропроцесорної системи

Узагальнена структурна схема мікропроцесорної системи наведена на рис. 1.10.

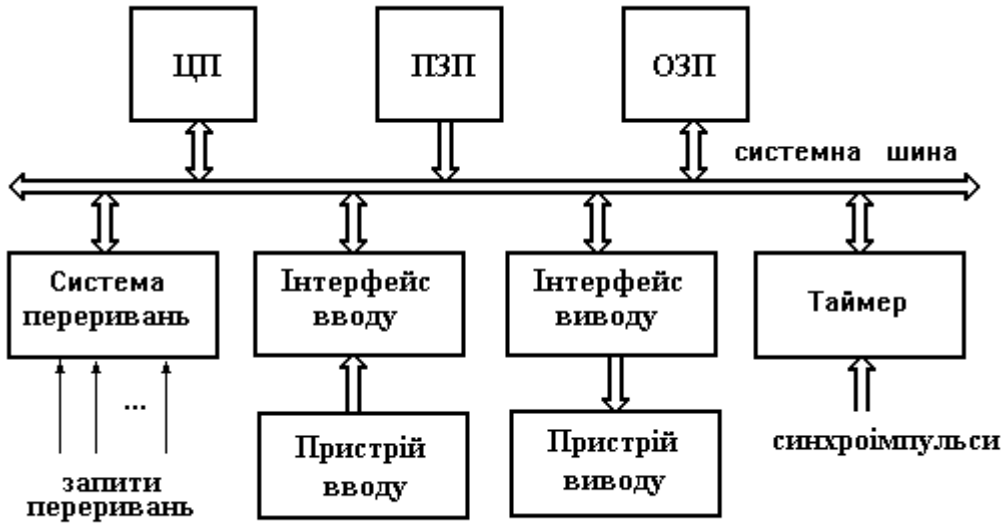


Рис. 1.10. Узагальнена структурна схема мікропроцесорної системи керування

До складу МПС входять: 1) центральний процесор ЦП; 2) постійний запам'ятовувальний пристрій ПЗП; 3) оперативний запам'ятовувальний пристрій ОЗП; 4) система переривань; 5) таймер; 6) пристрої введення/виведення ПВВ. Пристрої введення/виведення під'єднані до системної шини через інтерфейси введення/виведення.

ПЗП та ОЗП складають *систему пам'яті*, яка призначена для збереження інформації у вигляді двійкових чисел. ПЗП призначений для збереження

програм керування, таблиць, констант. ОЗП призначений для збереження проміжних результатів обчислень. Пам’ять організована у вигляді масиву комірок, кожна з яких має свою адресу і містить байт або слово. Байтом називається группа з 8 біт, а слово може мати будь-яку довжину в бітах. Найбільш часто під словом розуміють двійкове число довжиною 2 байти. Для звернення до комірки пам’яті необхідно видати її адресу на шину адреси. На рис. 1.11 наведена структура пам’яті із 8 однобайтових комірок, де кожній адресі відповідає певний вміст комірки. Так, комірка з адресою 000 має вміст $01011111_2 = 5F_{16}$.

Адреса	Дані
000	01011111
001	00010011
010	01110111
011	00001100
100	00000000
101	11111111
110	10101010
111	11110000

Рис. 1.11. Структура пам’яті із 8 однобайтових комірок

Модуль ***центрального процесора*** здійснює обробку даних і керує усіма іншими модулями системи. ЦП, крім ВІС мікропроцесора, містить схеми синхронізації та інтерфейсу із системною шиною. Він вибирає коди команд з пам’яти, дешифрує їх і виконує. Впродовж часу виконання команди, який має назву *командого циклу*, ЦП виконує наступні дії:

1) виставляє адресу команди на шину адреси АВ;

2) отримує код команди із пам’яті і дешифрує його;

3) обчислює адреси операнда і зчитує дані;

4) виконує операцію, що визначена командою;

5) сприймає зовнішні керуючі сигнали, наприклад, запити переривань;

6) генерує сигнали стану і керування, необхідні для роботи пам'яті і пристроїв введення/виведення.

Пристрої введення/виведення, або ***зовнішні пристрої*** – це пристрої, призначені для введення інформації у мікропроцесор або виведення інформації з нього. Прикладами пристроїв введення/виведення є дисплеї, друкувальні пристрої, клавіатура, ЦАП, АЦП, реле, комутатори. Для під'єднання пристроїв введення/виведення до системної шини їх сигнали повинні відповідати певним стандартам. Це досягається за допомогою інтерфейсів введення/виведення.

Інтерфейси введення/виведення називають також ***контролерами*** або ***адаптерами***. МП звертається до інтерфейсів за допомогою спеціальних команд введення/виведення. При цьому МП виставляє на шину адреси AB адресу інтерфейсу, а по шині даних DB зчитує дані з пристрою введення або записує у пристрій виведення. На рис. 2.6 показано один інтерфейс введення і один інтерфейс виведення.

Система переривань дозволяє МПС реагувати на зовнішні сигнали – запити переривань, джерелами яких можуть бути: 1) сигнали готовності від зовнішніх пристроїв; 2) сигнали від генераторів; 3) сигнали з виходів датчиків. При появі запиту переривання ЦП перериває основну програму і переходить до виконання підпрограми обслуговування запиту переривання. Для побудови системи переривань мікропроцесорні комплекти містять BIC спеціальних програмовних контролерів переривань.

Таймер призначений для реалізації функцій, пов'язаних з відліком часу. Після того, як мікропроцесор завантажує в таймер число, яке задає частоту, затримку або коефіцієнт ділення, таймер реалізує необхідну функцію.

ЛЕКЦІЯ 4

Архітектура мікропроцесорів

Поняття ***архітектури*** мікропроцесора включає його складові частини, а також зв’язки та взаємодію між ними. Описання архітектури включає: 1) описання структурної схеми самого мікропроцесора; 2) програмну модель мікропроцесора (описання функцій регістрів); 3) інформацію про організацію пам'яті (ємність пам’яті та способи її адресації); 4) описання організації процедур введення/виведення.

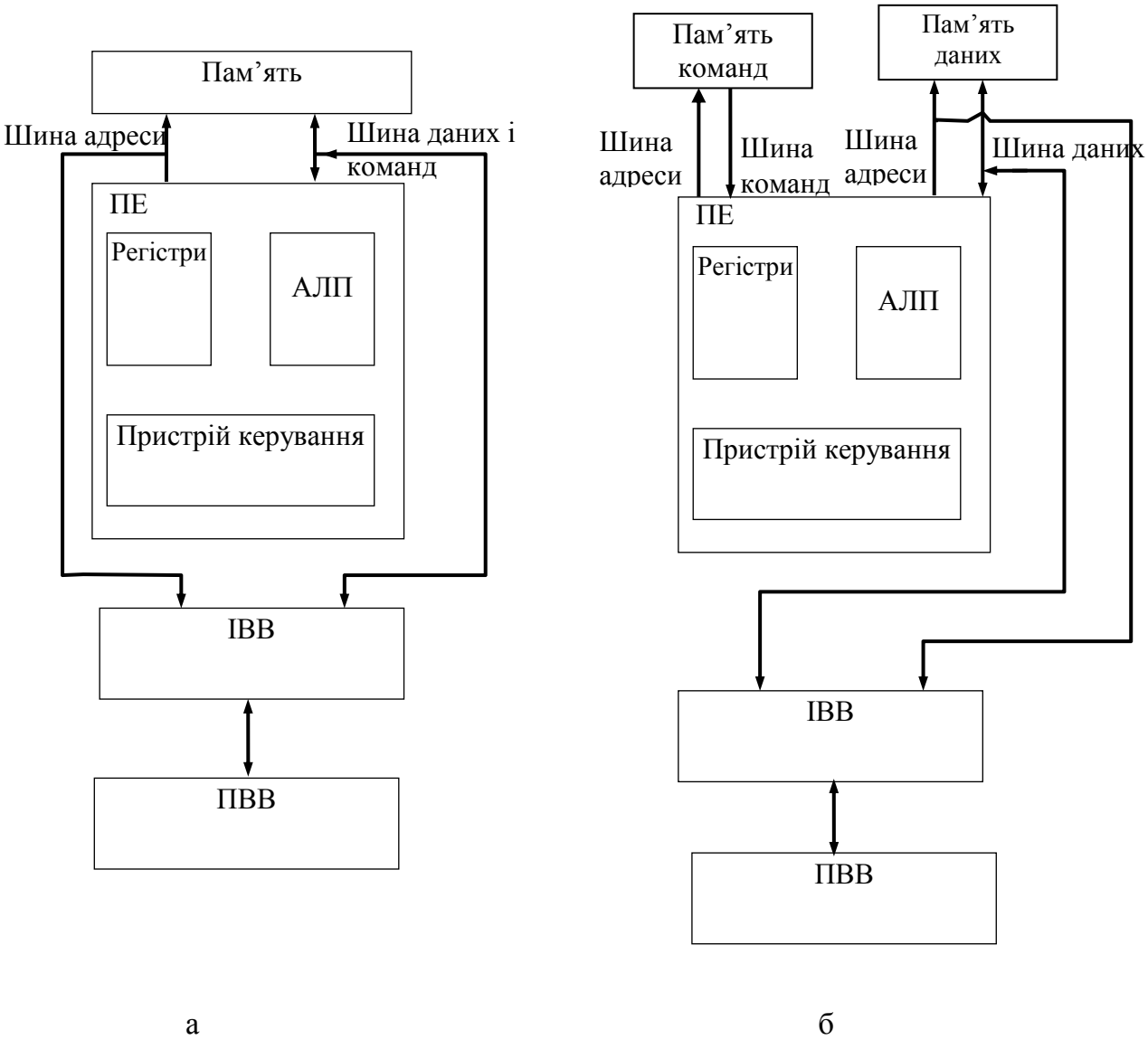
Існують два основних типи архітектури – фоннеймановська та гарвардська. ***Фоннеймановська*** архітектура (рис. 1.12,а) була запропонована в 1945 р. американським математиком Дж. фон Нейманом. Особливістю цієї архітектури є те, що програма і дані знаходяться у спільній пам’яті, доступ до якої здійнюється по одній шині даних і команд.

Гарвардська архітектура була вперше реалізована в 1944 р. у релейній обчислювальній машині Гарвардського університету (США). Особливістю цієї архітектури є те, що пам’ять даних і пам’ять програм розділені та мають окремі шину даних і *шину команд* (рис. 1.12,б), що дозволяє підвищити швидкодію мікропроцесорної системи.

Структурні схеми обох архітектур містять: процесор, пам’ять, інтерфейси введення/виведення (ІВВ) і пристрої введення/виведення (ПВВ). Пам’ять і ІВВ для різних типів МП можуть бути як внутрішніми, тобто розміщуватися на тому ж кристалі, що і процесорний елемент (ПЕ), так і зовнішніми. Процесорний елемент містить регістри, арифметико-логічний пристрій (АЛП) і пристрій керування та виконує функції обробки даних і керування процесами обміну інформацією.

Пам’ять забезпечує зберігання кодів команд програми і даних. Інтерфейси призначені для зв’язку з пристроями введення/виведення,

наприклад, з клавіатурою, дисплеєм, друкувальними пристроями, датчиками інформації. Усі елементи структурної схеми під'єднані за допомогою шин.



а б

Рис. 2.8. Основні типи архітектури: а – фоннеймановська; б - гарвардська

Більш докладна структурна схема з процесором фоннеймановської архітектури наведена на рис. 1.13.

Схема процесора містить пристрій керування, арифметико-логічний пристрій (АЛП) та наступні регістри: 1) акумулятор; 2) лічильник команд; 3) вказівник стека; 4) регістр адреси; 5) регістри даних; 6) регістр команд; 7) регістр стану.

Пристрій керування відповідно до кодів команд та зовнішніх керуючих і синхросигналів виробляє керуючі сигнали для всіх блоків структурної схеми МП, а також керує обміном інформацією між МП, пам'яттю і пристроями введення/виведення. Пристрій керування реалізує наступні функції: 1) початкового встановлення МП; 2) синхронізації; 3) переривань; 4) узгодження швидкодії модулів мікропроцесорної системи

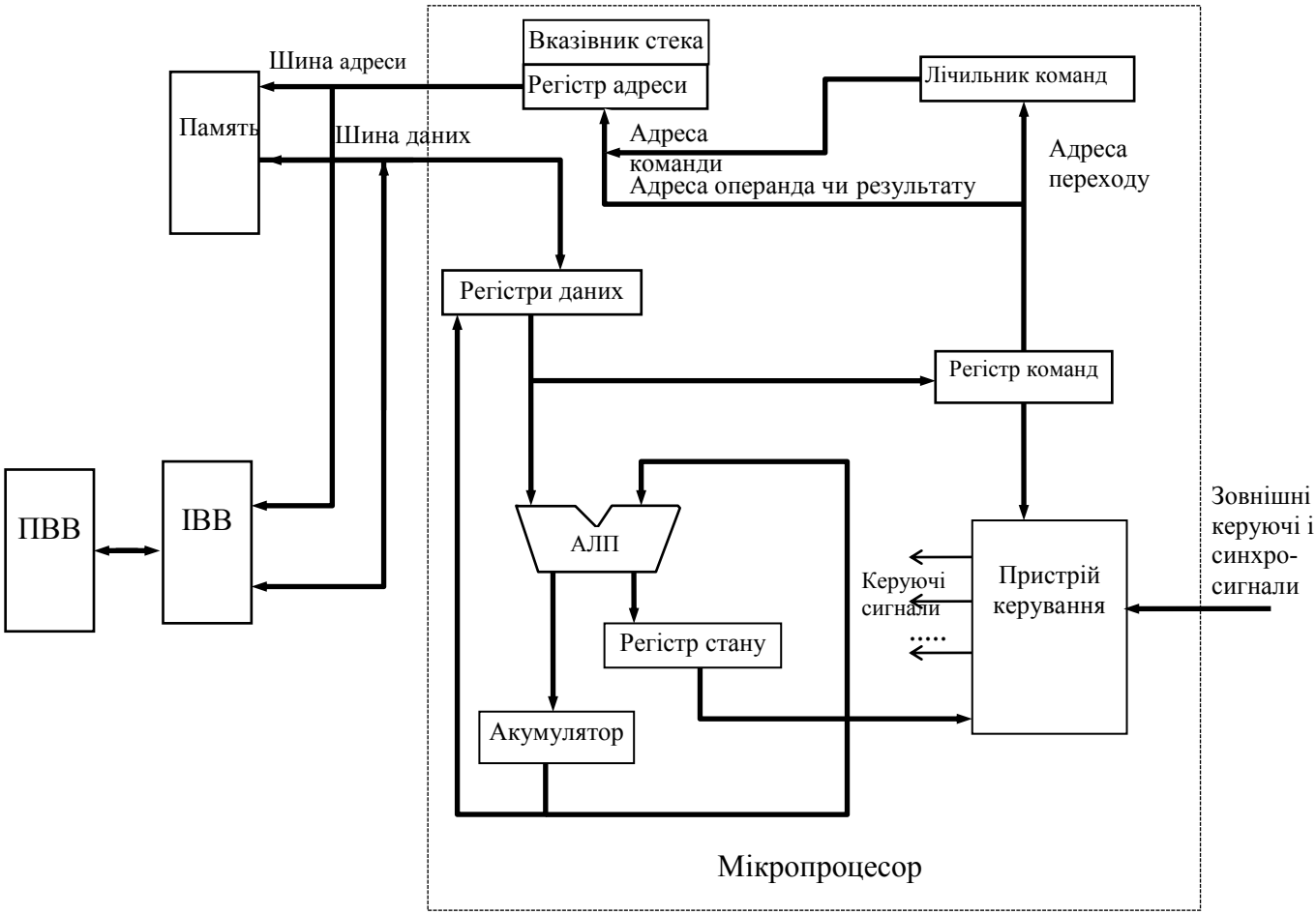


Рис. 1.13. Структурна схема з процесором фоннеймановської архітектури

Функція початкового встановлення МП. Зовнішній сигнал початкового встановлення процесора **RESET** формується при ввімкненні джерела живлення МП або при натисканні кнопки "RESET". При появі цього сигналу пристрій керування забезпечує завантаження нульового значення у програмний лічильник, що ініціює вибірку з пам'яті байту команди з нульовою адресою. Наприкінці вибірки вміст лічильника команд

збільшується на одиницю, і вибирається байт команд з наступною адресою. Таким чином виконується вся програма, що записана у пам’яті програм.

Функція синхронізації. Згідно з зовнішніми керуючими і синхросигналами пристрій керування синхронізує роботу усіх блоків МП.

Функція переривань. При надходженні сигналу переривання пристрій керування ініціює роботу підпрограми обробки відповідного переривання. Необхідність реалізації функцій переривань виникає у тих випадках, коли при виконанні основної програми треба перевести МП на розв’язання іншої задачі, наприклад, обробки аварійної ситуації чи роботи з пристроями введення/виведення.

Функція узгодження швидкодії модулів мікропроцесорної системи. При обслуговуванні пам’яті та ПВВ, які мають значно меншу швидкодію, ніж МП, узгодження швидкодії вирішується шляхом генерації тактів очікування МП. При обслуговуванні пристроїв, які мають більшу швидкодію, ніж МП, використовується режим безпосереднього доступу до пам’яті.

Арифметико-логічний пристрій являє собою комбінаційну схему на основі суматора, який сигналами з виходів пристрою керування налагоджується на виконання певної арифметичної або логічної операції: 1) додавання; 2) віднімання; 3) логічне І; 4) логічне АБО; 5) логічне НІ; 6) ВИКЛЮЧНЕ АБО; 7) зсуву; 8) порівняння; 9) десяткової корекції. Таким чином, АЛП виконує арифметичні або логічні операції над операндами, які пересилаються із пам’яті і/або регістрів мікропроцесора. ***Операнд*** - це об’єкт у вигляді значення даних, вмісту регістрів або вмісту комірки пам’яті, з яким оперує команда,.

Наприклад, в команді додавання операндами є доданки. Операнд може бути заданий у команді у вигляді числа або знаходитися в регістрі або комірці пам’яті. Отриманий після виконання команди в АЛП результат пересилається в регістр або комірку пам’яті.

Регістри призначені для зберігання n-розрядного двійкового числа, і являють собою n тригерів зі схемами керування читанням/записом та вибірки. Регістри утворюють внутрішню пам’ять МП і використовуються для зберігання проміжних результатів обчислень.

Акумулятор - це регістр, у якому зберігається один з операндів. Після виконання команди в акумуляторі замість операнда розміщується результат операції. У 8-розрядних процесорах акумулятор бере участь в усіх операціях арифметико-логічного пристрою. У 16-розрядних МП більшість команд виконуються без участі акумулятора, але в деяких командах (введення, виведення, множення, ділення) акумулятор діє так само, як і в 8-розрядних МП, тобто зберігає один з операндів, а після виконання команди – результат операції.

Лічильник команд, або ***програмний лічильник*** призначений для зберігання адреси комірки пам’яті, яка містить код наступної команди. Програма дій мікропроцесора записана у пам’яті у вигляді послідовності кодів команд. Для переходу до наступної команди вміст лічильника збільшується на одиницю в момент вибірки команди з пам'яті. Таким чином, наприкінці виконання команди в лічильнику команд зберігається адреса команди, яка повинна виконуватися наступною.

Вказівник стека – це регістр, який зберігає адресу останньої зайнятої комірки стека. ***Стеком***, або ***стековою пам’яттю,*** називається область пам’яті, організована за принципом “останній прийшов – перший вийшов”.

Регістр команд зберігає код команди протягом всього часу виконання команди.

Регістр адреси і ***регістри даних*** призначені для зберігання адрес і даних, які використовуються при виконанні поточної команди у МП.

Регістр стану, або ***регістр прапорців*** (ознак) призначений для зберігання інформації про результату операції в АЛП і являє собою декілька тригерів, що приймають одиничні або нульові значення.

Розділ 2 ОДНОКРИСТАЛЬНІ 16-РОЗРЯДНІ МІКРОПРОЦЕСОРИ

ЛЕКЦІЯ 5

Однокристальні 16-розрядні мікропроцесори

До 16-розрядних мікропроцесорів першого покоління відносяться мікропроцесори i8086/i8088 та i80186/i80188, до мікропроцесорів другого покоління – i80286. ВІС мікропроцесора i8086 з геометричними розмірами 5,5 х 5,5 мм має 40 контактів, містить близько 29000 транзисторів і споживає 1,7 Вт від джерела живлення +5 В, тактова частота – 5; 8 або 10 МГц.

Мікропроцесор виконує операції над 8- та 16-розрядними даними, наведеними у двійковому або двійково-десятковому вигляді, може обробляти окремі біти, а також рядки або масиви даних. Він має вбудовані апаратні засоби множення та ділення. Формати даних і операції, що виконуються наведені в табл. 2.1. МП має внутрішній надоперативний запам’ятовувальний пристрій ємністю 14х16 байт. Шина адреси є 20-розрядною, що дозволяє безпосередньо адресувати до 2^{20}=1048576 комірок пам'яті (1М байт).

Простір адрес введення/виведення складає 64К байт. У ВІС i8086 реалізована багаторівнева векторна система переривань з кількістю векторів до 256. Передбачена також організація прямого доступу до пам'яті, при якому МП припиняє роботу і переводить в третій стан шини адреси, даних та керування.

Середній час виконання команди займає 12 тактів. Особливістю МП i8086 є можливість часткової реконфігурації апаратної частини для забезпечення роботи в двох режимах – мінімальному і максимальному. Режими роботи задаються апаратно. У мінімальному режимі, що використовується для побудови однопроцесорних систем, МП самостійно формує всі сигнали керування внутрішнім системним інтерфейсом. У максимальному режимі, що використовується для побудови мультипроцесорних систем, МП формує на лініях стану двійковий код, що залежить від типу циклу шини. У відповідності до цього кодом системний

контролер К1810ВГ88 виробляє сигнали керування шиною. Контакти, які звільнилися у результаті кодування інформації, використовуються для керування мультипроцесорним режимом. При використанні арифметичного співпроцесора необхідно обирати максимальний режим

Таблиця 2.1. Формати даних і операцій, що виконуються МП i8086

Тип даних	Формат	Діапазон	Операції
Байт без знака	7 6 5 4 3 2 1 0 Старший значимий розряд	0... 255	+, -, x,:
Байт зі знаком	7 6 5 4 3 2 1 0 Старший значимий розряд Знаковий розряд	-128... +127	+, -, x,:
Слово без знака	15 14 ..... 1 0 Старший значимий розряд	0... 65 535	+, -, x,:
Слово зі знаком Ст. тетрада (0-9)	15 14 ..... 1 0 Старший значущій розряд Знаковий розряд	-32 768...+32 767	+, -, x,:
Упаковане двійково-десяткове число	7 6 5 4 3 2 1 0 Ст. тетрада (0-9) Мол. тетрада (0-9)	0...99	+, -з корекцією
Розпаковане двійково-десяткове число	7 6 5 4 3 2 1 0 Ст. тетрада (0) Мол. тетрада (0-9)	0...9	+, -, x , :з корекцією

Примітка. Знакові числа подають у додатковому коді (див.розд.2.9).

Структурна схема МП i8086

У МП i8086 застосовано конвеєрну архітектуру, що дозволяє сумістити у часі цикли виконання команди та вибірки з пам'яті кодів наступних команд. Це досягається паралельною роботою двох порівняно незалежних пристроїв -

операційного пристрою (ОП) та шинного інтерфейсу (ШІ). Структурну схему МП і8086 наведено на рис. 2.1. Операційний пристрій виконує команду, а шинний інтерфейс здійснює взаємодію із зовнішньою шиною - виставляє адреси, зчитує коди команд, операнди, записує результати обчислень в пам'ять або пристрої введення/виведення.

Операційний пристрій складається з регістрів загального призначення РЗП, призначених для зберігання проміжних результатів – даних та адрес; арифметико-логічного пристрою АЛП з буферними регістрами; регістра прапорців; схеми керування та синхронізації СК та С, яка дешифрує коди команд і виробляє керуючі сигнали для всіх блоків схеми МП. Шинний інтерфейс складається з шістибайтової регістрової пам’яті, яка називається чергою команд, чотирьох сегментних регістрів: CS, DS, ES, SS, вказівника команд IP, суматора, а також допоміжних регістрів зв’язку і буферних схем шин БШ адреси/даних. Черга команд працює за принципом FIFO (First Input - First Output, тобто *перший прийшов - перший вийшов*) і зберігає на виході порядок надходження команд. Довжина черги - 6 байт. Коли ОП зайнятий виконанням команди, ШІ самостійно ініціює випереджаючу вибірку кодів команд з пам'яті у чергу команд. Вибірка з пам'яті чергового командного слова здійснюється тоді, коли в черзі виявляється два вільних байта. Черга збільшує швидкодію процесора у випадку послідовного виконання команд. При вибірці команд переходів, викликів і повернень з підпрограм та обробці запитів переривань черга команд скидається і вибірка починається з нового місця програмної пам'яті.

Ще одним із завдань ШІ є формування фізичної 20-розрядної адреси з двох 16-розрядних слів. Першим словом є вміст одного з сегментних регістрів CS, SS, DS, ES, а друге слово залежить від типу адресації операнду або коду команди. Складання 16-розрядних слів відбувається зі зміщенням на 4 розряди і здійснюється за допомогою суматора, що входить до складу шинного інтерфейсу.

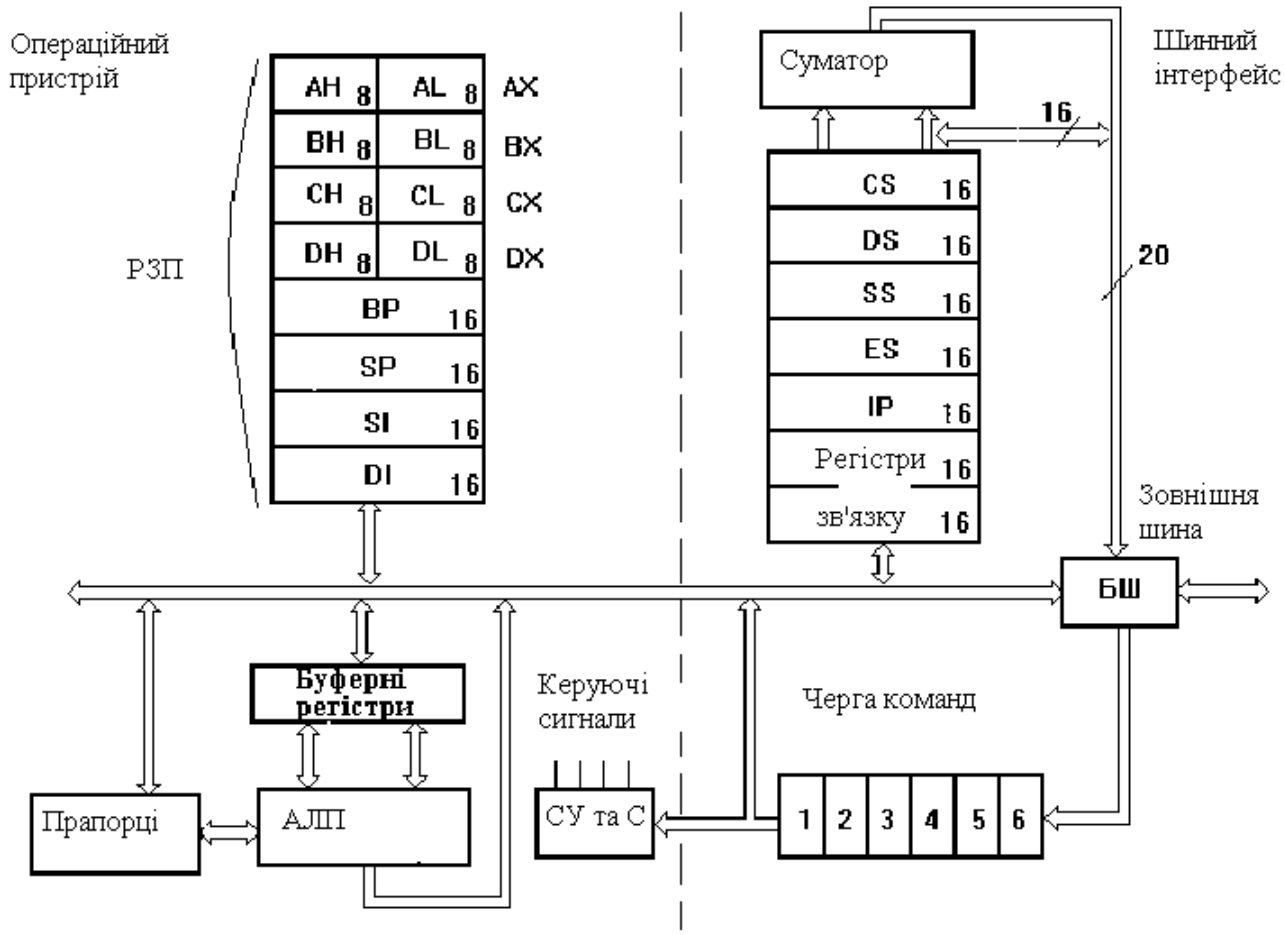


Рис. 2.1. Структурна схема мікропроцесора i8086

Організація пам'яті

Пам'ять являє собою масив ємністю 1М байт, тобто 2^{20} 8-розрядних комірок (рис.2.2). У пам'яті зберігаються як байти, так і двобайтові слова. Слова розташовуються у двох сусідніх комірках пам'яті – старший байт зберігається у комірці зі старшою адресою, молодший – з молодшою. Адресою слова вважається адреса його молодшого байта. На рис. 3.13 показаний приклад, коли з адресою 00000 зберігається байт 35H, а з адресою 00001 - слово 784AH. Початкові (00000H-003FFH) і кінцеві адреси (FFFF0H-FFFFFH) зарезервовані для системи переривань та початкового встановлення відповідно.

Наведена на рис. 2.2 організація пам’яті, коли кожній адресі відповідає вміст однієї комірки пам’яті, називається ***лінійною***. У МП i8086 прийнято

сегментну організацію пам’яті, яка характеризується тим, що програмно-доступною є не вся пам’ять, а лише деякі сегменти, тобто області пам’яті. Всередині сегмента використовують лінійну адресацію.

Застосування сегментної організації пояснюється наступним. Мікропроцесор i8086 являє собою 16-розрядний процесор, тобто він має 16-розрядну внутрішню шину, 16-розрядні регістри і суматори. Прагнення розробників ВІС адресувати якомога більший масив пам'яті призвело до використання 20-розрядної шини даних. Для порівняння: 16-розрядна шина адреси дозволяє адресувати 2^{16} = 64К байт; 20-розрядна - 2^{20}= 1М байт.

Для того, щоб сформувати 20-розрядну адресу у 16-розрядному процесорі використовують інформацію двох 16-розрядних регістрів. У МП i8086 прийнято наступний механізм формування 20-розрядної адреси з двох 16-розрядних адрес, які називаються ***логічними.***

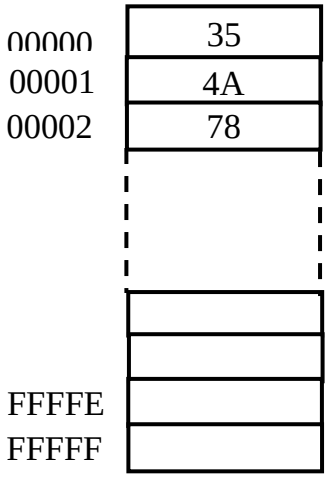


Рис 2.2 Програмна модель пам’яті

Перша логічна адреса, доповнена праворуч чотирма нулями, являє собою початкову адресу сегмента обсягом 64К байт. Друга логічна адреса визначає зміщення у сегменті, тобто визначає відстань від початку сегменту до комірки, що адресується. Якщо вона дорівнює 0000, то адресується перша комірка сегмента, якщо FFFFH - то остання. Таким чином, логічний адресний простір поділено на блоки суміжних адрес розміром 64К байт, тобто сегменти.

Такий підхід до організації пам'яті зручний ще й тому, що пам'ять звичайно логічно поділяється на області коду (програмної пам'яті), даних і стека. Фізична 20-розрядна адреса комірки пам'яті формується з двох 16-розрядних адрес - адреси сегмента Seg і виконавчої адреси EA (Executive

Address), що додаються зі зміщенням на чотири розряди, як показано на рис. 3.14.

Рис. 2.3. Формування фізичної адреси

Зміщення адреси сегмента на 4 розряди ліворуч еквівалентне його множенню на 2^4. Тоді фізична адреса дорівнює 16xSeg+EA. Як перша логічна адреса Seg використовується вміст одного з чотирьох сегментних регістрів: CS - (Code Segment – сегмент кодів), DS (Data Segment – сегмент даних), ES (Extended Segment – додатковий сегмент даних), SS (Stack Segment – сегмент стека). Друга логічна адреса EA, або зміщення, залежить від сегмента. Так, в сегменті кодів як EA використовується вміст лічильника інструкцій IP, в сегментах даних значення EA залежить від засобу адресації операнда, в сегменті стека використовуються регістри SP або BP.

Перетворення логічних адрес на фізичні завжди однозначне, тобто парі Seg і EA відповідає єдина фізична адреса. Зворотне перетворення неоднозначне: фізичну адреса можна подати за допомогою 4096 пар логічних адрес.

У подальшому будемо позначати фізичну адресу у вигляді Seg:EA, де як Seg і EA можуть використовуватися як позначення регістрів, так і 16-розрядні дані.

Приклад 2.1. *Знайти значення фізичної адреси за двома значеннями логічних адрес CS:IP.*

Нехай вмістом сегментного регістра CS є число 2002H, вмістом вказівника команд IP - 3175H. Додамо до значення CS чотири нулі праворуч:

CS(0000) = 0010 0000 0000 0010 0000B=20020H.

Виконавши операцію додавання отриманої величини з вмістом регістра IP, отримаємо фізичну адресу:

```
+0010 0000 0000 0010 0000
      0011 0001 0111 0101
 0010 0011 0001 1001 0101=23195H.
```

Таким чином, запис CS:IP при CS=2002H, IP=3175H відповідає фізичній адресі 23195H.

Приклад 2.2. *Знайти значення двох логічних адрес, які б відповідали фізичній адресі 23195H і не дорівнювали логічним адресам прикладу 2.1.*

Значення фізичної адреси 23195H дасть додавання двох інших логічних адрес 2100H: 2195H:

```
+0010 0001 0000 0000 0000
      0010 0001 1001 0101
 0010 0011 0001 1001 0101=23195H.
```

Простір пам'яті 1M байт, починаючи з нульової адреси, розбивається на ***параграфи*** по 16 байт. Сегмент може починатися тільки на межі параграфа, тобто у адресі сегмента молодші 4 біти адреси – нульові. Розміщення сегментів у пам'яті довільне: сегменти можуть частково або повністю перекриватися або не мати загальних частин. Змінюючи значення як першої, так і другої логічних адрес, можна адресувати будь-яку комірку із загальної пам’яті ємністю 1M байт.

На рис.2.4,а показано розташування у просторі 1M байт 4-х сегментів по 64К байт без перекриття. 20-розрядні початкові адреси сегментів визначаються вмістом 16-розрядних сегментних регістрів, які доповнені праворуч чотирма нульовими бітами. Зміщення в сегменті кодів визначається вмістом регістра IP, зміщення в сегменті даних і додатковому сегменті даних – ефективною адресою EA, яка наводиться у команді стека – вмістом регістра SP.

У сегментах кодів розташовано коди команд, тобто програма у машинних кодах; у решті сегментів – дані. Програма може звертатися тільки до даних у сегментах (заштриховані області на рис.2.4).

Змінюючи вміст сегментних регістрів можна пересувати сегменти в межах всієї пам’яті 1М байт. На рис.2.4,б показано розташування сегментів кодів, даних, стека та додаткового сегмента з частковим перекриттям. Цей випадок виникає тоді, коли вміст сегментних регістрів відрізняється менш ніж 64К/16=4096 байт.

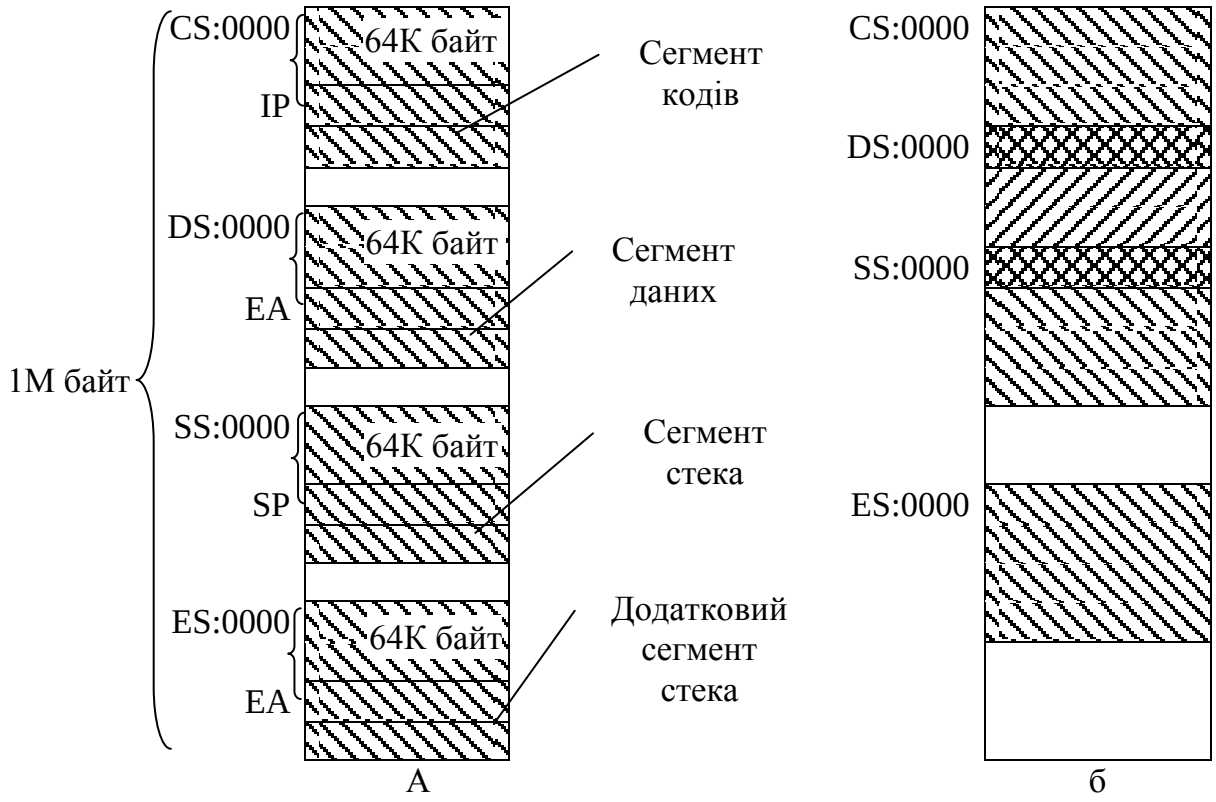


Рис. 2.4. Розташування сегментів у просторі пам’яті 1М байт: а – без перекриття; б – з частковим перекриттям

ЛЕКЦІЯ 6

Програмна модель МП

Програмна модель МП i8086 (рис.2.5) складається з регістрів загального призначення РЗП, сегментних регістрів, вказівника команд і регістра прапорців.

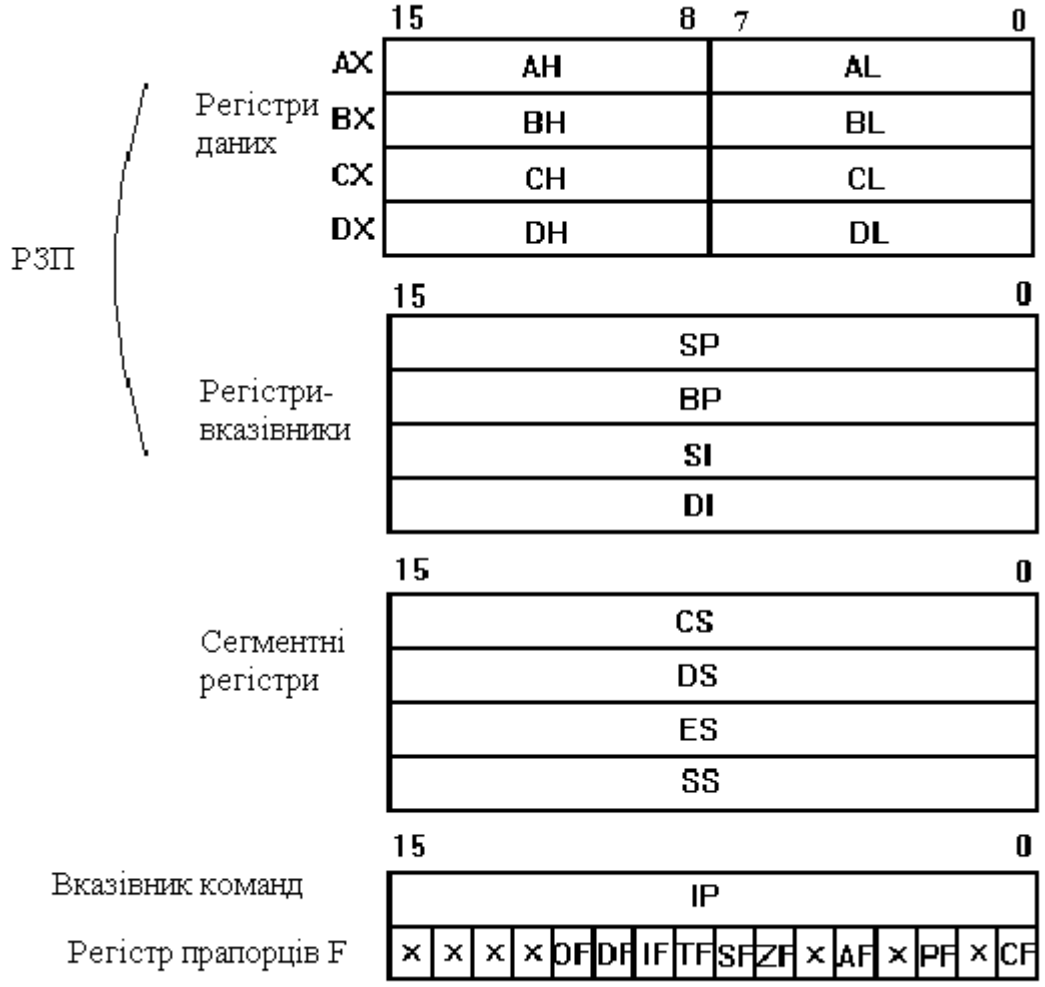


Рис. 2.5. Програмна модель мікропроцесора i8086

Регістри загального призначення РЗП поділяються на регістри даних і регістри-вказівники. До *регістрів даних* відносяться чотири 16-розрядних регістри AX, BX, CX, DX. Кожний з цих регістрів складається з двох 8-розрядних регістрів, які можна незалежно адресувати за символічними іменами AH, BH, CH, DH (старші байти - High) та AL, BL, CL, DL (молодші

байти - Low). *Регістри-вказівники* SP (Stack Pointer – вказівник стека), BP (Base Pointer – базовий регістр), SI (Source Index – індекс джерела), DI (Destination Index – індекс призначення) є 16-розрядними.

Всі РЗП можна використати для зберігання даних, але в деяких командах припускається використання певного регістра за замовчуванням: AX – при множенні, діленні, введенні і виведенні слів; AL – при множенні, діленні, введенні і виведенні байтів, десятковій корекції, перетворенні байтів (команда **XLAT**); AH – при множенні і діленні байтів; BX – при трансляції; CX – як лічильник циклів і вказівник довжини рядків у рядкових командах; CL – для зберігання зміщення з вказанням змінної; DX – при множенні і діленні слів, введенні і виведенні з непрямою адресацією; SP – при операціях зі стеком; SI, DI – при рядкових операціях. На відміну від 8-розрядних МП, регістр SP зберігає зміщення останньої зайнятої комірки стека відносно початку сегмента стека, а повна адреса стека визначається як SS:SP.

Сегментні регістри CS, DS, ES, SS визначають початкові адреси чотирьох сегментів пам'яті. Використання сегментних регістрів визначається типом звернення до пам'яті (табл. 2.2).

Таблиця 2.2. Використання регістрів при адресації пам'яті

Тип звернення до пам'яті	Сегментний регістр		Зміщення
	за замовчуванням	альтернативний	
Вибірка команд	CS	Немає	IP
Стекові операції	SS	Немає	SP
Адресація змінної	DS	CS, ES, SS	EA
Рядок-джерело*	DS	CS, ES, SS	SI
Рядок-приймальник*	ES	Немає	DI
Використання BP при зверненні до стека при читанні/запису даних	SS	CS, ES, DS	EA

*Рядок-джерело і рядок–приймальник – це рядки даних (масиви), які беруть участь у рядкових командах.

Для деяких типів звернень допускається заміна сегментного регістра за замовчуванням на альтернативний (див. табл. 2.2), яка реалізується префіксами команд CS:, DS:, SS:, ES:.

Приклад 2.3. *Переслати вміст комірки пам'яті з адресою DS:1000H в регістр-акумулятор AL.*

Для того, щоб переслати вміст комірки пам'яті у акумулятор, треба використати команду пересилки MOV dst, src, де операндом призначення dst (Destination) буде регістр AL, а операндом джерела інформації src (Source) – комірка пам'яті. Комірка пам'яті позначається квадратними дужками, всередині яких записується зміщення у сегменті, тобто друга логічна адреса. Перша логічна адреса за замовчуванням є вмістом регістра DS.

Після запису мнемоніки команди пересилки MOV записується операнд-призначення, а потім через кому – операнд-джерело. Після операндів через крапку з комою записується коментар до команди.

Таким чином, за командою

MOV AL, [1000H] ; AL←DS:[1000H]

в регістр AL пересилається байт з комірки пам'яті з адресою DS:1000H.

Зазначимо, що перед використанням цієї команди вміст регістра DS повинен бути визначеним.

Приклад 2.4. *Переслати вміст комірки пам'яті з адресою ES:1000H в регістр-акумулятор AL.*

За командою з префіксом ES

MOV AL, ES: [1000H] ; AL←ES:[1000H]

в AL пересилається вміст комірки пам'яті з адресою ES: 1000H.

На відміну від 8-розрядних МП, вказівник команд IP зберігає зміщення в сегменті кодів поточної команди.

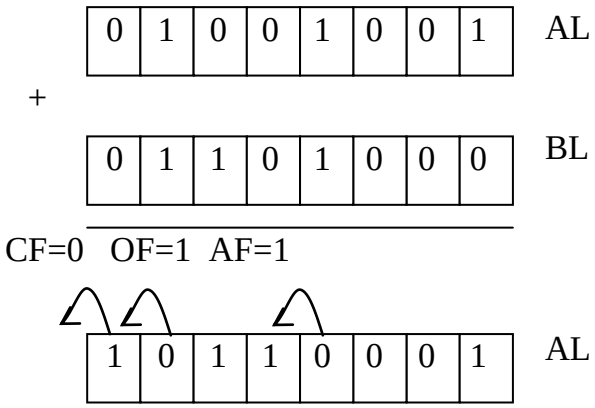
Регістр прапорців зберігає ознаки результатів виконання арифметичних і логічних операцій і керуючі ознаки, які можна встановити або скинути програмно. Типи прапорців подані в табл. 2.3.

Таблиця 2.3. Призначення прапорців

Позначення	Призначення	Розрядність операнду	
		8	16
AF	Auxiliary Flag - прапорець допоміжного перенесення/позики з молодшої тетради в старшу (з розряду D3 в розряд D4). Використовується при десятковій арифметиці	+	-
CF	Carry Flag – прапорець перенесення/позики. Встановлюється при виході результату додавання/віднімання беззнакових операндів за межу діапазону. У командах зсуву прапорець CF фіксує значення старшого біта	+	+
OF	Overflow Flag – прапорець переповнення, встановлюється при виході знакового результату за межи діапазону	+	+
SF	Sign Flag – прапорець знака. Дублює значення старшого біта результату. SF=0 для позитивних чисел і SF=1 - для негативних	+	+
PF	Parity Flag – прапорець паритету (парності). Встановлюється при парному числі одиниць в результаті	+	-
ZF	Zero Flag – прапорець нульового результату. Встановлюється при отриманні нульового результату операції	+	+
DF	Direction Flag – прапорець керування напрямком в рядкових операціях. При DF=1 індексні регістри SI, DI, що беруть участь у рядкових операціях, автоматично декрементуються на кількість байтів операнда, при DF=0 – інкрементуються		
IF	Interrupt-enable Flag – прапорець дозволу переривань. При IF=1 дозволяється виконання маскованих апаратних переривань		
TF	Trap Flag – прапорець трасування (покрокового режиму). При його встановленні після виконання кожної команди викликається внутрішнє переривання 1 (INT 1)		

Приклад 2.5. *Визначити значення прапорців після виконання команди* ***ADD AL, BL*** *(додавання вмісту 8-розрядних регістрів AL, BL; результат передається в AL), якщо в регістрі AL міститься число 49H, а в регістрі BL - 68H.*

Після виконання команди додавання прапорці встановляться таким чином:



AF=1, CF=0, OF=1, SF=1, PF=1, ZF=0.

Пояснимо встановлення прапорців. Результат операції додавання – ненульовий, тому прапорець ZF скинутий, тобто ZF=0. Старший розряд результату дорівнює одиниці, тому SF=1. Кількість одиниць у результаті 4, тобто парне число, тому PF=1. В результаті додавання виникло переповнення із молодшої тетради у старшу (AF=1), у знаковий розряд (OF=1). Переповнення розрядної сітки не відбулося, тому CF=0.

Адресація портів введення/виведення

Простір адрес портів введення/виведення несегментований, займає 64К байт і адресується 16 молодшими розрядами 20-розрядної шини адреси. Порти можуть бути як 8-, так і 16-розрядними. Будь-які два суміжних восьмибітових порти можна вважати шістнадцятибітовим портом аналогічно слову у пам'яті. При цьому для обміну з 8-розрядними портами використовується регістр AL, а з 16-розрядними - регістр AX. Перші 256 портів (з номерами 0-0FFH) можна адресувати за допомогою прямої адресації.

Приклад 2.6. *Ввести інформацію з 8-розрядного порту з адресою 56H у регістр-акумулятор AL.*

Для того, щоб ввести інформацію з 8-розрядного порту з адресою 56H у акумулятор AL, треба виконати команду введення IN (введення). Першим

операндом команди є позначення акумулятора AL, якщо вводиться байт інформації, або AX, якщо вводиться слово. У даному випадку треба використати операнд AL. Другим операндом є номер порту 56H. Таким чином, за командою

IN AL, 56H ; AL←P_8(56H)

відбудеться введення інформації з 8-розрядного порту з адресою 56H до акумулятора.

Зазначимо, що допускається позначення номера порту у квадратних дужках:

IN AL, [56H] ; AL←P_8(56H).

Приклад 2.7. *Вивести інформацію із регістра-акумулятора AX у 16-розрядний порт з адресою 34H.*

Для того, щоб вивести інформацію із акумулятора AX до 16-розрядного порту з адресою 34H, треба виконати команду виведення OUT. Першим операндом команди є номер порту 34H, другим – позначення акумулятора AL, якщо виводиться байт інформації, або AX, якщо виводиться слово. У даному випадку треба використати операнд AX. Таким чином, за командою

OUT 34H, AX ; AX → P_{16}(34H)

відбудетьсявиведення інформації із регістра AX на 16-розрядний порт з адресою 34H.

Всі 64K байт портів адресуються непрямо – за допомогою регістра DX.

Приклад 2.8. *Ввести інформацію з 8-розрядного порту з адресою, що знаходиться у регістрі DX, у регістр-аккумулятор AL.*

Для того, щоб ввести інформацію, треба виконати команду введення IN, першим операндом якої є позначення акумулятора AL, а другим - позначення регістра DX. Таким чином, за командою

IN AL, DX ; AL←P_8(DX)

відбудеться введення інформації у акумулятор AL з 8-розрядного порту з адресою, що знаходиться в регістрі DX. Вміст регістра DX повинен бути визначений до моменту виконання команди введення.

Допускається запис команди з непрямою адресацією у вигляді:

IN AL, [DX]

ЛЕКЦІЯ 7

Типи адресації

У МП і8086 використовуються такі ж основні типи адресації - пряма, регістрова, безпосередня та непряма, однак непряма адресація має наступні різновиди: 1) базова; 2) індексна; 3) базово-індексна.

Базова адресація. Ефективна адреса операнда ЕА обчислюється шляхом складання вмісту базових регістрів BX або BP і зміщення (8- або 16-розрядного знакового числа). В окремому випадку зміщення може бути відсутнє.

Приклад 2.9. *Переслати у регістр-акумулятор AX вміст комірки пам'яті, що розташована у сегменті даних і має ефективну адресу (зміщення у сегменті), яка дорівнює сумі вмісту регістра BX і числа 2000H.*

Для того, щоб переслати вміст комірки пам'яті у акумулятор, треба використати команду пересилки MOV dst, src, де операндом призначення dst (destination) буде регістр AX, а операндом джерела інформації src (source) – комірка пам'яті. Комірка пам'яті позначається квадратними дужками, всередині записується значення ефективної адреси, тобто BX+2000H.

Таким чином, за командою

MOV AX, [BX+2000H] ; AX←DS:[BX+2000H]

у регістр AX пересилається байт з комірки пам'яті з адресою DS: BX+2000H.

Зазначимо, що перед використанням цієї команди вмісти регістрів DS і BX повинні бути визначені.

Індексна адресація. При індексній адресації в якості адреси зміщення використовується вміст індексних регістрів SI або DI та зміщення.

Приклад 2.10. *Переслати у регістр-акумулятор AX вміст комірки пам'яті, що розташована у сегменті даних і має ефективну адресу (зміщення у сегменті), яка дорівнює сумі вмісту регістра SI і числа 5000H.*

За командою

MOV AX, [SI+5000H] ; AX←DS:[SI+5000H]

у регістр AX пересилається байт з комірки пам'яті з адресою DS: SI+5000H.

Перед використанням цієї команди вмісти регістрів DS і SI повинні бути визначені.

Базово-індексна адресація. Ефективна адреса операнда EA дорівнює сумі вмісту базових регістрів BX або BP, індексних регістрів SI або DI та зміщення.

Приклад 2.11. *Переслати у регістр-акумулятор AX вміст комірки пам'яті, що розташована у сегменті даних і має ефективну адресу, яка дорівнює сумі вмісту двох регістрів SI і BX.*

За командою

MOV AX, [SI+BX] ; AX←DS:[SI+BX]

у регістр AX пересилається байт з комірки пам'яті з адресою DS: SI+BX.

Перед використанням цієї команди вмісти регістрів DS, SI і BX повинні бути визначені.

Базова та індексна адресації застосовуються для звернення до елементів одновимірного масиву, базово-індексна - до двовимірного масиву.

Цикли шини процесора

Протягом ***циклу шини*** МП виставляє адресу комірки пам'яті або ПВВ на шину адреси, формує керуючі сигнали читання/запису, а після цього зчитує або записує дані. Окрім циклів ЧИТАННЯ і ЗАПИСУ ПАМ'ЯТІ або ПВВ, існують цикли ПІДТВЕРДЖЕННЯ ПЕРЕРИВАННЯ і ЗАХОПЛЕННЯ

ШИН. Цикл шини може ініціювати не тільки незалежний процесор i8086, але і арифметичний співпроцесор або співпроцесор введення/виведення. Розрізняють цикли шини в мінімальному і максимальному режимах. Часові діаграми циклів ЧИТАННЯ та ЗАПИСУ у мінімальному режимі наведені на рис.2.6.

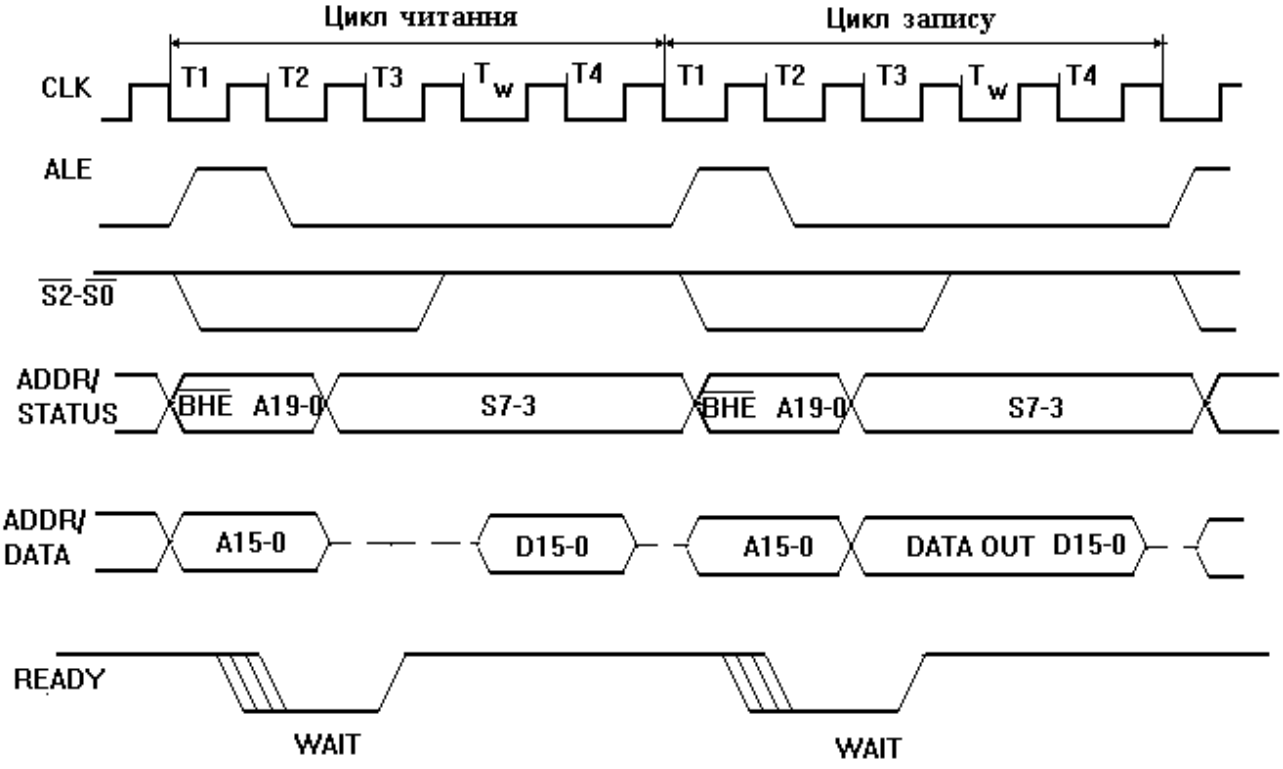


Рис. 2.6. Цикли ЧИТАННЯ і ЗАПИСУ МП i8086 в мінімальному режимі

Цикл шини складається, як мінімум, з чотирьох тактів. Такт визначається як проміжок часу між задніми фронтами двох сусідніх імпульсів CLK. Будь-який цикл шини може бути необмежено розтягнуто за допомогою сигналу готовності READY, при цьому процесор вводить необов’язкові такти очікування T_w.

Цикли звернення до порту відрізняються від циклів пам'яті тим, що старші розряди шини адреси мають нульове значення (при непрямій

адресації за допомогою DX сигнали на лініях A19-A16 набувають значень L-рівня, при прямій адресації сигнали на лініях A19-A8=0.

Цикл ПІДТВЕРДЖЕННЯ ПЕРЕРИВАННЯ формується аналогічно циклу ЧИТАННЯ порту, але замість активного сигналу $\overline{IOR}$ активним є сигнал $\overline{INTA}$, а шина адреси процесором не керується.

ЛЕКЦІЯ 8

Типи переривань

Процесор i8086 може обробляти до 256 типів переривань. Кожному перериванню відповідає свій вектор – подвійне слово, що містить адресу CS: IP підпрограми, що викликається. Під вектори переривань у загальному просторі адрес пам'яті відводиться 1К байт, починаючи з нульової адреси (рис.2.7).

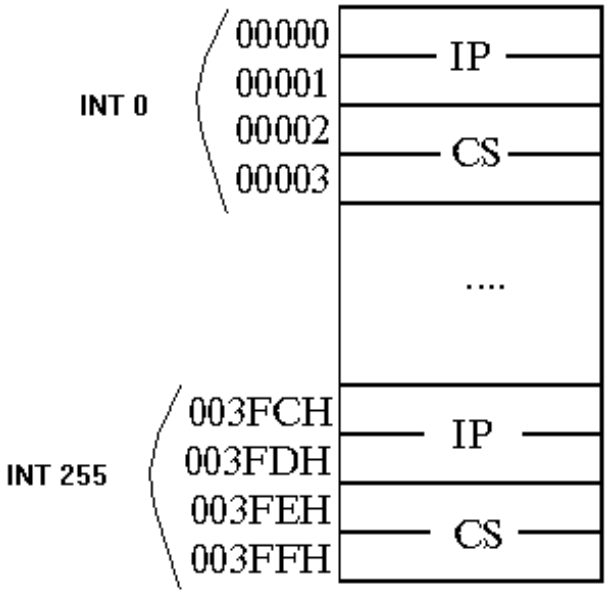


Рис. 2.7. Карта векторів переривань

При переході на підпрограму обробки переривань INT *n* (*n* – тип переривання) процесор переміщує у стек вміст регістрів IP, CS, регістр прапорців F і скидає прапорець дозволу переривання IF; обчислює адресу 4 х

n і перше слово за цією адресою переміщує в IP, друге – в CS. Послідовність цих дій еквівалентна командам:

```
PUSHF                  ; запам‘ятовування у стеку прапорців
CALL  FAR i_proc_4n    ;далекий виклик підпрограми обробки
                       ; переривання
```

Скидання прапорця переривання IF не дозволяє перервати виконання підпрограми обробки переривання до її завершення або виконання команди дозволу **STI**. Останньою командою підпрограми обробки переривання є команда **IRET.** За цією командою процесор витягає зі стека адресу повернення (адресу команди, наступної за командою **INT**) і вміст регістра прапорців.

Типи переривань наведено на рис. 2.8. Переривання поділяються на *зовнішні апаратні* і *внутрішні*. Запити IRQi зовнішніх апаратних переривань надходять на систему переривань або на вивід немаскованого переривання NMI мікропроцесора. Система переривання формує сигнал INTR маскованого переривання МП. Зазначимо, що масковане переривання відрізняється від немаскованого тим, що перше може бути заборонено програмним шляхом – командою скидання прапорця дозволу переривань IF. В цьому разі при надходженні запитів переривання вони будуть ігноруватися. Внутрішні переривання процесора поділяються на *програмні* і *апаратні*. Джерелами *внутрішніх програмних* переривань (див.рис.3.19) є: 1) помилка ділення (тип 0); 2) покроковий режим (тип 1); 3) команда INTO (тип 4).

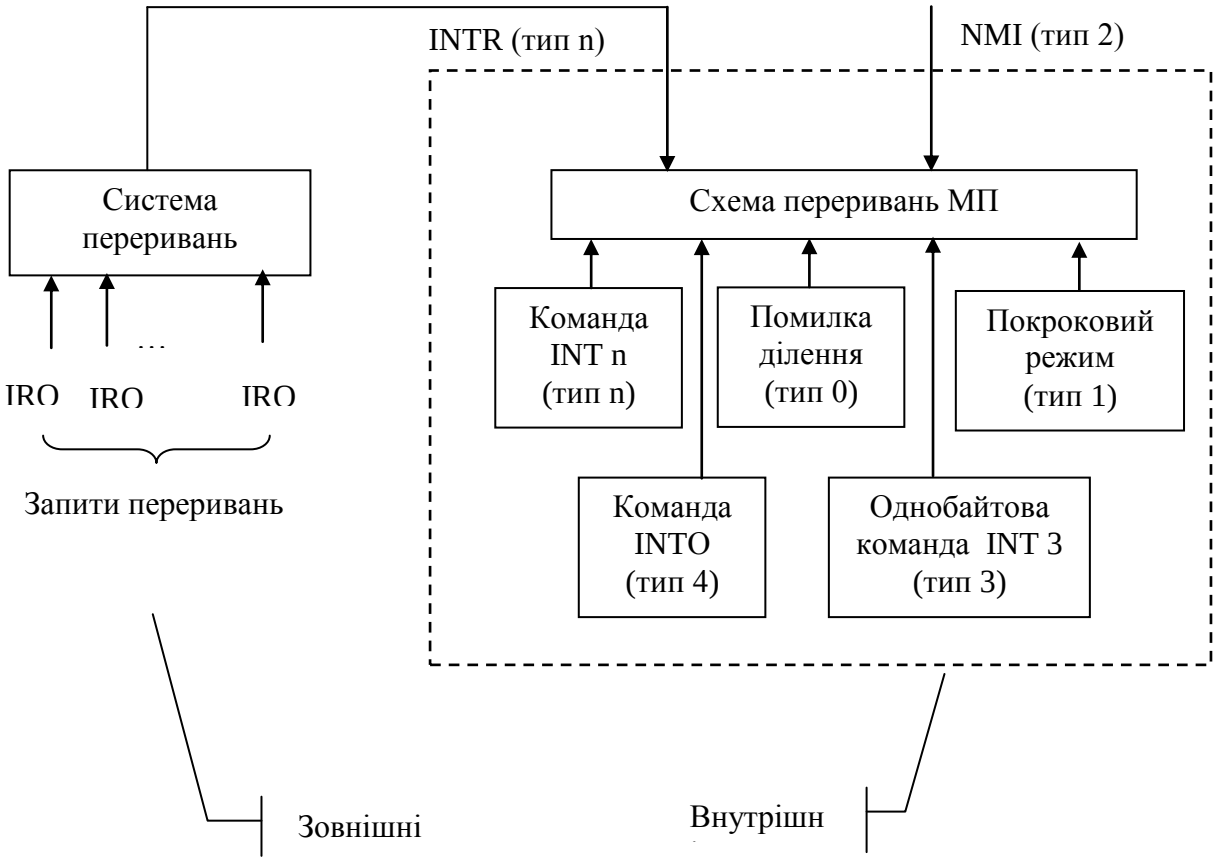


Рис. 2.8. Типи переривань

Внутрішні програмні переривання **INT n** і **INT 3** виконуються за командами переривання і дозволяють викликати підпрограми обробки переривань (наприклад, сервісні підпрограми BIOS і DOS) без застосування дальніх викликів. Переривання **INT 3** є однобайтовою командою на відміну від **INT n** і звичайно використовується для передачі керування підпрограмі-налагоджувачу. Виконання програмних переривань не залежить від прапорця дозволу переривань IF.

Внутрішні апаратні переривання процесора виникають в наступних особливих випадках:

- при діленні на 0 (тип 0);
- при встановленому прапорці трасування (тип 1). В цьому випадку переривання відбувається після виконання кожної команди;

- після команди **INTO** (тип 4), якщо встановлений прапорець переповнення OF.

Апаратні переривання виникають при активному рівні сигналів на контактах МП - NMI (немасковане переривання – тип 2) і INTR (масковані, типи 5-255). Масковані переривання виконуються при встановленому прапорці IF. При переході до підпрограми обробки апаратного переривання процесор виробляє два цикли підтвердження переривання, що йдуть один за одним, в яких генерується сигнал $\overline{\text{INTA}}$. За другим імпульсом $\overline{\text{INTA}}$ контролер переривань передає по шині даних номер вектора переривання *n*. Далі дії процесора аналогічні виконанню програмного переривання. Обробка поточного переривання може бути перервана немаскованим перериванням або іншим маскованим перериванням вищого пріоритету у тому разі, якщо підпрограма-обробник встановить прапорець дозволу переривання IF. Немасковане переривання виконується незалежно від стану прапорця IF.

Розділ 3. ПОБУДОВА ОДНОПРОЦЕСОРНИХ СИСТЕМ НА ОСНОВІ 16-РОЗРЯДНИХ МІКРОПРОЦЕСОРІВ

ЛЕКЦІЯ 9

Модуль центрального процесора

Для побудови модуля центрального процесора необхідно забезпечити синхронізацію роботи системи та узгодження роботи центрального процесора з системною шиною.

Схема синхронізації. Для синхронізації використовується генератор тактових імпульсів (ГТІ) і8284, який виробляє сигнали синхронізації для центрального процесора і периферійних пристроїв, а також синхронізує зовнішні сигнали готовності READY і початкового встановлення RESET з тактовими сигналами МП. Генератор тактових імпульсів (рис.3.25) містить подільники частоти на 3 і на 2 та логіку керування сигналів скидання і готовності. Робота генератора стабілізується кварцовим резонатором, який під'єднується до входів X1, X2 (див. рис. 3.1). Вхід TANK використовується для додаткового під'єднання паралельного резонансного LC-контура, що дозволяє працювати на вищих гармоніках кварцового резонатора. При цьому опорна частота генератора визначається параметрами контура і дорівнює $\frac{1}{2\pi\sqrt{LC}}$. Вхід $F/\overline{C}$ дозволяє обрати зовнішній ($F/\overline{C}=1$) або внутрішній ($F/\overline{C}=0$) генератор. У випадку обрання зовнішнього генератора з частотою імпульсів F_{EFI} він під'єднується до входу EFI.

Схема формування тактових імпульсів виробляє сигнали: CLK – тактової частоти F_{CLK} для ЦП, PCLK – тактової частоти F_{PCLK} для керування периферійними ВІС, OSC – тактової частоти генератора, які необхідні для керування пристроями та контролю частоти. Частоти цих сигналів пов'язані співвідношеннями: $F_{OSC}=3F_{CLK}= 6F_{PCLK}$, в режимі внутрішнього генератора та

$F_{EFI}=3F_{CLK}= 6F_{PCLK}$ в режимі зовнішнього генератора. Вихідний сигнал CLK формується одним з трьох способів: 1) з коливань основної частоти кварцового резонатора, під'єднаного до входів X1 та X2; 2) з третьої гармоніки кварцового резонатора, що виділяється LC-фільтром, під'єднаним до входу TANK; 3) від зовнішнього генератора, під'єднаного до входу EFI. Схема формування тактових імпульсів має вхід зовнішньої синхронізації CSYNC, за допомогою якого можна синхронізувати роботу декількох ГТІ, які входять до складу системи. Сигнал CSYNC впливає також на подільник частоти на 2 – при CSYNC=0 робота подільника зупиняється, при CSYNC=1 – відновлюється.

Вихідний сигнал генератора READY використовується для підтвердження готовності до обміну. Високий рівень цього сигналу вказує на наявність даних на ШД. Схема формування сигналу READY побудована так, щоб спростити вмикання системи у інтерфейсну шину стандарта Multibus, і містить дві ідентичні пари сигналів RDY1, $\overline{\text{AEN1}}$ та RDY2, $\overline{\text{AEN2}}$, об'єднаних схемою АБО. Сигнали RDY1 та RDY2 формуються елементами, які входять до складу системи, та свідчать про їхню готовність до обміну. Сигнали $\overline{\text{AEN1}}$ та $\overline{\text{AEN2}}$ дозволяють формування сигналу READY за сигналами RDY1 та RDY2, підтверджуючи адресацію елементів.

Схема формування вихідного сигналу скидання RESET має на вході тригер Шмітта, а на виході – тригер, що формує фронт сигналу RESET по зрізу сигналу CLK. Звичайно до входу $\overline{\text{RES}}$ під'єднується RC-ланцюг, що забезпечує автоматичне формування сигналу при вмиканні джерела живлення.

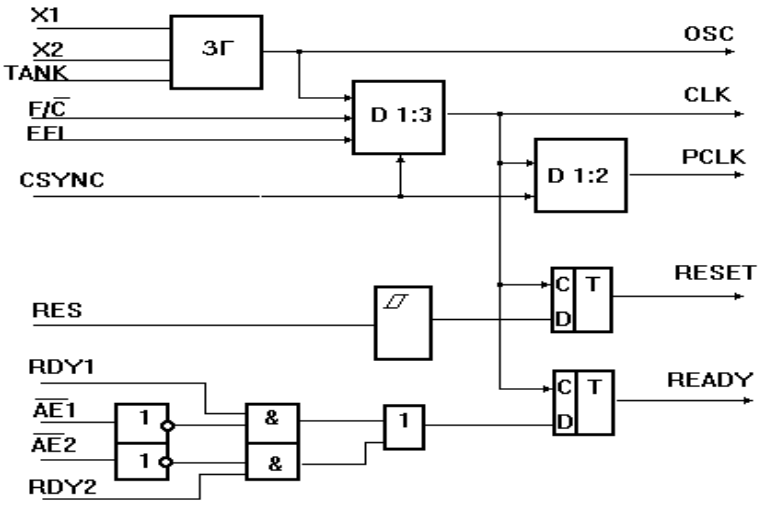


Рис. 3.1. Структурна схема ВІС генератора і8284

Графічне позначення генератора наведено на рис. 3.2.

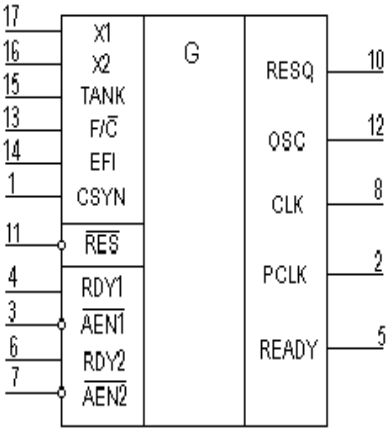


Рис. 3.26 Графічне позначення ВІС генератора і8284

Інтерфейс ЦП з системною шиною виконує такі функції:

- Демультиплексування шини адреси/даних (розподіл її на шину адреси AB та шину даних DB);

- Буферизація шин (збільшення навантажувальної здатності ліній шин, забезпечення можливості їхнього переходу в z-стан);
- Формування сигналів керування.

Виконання першої функції здійснюється за допомогою регістрів-замків, наприклад, буферних регістрів і8282, і8283. Узагальнена структурна схема регістра-замка (рис. 3.3) містить вісім D-тригерів з вихідними схемами SW, які мають три стани. Сигнали запису інформації STB і дозволу вибірки $\overline{OE}$ є спільними для всіх тригерів ВІС. У буферному регістрі і8282 (рис. 3.28) до схем SW під'єднуються прямі виходи D-тригерів, а в буферному регістрі і8283 (рис. 3.29) – інверсні виходи. При сигналі високого рівня на вході STB стан вхідних ліній DI7-DI0 передається на вихідні лінії DO7–DO0. Запитування інформації в D-тригерах здійснюється по зрізу сигналу STB. Малий вхідний і достатньо великий вихідний струми дозволяють використовувати ВІС буферних регістрів як регістри-замки або шинні формувачі. При використанні буферних регістрів як шинних формувачів вхід STB під'єднується до виводу живлення +5 В через резистор опором 1 кОм, а вхід $\overline{OE}$ - до спільної шини.

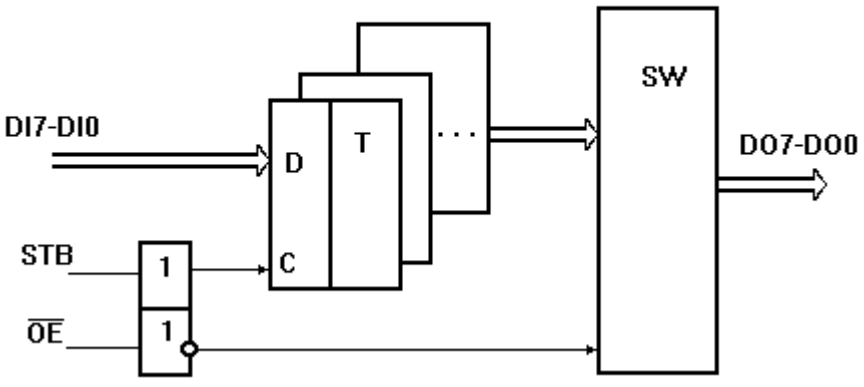


Рис. 3.3.Структурна схема буферного регістра

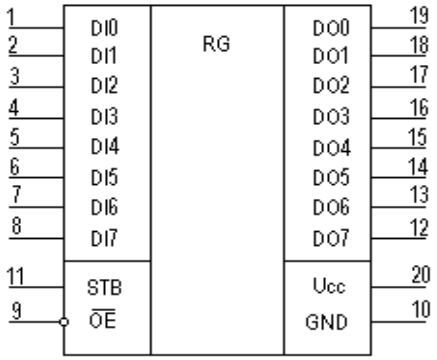


Рис. 3.4. Графічне позначення буферного регістра i8282

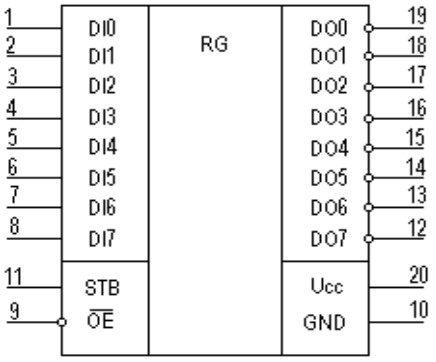


Рис.3.5. Графічне позначення буферного регістра i8283

Для збільшення навантажувальної здатності двонапрямленної шини даних використовують 8-розрядні шинні формувачі (ШФ) i8286, i8287. Формувач i8286 не інвертує дані, а i8287 інвертує.

Структурна схема шинного формувача (рис. 3.6) містить вісім однакових функціональних блоків з трьома станами і спільними сигналом керування напрямком передачі T та сигналом дозволу передачі $\overline{OE}$.

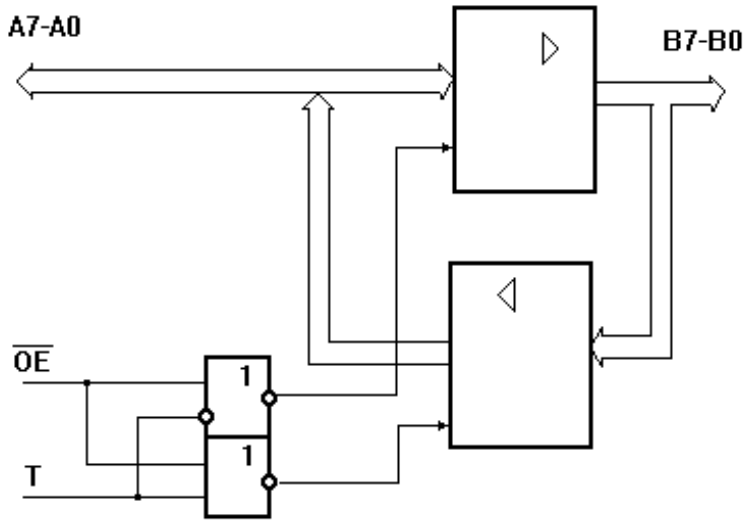


Рис. 3.6. Структурна схема шинного формувача

При низькому рівні сигналу T (T=0) здійснюється передача даних з ліній B7-B0 на лінії A7-A0, при високому рівні сигналу (T=1) – передача з ліній A7-A0 на лінії B7-B0. При $\overline{OE}$=0 передача дозволена, при $\overline{OE}$=1 – заборонена. Графічні позначення формувачів i8286, i8287 наведені на рис. 3.7 і 3.8 відповідно.

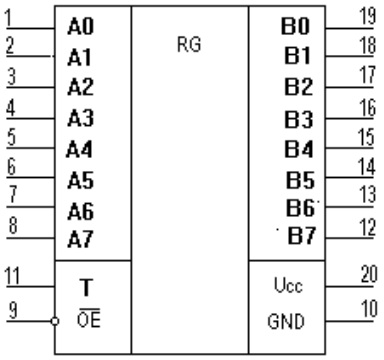


Рис. 3.7. Графічне позначення шинного формувача i8286

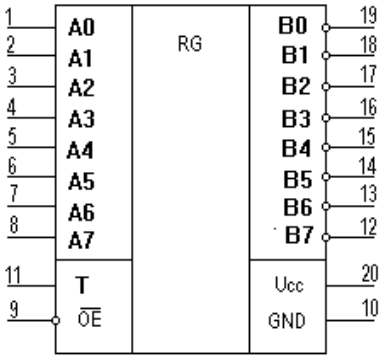


Рис.3.8. Графічне позначення шинного формувача i8287

На рис. 3.9 показаний приклад функціональної схеми модуля центрального процесора для однопроцесорних систем. Мікропроцесор і8086 ввімкнений у мінімальному режимі. Схема синхронізації реалізована на базі ВІС тактового генератора і8284, на вхід RDY1 якої подається сигнал готовності зовнішніх пристроїв або пам'яті до обміну. У мінімальному режимі використовується одна шина, тому вхід RDY2 під'єднаний через резистор до виводу живлення. Демультиплексування шини адреси/даних і шини адреси/стану на дві шини здійснюється за допомогою трьох буферних регістрів і8282. Відзначимо, що сигнал дозволу старшого байта $\overline{BHE}$, що з'являється водночас з дійсною адресою, також фіксується в одному з розрядів регістрів/замків. Сигнали $\overline{BHE}$ і A0 використовуються для вибірки банків системи пам'яті. Формувач 16-розрядної шини даних виконаний на двох ВІС шинних формувачів і8286.

У мінімальному режимі процесор формує керуючі сигнали шин формувачів і регістрів-замків, а також сигнали $M/\overline{IO}, \overline{R}, \overline{W}$, з яких за допомогою логічних елементів формуються чотири сигнали керування записом/читанням для пам'яті і пристроїв введення/виведення. Шини адреси, даних та керування переводяться у z-стан сигналом $\overline{BUSEN}$, що формує контролер прямого доступу до пам'яті.

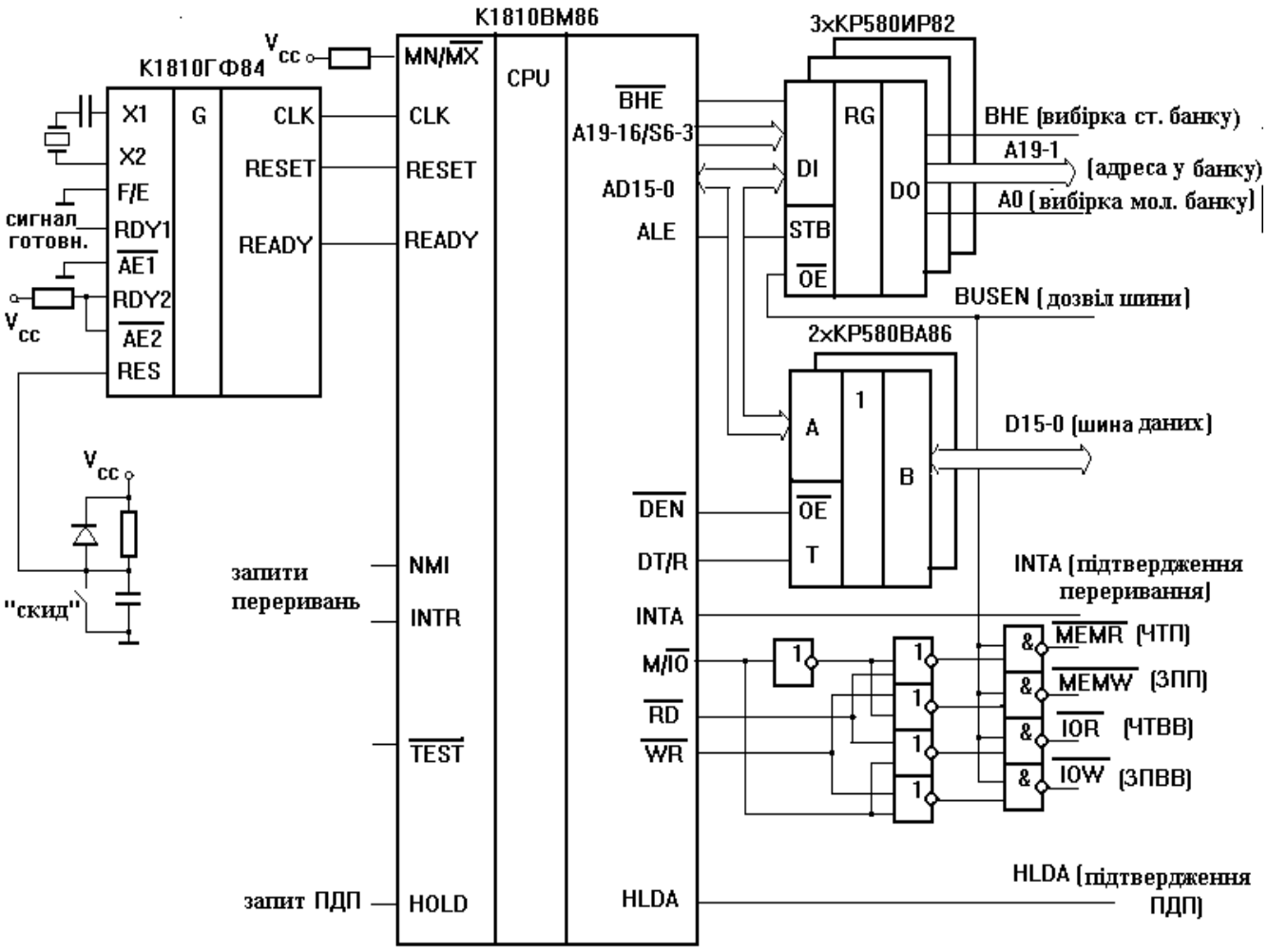


Рис. 3.9. Функціональна схема модуля центрального процесора

ЛЕКЦІЯ 10

Система пам’яті.

Побудова модулів ПЗП Основною складовою частиною ПЗП є ***елемент пам'яті (ЕП)***, який зберігає 1 біт інформації. Елементи пам’яті об'єднані у матрицю накопичувача інформації. Сукупність з *n* елементів пам’яті, у якій розміщується *n*-розрядне слово, називається ***коміркою пам'яті (КП)***, при цьому величина *n* визначає ***розрядність комірки***. Кількість комірок пам'яті дорівнює 2^m, де *m* – кількість адресних входів, а інформаційна ємність мікросхеми - 2^mx*n* біт. Кожна комірка пам’яті має свою

адресу. Більшість ПЗП мають словникову організацію, тобто припускають паралельне зчитування *n* розрядів слова Dn-1-D0.

Для зчитування інформації з комірки необхідно на адресні входи мікросхеми ПЗП подати код адреси Am-1-A0, які через дешифратор (ДШ) рядків обирають відповідну комірку. Зчитування інформації відбувається при активному (нульовому) рівні сигналу $\overline{CS}$. При $\overline{CS}$=1 виходи D_{n-1}-D_0 знаходяться у третьому (високоімпедансному) стані – z-стані, що відображається у позначенні ВІС знаком ◊. Якщо ЗП має виходи з трьома станами або з відкритим колектором, то вихід ВІС ПЗП може бути під'єднаний безпосередньо до шини. Якщо на виході ВІС ПЗП немає активних пристроїв, використовуються *підтягуючі* резистори, вмикання яких забезпечує високий рівень вихідного сигналу. Якщо ЗП не має виходів з трьома станами, то необхідно застосовувати мікросхеми шинних формувачів, наприклад, ВІС і8286 або К580ВА86.

За способом програмування, тобто занесення інформації, розрізняють наступні типи ПЗП: масочні; однократно програмовні; багаторазово програмовні з ультрафіолетовим стиранням; багаторазово програмовні з електричним стиранням, або флеш-пам'ять.

Розглянемо побудову модуля постійної пам'яті для мікропроцесорних систем на базі ***16-розрядних*** процесорів, які можуть оперувати як з 8-, так і з 16-розрядними комірками пам'яті. Для використання 8-розрядних ВІС у модулях пам'яті 16-розрядних процесорів, наприклад, з інформаційною ємністю 1 Мх8, постійна пам'ять виконується у вигляді двох ***банків*** по 512К байт кожний. Один з банків під'єднується до молодшої половини шини даних, тобто до розрядів *D7-D0* і називається ***молодшим***, другий – до старшої половини шини даних (розряди *D15-D8*) і називається ***старшим***. Молодший банк містить байти з парними адресами (*A0=0*), старший – з непарними (*A0=1*). Для адресації байта всередині банку використовуються адресні розряди *A19-A1*. Зчитування з ПЗП організовано таким чином, що

при зверненні до ПЗП на шину даних мікропроцесора завжди надходять два байти, тобто зчитується вміст обох банків одночасно. У разі необхідності процесор може обирати один потрібний байт з двох. На рис.3.10 наведена система пам'яті, виконана у вигляді двох банків. Кожний з банків може бути виконаний за структурною схемою модуля ПЗП для 8-розрядних процесорів, розглянутих вище.

У мікропроцесорних системах з 32-розрядною шиною даних модуль ПЗП виконується у вигляді 4-х банків. Зчитування інформації відбувається одночасно з усіх чотирьох банків, після чого мікропроцесор обирає 1-, 2- або 4-байтове слово залежно від команди, що виконується.

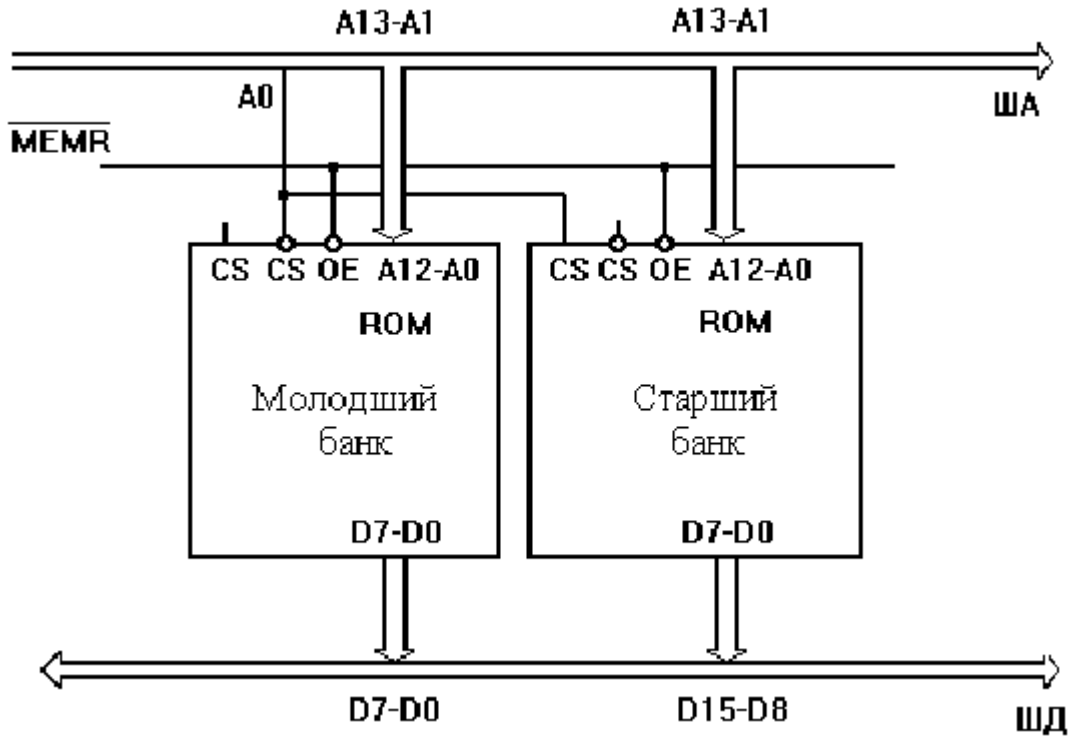


Рис.3.10. Модуль ПЗП в 16-розрядних мікропроцесорних системах

Побудова модулів ОЗП статичного типу Елементом пам'яті ОЗП статичного типу є тригер. Кожний ЕП має свою адресу. Для того, щоб зчитати інформацію з ЕП або записати в нього нове значення, його необхідно обрати шляхом подання на ВІС ОЗП відповідної адреси A_{m-1}-A_0. Частина

адресних розрядів надходить на ДШ рядків, а частина – на ДШ колонок. Таким чином, визначається положення ЕП у матриці накопичувача. Запам'ятовувальний пристрій, схему якого наведено на рис 5.11, припускає звернення до будь-якого ЕП в довільному порядку. Такий ЗП називають ***ОЗП з довільною вибіркою*** (RAM - Random Access Memory - пам'ять з довільним доступом).

Інформаційна ємність мікросхеми ОЗП, так само, як і для ПЗП, дорівнює 2^mxn біт.

Приклад 3.1. *Розробити схему модуля ОЗП інформаційною ємністю 16Кх16 для мікропроцесорної системи з 16-розрядним МП.*

У схемній реалізації модуля ОЗП повинне бути забезпечене зчитування/запис як 8-розрядних, так і 16-розрядних даних. Модуль ОЗП складається з двох банків – молодшого та старшого (рис. 3.10). Адресні розряди *A13-A1* під'єднані до адресних входів *A12-A0* обох банків. Вибірка банків здійснюється одиничним значенням сигналу на виході дешифратора DC, сигналами *A0* для вибірки молодшого банку або $\overline{BHE}$ (Bus High Enable - дозвіл старшого байта шини) для вибірки старшого банку. Одиничний рівень сигналу на виході DC з'являється при надходженні на шину адреси ША відповідної адреси ОЗП. Сигнал $\overline{\text{MEMR}}$ або $\overline{\text{MEMW}}$ визначає виконання циклу ЧИТАННЯ ПАМ'ЯТІ або ЗАПИС У ПАМ'ЯТЬ відповідно.

Комбінація значень сигналів *A0* і $\overline{BHE}$ визначає чотири можливих випадки звернення до пам'яті: 1) ***вибірка байта за парною адресою***; 2)***вибірка байта за непарною адресою***; 3)***вибірка слова за парною адресою***; 4)***вибірка слова за непарною адресою***.

При ***вибірці байта за парною адресою*** *A0=0,* $\overline{BHE}$ *=1*. Байт з парною адресою передається по лініях *D7-D0*, тобто відбувається зчитування або запис байта. У випадку запису байта в молодший банк інформація в старшому банку захищена від стирання, тобто одиничне значення сигналу

$\overline{BHE}$ забороняє звернення до старшого банку.

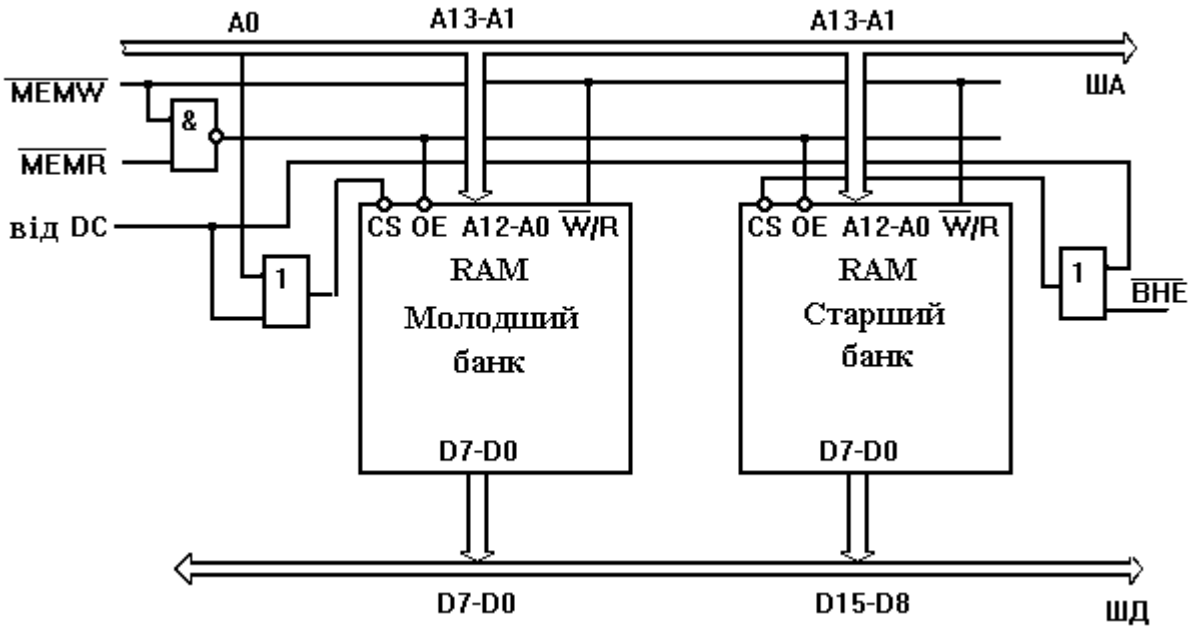


Рис. 3.10. Модуль ОЗП в 16-розрядних мікропроцесорних системах

При ***вибірці байта за непарною адресою*** A0*=1*, $\overline{BHE}$ *=0*. Наприклад, за командою

MOV BL, BYTE PTR [10001H]

вміст комірки пам'яті з адресою *DS:10001H* пересилається в молодшу половину 16-розрядного регістра *BX*, тобто у 8-розрядний регістр *BL*. При цьому вміст комірки пам'яті записується у розряди *D15-D8*, тобто старшу половину шини даних, потім, в процесі виконання команди, на молодшу половину внутрішньої 16-розрядної шини мікропроцесора, а після цього - в регістр *BL*. Цей процес, що називається ***маршрутизацією байта,*** відбувається автоматично і непомітний для програміста.

При ***вибірці слова за парною адресою*** $\overline{BHE}$ *=0, A0=0***.** В цьому випадку водночас обираються два банки, і 16-розрядне слово передається по лініях *D15-D0* за один цикл шини.

Якщо слово має непарну адресу, то його молодший байт розміщений в старшому банку пам'яті, старший байт - в молодшому банку. При ***вибірці слова за непарною адресою*** спочатку *A0=1,* $\overline{BHE}$ =0, по лініях шини *D15-D8* передається молодший байт. Після цього генеруються сигнали *A0=0,* $\overline{BHE}$ *=1*, здійснюється *інкремент* (збільшення на одиницю) повної адреси *A19-A0*, старший байт слова передається з молодшого банку або у молодший банк по лініях шини *D7-D0*. Таким чином, вибірка слова за непарною адресою вимагає два цикли шини. Тому слова доцільно розміщувати за парними адресами, особливо при організації операцій зі стеком.

ЛЕКЦІЯ 11

Інтерфейс введення-виведення

Одним з найважливіших завдань проектування мікропроцесорних систем є організація взаємодії з *зовнішніми пристроями* - джерелами і приймачами даних. Прикладами ***пристроїв введення/виведення*** (ПВВ), що є і джерелами, і приймачами інформації, є накопичувачі на гнучких та твердих магнітних дисках. До пристроїв введення відносяться перемикачі, клавіатура, аналого-цифрові перетворювачі, датчики двійкової інформації, а до пристроїв виведення - індикатори, світлодіоди, дисплеї, друкувальні пристрої, цифро-аналогові перетворювачі, транзисторні ключі, реле, комутатори. ПВВ відрізняються: розрядністю даних; швидкодією; протоколами, тобто визначеним порядком обміну; керуючими сигналами. Зміна даних у ПВВ відбувається у довільний або чітко визначений момент часу. З'єднання ПВВ із системною шиною мікропроцесорної системи здійснюється за допомогою ***інтерфейсу введення/виведення***, який вирішує задачу узгодження ПВВ з системною шиною МПС. Звичайно інтерфейс

складається з одного або декількох ***портів введення/виведення*** та схем керування ними.

При проектуванні інтерфейсу введення/виведення необхідно забезпечити:

1) зберігання інформації, яка надходить від ПВВ;

2) доступ до інформації з боку МП;

2) керування обміном;

3) перетворення форматів даних.

Зберігання інформації та доступ до неї з боку МП. Введення та виведення інформації виконується за допомогою портів введення/виведення, які являють собою 8- або 16-розрядні регістри зі схемами вибірки та керуванням читанням/записом. Як порти можуть бути використані буферні регістри, наприклад, і8282, і8285, КР580ИР82, КР589ИР12, КР580ВВ55. Використання регістра КР580ИР82 для з'єднання з пристроєм введення та пристроєм виведення наведено на рис.3.11,а і б відповідно.

Якщо регістр використовується як порт введення (див. рис. 6.1,а), то дані від пристрою введення надходять у регістр по лініях DI7-DI0 і записуються за стробом STB. Вихідні дані DO7-DO0 порту надходять у мікропроцесорну систему по шині даних. МП формує також сигнал керування читанням і вибіркою порту, який надходить на вхід $\overline{\text{OE}}$. Якщо регістр використовується як порт виведення (рис. 6.1,б), то дані від МП надходять по шині даних на входи DI7-DI0 порту і супроводжуються сигналами керування записом і вибірки ВІС. Вихідні дані DO7-DO0 порту надходять у пристрій виведення.

Введення або виведення даних можна здійснювати двома способами:

1) з використанням окремого адресного простору ПВВ;

2) з використанням спільного з пам'яттю адресного простору, тобто з відображенням на пам'ять.

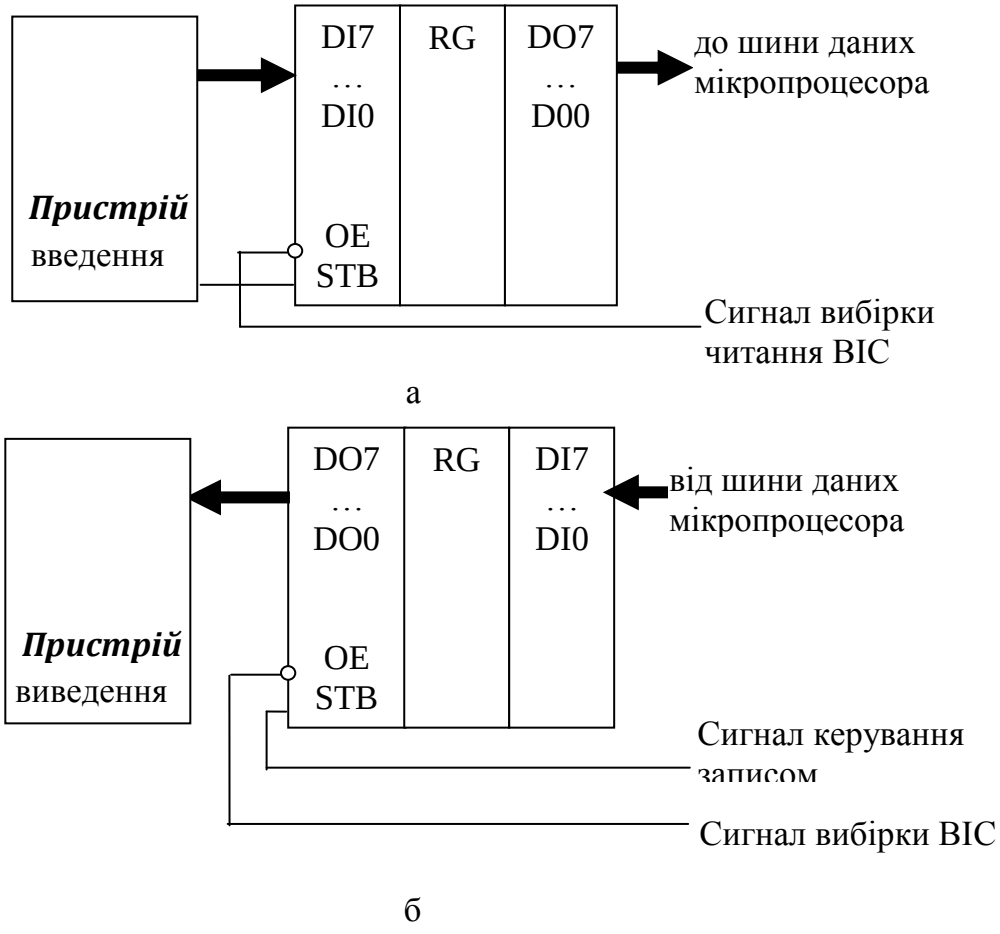


Рис.3.11. Використання регістра КР580ИР82 для з'єднання з: а - пристроєм введення; б - пристроєм виведення

У першому випадку введення і виведення даних виконується за командами **IN** та **OUT**.

Керування обміном. Існують три способи керування обміном:

1) програмний обмін;

2) обмін за перериванням;

3) обмін у режимі ПДП.

Програмний обмін ініціюється мікропроцесором та здійснюється під його керуванням. Розрізняють *простий програмний обмін* та *програмний обмін за стробом готовності*. При простому програмному обміні вважається, що ПВВ в будь-який момент готовий до обміну за командами IN або OUT. При обміні за стробом готовності ПВВ сповіщає про свою

готовність до обміну стробом. Наприклад, видача 8-розрядних даних супроводжується дев'ятим бітом - стробом. При такому обміні схема інтерфейсу містить тригер або порт керування для зберігання інформації про готовність зовнішнього пристрою до обміну. Процесор опитує відповідний розряд порту керування для визначення стану готовності зовншінього пристрою.

Приклад 3.2. *Розробити функціональну схему введення і виведення 8-розрядних даних за стробом готовності. Адреса порту введення - 02H, порту керування – 03H, порту виведення – 04H.*

Функціональна схема обміну за стробом готовності подана на рис. 3.12. Схема містить: пристрій введення, під'єднаний до порту введення; пристрій виведення, під'єднаний до порту виведення; порт керування для зберігання сигналів готовності пристроїв введення і виведення. Пристрій введення має 8 інформаційних вихідних ліній і 1 вихідну лінію стробу супроводження даних. Поява цього стробу сигналізує про те, що дані на інформаційних линіях є дійсними (коректними). Пристрій виведення має 8 інформаційних вхідних ліній і 1 вихідну лінію стробу підтвердження прийому даних. Поява цього стробу сигналізує про те, що дані прийняті пристроєм і МП може передавати нову порцію даних. Порт керування зберігає інформацію про строби від двох пристроїв.

Програма введення за стробом готовності буде мати вигляд:

```
M1:     IN AL,03              ; AL ←порт керування (адреса 03)
        AND AL, 00000001B     ; маскування усіх розрядів, крім D0
        JZ M1                 ; якщо D0=0 (порт не готовий), то на M1
        IN AL, 02             ; інакше – введення інформації з порту
                                  введення (адреса 02)
```

Програма виведення за стробом готовності має вигляд:

```
M2:     IN AL,03              ; AL ←порт керування (адреса )03
        AND AL, 00000010B     ; маскування усіх розрядів, крім D1
        JZ M2                 ; якщо D1=0 (порт не готовий), то на M2
        OUT 04, AL            ; інакше - виведення інформації на порт
                                  виведення (адреса 04)
```

Якщо ПВВ має вбудований апаратний засіб для визначення готовності до обміну, стан пристрою характеризується прапорцем готовності READY або прапорцем готовності/зайнятості READY/BUSY. Інформація про готовність пристроїв належить до ***статусної інформації*** і входить до складу ***слова стану*** пристрою. Іноді стан готовності і зайнятості ідентифікується окремими прапорцями: READY і BUSY. Прапорець READY замінює біт порту керування.

На рис. 3.13 наведено схему алгоритму програмного обміну даними за значенням прапорця READY. Якщо ПВВ не готовий до обміну, то мікропроцесор знаходиться у режимі програмного очікування готовності зовнішнього пристрою, виконуючи команди блоків 1 і 2. Після виявлення стану готовності мікропроцесор передає дані по командах блока 3, а потім працює з продовженням основної програми.

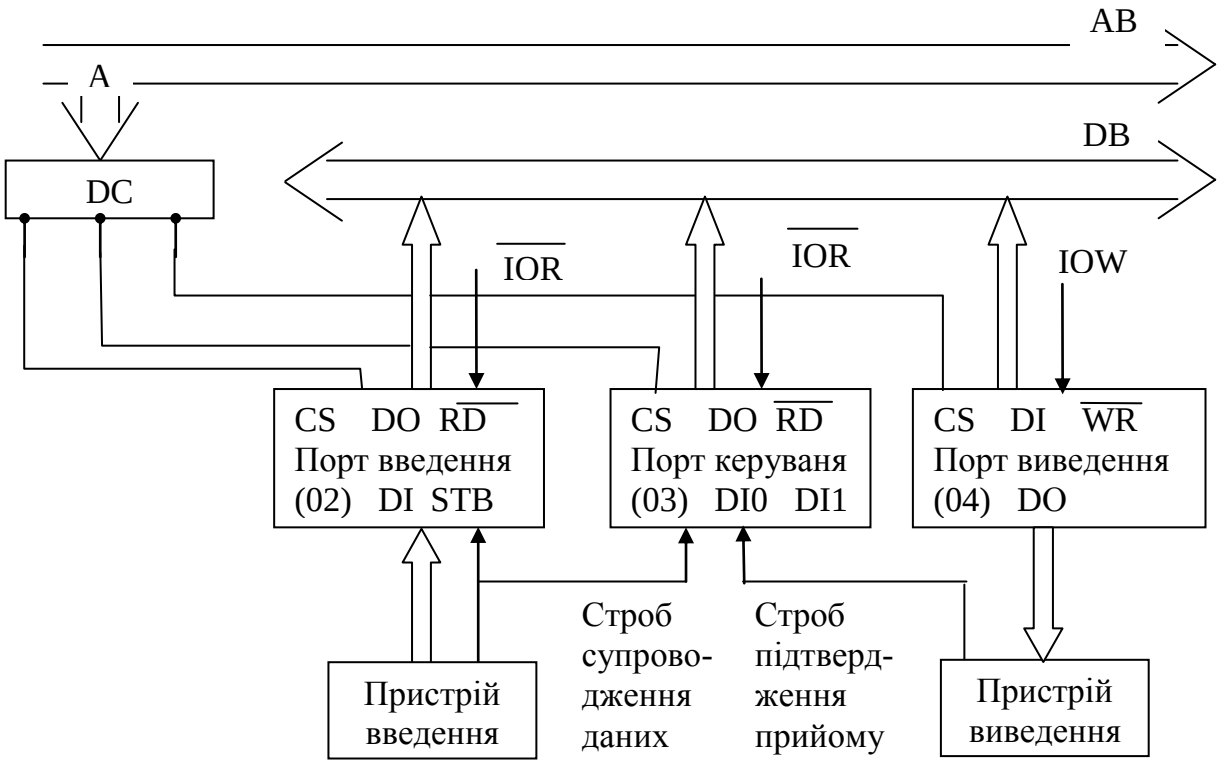


Рис. 6.3. Схема введення/виведення даних за стробом готовності

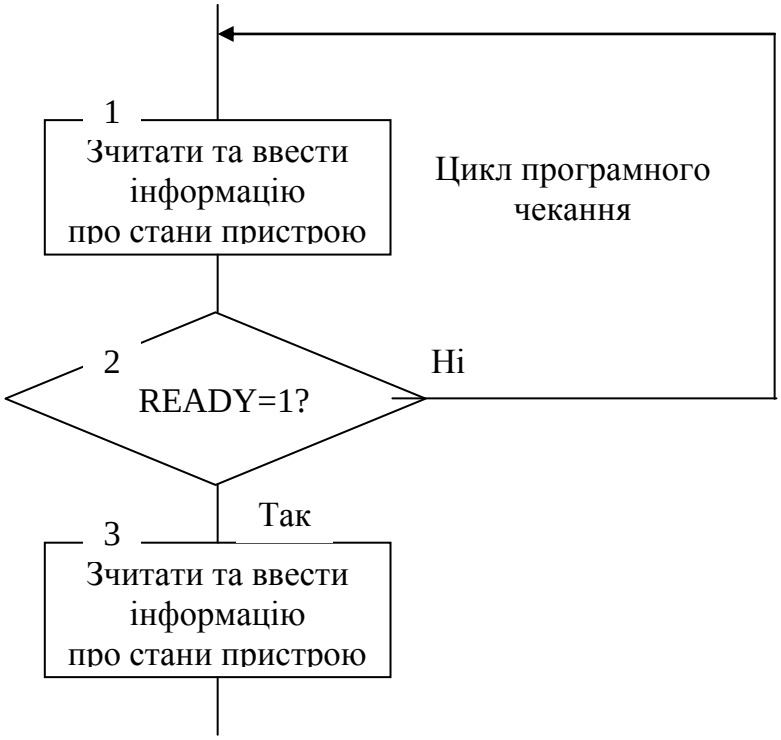


Рис. 3.13. Алгоритм введення даних за стробом

На читання мікропроцесором статусної інформації і її аналіз витрачається декілька циклів роботи мікропроцесора, що призводить до непродуктивних втрат часу процесора. До недоліків програмного обміну за стробом готовності відноситься те, що даний спосіб інформацією не дозволяє зовнішнім пристроям ініціювати обмін. Перевагою програмного обміну є простота реалізації та відсутність необхідності застосування додаткових апаратних засобів.

Програмний обмін використовується для обміну з ПВВ, продуктивність яких менша від продуктивності мікропроцесора.

Обмін за перериванням за ініціюється ПВВ та здійснюється під керуванням мікропроцесора. У цьому випадку сигнал готовності ПВВ до обміну використовується як запит переривання і надходить до програмовного контролера переривань (ПКП) (рис. 3.14). Введення або виведення здійснюється за підпрограмою обробки запиту переривання.

ПВВ формує сигнал готовності IRQ, коли він готовий до обміну. ПКП здатний сприйняти 8 сигналів IRQ7-IRQ0. На рис. 3.14 сигнал готовності ПВВ надходить на вхід IRQ6. Сигнал готовності ПВВ являє собою вихідний сигнал тригера, що фіксує стан готовності READY. На виході ПКП асинхронно відносно дій мікропроцесора формується сигнал INT. Заздалегідь не відомо, у якій момент і які периферійні пристрої ініціюють переривання.

Реагуючи на сигнал INT, мікропроцесор перериває виконання програми, ідентифікує пристрій, переходить до підпрограми обслуговування переривань роботи цього пристрою, а після її завершення відновлює виконання перерваної програми. За командою INT вміст програмного лічильника та прапорців автоматично запам’ятовується у стеку. Вміст акумулятора та РЗП необхідно занести у стек за допомогою команд **PUSH** в підпрограмі обробки переривання.

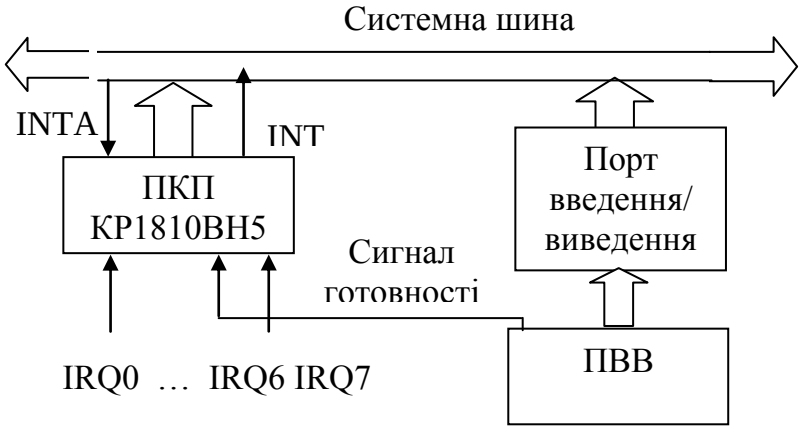


Рис. 3.14. Схема обміну за перериванням

Практично в кожному мікропроцесорі реалізована особлива структура системи переривань. Однак загальна послідовність обміну за перериванням містить наступні дії:

- ПВВ генерує сигнал готовності, який викликає появу сигналу переривання, що подається на вхід INT мікропроцесора;
- МП завершує виконання поточної команди і, якщо переривання дозволені (не замасковані), формує сигнал INTA підтвердження переривання;
- МП здійснює запам'ятовування вмісту акумулятора, програмного лічильника, РЗП у стеку;
- МП ідентифікує пристрій, що викликав переривання, і виконує відповідну підпрограму обслуговування переривання;

- За допомогою команд **POP** відновлюються значення вмісту акумулятора та РЗП зі стека;
- За командою повернення з переривання **RET**, що є останньою командою підпрограми обслуговування переривання, відновлюються значення програмного лічильника та прапорців, і продовжується виконання перерваної програми.

Обмін за перериванням є більш продуктивним, ніж програмний обмін, оскільки не потребує часу для опитування стану готовності ПВВ до обміну.

Обмін у режимі ПДП ініціюється ПВВ та здійснюється під керуванням контролера прямого доступу до пам’яті (КПДП) без участі МП. При необхідності обміну між ПВВ і пам'яттю немає потреби пересилати дані через мікропроцесор. Дані за допомогою КПДП пересилаються безпосередньо з ПВВ у пам’ять або навпаки. Прямий доступ до пам'яті при виконанні операцій введення/виведення дозволяє значно збільшити швидкість передачі даних і підвищити ефективність використання засобів мікропроцесора. Схема обміну в режимі ПДП наведена на рис. 3.15.

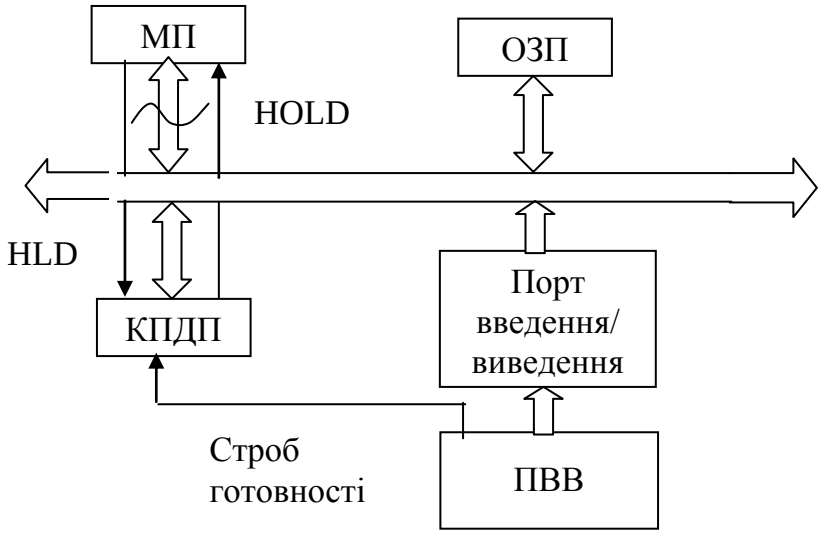


Рис. 3.15. Схема обміну у режимі ПДП

КПДП приймає запит від ПВВ, формує сигнал запиту захоплення шин МП HOLD і, отримавши від МП дозвіл HLDA, формує адреси пам’ятті та

керуючі сигнали $\overline{MEMR}, \overline{IOW}$ - в разі читання пам'яті, або $\overline{MEMW}, \overline{IOR}$ - в разі запису у пам'ять. Інформацію про область пам'яті, що використовується при обміні у вигляді початкової адреси і довжини масиву, завантажують у КПДП при програмуванні. Продуктивність обміну у режимі ПДП є найбільш високою.

Перетворення форматів даних. Якщо розрядність даних, з якими оперує МП, менше розрядності даних, з якими оперує ПВВ, то для узгодження розрядності збільшується кількість портів введення/виведення. Якщо розрядність даних, з якими оперує МП, більш за розрядність даних, з якими оперує ПВВ, то для узгодження розрядності виконується пакування даних, отриманих з декількох джерел, у одне слово необхідної розрядності або доповнення нулями. Для перетворення послідовного коду на паралельний і навпаки використовують контролер послідовного обміну.

ЛЕКЦІЯ 12

Програмовний паралельний інтерфейс

Принцип роботи програмовного паралельного інтерфейсу (ППІ) розглянемо на прикладі ВІС КР580ВВ55, яка призначена для організації введення/виведення паралельної інформації у 8-байтовому форматі і дозволяє реалізувати більшість відомих протоколів обміну по паралельних каналах. ВІС програмовного паралельного інтерфейсу може використовуватися для з'єднання мікропроцесора зі стандартним периферійним устаткуванням (дисплеєм, телетайпом, накопичувачем тощо). Структурна схема ВІС наведена на рис. 3.16. До складу ВІС входять: двонапрямлений 8-розрядний буфер даних Bufer of Data (BD), що пов'язує лінії даних ВІС із системною шиною даних; блок керування читанням/записом Read/Write Control Unit (RWCU), що забезпечує керування зовнішнім і внутрішнім передаванням даних та керуючих слів; три 8-розрядних порти введення/виведення (Port A, Port B, Port C) для обміну

інформацією, причому порт С поділений на два 4-розрядних: С′ (PC7-PC4) і С′′(PC3-PC0). Порти А і С′ об’єднані у групу В, порти В і С′′ - у групу В. Схема ВІС ППІ містить також блоки керування групою А Control Unit A (CUA) та групою В (CUB), що виробляють сигнали керування для відповіднх групами. Блок RWCU (Register of Control Word Unit) містить регістр керуючого слова, який зберігає керуючі слова, що надходять від МП.

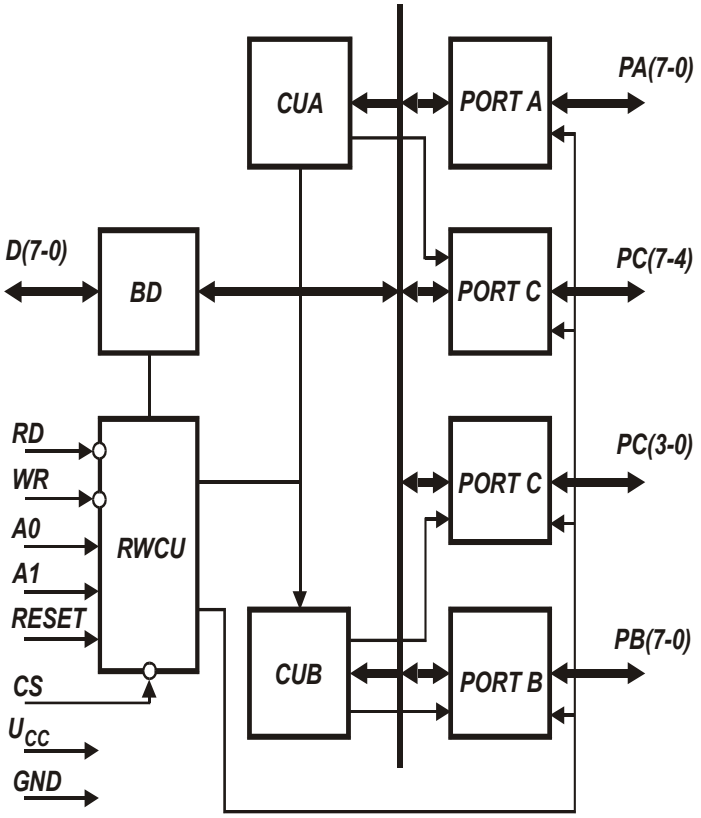


Рис. 3.16. Структурна схема ВІС КР580ВВ55

Адресні розряди А1, А0 дозволяють обирати один з портів або регістр керуючого слова RCW:

A1	A0	Порт
0	0	A
0	1	B
1	0	C
1	1	RCW

Сигнал керування третім станом шини даних $\overline{CS}$, сигнал читання $\overline{RD}$, сигнал запису $\overline{WR}$ та сигнал скидання RESET подаються на блок RWCU і разом із сигналами на адресних лініях A0, A1 задають вид операції, що виконується (табл.3.1).

Таблиця 3.1. Визначення виду операцій залежно від сигналів керування та адресних розрядів A1, A0

Операція	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A1	A0
Запис керуючого слова із МП	0	1	0	1	1
Запис даних у порт A	0	1	0	0	0
Запис даних у порт B	0	1	0	0	1
Запис даних у порт C	0	1	0	1	0
Читання даних з порту A	0	0	1	0	0
Читання даних з порту B	0	0	1	0	1
Читання даних з порту C	0	0	1	1	0
Вимикання ППІ від D7-D0	1	X	X	X	X

Примітка. X – будь-яке значення (0 або 1)

Функціональна схеми під'єднання ВІС до системної шини МП показана на рис. 3.17.

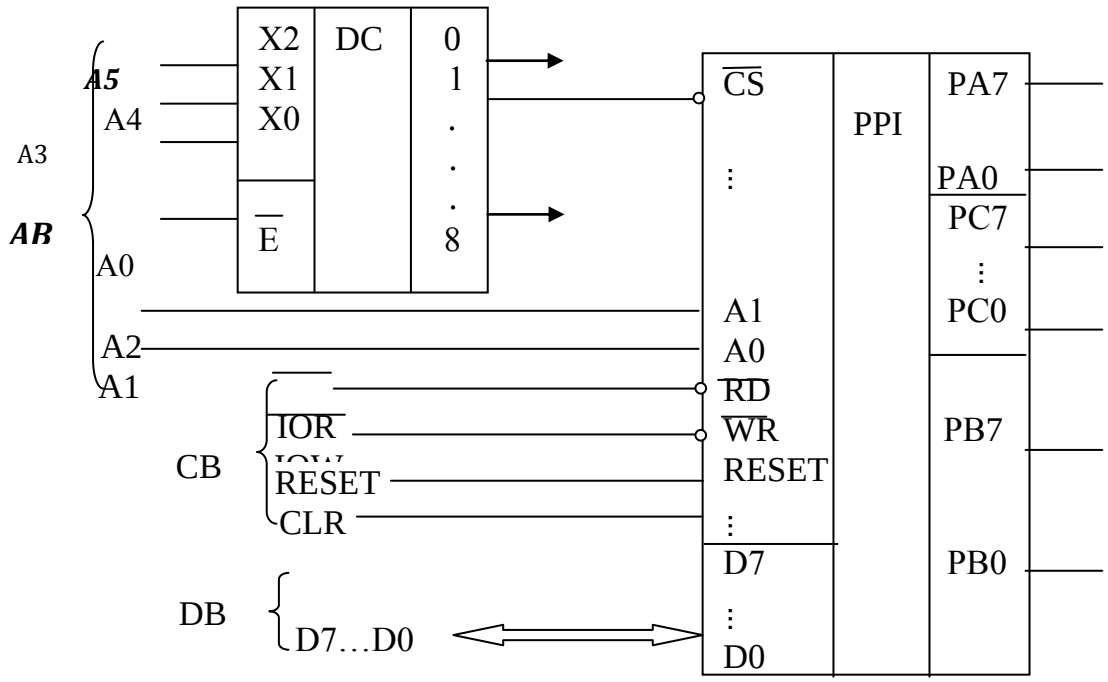


Рис. 3.17. Функціональна схема під'єднання ВІС КР580ВВ55 до системної шини

Відповідно до схеми на рис. 3.17 і табл. 3.1 визначаються адреси портів та регістра керуючого слова RCW (табл.3.2).

Таблиця 3.2. Адреси портів та регістра RCW

	A7	A6	A5	A4	A3	A2	A1	A0	Адреса
Порт А	0	0	0	0	1	0	0	0	08H
Порт В	0	0	0	0	1	0	1	0	0AH
Порт С	0	0	0	0	1	1	0	0	0CH
Регістр RCW	0	0	0	0	1	1	1	0	0EH

Керуюче слово визначає один з трьох режимів портів ППІ: режим 0 - основний режим введення/виведення; режим 1 – режим введення/виведення за стробом готовності; режим 2 - режим двонапрямленої передачі інформації. Керуюче слово може встановлювати різні режими роботи для кожного з портів. Порт А може працювати в будь-якому з трьох режимів, порт В - в режимах 0 та 1. Порт С може бути використаний для передачі даних тільки в режимі 0, в інших режимах він застосовується для передачі керуючих сигналів, що супроводжують процес обміну по портах А і В.

Програмування ВІС полягає у завантаженні керуючого слова режиму при A1=1, A0=1. Формат керуючого слова режиму наведено на рис. 3.18.

A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	M1	M0	IOA	IOC`	M	IOB	IOC``
			Режим групи А 00 – реж. 0 01 – реж. 1 1x – реж. 2		введення/ виведення порту А: 1 – введення 0–виведення	введення/ виведення порту С[1] 1 – введення 0–виведення	Режим групи В 0- реж. 0 1- реж. 1	введення/ виведення порту В 1 – введення 0–виведення	введення/ виведення порту С[11] 1 – введення 0–виведення

Рис. 3.18. Формат керуючого слова режиму

Окремі розряди порту С можна встановлювати або скидати програмно за допомогою керуючого слова встановлення /скидання, формат якого наведено на рис. 3.19.

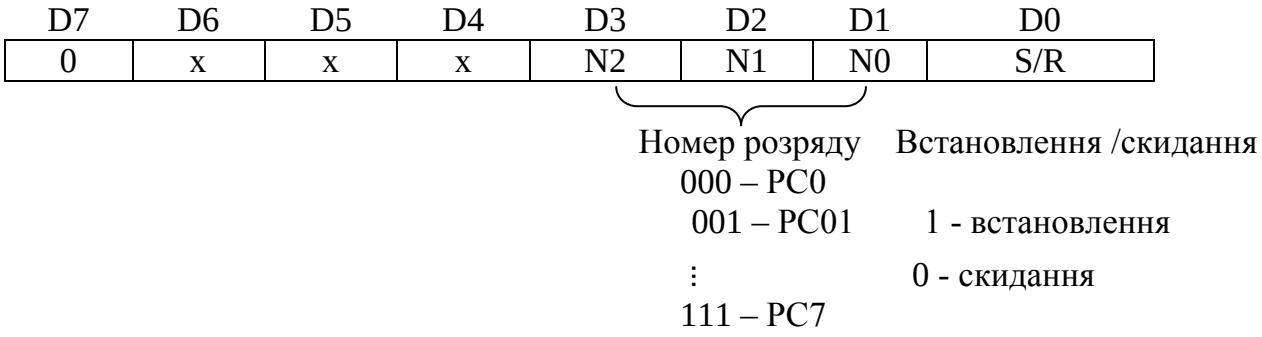


Рис. 3.19. Формат керуючого слова встановлення/скидання порту С

Для кожної групи А або В у ППІ є тригер дозволу переривання INTE, встановлення/скидання якого здійснюється керуючим словом встановлення/скидання визначеного розряду порту С. Якщо тригер дозволу переривання відповідної групи встановлений (INTE=1), то ППІ може сформувати запит переривання при готовності ПВВ до обміну.

Розглянемо режими роботи портів ППІ.

Режим 0 застосовується при синхронному обміні або при програмній організації асинхронного обміну. У цьому режимі ВІС являє собою пристрій, що складається з чотирьох портів (два 8-розрядних і два 4-розрядних), які можуть незалежно налагоджуватись на введення або виведення інформації. Виведення інформації здійснюється за командою OUT з фіксацією виведеної інформації у регістрах портів, а введення - за командою IN без запам'ятовування інформації.

Приклад 3.3. *Встановити порт А у режим введення 0, порт В - у режим виведення 0, порт С` - режим введення 0, порт С`` - режим виведення 0, а потім здійснити введення або виведення інформації через порти А та В відповідно.*

Керуюче слово режиму у цьому випадку має вигляд:

D7	D6	D5	D4	D3	D2	D1	D0
1	M1	M0	IOA	IOC`	M	IOB	IOC``
1	0	0	1	1	0	0	0

= 98H

Програма має вигляд:

```
MOV  AL, 98H      ; формування керуючого слова режиму 98H у AL
OUT  OEH,AL       ; запис до регістра RWC
 ⋮
IN  AL,08H        ; введення з порту A
 ⋮
OUT  0AH,  AL     ; виведення на порт B
```

Зазначимо, що в цьому прикладі адреси портів прийняті згідно зі схемою рис. 3.17.

Режим 1 забезпечує однонапрямлений обмін інформацією МП з ПВВ за стробом готовності. Передача інформації проводиться по портах A і B, а лінії порту C керують передачею. Роботу порту в режимі 1 супроводжують три керуючі сигнали. Якщо один із портів запрограмований у режим 1, то інші 13 інтерфейсних ліній можна використовувати у режимі 0. Якщо обидва порти запрограмовані на режим 1, то 2 інтерфейсні лінії порту C, що залишилися, можуть бути налагоджені на введення або виведення. Призначення розрядів порту C при введенні даних з портів A і B у режимі 1 наведено на рис. 3. 20.

	Група A					Група B		
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
IO	IO	IBF A	INTE A	INTR A	INTE B	IBF B	INTR B

Рис. 3.20. Призначення розрядів порту C при введенні даних з портів A і B у режимі 1

На рис. 3.20 позначено:

IBF (Input Buffer Full) - вихідний сигнал ППІ, що повідомляє про заповненість вхідного буферу порту даними;

INTR (INTeRrupt) - вихідний сигнал ППІ, що повідомляє про завершення приймання інформації;

INTE (INTerrupt Enable) – сигнал дозволу переривання (вхід стробу прийому).

На рис.3.21 наведено приклад схеми під'єднання пристрою введення до порту А, пристрою виведення – до порту В у режимі 1.

Введення даних у режимі 1 на рис. 3.21 здійснюється по каналу А, а керуючі сигнали передаються по лініях РС4 і РС5. Пристрій введення видає строб прийому $\overline{\text{STB}}$, який вказує на готовність до введення інформації. Цей строб надходить на вхід дозволу переривання від каналу А – РС4. Вихідний сигнал ППІ IBF з вивода РС5 використовується для підтвердження прийому. Він формується за спадом STB та повідомляє пристрій введення про закінчення приймання даних

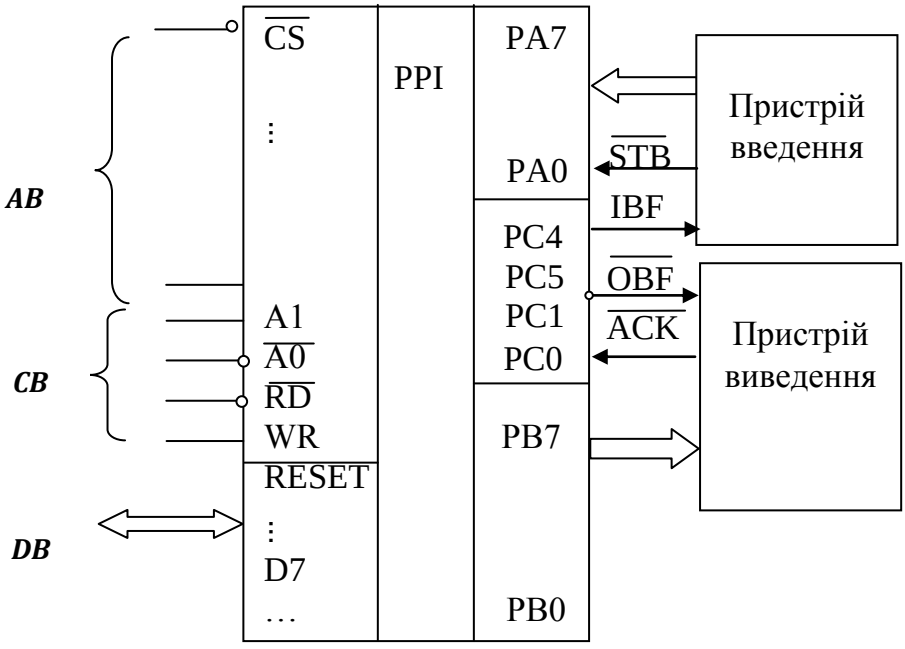


Рис. 3.21. Схема під'єднання пристрою введення до порту А, пристрою виведення – до порту В у режимі 1

Крім показаних на рис.3.21 сигналів, ППІ формує також сигнал запиту переривання INTR, який інформує мікропроцесор про завершення приймання інформації. При обміні за перериванням цей сигнал використовується як запит переривання, а при програмному обміні може ігноруватися. Високий рівень цього сигналу встановлюється при $\overline{STB}$=1, IBF=1. Нульовий рівень сигналу INTR встановлюється при надходженні сигналу $\overline{IOR}$.

Призначення розрядів порту С у режимі виведення 1 наведено на рис. 3.22.

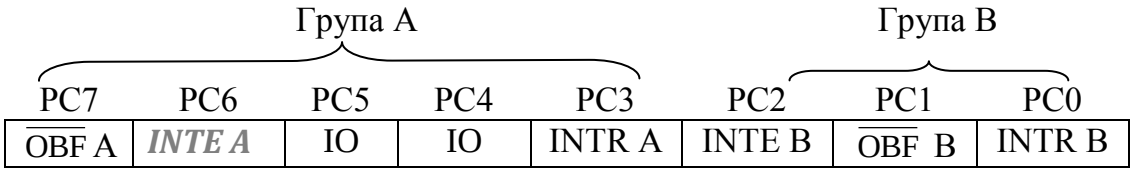


Рис. 3.22. Призначення розрядів порту С при виведенні даних на порти А і В у режимі 1

На рис. 3.22 позначено:

$\overline{OBF}$ (Output Buffer Full) - вихідний сигнал ППІ, що повідомляє про заповненість вихідного буферу порту даними. Інші сигнали мають такий самий сенс, що і на рис. 3.20.

Приклад виведення даних на порт В у режимі 1 показано на рис. 6.13. Для виведення даних в цьому режимі використовуються наступні керуючі сигнали: $\overline{OBF}$ - вихідний сигнал, що формується за фронтом $\overline{WR}$ та повідомляє ПВВ про готовність до виведення; $\overline{ACK}$ - вхідний сигнал, що підтверджує прийом інформації з ППІ; INTR - вихідний сигнал ППІ, що повідомляє мікропроцесор про завершення виведення. Сигнал INTR встановлюється у 1 при $\overline{OBF}$=1 і $\overline{ACK}$=1, і скидається у 0 сигналом $\overline{IOW}$ при запису у ППІ.

Розряди PC6, PC7 при введенні (рис. 3.20) та PC5, PC4 при виведенні (рис.3.22) не беруть участі у керуванні обміном і можуть бути запрограмовані на просте введення або виведення (I/O). Введення

здійснюється читанням порту С, а виведення – записом керуючих слів встановлення/скидання окремих розрядів.

Обмін за стробом готовності може здійснюватися за перериванням або за програмою. При обміні за перериванням сигнал INTR надходить до системи переривання та ініціює обмін. При обміні за програмою готовність портів А або В визначається шляхом опитування INTR А або В.

Приклад 3.4. *Написати програму встановлення порту А у режим введення 1 та ввести дані за стробом готовності. Адреси порту А визначаються рис.3.21.*

Керуюче слово режиму у цьому випадку має вигляд:

D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0	
1	M 1	M 0	IO A	IOC`	M	IOB	IOC``	
1	0	1		1	0	0	0	0 = 0B0H

Програма має вигляд:

```
        MOV  AL, 0B0H     ;  формування керуючого слова режиму  в AL
        OUT  OEH,AL       ;  запис до регістра RWC ВІС КР580ВВ55
        MOV  AL,09        ; формування керуючого слова встановлення розряду
                          ; РС4 – INTE A – дозвіл переривань
        OUT  OEH,AL       ; запис  вмісту AL до регістра керуючого слова
         ⋮
M1:     IN AL,0CH          ; AL ← вміст порту С
        AND AL, 00001000B  ; маскування усіх розрядів, крім РС3 (INTR A)
        JZ M1              ; Якщо дані  не готові, то на М1
         IN AL, 08H        ; інакше - введення інформації з порту А
         ⋮
```

Режим 2. забезпечує двонапрямлену передачу інформації з порту А до зовнішнього пристрою та навпаки. Процес обміну супроводжують 5 керуючих сигналів, що подаються по лініях РС7-РС3 . 11 інтерфейсних ліній, що залишились, можуть налагоджуватися на режим 0 або режим 1. Призначення розрядів порту С у режимі 2 наведено на рис. 3.23, а схема під’єднання ПВВ – на рис. 3.24.

Призначення керуючих сигналів у режимі 2 аналогічне сигналам у режимі 1. Керування формуванням внутрішнього сигналу INTE для операції введення здійснюється по лінії PC4, для операції виведення - по лінії PC6.

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
OBF_A	$INTE_1$	IBF_A	$INTE_2$	$INTR_A$			

залежать від режиму порту B

Рис. 3.23. Призначення розрядів порту C у режимі 2

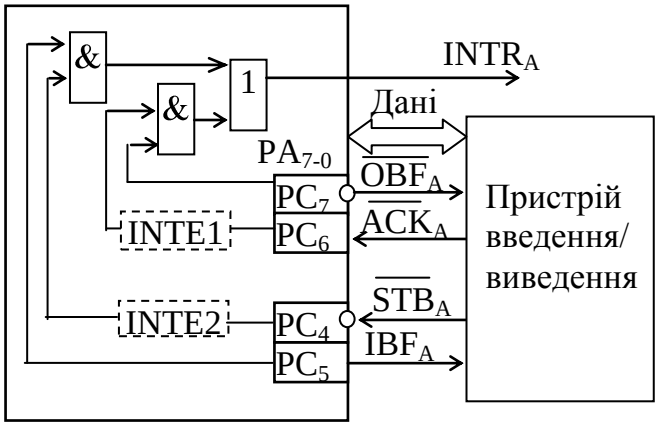


Рис. 3.24. Схема під'єднання ПВВ до ВІС КР580ВВ55 у режимі 2

Вивід ВІС INTR A використовується як запит переривань як при введенні, так і при виведенні інформації. Розподіл сигналів по інтерфейсних лініях, керуюче слово режиму 2 і часові діаграми роботи подані на рис 3.25.

Приклад 3.5. *Написати програму встановлення порту А у режим введення 2, а потім здійснити введення (виведення) інформації через порт А в цьому режимі.*

Керуюче слово режиму у цьому випадку буде дорівнювати 0C0H:

D7	D6	D5	D4	D3	D2	D1	D0
1	M1	M0	IOA	IOC`	M	IOB	IOC``
1	1	0	0	0	0	0	0

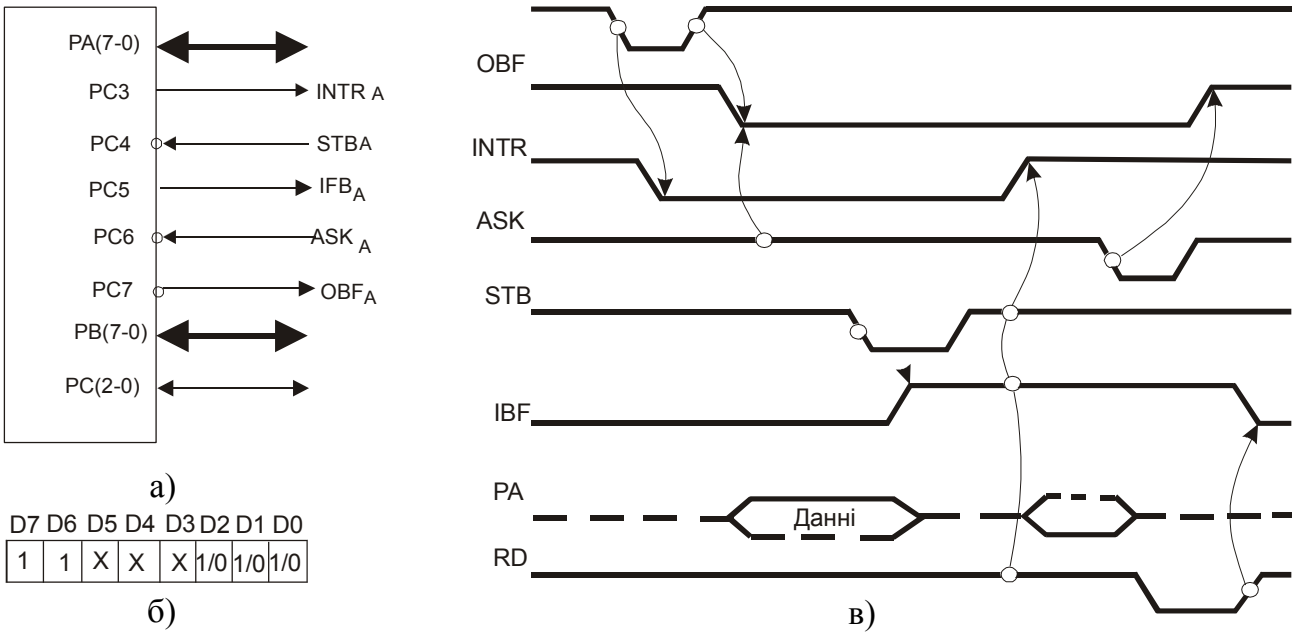


Рис. 3.25. Робота ППІ у режимі 2: а - розподіл сигналів по інтерфейсних лініях; б - керуюче слово; в - часові діаграми роботи

Програма двонапрямленного введення/виведення за стробом готовності має спочатку виявити готовність порту до введення або виведення за одиничним станом сигналу INTR A (лінія PC3), а потім встановити, які саме дані готові – для введення (одиничний стан лінії PC4) чи виведення (одиничний стан лінії PC6). Після цього можна здійснювати обмін даними. Програма має вигляд:

```
      MOV  AL, 0C0H       ;  формування керуючого слова режиму  в AL
      OUT  OEH,AL         ;  запис до регістра RWC ВІС КР580ВВ55
       ⋮
M1:  IN AL,0CH            ; AL ←вміст порту С
      MOV BL,AL           ; зберігання вмісту AL у регістрі BL.
      AND AL, 00001000B   ; маскування усіх розрядів, крім PC3 (INTR A)
      JZ M1               ; Якщо дані  не готові, то на M1
      MOV AL,BL           ; AL ←вміст порту С
      AND AL, 00010000B   ; маскування усіх розрядів, крім PC4 (INTR A)
      JZ M2               ; Якщо дані готові не для введення, то на виведення
        IN AL, 08H        ; інакше - введення інформації з порту А
        JMP M3
M2:  OUT  08H,AL          ; виведення інформації на порт А
M3:          …            ; продовження програми
```

Програмовний інтерфейс клавіатури та індикації

Програмовний інтерфейс клавіатури та індикації призначений для реалізації обміну інформацією між МП та матрицею клавіш/датчиків та індикації. На відміну від ВІС паралельного інтерфейсу, який може використовуватись для будь-якого пристрою, що здійснює введення/виведення даних у паралельному форматі, програмовний інтерфейс клавіатури та індикації є спеціалізованим і призначений для обміну інформацією лише з деякими типами клавіатури та індикаторів. Розглянемо принцип дії програмовного інтерфейсу клавіатури та індикації на прикладі ВІС КР580ВВ79, структурна схема якої наведена на рис. 3.26.

Схема містить: 1) двонапрямлений восьмирозрядний буфер даних BD, що пов'язує лінії даних ВІС із системною шиною даних; 2) блок RWCU, що забезпечує керування зовнішнім і внутрішнім передаванням даних та керуючих слів; 3) блок керування; 4) блок інтерфейсу індикації; 5) блок інтерфейсу клавіатури.

Блок керування містить схему керування та синхронізації і лічильник сканування СТ. Схема керування та сигналізації виробляє сигнали, які керують усіма блоками ВІС, сигнали внутрішньої синхронізації та сигнал $\overline{BD}$ для гасіння індикатора під час зміни символів. До складу схеми входить подільник частоти з програмовним коефіцієнтом ділення для генерації внутрішніх імпульсів синхронізації частотою до 100 кГц. Лічильник сканування формує коди на лініях S3-S0 для опитування матриці клавіатури та керування індикацією. При цьому залежно від керуючих слів можна налагоджувати схеми видачі стану лічильника сканування або на безпосереднє виведення вмісту чотирьох молодших розрядів лічильника, або на виведення вмісту двох молодших розрядів через дешифратор з чотирма виходами.

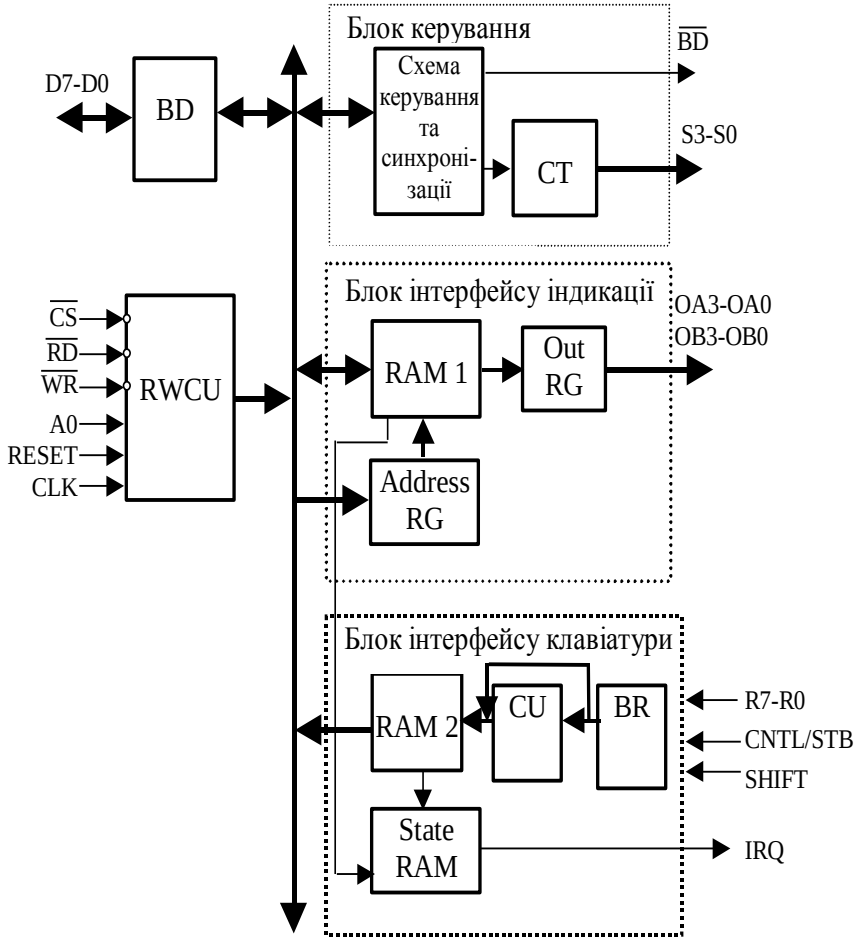


Рис. 3.26. Структурна схема програмовного інтерфейса клавіатури і ндикації

Блок інтерфейсу індикації містить ОЗП індикації RAM 1 інформаційною ємністю 16x8, адресний регістр Address RG та вихідний регістр Out RG. ОЗП складається з двох незалежних частин по 16 4-розрядних слів кожний та зберігає коди символів для 8- або 16-розрядної індикації. Тип індикації задається відповідним керуючим словом. Дані з ОЗП передаються через вихідні регістри на виходи OA3-OA0 (старша частина 8-розрядного слова) та OB3-OB0 (молодша частина).

До *блоку інтерфейсу клавіатури* входить ОЗП RAM 2 клавіатури/датчиків, буфер повернення BR, схема усунення брязкіту контактів CU, схема аналізу стану ОЗП State RAM. Блок забезпечує введення інформації через лінії повернення (R7-R0) з клавіатури. Збереження введеної інформації здійснюється у ОЗП RAM 2, який являє собою стек ємністю 8x8 біт. Вхідні лінії R7-R0 мають високий внутрішній опір, що дає можливість

безпосереднього під'єднання до них матриці клавіатури або датчиків. Для забезпечення режиму введення даних з датчиків за стробом готовності передбачена лінія CNTL/STB.

До виходів буфера BR під'єднані входи схеми усунення брязкіту контактів, яка виявляє заборонені ситуації при натисканні клавіш і не допускає повторного введення коду клавіш, що може статися внаслідок брязкіту контактів. Схема аналізу стану ОЗП формує статусну інформацію про роботу ОЗП та сигнал запиту переривання IRQ.

Програмування мікросхеми КР580ВВ79 здійснюється завантаженням ***керуючого слова ініціалізації клавіатури та дисплея*** у відповідний регістр керуючих слів, який розташований у блоку керування. При запису керуючих слів на вхід A0 необхідно подавати сигнал логічної одиниці. Формат керуючого слова ініціалізації клавіатури та дисплея наведено на рис.3.27.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	D	D	K	K	S

Рис. 3.27. Формат керуючого слова ініціалізації клавіатури та дисплею

Розряди D3 та D4 керуючого слова (див. рис.3.27) визначають режим роботи дисплея:

D4	**D3**	**Режим роботи дисплея**
0	0	дисплей на 8 символів із введенням з лівого боку
0	1	дисплей на 16 символів із введенням з лівого боку
1	0	дисплей на 8 символів із введенням з правого боку
1	1	дисплей на 16 символів із введенням з правого боку

Розряди D1 та D2 визначають режим роботи клавіатури:

D2	**D1**	**Режим роботи клавіатури**
0	0	клавіатура в режимі одиночного натискання клавіш
0	1	клавіатура в режимі N–клавішного натискання
1	0	сканування матриці датчиків
1	1	режим стробованого введення

Розряд S визначає режим сканування: при S=0 – сканування в режимі 4-розрядного двійкового лічильника; при S=1 – сканування в режимі інверсного дешифратора по лініях S3-S0. Якщо сканування здійснюється у режимі дешифратора, то дисплей містить не більше 4 символів, а клавіатура містить не більше 8x4 = 32 клавіш.

У керуючому слові ***ініціалізації опорної частоти*** (рис.3.28) розряди D4-D0 визначають коефіцієнт PPPPP ділення частоти зовнішнього синхросигналу CLK для одержання внутрішнього опорного сигналу з частотою не більше 100 кГц. Після скидання ВІС сигналом RESET встановлюється максимальний коефіцієнт PPPPP=11111.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	P	P	P	P	P

Рис. 3.28. Формат керуючого слова ініціалізації опорної частоти

Приклад 3.5. *Запрограмувати ВІС контролера клавіатури та індикації для роботи з клавіатурою 8x8 клавіш у режимі N-клавішного натискання та з дисплеєм з 8 символів в режимі введення з лівого боку. ВІС КР580ВВ79 має адреси 00Н для даних і 02Н для запису керуючих слів і читання статусної інформації. Частота імпульсів на вході CLK - 2 МГц.*

Визначимо керуючі слова.

Згідно умов прикладу та рис.3.27 керуюче слово ініціалізації клавіатури та дисплею має вигляд:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	0

і дорівнює 02Н. Керуюче слово ініціалізації опорної частоти (див. рис.3.28) має вигляд:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	1	0	0

і дорівнює 34Н. Значення PPPPP=10100_2 визначає двійковий код коефіцієнта ділення імпульсів з частотою 2 МГц для забезпечення внутрішньої частоти 100 кГц (2000/100=20= 10100_2).

Програма ініціалізації ВІС має вигляд:

```
MOV   AL, 02      ; формування першого керуючого слова режиму в AL
OUT   02,AL       ; виведення в інтерфейс
MOV   AL, 34H     ; формування другого керуючого слова режиму в AL
OUT   02,AL       ; виведення в інтерфейс.
```

Після такої послідовності команд інтерфейс клавіатури та індикації готовий до роботи у запрограмованому режимі.

Під'єднання ВІС інтерфейсу клавіатури та індикації наведено на рис. 3.29.

Пунктиром показано під'єднання зовнішнього дешифратора DC. У випадку, якщо клавіатура містить менше, ніж 4х8 клавіш, і кількість символів дисплея менше 4, то сигнали на виводах S3-S0 можуть бути безпосередньо використані для керування клавіатурою та індикацією, оскільки на виводах S3-S0 формуються сигнали дешифратора з 4 виходами. Під'єднання зовнішнього дешифратора (до 16 виходів) дозволяє керувати клавіатурою 16х8 клавіш та 16 символами дисплея.

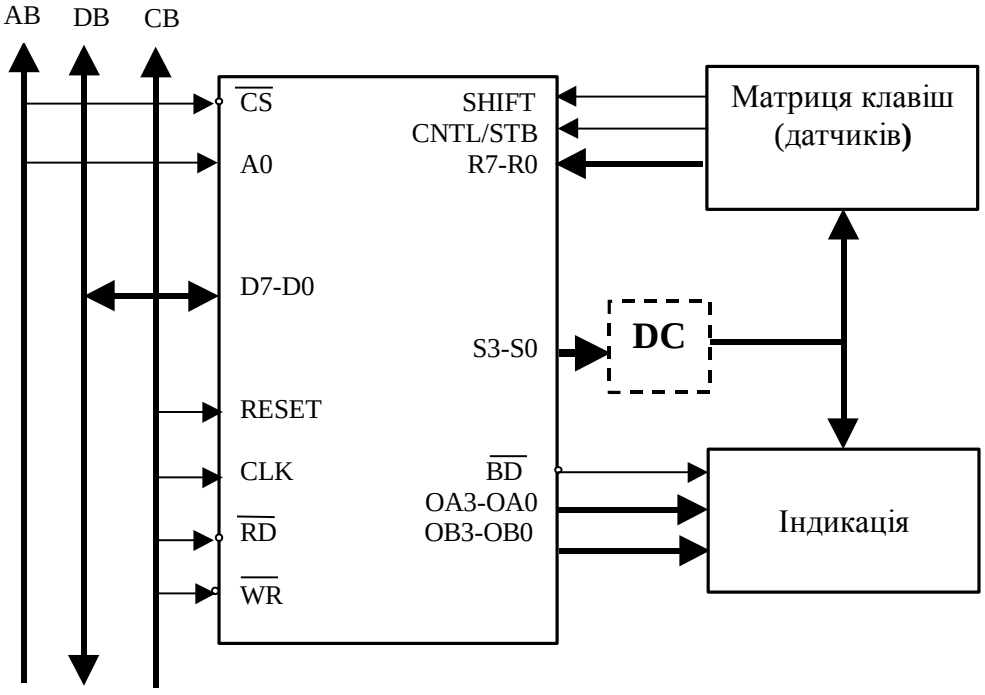


Рис. 3.29. Структурна схема підключення клавіатури і індикації

Функціонування блока інтерфейсу індикації. Після запису керуючого слова ініціалізації клавіатури та дисплею (див. рис. 3.27) блок інтерфейсу індикації встановлюється в один з чотирьох режимів, які визначаються розрядами D3 та D4. В усіх режимах для того, щоб висвітлити символ на індикації, треба завантажити керуюче слово ***запису в ОЗП індикації*** (рис. 3.30) при адресі з A0=1, а потім завантажити дані при A0=1. При індикації дані з ОЗП передаються на 8 ліній OA3-OA0, OB3-OB0. Дані можуть бути представлені семисегментним кодом при безпосередньому під'єднанні індикаторів до ліній OA3-OA0, OB3-OB0 або двома 4-розрядними кодами при підключенні зовнішніх шифраторів.

Розряди D3-D0 керуючого слова (див.рис.3.30) містять адресу AAAA позиції дисплея, яка має бути прочитана. Розряд D4 містить ознаку автоінкрементної адресації I. Якщо I=1, то адреса буде інкрементуватися після кожної операції читання.

D7	D6	D5	D4	D3	D2	D1	D0
0	1	1	I	A	A	A	A

Рис. 3.30. Формат керуючого слова запису в ОЗП індикації

Для читання даних з ОЗП індикації необхідно завантажити керуюче слово ***читання з ОЗП індикації*** (рис. 3.31) при A0=1, а потім прочитати інформацію з ОЗП при A0=0.

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	I	A	A	A	A

Рис. 3.31. Формат керуючого слова читання з ОЗП індикації

Призначення розрядів керуючого слова (див. рис. 3.31), аналогічно призначенню розрядів керуючого слова (див. рис. 3.39). Поле AAAA визначає адресу позиції в ОЗП індикації, яка має бути прочитана.

При використанні дисплея на 8 символів блок керування мікросхеми КР580ВВ79 сканує дисплей за 5,1 мс при внутрішній частоті 100 кГц, при використанні дисплея на 16 символів – за 10,23 мс. Процес сканування дисплея полягає у видачі у вихідний регістр індикації OUT RG (див. рис. 3.26) вмісту кожної комірки ОЗП індикації. Функціонування блоку інтерфейсу індикації залежить від способу видачі кодів сканування S3-S0 (див. рис. 3.26). При скануванні в ***режимі інверсного дешифратора*** інформація з'являється тільки в перших чотирьох розрядах дисплею, у режимі 4-розрядного двійкового лічильника при використанні зовнішнього дешифратора – на 16 розрядах. Одночасно зі зміною станів лічильника сканування і вмісту вихідного регістру індикації на виводі $\overline{BD}$ з'являється сигнал логічного нуля тривалістю 150 мкс, що використовується для гасіння індикації при зміні символів.

У режимах виведення інформації на індикацію із введенням нових символів *з лівого боку* кожному розряду дисплея відповідає один байт у ОЗП індикації. Комірці ОЗП з нульовою адресою відповідає крайній лівий розряд, а комірці з адресою 7 (або 15) - крайній правий розряд у 8-розрядній або 16-розрядній індикації відповідно. У режимах виводу інформаціїі з введенням нових символів *з правого боку* запис коду нового символу відбувається у комірку ОЗП з адресою 0, при цьому раніше записана інформація зсувається вліво. У цьому режимі немає прямої відповідності між адресою комірки ОЗП і розрядом індикації.

Програмним шляхом можна заборонити видачу однієї або обох тетрад вмісту вихідних регістрів. Керуюче слово ***заборони запису у ОЗП індикації-гасіння*** має вигляд (рис. 3.32):

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	X	IWA	IWB	BLA	BLB

Рис. 3.32.Формат керуючого слова заборони запису у ОЗП індикації-гасіння

Байти ОЗП індикації розділяються на тетради: А (старша), В (молодша). Розряди D3 (IWA) та D3 (IWB) забороняють запис інформації у тетради А та В ОЗП індикації відповідно; розряди D1 (BLA) та D3 (BLB) - біти гасіння або бланкування (встановлення спеціального коду, наприклад, коду пробілу). Керуюче слово (див. рис.3.32) дозволяє маскувати одну з тетрад у випадку подвійного 4-х позиційного дисплею. При забороні запису в одну з тетрад тривалість низького рівня сигналу гасіння залишається не менше 150 мкс, а у випадку заборони запису в обидві тетради сигнал низького рівня на виході залишається на час дії керуючого слова.

Для встановлення коду бланкування, а також сканування ОЗП індикації та скидання байту стану використовується керуюче слово, формат якого наведений на рис. 3.33.

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	CD	BC_1	BC_2	CF	CA

Рис. 3.33.Формат керуючого слова встановлення коду бланкування,сканування ОЗП індикації і скидання байту стану

Розряди D3 (BC_1) та D2 (BC_2) дозволяють обрати один з кодів бланкування:

BC_1	BC_2	Коди бланкування
0	X	00
1	0	20H (код пробілу)
1	1	0FFH

При встановленні розряду D4 (CD) здійснюється процедура скидання ОЗП індикації шляхом заповнення кодами бланкування. Встановлення розряду D1(CF) скидає байт стану, сигнал переривання і встановлює вказівник пам‘яті матриці клавіатури на рядок 0. Дія розряду D0(CA) аналогічна одночасній дії розрядів D4 та D1.

Приклад 3.6. *Інтерфейс клавіатури і індикації запрограмований на режим сканування 8-розрядного дисплея із введенням з лівого боку. До ВІС інтерфесу під'єднані три семисегментних індикатори в позиціях 0, 1, 2. Навести функціональну схему під'єднання дисплея та написати програму видачі на дисплей вмісту трьох 8-байтових комірок пам'яті з початковою адресою DS:SI, в яких записані семисегментні коди. Адреси ВІС контролера клавіатури і індикації такі, як в прикладі 3.5.*

Функціональна схема під'єднання дисплея наведена на рис 3.34. Дисплей складається з трьох семисегментних індикаторів типу АЛС321Б, схема кожного з яких містить собою вісім світлодіодів, з'єднаних по схемі зі спільним анодом (рис. 3.35,а). Світлодіод висвітлюється при надходженні на входи a-g сигналу низького рівня (рис.3.35,б), а на спільний анод – сигналу високого рівня. Використання семисегментних індикаторів цього типу вимагає подання інверсних семисегментних кодів, а гасіння індикаторів здійснюється при надходженні коду бланкування 0FFH

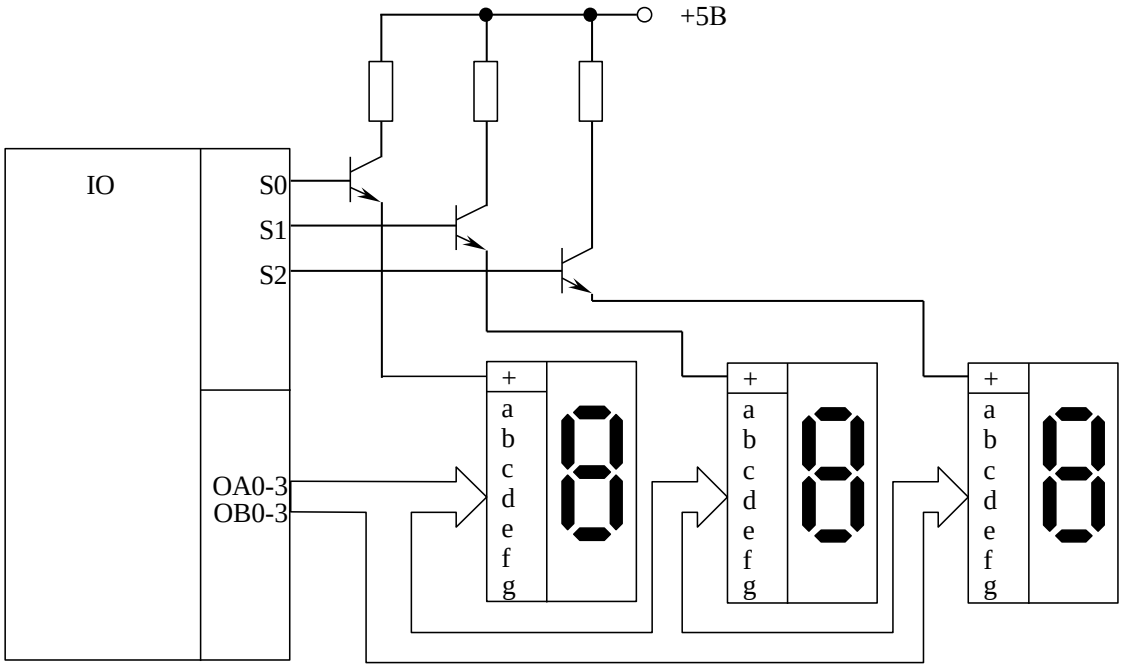


Рис 3.34. Функціональна схема під'єднання дисплея

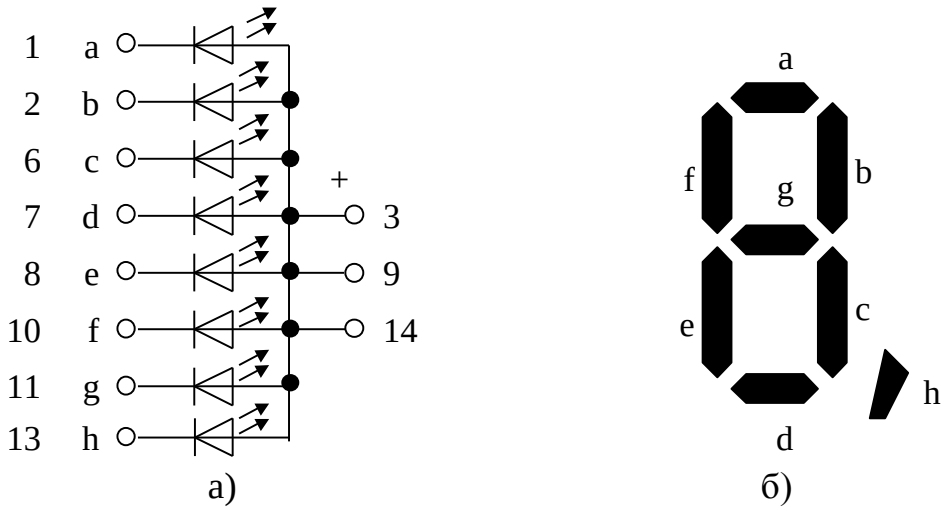


Рис.3.35. Семисегментний індикатор АЛС321Б: а – принципова схема, б – розташування світлодіодів

Програма виведення символів на індикацію починається з гасіння дисплею шляхом запису коду 0FFH у ОЗП індикації. Керуюче слово встановлення коду бланкування, сканування ОЗП індикації і скидання байту стану (див. рис. 3.33) повинно містити значення CD = 1, BC_1 = BC_2 = 1. Тоді значення керуючого слова дорівнює 0DCH.

Виведення інформації на дисплей здійснюється після завантаження керуючого слова запису в ОЗП індикації (див. рис.3.30), в якому AAAA=0000, I=1. Тоді значення керуючого слова дорівнює 70H.

Програма має вигляд

```
        MOV  AL, 0DCH       ;формування керуючого слова встановлення коду
                            ; бланкування, сканування ОЗП індикації і скидання
                            ; байту стану
        OUT     02, AL      ; виведення на інтерфейс
        CALL    DELAY1      ;затримка на час бланкування
        MOV     AL, 70H     ;формування керуючого слова запису в ОЗП індикації
        OUT     02, AL      ;виведення на інтерфейс
        MOV     CX, 3       ;завантаження лічильника байтів
M0:     MOV     AL, [SI]    ;зчитування з ОЗП індикації
        OUT     00, AL      ;виведення даних на інтерфейс
        INC     SI          ;збільшення адреси на 1
        LOOP    M0
        CALL    DELAY2      ;затримка на час сканування
```

Оскільки нові значення на індикацію можна передавати лише після затримки на час сканування даних в ОЗП дисплея (5,1 мс), то наведена програма містить підпрограму часової затримки DELAY2.

Функціонування блоку інтерфейсу клавіатури/датчиків. Всі режими роботи блоку інтерфейсу клавіатури/датчиків, що визначаються розрядами D1 та D2 керуючого слова ініціалізації клавіатури та дисплею (див. рис.3.27), можна розділити на три групи: 1) опитування матриці клавіатури; 2) опитування матриці датчиків; 3) введення даних за стробом.

У *режимі опитування матриці клавіатури* (при D1=D2=0 та при D1=1, D2=0) натискання будь-якої клавіші ініціює генерацію високого рівня сигналу переривання на виводі IRQ, а код натискнутої клавіші записується у ОЗП клавіатури/датчиків. При опитуванні матриці клавіатури функціонує схема усунення брязкоту контактів. Звернення до ОЗП відбувається за принципом черги: код, записаний в ОЗП першим, зчитується з нього першим. Щоб прочитати код клавіші з ОЗП, треба завантажити в інтерфейс клавіатури і індикації керуюче слово ***читання ОЗП клавіатури/датчиків*** (рис. 3.36) при A0=1, а потім прочитати дані з ОЗП при A0=0.

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	I	X	A	A	A

Рис. 3.36. Формат керуючого слова читання ОЗП клавіатури/датчиків

Розряд D4 (дис. рис.3.36) містить ознаку автоінкрементної адресації I, розряди D2-D0 – адресу AAA байта ОЗП клавіатури/датчиків, що має бути прочитаним. Якщо біт I встановлений, то наступні команди читання даних будуть викликати автоматичне інкрементування адреси. Для читання вмісту всього ОЗП необхідно завантажити керуюче слово читання ОЗП клавіатури/датчиків при I=1, а після цього 8 разів зчитати дані.

Формат даних при читанні ОЗП клавіатури/датчиків наведений на рис.3.37. У розрядах D5-D3 розміщується номер рядку матриці натиснутої клавіші - значення розрядів S2-S0 лічильника сканування; у розрядах D2-D0 розміщений номер колонки матриці натиснутої клавіші – значення розрядів

R2-R0. Розряди D7-D6 можуть використовуватися при введенні з розширеної клавіатури – в них записуються стан додаткових клавіш, під'єднаних до виводів SHIFT і CNTL.

D7	D6	D5	D4	D3	D2	D1	D0
CNTL	SHIFT	S2	S1	S0	R2	R1	R0

Рис. 3.37. Формат даних при читанні ОЗП клавіатури/датчиків

При читанні даних із ОЗП клавіатури/датчиків сигнал переривання IRQ скидається, але в тому випадку, якщо ОЗП клавіатури/датчиків містить ще не прочитані дані, на виводі IRQ знов генерується сигнал високого рівня.

Режим опитування матриці клавіатури містить: а) режим одиночного натискання клавіш із забороною введення кодів при натисканні двох або більше клавіш (D1=D2=0); б) N–клавішного натискання із дозволом введення кодів при натисканні N клавіш (D1=1, D2=0).

У *режимі одиночного натискання*, якщо натиснуто дві або більше клавіш, у ОЗП клавіатури/датчиків записується код лише однієї з клавіш – першої натиснутої або тієї, що опитується першою.

У *режимі N–клавішного натискання* в ОЗП заносяться коди усіх натиснутих клавіш у порядку їхнього опитування при скануванні матриці клавіатури. В цьому режимі можливо запрограмувати ВІС інтерфейса на спеціальний *режим помилки сканування клавіатури* шляхом встановлення одиничного значення розряду D4(E) у ***керуючому слові скидання переривання/встановлення режиму помилки сканування*** (рис. 3.38). Це керуюче слово також встановлює низький рівень сигнал переривання на лінії IRQ.

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	E	X	X	X	X

Рис. 3.38. Формат керуючого слова скидання переривання/встановлення режиму помилки сканування

При одночасному натисканні кількох клавіш буде встановлено прапорець помилки у байті стану і генерований високий рівень сигналу на виводі IRQ. ***Формат байта стану*** наведено на рис. 3.39. Одиничне значення розряду D7(DU) вказує на недоступний дисплей, тобто на те, що не закінчена операція очищення ОЗП індикації; одиничне значення розряду D6(S/E) – на те, що датчик замкнений (у режимі опитування матриці датчиків) або на помилку багатоклавішного натискання (у режимі опитування клавіатури); одиничне значення розряду D5(O) – на помилку переповнення. Цей розряд встановлюється тоді, коли відбувається спроба запису у заповнену пам'ять клавіатури. Одиничне значення розряду D4(U) вказує на помилку спустошення і встановлюється тоді, коли відбувається спроба читання даних з порожньої ОЗП клавіатури/датчиків. Одиничне значення розряду D3(F) вказує на заповненість ОЗП клавіатури/датчиків. Розряди D2-D0 (N2-N0) визначають кількість символів у ОЗП клавіатури/датчиків.

D7	D6	D5	D4	D3	D2	D1	D0
DU	S/E	O	U	F	N2	N1	N0

Рис. 3.39. Формат байта стану

У *режимі опитування матриці датчиків* зміна стану одного з датчиків ініціює генерацію високого рівня сигналу на виводі IRQ. При цьому значення розрядів R7-R0 безпосередньо записуються у ОЗП клавіатури/датчиків без передачі керування схемі усунення брязкоту контактів.

У *режимі введення за стробом* значення розрядів R7-R0 записуються у ОЗП клавіатури/датчиків, але введення стробується сигналом на виводі CNTL/STB.

Приклад 3.7. *Навести функціональну схему під'єднання до інтерфейсу клавіатури та індикації клавіатури, яка містить клавіші 10 цифр та клавішу ENTER. Визначити коди клавіш. Інтерфейс клавіатури і індикації*

запрограмовано на режим опитування матриці клавіатури із забороною введення кодів при натисканні N клавіш.

Функціональна схема під'єднання клавіатури наведена на рис. 3.40.

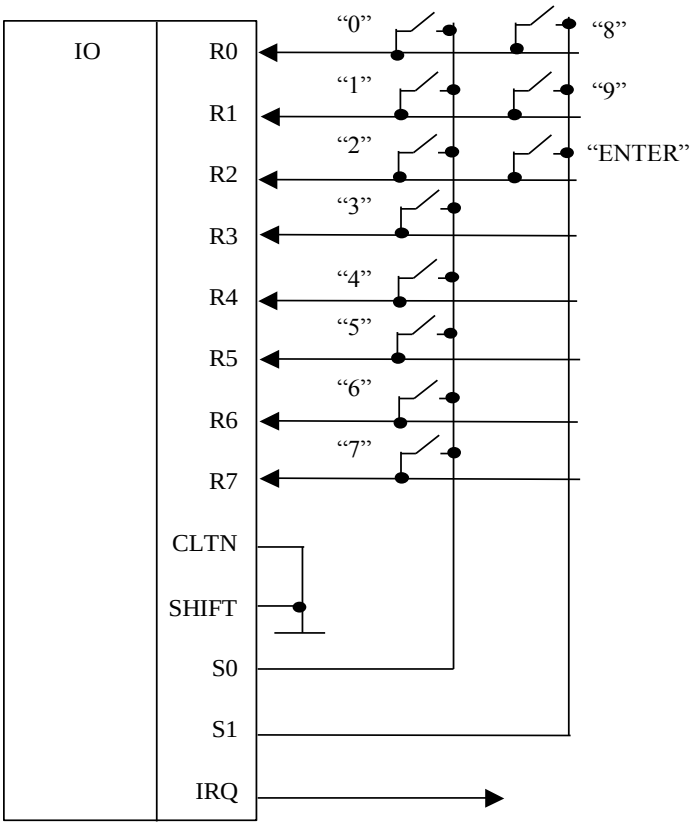


Рис.3.40. Функціональна схема під'єднання клавіатури

При такому під'єднанні коди клавіш, які записуються в ОЗП клавіатури, визначаються згідно рис. 3.37. Для цифр від 0 до 9 коди збігаються з цифрами, позначеними в лапках (див. рис. 3.40). Код клавіші *ENTER* визначається як 00 001 010 і дорівнює 0АН.

ЛЕКЦІЯ 13

Програмовний таймер

Принцип роботи програмовного таймера (ПТ) розглянемо на прикладі ВІС КР1810ВИ54, яка призначена для організації роботи мікропроцесорних систем і дозволяє формувати сигнали з різноманітними часовими і частотними характеристиками. Структурна схема ВІС наведена на рис. 3.41.

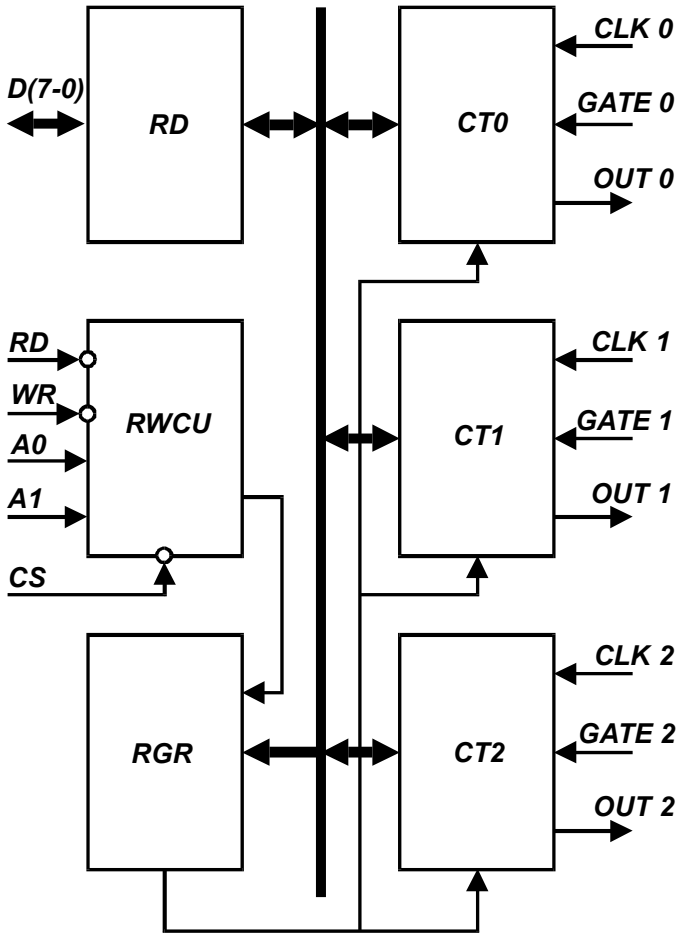


Рис. 3.41. Структурна схема ВІС КР580ВИ54

Схема таймера містить: блок керування читанням/записом RWCU з регістром керуючого слова RCW; тристабільний буфер даних BD; три канали на базі 16-розрядних від'ємних лічильників CT0-CT2. Кожний канал містить лічильник, вхідні та вихідні буферні регістри. Лічильники можуть працювати у двійковому або двійково-десятковому коді. Максимальна частота лічильника складає 2 МГц для КР580ВИ53 та 5МГц для КР 1810ВИ54.

На рис 3.41 позначено: CLK – входи тактових (лічильних) імпульсів; GATE – входи дозволу рахування, дія яких залежить від режиму роботи каналу; OUT – виходи лічильника. Розряди A1, A0 обирають звернення до лічильників або до регістра керуючого слова RCW:

A1	A0	Звернення
0	0	CT0
0	1	CT1
1	0	CT2
1	1	RCW

Сигнали керування роботою ВІС WR, RD, CS подаються на блок RWCU і разом із адресними розрядами A0, A1 задають вид операції, що виконується, згідно з табл.3.3.

Таблиця 3.3. Вид операції ПТ залежно від сигналів керування та адресних розрядів

Операція	Сигнали керування				
	$\overline{WR}$	$\overline{RD}$	$\overline{CS}$	A0	A1
Запис керуючого слова в RCW	0	1	0	1	1
Завантаження CT0 з D7-D0	0	1	0	0	0
Завантаження CT1 з D7-D0	0	1	0	0	1
Завантаження CT2 з D7-D0	0	1	0	1	0
Читання CT0 по D7-D0	1	0	0	0	0
Читання CT1 по D7-D0	1	0	0	0	1
Читання CT2 по D7-D0	1	0	0	1	0
Вимкнення ПТ від D7-D0	1	1	0	X	X

Примітка. X – будь-яке значення (0 або 1)

Встановлення режиму роботи кожного каналу програмовного таймера здійснюється програмно шляхом запису керуючого слова (рис. 3.42) і початкового вмісту лічильника.

D7	D6	D5	D4	D3	D2	D1	D0
CNT1	CNT0	RW1	RW0	M2	M1	M0	K

Рис.3.42. Формат керуючого слова програмовного таймера

Розряди D7 (CNT1) та D6 (CNT0) обирають лічильник:

D7	**D6**	**Лічильник**
0	0	Лічильник (CT0)
0	1	Лічильник (CT1)
1	0	Лічильник (CT2)
1	1	Заборонена комбінація для КР580ВИ53 та команда

		читання слову стану для КР1810ВИ54

Розряди D5 (RW1) та D4 (RW2) обирають спосіб читання/запису:

D5	**D4**	**Спосіб читання/запису**
0	0	читання вмісту лічильника
0	1	запис тільки молодшого байта
1	0	запис тільки старшого байта
1	1	запис молодшого, а потім старшого байтів

Розряди D3-D1 (M2-M0) обирають один з 6 режимів роботи лічильника:

M2	**M1**	**M0**	**Режим**
0	0	0	режим 0
0	0	1	режим 1
x	1	0	режим 2
x	1	1	режим 3
1	0	0	режим 4
1	0	1	режим 5

Розряд D0 (K) визначає спосіб кодування:

D0=0 – двійковий лічильник;

D0=1 – двійково-десятковий лічильник.

Приклад 3.8. *Запрограмувати лічильник 0 в режим 1. Адреса лічильника 0 - 10H, регістра керуючого слова - 16H.*

Визначимо керуюче слово:

0 0 1 1	0 0 1 0	=32H

Програма буде мати вигляд:

```
MOV AL,32H              ;формування керуючого слова
OUT 16H,AL              ;виведення в RCW
MOV AL, "молодший байт" ;завантаження молодшого байта коду
OUT 10H,AL              ;передвстановлення
MOV AL, "старший байт"  ;завантаження старшого байта коду
OUT 10H,AL              ;передвстановлення
```

Порядок програмування каналів таймера вельми гнучкий. Можна записати керуючі слова режимів в усі канали, а потім у довільному порядку

завантажувати коди передвстановлення, а можна запрограмувати окремо кожний канал (як у прикладі).

У процесі роботи програмовного таймера вміст будь-якого з лічильників можна прочитати двома способами:

-призупинити роботу лічильника надходженням сигналу GATE=0, або блокуванням тактових імпульсів, а потім прочитати вміст лічильника, починаючи з молодшого байта за допомогою двох команд введення. Перша команда введення прочитає молодший байт, друга – старший;

-записати у програмовний таймер керуюче слово, що містить нулі в розрядах D4, D5 (нулі в цих розрядах вказують на виконання операції "замкнення" вмісту лічильника у вихідному регістрі каналу в момент запису керуючого слова). Потім прочитати вміст лічильника за допомогою команд введення.

Приклад 3.9. *Прочитати вміст лічильника CT0 і записати його у регістр BX.*

Візьмемо адреси таймера такі, як у прикл. 6.12. Запрограмувати лічильник 0 в режимі 1.

Визначимо керуюче слово для фіксації вмісту лічильника:

0 0 0 0	0 0 1 0	=02H

Програма буде мати вигляд:

```
MOV AL,02H      ;формування керуючого слова
OUT 16H,AL      ;виведення в RCW
IN  AL,10H      ;читання молодшого байта
MOV BL,AL       ; переслати молодший байт у BL
IN  AL,10H      ;читання старшого байта
MOV BH,AL       ; переслати старший байт у BH
```

Таким чином, після виконання програми у BX буде вміст лічильника на момент його читання, а лічильник буде продовжувати рахунок.

У ВІС К1810ВИ54 крім того можна прочитати слова стану лічильника. Для цього необхідно записати керуюче слово наступного типу (рис.3.43):

D7	D6	D5	D4	D3	D2	D1	D0
1	1	COUNT	STAT	CT2	CT1	CT0	0

Рис. 3.43. Вигляд керуючого слова ВІС К1810ВИ54

На рис. 3.43 позначено: CT0, CT1, CT2 - вибір лічильника (лічильник вибирається при запису 1 у відповідний двійковий розряд). Значення STAT=0 вказує на те, що буде прочитано слово стану каналу, зазначеного в розряді D3-D1. Значення COUNT=0 вказує на те, що буде запам'ятовано вміст лічильників, що зазначені в D3-D1, у вихідних регістрах каналів.

Слово стану каналу має вигляд, показаний на рис.3.44.

D7	D6	D5	D4	D3	D2	D1	D0
OUT	FN	RW1	RW0	M2	M1	M0	K

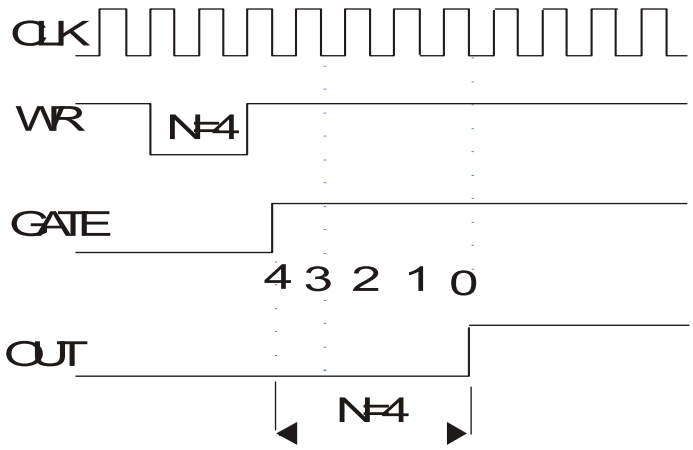
Рис. 3.44. Вигляд слова стану каналу

На рис.3.44 прийнято наступні позначення: OUT - стан виходу OUT (0,1); FN - прапорець перевантаження (FN=1, якщо було перезавантаження коду передвстановлення); розряди RW1,RW0,M2,M1,M0 та К дублюють розряди керуючого слова (див. рис. 3.42).

При записі керуючого слова в лічильник завантажується спочатку молодший, а потім старший байт коду передвстановлення. Надалі робота таймера залежить від обраного режиму роботи.

Режими роботи таймера. Лічильники таймера можуть працювати в шести режимах: режимі 0 - програмовна затримка; 1 – програмовний мультивібратор; 2 - програмовний генератор тактових імпульсів; 3 - генератор прямокутних сигналів; 4 - програмно-керований строб; 5 - апаратно-керований строб. Вплив сигналу GATE на відповідний лічильник залежить від режиму роботи.

Режим 0. Програмова затримка. У цьому режимі (рис. 3.45) на виході вибраного каналу таймера формується сигнал Н-рівня із програмно-



керованою затримкою. Затримка відраховується від заднього фронту першого імпульсу CLK після запису молодшого байта коду перестановки константи.

Рис. 3.45 Діаграма роботи ПТ в режимі 0

Після запису керуючого слова на виході OUT вибраного каналу таймера встановлюється сигнал L-рівня. Цей же стан зберігається при записі молодшого байта константи. Якщо під час рахування GATE = 0, рахування припиняється, а з появою GATE=1 - відновлюється з перерваного значення. Після закінчення рахування на виході OUT встановлюється сигнал Н-рівня. Завантаження у лічильник нового значення молодшого байта в процесі рахування зупиняє рахування, а завантаження старшого байта починає новий цикл рахування.

Режим 1. Програмовий мультивібратор. На виході лічильника формується імпульс L-рівня з програмно-керованою тривалістю, причому точкою початку відліку є задній фронт першого імпульсу CLK після появи сигналу GATE (рис. 3.46). При значенні сигналу GATE=1 на виході OUT формується імпульс L-рівня тривалістю N періодів тактових імпульсів CLK.

Завантаження в процесі рахування нового значення N не змінює поточного режиму рахування.

Мультивібратор автоматично перезапускається по кожному передньому фронту сигналу GATE .

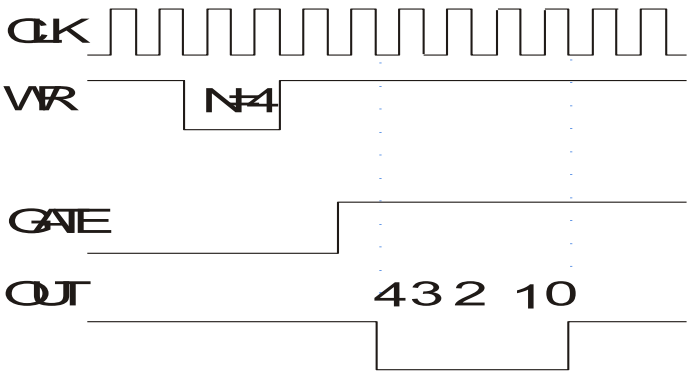


Рис. 3.46 Діаграма роботи ПТ в режимі 1

Режим 2. Програмовний генератор тактових імпульсів. У цьому режимі (рис. 3.47) обраний канал здійснює розподіл частоти імпульсів CLK на програмно-керований коефіцієнт N, тобто програмовний таймер генерує періодичний сигнал із частотою, у N разів меншою від частоти тактових імпульсів CLK. Вихідний сигнал L-рівня встановлюється на останньому такті періоду. Завантаження лічильника новим значенням N у процесі рахування призводить до зміни розміру періоду.

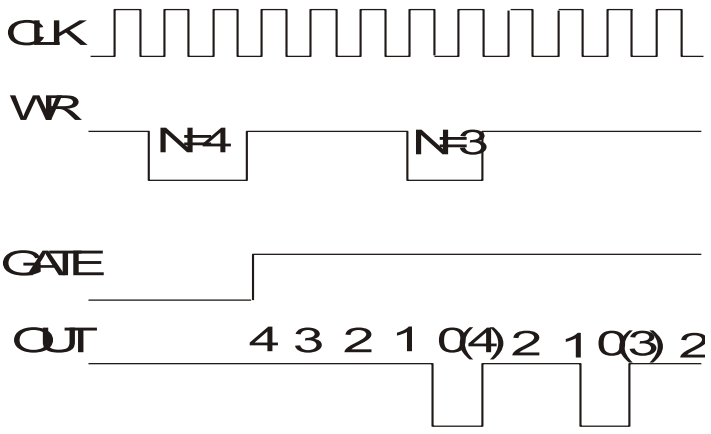


Рис. 3.47 Діаграма роботи ПТ в режимі 2

Сигнал GATE можна використовувати для зовнішньої синхронізації програмовного таймера, тому що значення GATE=0 забороняє рахування, встановлюючи значення сигналу OUT=1, а значення GATE=1 починає рахування спочатку.

Режим 3. Генератор прямокутних імпульсів. Обраний канал формує прямокутні імпульси з програмно-керованим періодом. Дія сигналу GATE аналогічна режиму "**0**". При парному значенні N на виході лічильника генерується сигнал H-рівня протягом першої половини періоду, і сигнал L-рівня протягом іншої половини. При непарному N тривалість сигналу H-рівня на один такт більше, ніж для сигналу L-рівня. (рис. 3.48). У режимі 3 число N=3 не можна завантажувати у лічильник.

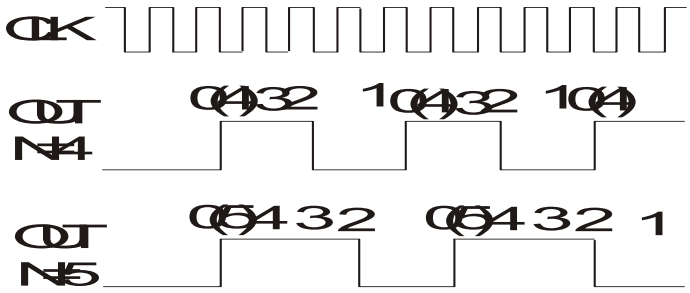


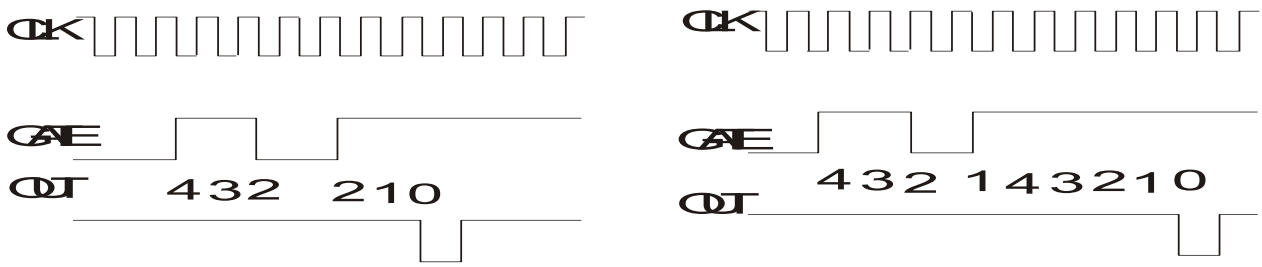
Рис. 3.48 Діаграма роботи ПТ в режимі 3

Режим 4. Програмно-керований строб. У цьому режимі на виході лічильника формується строб L-рівня тривалістю T_{CLK} з програмно-керованою затримкою щодо моменту запису молодшого байта команди. Перезавантаження молодшого байта в процесі рахування не впливає на поточне рахування, а завантаження старшого байта починає новий цикл рахування.

Режим 5. Апаратно - керований строб. Цей режим аналогічний режиму 4. Відмінність від режиму 4 полягає в тому, що початком відліку

програмно-керованої затримки є передній фронт сигналу GATE. Запуск лічильника здійснюється переднім фронтом сигналу GATE. Завантаження у лічильник нового значення N в процесі рахування не впливає на тривалість поточного циклу, але такий цикл буде відповідати новому значенню N.

На рис. 3.49 наведено діаграми роботи таймера, які ілюструють дію сигналу GATE для таймерів КР580ВИ53 (а) та КР1810ВИ54 (б).



а б

Рис.3.49. Діаграми роботи таймера: а – для таймера КР580ВИ53;
б – для таймера КР1810ВИ54

Для таймера КР580ВИ53 з появою L-рівня сигналу GATE рахування припиняється, а з появою Н-рівня – відновлюється з перерваного значення. Для таймера КР1810ВИ54 з появою L-рівня сигналу GATE рахування також припиняється, а з появою Н-рівня – починається із значення коду передвстановлення.

Приклад 3.10. *Запрограмувати лічильник СТ0 у режим генератора прямокутних імпульсів для отримання частоти* $f_{вих}$*=1 кГц.*

Приймемо адреси таймера такі, як у прикл.6.12. Для отримання послідовності імпульсів 1 кГц підключимо до виводу G0 сигнал Н-рівня, на вивід CLK0 – тактові імпульси з частотою 5 МГц.

Знайдемо значення коефіцієнта ділення як

$$N = \frac{f_{CLK}}{f_{вих}} = \frac{5000(кГц)}{1(кГц)} = 5000$$

Визначимо керуюче слово згідно з рис. 6.44 для програмування лічильника 0 у режим 3, з двійково-десятковим засобом кодування як:

00	11	011	1

=37H

Тоді програма матиме вигляд:

```
MOV  AL,37H;    програмування
OUT  16H,AL;    таймера
MOV  AL,00;     запис молодшого
OUT  10H,AL;    байту 00 передвстановлення
MOV  AL,50H;    запис старшого байта
OUT  10H,AL;    передвстановлення
```

Після виконання програми на виводі OUT0 будуть присутні прямокутні імпульси з частотою 1 кГц доти, доки не буде перепрограмовано таймер або вимкнене живлення таймера.

Приклад розробки мікропроцесорної системи

Розробити принципову схему і програмне забезпечення (ПЗ) мікропроцесорної системи керування (МСК) широтно-імпульсного стабілізатора напруги, з частотою f=1 кГц. МСК забезпечує обробку сигналу зворотного зв'язку U із виходу АЦП за законом пропорційного регулятора:

$$\gamma = 0{,}5 + (U - 20h)/256,$$

де γ – коефіцієнт заповнення імпульсів ШІП, 20h – код опорного сигналу.

Структурна схема МПС наведена на рис.3.50. До структурної схеми входять: ЦП – центральний процесор, ПЗП – постійний запам'ятовувальний пристрій, АЦП – аналого-цифровий перетворювач, ПТ - програмовний таймер.

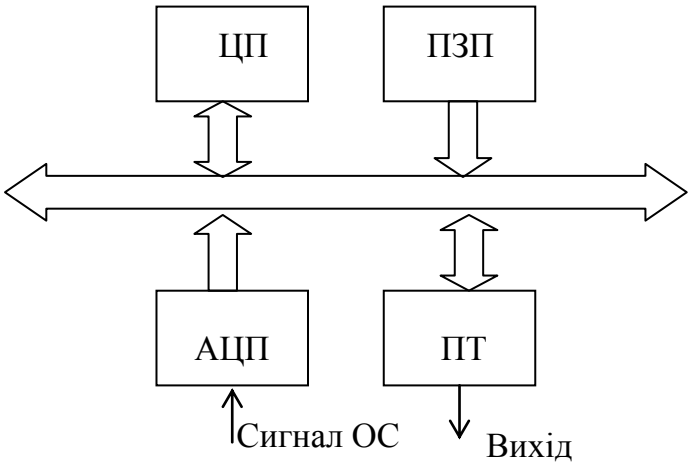


Рис. 3.50. Структурна схема МПС

Сигнал зворотного зв'язку надходить на АЦП, де він перетворюється на двійковий код. Цей код по системній шині надходить у ЦП, де за програмою, яка записана у ПЗП, він обробляється за законом пропорційного регулятора. Отримані дані - коефіцієнт заповнення ШІП - завантажуються в регістр таймера як константа. На виході таймера одержуємо логічні рівні імпульсів керування ШІП.

Функціональна схема мікропроцесорної системи керування наведена на рис. 3.51.

При розробці функціональної схеми *центрального процесора* виникають задачі демультиплексування шини адреси/даних, буферизація шин адреси (AB) і шин даних (BD), а також формування системних керуючих сигналів пам'яті і зовнішніх пристроїв. Перша задача розв’язується за допомогою двох ВІС К1810ИР82, що виконують функції адресного замка і буфера шини АВ. Друга задача - за допомогою двонапрямлених шинних формувачів К1810ВА86, що підсилюють сигнали шини даних. Третя - за допомогою комбінаційних логічних елементів, які формують керуючі сигнали на базі сигналів $\overline{RD}, \overline{WR}, M/\overline{IO}$, вироблених МП. На виході цих схем формуються сигнали $\overline{MEMR}$, $\overline{MEMW}, \overline{IOR}, \overline{IOW}$.

Внаслідок того, що в розроблювальній мікропроцесорній системі, не потрібні режими ПДП, переривань та обміну за сигналом готовності, то порівняно зі схемою модуля центрального процесора, яка наведена на рис. 3.40,, у схемі (рис. 3.51) відсутні сигнали переривань, готовності, запиту ПДП і дозволу шин BUSEN. Тому на входи ВІС генератора, ЦП, регістрів-замків та буферних регістрів подано постійні логічні рівні нуля або одиниці. Відсутні також кінцеві каскади схеми формування керуючих сигналів (у даній схемі нема потреби переводити керуючі сигнал у третій стан).

.

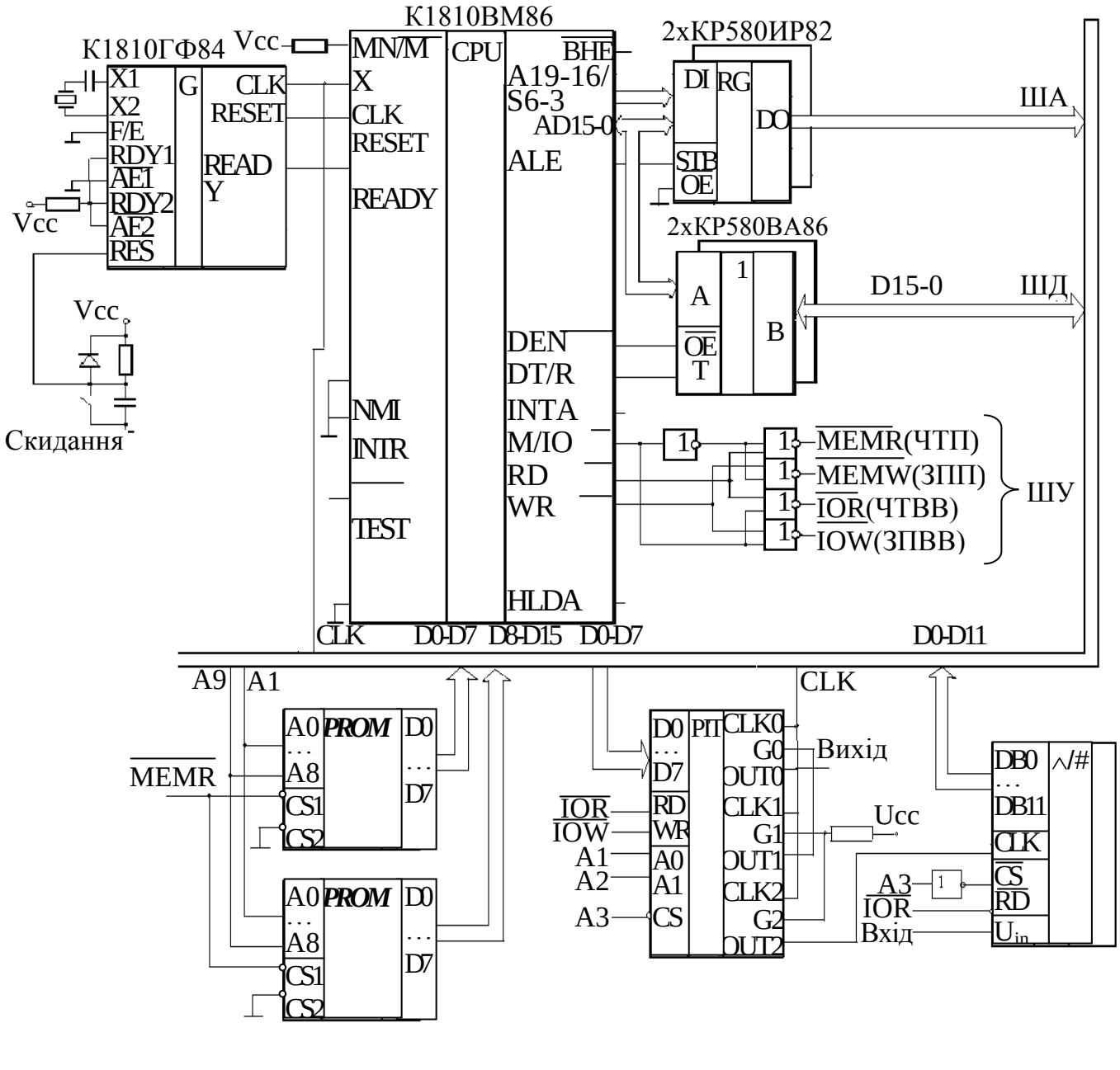


Рис.3.51. Функціональна схема МПС

Модуль ПЗП виконаний на базі двох ВІС КР556РТ5 ємністю 512 х 8 біт кожна. Пам'ять організована у вигляді двох банків пам'яті - молодшого і старшого. Молодший банк вмикається до молодшої половини шини даних D7-D0 і містить тільки комірки пам'яті з парними адресами; старший банк - до старшої половини шини даних D15 - D8 і містить тільки комірки пам'яті з непарними адресами. Читання з ПЗП відбувається при виконанні циклу читання пам'яті, при цьому формується сигнал $\overline{MEMR} = 0$, який і переводить

виходи ВІС ПЗП у активний стан. З ПЗП завжди зчитується слово. Для даного прикладу початкову адресу ПЗП визначимо при нульових значеннях A9… A1, A0, а кінцеву – при одиничних Таким чином, початкова адреса ПЗП - 00000H; кінцева адреса ПЗП - 003FFH

Функціональна схема містить також *АЦП* К572ПВ1, який являє собою 12-розрядний перетворювач напруги на двійковий код низької швидкодії. Внаслідок того, що АЦП має внутрішній регістр з входом керування третім станом, зовнішній порт введення не потрібний. Вихід АЦП ввімкнений до ліній D11-D0. З точки зору процесора АЦП являє собою 16-розрядний порт. Адреса 16-розрядного порту повинна бути парною. Як видно з рис. 3.51, АЦП вибирається при A3=1. Таким чином, адреса АЦП може бути будь-яка при A3=1, A0=0. Наприклад, оберемо адресу АЦП, що дорівнює 08H.

Програмовний таймер К1810ВИ54 в даній схемі призначений для генерації імпульсів керування ШІП. Таймер містить три незалежних канали, кожний з яких може бути запрограмований на роботу в одному з шести режимів для двійкового та двійково-десяткового обчислення. У даному прикладі використовуються наступні режими роботи каналів:

- канал 0 - програмовний мультивібратор;
- канал 1 - імпульсний генератор частоти для запуску каналу 0;
- канал 2 - імпульсний генератор для задавання частоти роботи АЦП.

Як видно із рис.3.51, таймер обирається при адресі з A3=0. Лінії A1 і A2 обирають один з трьох каналів таймера або регістр керуючого слова. Таким чином, адреси таймера будуть:

- адреса каналу 0 - 00H;
- адреса каналу 1- 02H;
- адреса каналу 2- 04H;
- адреса РКС - 06H.

Розрахуємо константи завантаження таймера таким чином:

Перетворімо коефіцієнт заповнення *ШІП* gamma на константу перерахунку, що завантажується в таймер, з урахуванням того, що частота напруги дорівнює 1 кГц, частота тактових імпульсів $f_{CLK} = 5$ МГц .

Канал 0. Згідно з завданням, коефіцієнт заповнення імпульсів ШІП дорівнює

$$\text{gamma} = 0{,}5 + \frac{1}{256}(\text{I-20h}).$$

Визначимо період роботи ШІП як

$$\text{T}= \frac{1}{f} = \frac{1}{10^3} = 1 \text{ (мс)}.$$

Тоді тривалість імпульсу буде:

$$\tau_i = \frac{T}{2} + \frac{T}{256}(U - 20h) = 0{,}5 + \frac{1}{256}(U - 20h) \text{ (мс)}.$$

Тривалість лічильних імпульсів каналів (CLK) при частоті роботи процесора 5 МГц:

$$T_{CLK}=1/(5*10^6)=200 \text{ (нс)}.$$

Код завантаження каналу 0 таймера визначиться як відношення τ_i до T_{CLK}, тобто

$$N_0= \frac{0{,}5 \cdot 10^{-3}}{200 \cdot 10^{-9}} + \frac{10^{-3}}{256 \cdot 200 \cdot 10^{-9}}(U - 20h) \approx 2500 + 20(U - 20h).$$

Канал 1. Коефіцієнт ділення каналу 1 визначається як відношення тривалості періоду ШІП до тривалості періоду $T_{CLK,}$ тобто

$$N_1 = \frac{T}{T_{CLK}} = \frac{1000000}{200} = 5000.$$

Канал 2. Коефіцієнт ділення каналу 2 визначається як відношення тривалості періоду тактових імпульсів АЦП (170 кГц) до тривалості періоду $T_{CLK,}$ тобто

$$N_2 = \frac{T_{АЦП}}{T_{CLK}} = 29.$$

Алгоритм ПО наведений на рис. 3.52.

Рис. 6.65. Алгоритм ПО

Спочатку в усі три канали таймера завантажуються керуючі слова. Керуюче слово для каналу 0 дорівнює 00110010B і визначає режим 1 лічильника 0, завантаження спочатку молодшого, а потім старшого байтів, рахунок – двійковий . Керуюче слово для каналів 1 і 2 визначають режим 1, завантаження спочатку молодшого, а потім старшого байтів, рахунок – двійковий. Керуючі слова дорівнюють 01110100B і 10110100B відповідно.

Після завантаження керуючих слів програма завантажує початкові значення у канали 1 та 2 таймера. На цьому ініціалізація закінчується і починається нескінченний цикл введення коду зворотної напруги з АЦП, перерахування її у код завантаження каналу 0 і виведення у канал 0.

Зазначимо, що після системного скидання регістри CS і IP мікропроцесора набудуть значень 0FFFFH і 0000H відповідно. Це призведе до того, що в адресі першої комірки ПЗП, до якої після скидання звернеться мікропроцесор у першому машинному циклі ВИБІРКА КОМАНДИ, чотири

молодших розряди будуть нульовими, а усі інші – одиничними. У даному випадку ця адреса буде дорівнювати 3F0H.

Програма має вигляд:

```
        ORG 3F0H
        JMP START
        ORG 100H
START:  MOV AL, 00110010B   ;формування в AL керуючого слова режиму
                                ;1 каналу 0
        OUT 06, AL          ;виведення керуючого слова в РКС
        MOV AL, 01110100B   ;формування в AL керуючого слова в РКС
        OUT 06, AL          ;виведення керуючого слова режиму 2
                            ;каналу 1
        MOV AL,10110100B    ;формування в AL керуючого слова режиму
                            ;каналу 2
        OUT 06, AL          ;виведення керуючого слова режиму РКС
        MOV AX, 5000         ;задавання частоти перетворення f =1кГц
        OUT 02H, AL         ;запис молодшого байта коду
                            ;передвстановлення в канал 1
        MOV AL, AH          ;запис старшого байта коду
                            ;передвстановлення  в канал1
        OUT 02H,AL          ;задавання частоти АЦП f  = 170 кГц
        MOV AL,29           ;запис коду передвстановлення в канал 2
        OUT 04H,AL
        MOV AL,00
        OUT 04H,AL
L:      IN AX,08H           ;введення сигналу зворотного зв'язку U;
        AND AX,0000 1111 1111 1111B      ; виділення значущих цифр U
        SUB AX,20H                  ;віднімання U - 20H, результат у AX;
        MOV BX,20
        MUL BX              ;множення на 20:20(U-20H), результат в AX
        ADD AX,2500         ;додавання 2500+20(U-20H), результат в AX
        OUT 00H,AL          ;запис молодшого байта коду
                            ;передвстановлення в канал 0
        MOV AL,AH
        OUT 00H,AL          ;запис старшого байта коду
                            ;передвстановлення в канал 0
        MOV CX,100          ;затримка на час, більше перетворення АЦП
D:      LOOP D              ;16x100x200 (нс)
        JMP L               ;перейти за адресою з міткою L (цикл);
        END.
```

Розділ 4. СТАРШІ МОДЕЛІ ОДНОКРИСТАЛЬНИХ МІКРОПРОЦЕСОРІВ

ЛЕКЦІЯ 14

Мікропроцесор i80286

Загальні відомості. Мікропроцесор i80286 відноситься до другого покоління 16-розрядних мікропроцесорів. Він виконаний за технологією 1,5 мкм, містить 134000 транзисторів і працює з тактовою частотою 12,5 МГц. За рахунок вдосконалення архітектури швидкодія i80286 в 6 раз вища, ніж i8086 з тактовою частотою 5 МГц. Розрядність регістрів дорівнює 16. Шина адреси є 24-розрядною, що дозволяє адресувати 2^{24}=16М байт пам'яті. Простір адрес введення/виведення складає 64К байт. Система команд містить всі команди i8086, декілька нових команд загального призначення і групу команд керування захистом. Мікропроцесор має спеціальні засоби для роботи у системах з багатьма користувачами та багатозадачних режимах. Найбільш суттєвою відмінністю від мікропроцесорів серії i8086/88 є механізм керування адресацією пам’яті, що забезпечує чотирирівневу систему захисту та підтримку віртуальної пам’яті. Спеціальні засоби призначені для підтримки механізму перемикання задач. Мікропроцесор i80286 має засоби контролю за переходом через межу сегмента, які працюють в реальному режимі.

Мікропроцесор може працювати в двох режимах:

1. ***8080 Real Address Mode*** (або ***Real Mode***) - режим реальної адресації або реальний режим. У цьому режимі МП i80286 фактично являє собою високошвидкісний i8086 і адресує 1 Мбайт пам'яті.
2. ***Protected Virtual Address Mode*** (або ***Protected Mode***) – захищений режим віртуальної адресації, або просто захищений режим. В цьому режимі МП

адресує до 16М байт пам'яті, а при використанні механізму сторінкової адресації до 1Г байт віртуальної пам'яті кожної задачі.

Перемикання в захищений режим здійснюється швидко – за допомогою однією команди (із заздалегідь підготовленими таблицями дескрипторів), а в режим реальної адресації – повільно, тільки через апаратне скидання процесора. В MS DOS використовується реальний режим. Захищений режим використовується в операційних системах типу XENIX, UNIX, OS/2, NetWare286, MS Windows.

Для процесора i80286 можливі 256 різноманітних типів переривань. Відміною від системи переривань МП i8086 є переривання при виникненні особливих умов, в ході виконання команд наприклад, при розміщенні двобайтового операнда у останній комірці сегменту даних зі зміщенням FFFFH. Таке переривання називається ***особливим випадком***, або ***винятком***. На відміну від переривань після обробки винятків (крім винятку 9, що стосується співпроцесора) керування передається знову тій самій команді (включаючи всі префікси), що викликала переривання. Після усунення умов, що викликали виняток, відбувається повторне виконання команди.

Організація пам'яті. У реальному режимі адресація пам'яті переважно така ж, як і у МП i8086. Відмінність полягає у можливості використання додаткового блока пам'яті ємністю 64К байт. Якщо в ході виконання команди при обчисленні адреси комірки пам’яті виникає переповнення у двадцятий розряд шини адреси A20, процесор починає працювати з комірками пам’яті, адреси яких лежать в діапазоні 100000H-10FFFFH.

Приклад 4.1. *Знайти значення фізичної адреси операнду в реальному режимі при виконанні команди* **MOV AL, [SI];** *пересилання у регістр AL вмісту комірки пам’яті з адресою DS:SI, якщо вміст сегментного регістра DS є число 0F802H, вміст регістра SI – 0B175H.*

Для того, щоб обчислити фізичну адресу, додамо до значення DS чотири нулі праворуч:

DS(0000) = 1111 1000 0000 0010 0000B=0F8020H.

Виконавши операцію додавання отриманої величини з вмістом регістра SI, отримаємо фізичну адресу:

```
1 1111              11          - рядок перенесень
+ 1111 1000 0000 0010 0000
     1011 0001 0111 0101
1 0000 0011 0001 1001 0101=103195H.
```

Таким чином, при обчисленні фізичної адреси отримане одиничне значення розряду A20, означає що операнд буде розташований у другому мегабайті фізичної пам’яті.

Ще однією особливістю реального режиму i80286 є можливість контролю за переходом за межі сегмента. При адресації слова зі зміщенням 0FFFFH генерується виняток 13 (Segment Overrun Exception). При спробі виконання команди ESC з операндом пам'яті, що не вміщається у сегменті, виробляється виняток 9 – Processor Extension Segment Overrun Interrupt.

У захищеному режимі також використовується сегментна адресація, кількість сегментів може бути від 1 до 16М байт, довжина сегментів задається і може варіюватися від 1 до 64К байт, оговорюються атрибути чи права доступу до сегмента (дозвіл запису чи тільки читання, рівні привілеїв тощо). Кожний сегмент характеризується 8-розрядною структурою даних – ***дескриптором сегмента***, яка містить інформацію про базову адресу сегмента, його межу та атрибути. Дескриптори розміщуються у спеціальних таблицях – глобальній дескрипторній таблиці GDT або локальній дескрипторній таблиці LDT, які зберігаються в ОЗП. Незалежно від рівня привілею програма не може звертатися до сегмента доти, доки він не описаний у дескрипторній таблиці.

У захищеному режимі вміст сегментних регістрів називається ***селекторами сегментів.*** Вони використовуються для пошуку базової (початкової) адреси сегмента в одній з дескрипторних таблиць. Формат селектора наведено на рис. 4.1.

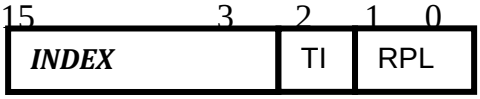


Рис.4.1. Формат селектора сегмента

Селектори, що завантажуються у 16-бітові сегментні регістри CS, DS, SS, ES, мають три поля:

- RPL (Requested Privilege Level) (біти 0 і 1) - запрошений рівень привілеїв сегмента;
- TI (Table Indicator) (біт 2) - індикатор використання таблиці дескрипторів (при TI=1 використовується глобальна таблиця, при 0 - локальна);
- INDEX (біти 3-15) - номер дескриптора в таблиці.

Глобальна дескрипторна таблиця єдина. Вона містить дескриптори для усіх задач, які виконує МП у багатозадачному режимі. Локальних дескрипторних таблиць може бути кілька - для кожної задачі може бути задана своя локальна дескрипторна таблиця.

Фізична 24-розрядна адреса операнда знаходиться шляхом додавання початкової 24-розрядної адреси сегмента із дескрипторної таблиці і адреси-зміщення, яка вказана у команді. Формування фізичної 24-розрядної адреси у захищеному режимі ілюструється рис. 4.2.

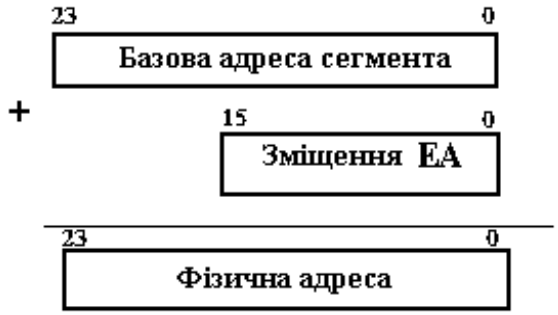


Рис. 4.2. Формування фізичної 24-розрядної адреси у захищеному режимі

Приклад 4.2. *Знайти значення фізичної адреси комірки пам'яті [DS:SI], якщо базова адреса сегмента даних дорівнює 456789H, а вміст регістра SI – 1234H.*

Виконавши операцію додавання 24-розрядної базової адреси з адресою-зміщенням, тобто з вмістом регістра SI, отримаємо фізичну адресу:

```
  0100 0101 0110 0111 1000 1001
+                 0001 0010 0011 0100
  0100 0101 0111 1001 1011 1101      = 4579BBH.
```

Таким чином, фізична адреса дорівнює 4579BBH.

Програмна модель.

До програмної моделі МП i80286 (рис. 4.3) входять 19 програмно-доступних регістрів і 6 недоступних (тіньових).

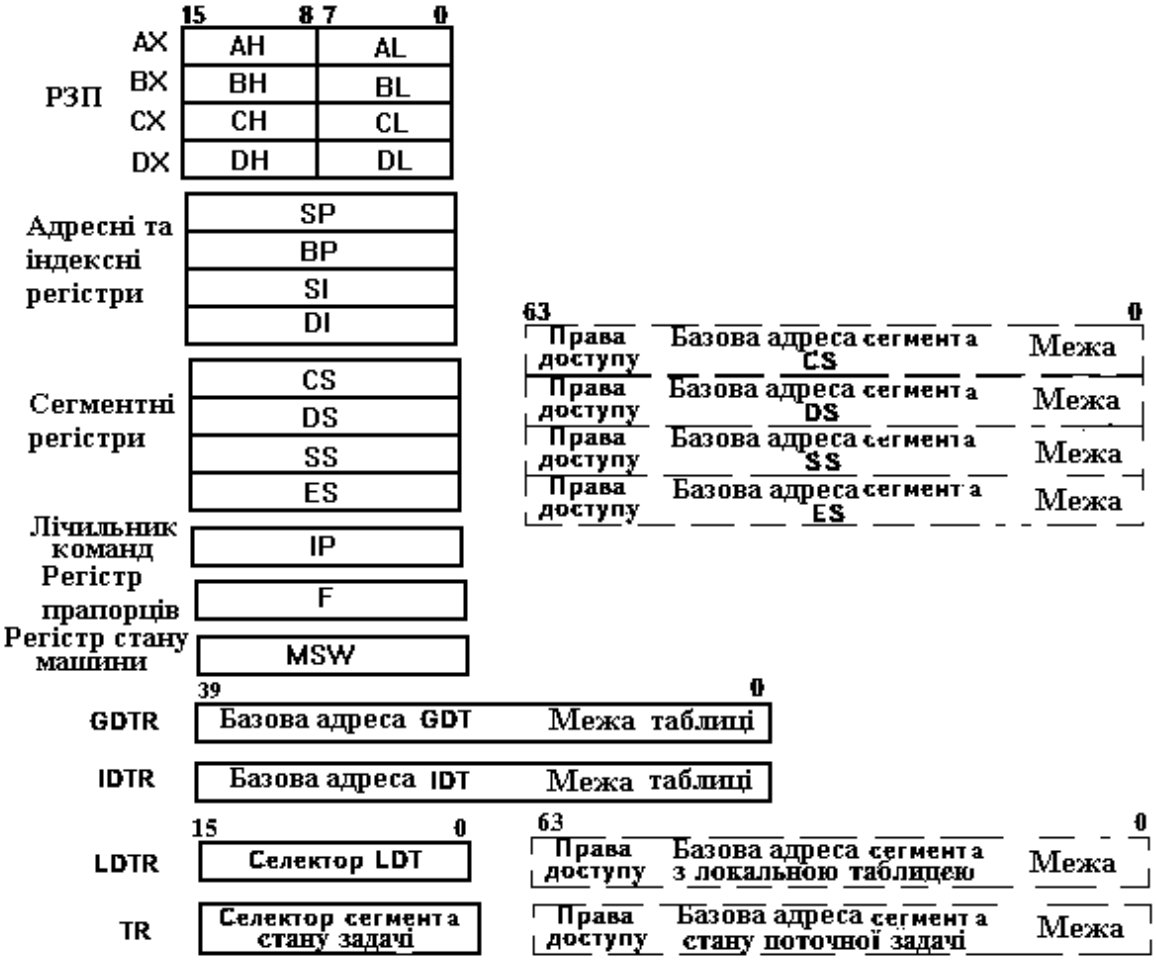


Рис. 4.3. Програмна модель МП i80286

З 19 програмно-доступних регістрів МП i80286 14 повторюють регістри процесора i8086. П’ять нових регістрів наступні:

- 40-розрядний регістр GDTR (Global Descriptor Table Register) - регістр глобальної дескрипторной таблиці. Призначений для завдання розташування глобальної дескрипторної таблиці в пам'яті. Регістр GDTR безпосередньо містить базову адресу і значення межі таблиці, яке задає її розмір. Розмір таблиці більше від значення межі на одиницю;
- 16-розрядний регістр LDTR (Local Descriptor Table Register) - регістр-селектор локальної дескрипторної таблиці. Його вміст вказує, де в глобальній дескрипторній таблиці знаходиться інформація про початкову адресу, межу і права доступу до локальної таблиці. Ця інформація переписується у тіньовий програмно-недоступний регістр;
- 40-розрядний регістр IDTR (Interrupt Descriptor Table Register) - регістр дескрипторної таблиці переривань. Завдяки цьому регістру в МП i80286 з'явилася можливість розміщувати таблицю векторів переривань в довільному місці ОЗП, а не з нульових адрес, як в МП i8086. Регістр IDTR за структурою аналогічний регістру GDTR;
- 16-розрядний регістр задачі TR (Task Register) – регістр-селектор сегмента стану поточної задачі TSS (Task State Segment). Такі сегменти асоціюються з кожною задачею і призначені для збереження контексту задачі при переключенні задач;
- 16-розрядний регістр стану машини MSW (Machine State Word), що керує режимом процесора і містить наступні біти (рис. 4.4):

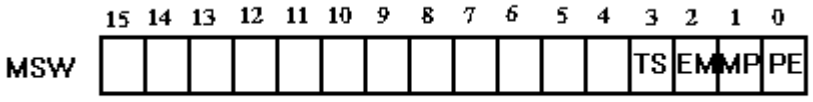


Рис. 4.4. Регістр стану машини MSW

PE (Protecrion Enable) – дозвіл захисту. Встановлення цього прапорця переводить процесор у захищений режим. Повернення до реального режиму можливе тільки за сигналом RESET.

MP (Monitor Processor Extension) – присутність співпроцесора. Якщо арифметичний співпроцессор i80287 під'єднаний до процессору i80286, то при ініціалізації операційна система повинна встановити цей біт в одиничний стан. Тоді при виконанні команди WAIT, а також при виконанні команди ESC і значенні TS=1 процесор буде генерувати виняток 7.

EM (Processor Extension Emulated) - емуляція співпроцесора. Встановлення цього прапорця викликає виняток 7 при виконанні команд арифметичного співпроцесора.

TS (Task Switch) - переключення задач. При встановленні прапорця наступна команда, яка належить до співпроцесора, викличе виняток 7.

Тіньові регістри відіграють роль надоперативної пам'яті і призначені для підвищення швидкодії роботи МП. На рис. 4.3 тіньові регістри показані пунктиром.

Крім того, у програмну модель порівняно з моделлю i8086, додані нові біти у регістрі прапорців і змінене використання сегментних регістрів у захищеному режимі. У регістрі прапорців біт 14 визначений як NT (Nested Task Flag) - прапорець вкладеної задачі і біти 13-12 як 2-бітове поле IOPL (Input/Output Privilege Level) рівня привілеїв. Ці прапорці діють тільки в захищеному режимі. Прапорець NT встановлюється в 1 при переключенні задач за допомогою команди CALL. При виконанні команди IRET перевіряється стан прапорця NT і якщо NT=1, здійснюється переключення задач, в протилежному випадку відбувається звичайне повернення з переривання. Двобітове поле IOPL вказує рівень привілею поточної задачі, при якому дозволяється виконання певних операцій.

Сегментні регістри CS, SS, ES і DS визначають початкові адреси сегментів. У реальному режимі 20-розрядна початкова адреса сегмента

визначається як вміст 16-розрядного сегментного регістра, доповненого праворуч чотирма нульовими бітами. У захищеному режимі початкова 24-розрядна базова адреса сегмента знаходиться у дескрипторній таблиці в ОЗП, а вміст сегментних регістрів є селекторами, які вказують на тип таблиці і номер запису у таблиці (див. рис.4.1).

При завантаженні нового значення селектора дескриптори зчитуються з ОЗП і запам’ятовуються у внутрішніх програмно-недоступних, або тіньових регістрах процесора. Це дозволяє підвищити швидкодію процесора, бо зміна значень базових адрес сегментів відбувається порівняно рідко.

Регістри GDTR, LDTR, IDTR задають розташування дескрипторних таблиць у пам'яті (рис. 4.5).

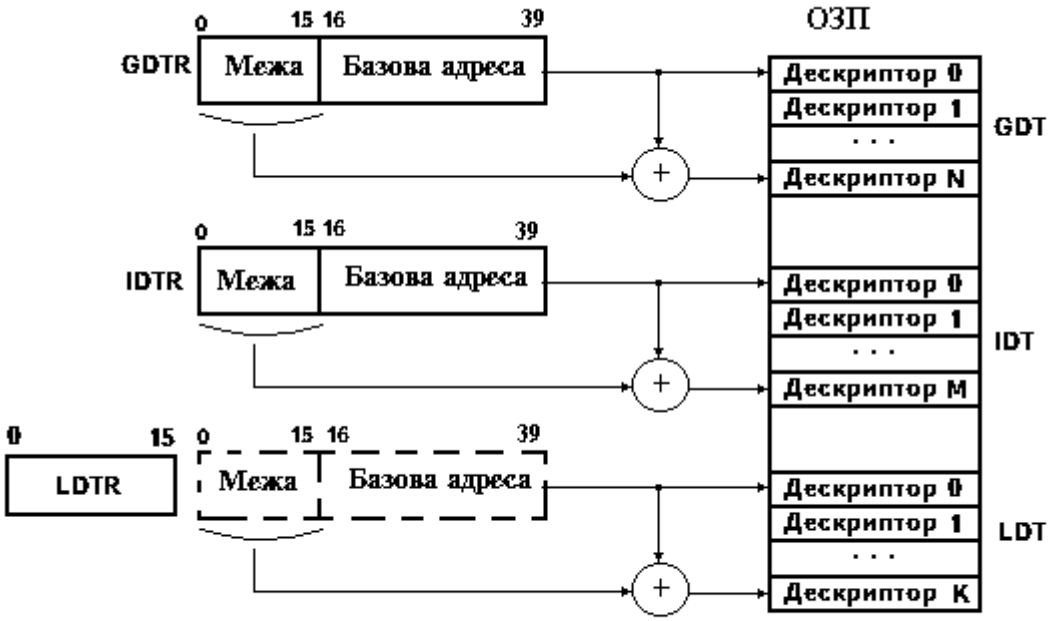


Рис. 4.5. Розташування дескрипторних таблиць у пам'яті

На рис. 4.5 показано, що глобальна дескрипторна таблиця містить N+1 дескрипторів, локальна – K+1, дескрипторна таблиця переривань – M+1. Початкова адреса глобальної дескрипторної таблиці визначається бітами 39-16 регістра GDTR, остання - визначається додаванням значень початкової

адреси і межі таблиці (біти 15-0). Аналогічно початкова і остання адреси таблиці переривань визначаються вмістом регістра IDTR. Для адресації дескрипторів локальної таблиці використовують вміст регістра LDTR, який є селектором. Він визначає номер дескриптора у глобальній дескрипторній таблиці. Цей дескриптор завантажується у тіньовий регістр (на рис.4.5 показано пунктиром). Перша та остання адреси локальної дескрипторної таблиці визначаються вмістом цього регістра.

Приклад 4.3. *Знайти значення, яке треба завантажити у регістр* GDTR, *щоб задати у пам'яті глобальну дескрипторну таблицю з 21 дескрипторів і з початковою адресою 000100H.*

Біти 39-16 регістра GDTR задають початкову 24-розрядну адресу, отже повинні дорівнювати 000100H. Кожний запис у таблиці займає 8 байт, отже адреса 21 дескриптора буде: 000100H+20*8= 0001A0H. У бітах 15-0 регістра GDTR треба розмістити число 0A0H. Таким чином, вміст регістра GDTR дорівнює

0001000AH.

Зазначимо, що команди завантаження регістрів таблиць GDTR, LDTR, IDTR - LGDT, LLDT, LIDT є привілейованими і виконуються у програмах з вищим рівнем пріоритету.

.

Пристрої введення-виведення

Адресний простір портів МП i80286 так само, як і для МП i8086, складає 64×2^{10} однобайтових або 32×2^{10} двобайтових портів. Додаткові рядкові команди REP INSB/INSW, REP OUTSB/OUTSW забезпечують блокову обробку зі швидкістю, що перевищує аналогічні операції в режимі ПДП. У захищеному режимі команди є привілейованими, тобто вони можуть виконуватися тільки з певним рівнем привілею, що визначаються полем IOPL регістра прапорців. В протилежному випадку викликається виняток 13 – порушення захисту

Цикли шини

МП i80286 містить шинний інтерфейс з 6-байтовою чергою команд, що забезпечує конвеєрну адресацію (Pipelined Addressing). Це забезпечує вибірку кодів команд або даних з пам'яті з випередженням, що дає можливість почати фазу ідентифікації або адресації нового циклу, не дочекавшись закінчення попереднього.

Розрізняють 6 типів циклів шини:

1. ЧИТАННЯ ПАМ'ЯТІ.
2. ЗАПИС У ПАМ'ЯТЬ.
3. ЧИТАННЯ ПОРТУ.
4. ЗАПИС У ПОРТ.
5. ПІДТВЕРДЖЕННЯ ПЕРЕРИВАННЯ.
6. СТОП/ЗУПИН.

Відзначимо, що слово з непарною адресою передається за два цикли шини, з парною – за один. На рис. 4.6 наведені цикли шини ЧИТАННЯ і ЗАПИС слова у комірку пам'яті з парною адресою. Читання слова з ОЗП здійснюється, як мінімум, за 4 періоди тактових імпульсів CLK або за 2 стани процесора (без врахування сигналу готовності READY). Кожний стан МП триває 2 такти CLK. Під час першого стану, позначеного Ts, процесор виставляє на шину адреси значення адреси комірки пам'яті, по якій буде читатися слово. Завдяки конвеєрній адресації адреса комірки виставляється з деяким випередженням, однак вона не зберігається на шині адреси протягом циклу. Для сумісності з шиною ISA сигнали шини адреси запам’ятовуються у регістрах-замках за стробом ALE. Керуючі сигнали читання пам'яті $\overline{\text{MRDC}}$ і адресний строб ALE формуються системним контролером 82288 на початку другого стану Tc. Сигнали керування і адреси обробляються схемою керування пам'яттю, в результаті чого, починаючи з середини другого стану

Тс, на шині даних з'являється слово з ОЗП, і процесор зчитує його з шини даних.

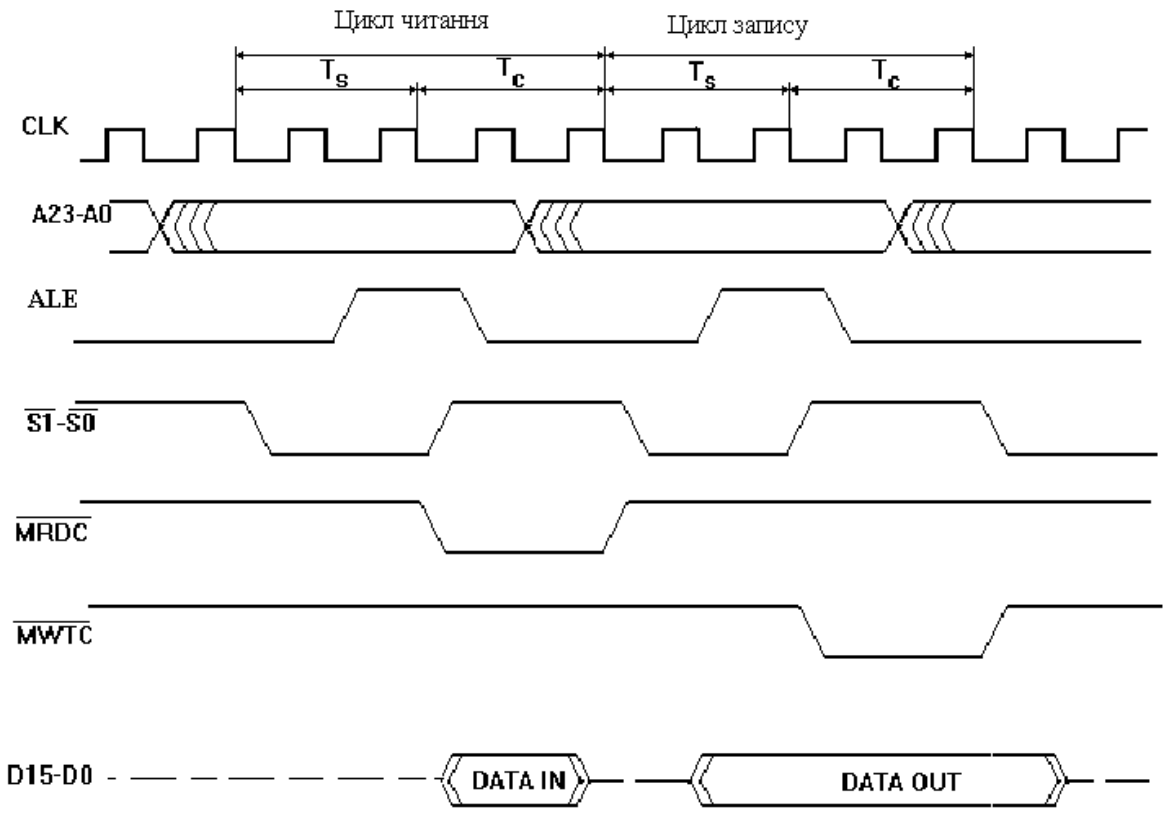


Рис. 4.6. Цикли шини ЧИТАННЯ і ЗАПИС слова у комірку пам'яті з парною адресою

При зчитуванні слова з непарною адресою в першому циклі переноситься байт за старшою половиною шини даних D15-D8, в другому – передається другий байт за молодшою половиною D7-D0.

Цикл шини ЗАПИС У ПАМ'ЯТЬ з парною адресою також триває 2 стани. У першій половині циклу Ts виставляється адреса і дані, в другій - відбувається запис в ОЗП.

Виконання циклів шини ЧИТАННЯ ПОРТУ та ЗАПИС У ПОРТ аналогічне розглянутим вище циклам ЧИТАННЯ ПАМ'ЯТІ та ЗАПИС У ПАМ'ЯТЬ, однак в цьому випадку на шину A15-A0 виставляється адреса порту і замість сигналів читання або запису пам’яті генеруються сигнали

$\overline{IOR}$ - для читання портів введення/виведення, або $\overline{IOW}$ - для запису у порти введення/виведення.

Цикл шини ПІДТВЕРДЖЕННЯ ПЕРЕРИВАННЯ (рис. 4.7) виконується при виникненні апаратного переривання (наявності активного рівня на виводі INTR). Цикл складається з двох процесорних циклів, поділених трьома тактами очікування T_W.

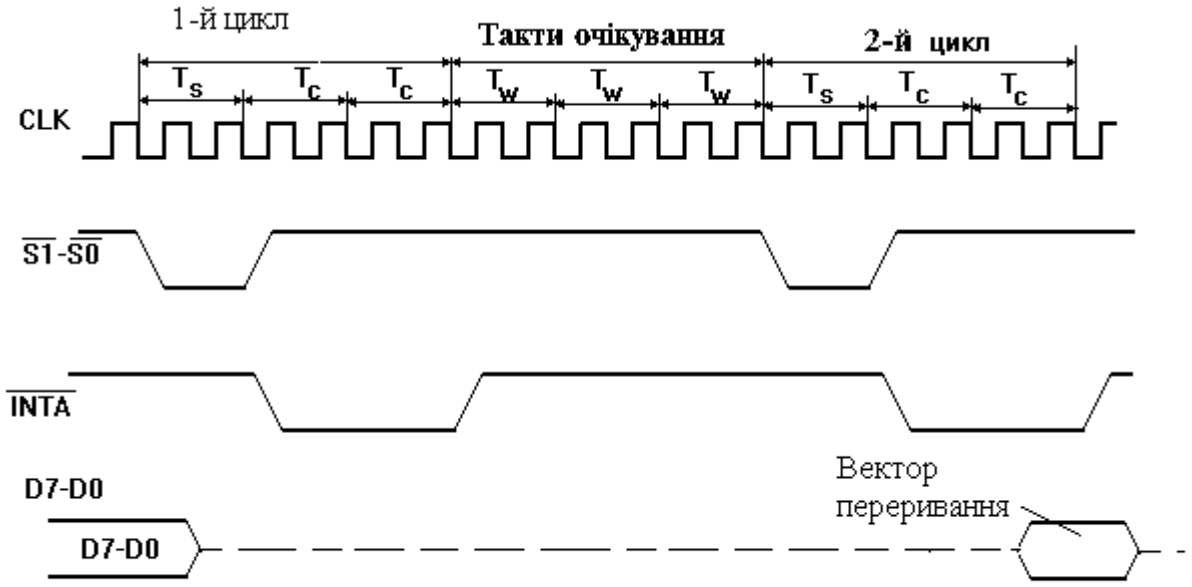


Рис. 4.7. Цикл шини ПІДТВЕРДЖЕННЯ ПЕРЕРИВАННЯ

Кожний процесорний цикл складається з трьох станів – Ts, Tc, Tc. Це необхідно для того, щоб подовжити дію сигналу INTA, що посилається на контролер переривань 8259A. Перший цикл дозволяє ведучому контролеру визначити, який з ведених контролерів викликав переривання. У другому циклі МП читає з шини даних вектор переривання, що використовується для знаходження адреси у таблиці векторів переривань.

Цикл шини СТОП/ЗУПИНКА (Halt/Shutdown) виникає або при виконанні команди HLT, або при обробці особливого випадку 8 у захищеному режимі.

Переривання та виключення

Типи переривань. Переривання і винятки в реальному і захищеному режимах наведені в табл. 4.2.

Таблиця 4.2. Переривання і винятки МП i80286

Номер	Адреса повернення	Функції	
		Реальний режим	Захищений режим
1	2	3	4
0	Перший байт команди	Помилка ділення - виникає, якщо частка занадто велика або дільник дорівнює нулю	
1	Наступна команда	Переривання покрокової роботи	
2	Немає	Немасковане переривання	
3	Наступна команда	Контрольна зупинка (INT 3)	
4	Наступна команда	Особливий випадок переповнення (INTO)	
5	Перший байт команди	Особливий випадок виходу з діапазону - виникає при виконанні команди BOUND	
6	Перший байт команди	Неприпустимий код операції – виникає, якщо зустрічається недійсний код операції або команда довжиною більше 10 байт	
7	Перший байт команди	Співпроцесор недоступний	Співпроцесор недоступний або відбулося переключення задач
8	Перший байт команди	-	Подвійна відмова - ситуація, коли МП виявляє два незалежних винятки при відпрацюванні однієї команди. Якщо під час його обслуговування відбудеться виняток, відбувається вимкнення процесора (Shutdown). У цьому стані процесор припиняє свої дії і виводиться з нього сигналами скидання або немаскованого переривання
9	Невизначена	Порушення межі сегмента співпроцесором - виникає, якщо операнд співпроцесора не поміщається у сегмент, наприклад, 16-розрядний операнд має зміщення 0FFFFH	

10	Перший байт команди	-	Неприпустимий сегмент стану задачі
11	Перший байт команди	-	Сегмент відсутній
12	Перший байт команди	Порушення межі сегмента стека	Порушення межі сегмента стека або стек відсутній
13	Перший байт команди	Порушення межі сегмента даних або коду – виникає, якщо операнд або код операції не поміщається в сегмент	Порушення захисту – виникає в наступних випадках: при виході за межі таблиці дескрипторів; при порушенні привілеїв; при завантаженні невірного дескриптора або типу сегмента; при спробі запису у сегмент коду або сегмент даних, які тільки для читання, при читанні тільки з виконуваного сегмента кодів; при спробі виконати привілейовані команди, що можна виконувати тільки при певних рівнях CPL і IOPL
14	Зарезервовано		

1	2	3	4
15	Зарезервовано		
16	Наступна команда ESC або WAIT	Особливий випадок співпроцесора – виникає в МП i80286 при будь-якому незамаскованому особливому випадку у співпроцесорі	
17	Зарезервовано		
.....			
31	Зарезервовано		

Кожному номеру переривання відповідає елемент у таблиці IDT дескрипторів переривань, яка містить вектори переривань. Після скидання МП, а також під час роботи МП в реальному режимі, таблиця векторів IDT розташується, починаючи з нульової адреси, однак командою завантаження регістра IDTR можна змінити місце її розміщення у межах першого мегабайта і зменшити розмір таблиці.

Приклад 4.5. *Визначити, чи виникне переривання під час роботи МП у реальному режимі при виконанні команди*
MOV AX,[0FFFFH].

При виконанні цієї команди в акумулятор AX пересилається слово з сегмента даних, причому молодший байт слова має зміщення 0FFFFH, а старший – 0000H. Цей випадок є порушенням межі сегмента і виникає переривання 9.

ЛЕКЦІЯ 15

Архітектура 32-розрядних мікропроцесорів

Існуючі 32-розрядні МП i386, i486, Pentium, Pentium Pro і Pentium II мають розрядність регістрів та шини адреси, яка дорівнює 32. Шина даних для процесорів i386, i486 є 32-розрядною, а для процесорів Pentium, Pentium Pro і Pentium II – 64-розрядною. Вони дозволяють адресувати 4 Гбайт пам'яті, мають засоби підтримки сегментної і сторінкової адресації пам’яті. Процесори мають чотирирівневу систему захисту пам'яті і портів введення/виведення, можуть працювати у багатозадачному режимі. До режимів роботи МП i80286 доданий Virtual Real Mode - режим віртуального процесора i8086. Мікропроцесори припускають паралельну роботу декількох віртуальних процесорів i8086 під керуванням операційної системи типу Windows, OS/2, Unix. Процесори оперують з бітами, полями бітів, 8-, 16- та 32-бітовими операндами, рядками бітів, байтів, слів (16-розрядних даних) і подвійних слів (32-розрядних даних). У архітектуру процесорів введені засоби налагодження і тестування.

Програмна модель

Програмна модель 32-розрядного процесора подана на рис. 4.8. Вона містить наступні групи регістрів: 1) регістри загального призначення; 2) лічильник команд; 3) регістр прапорців; 4) сегментні регістри; 5) регістри керування; 6) системні адресні регістри; 7) регістри налагодження; 8) регістри тестування.

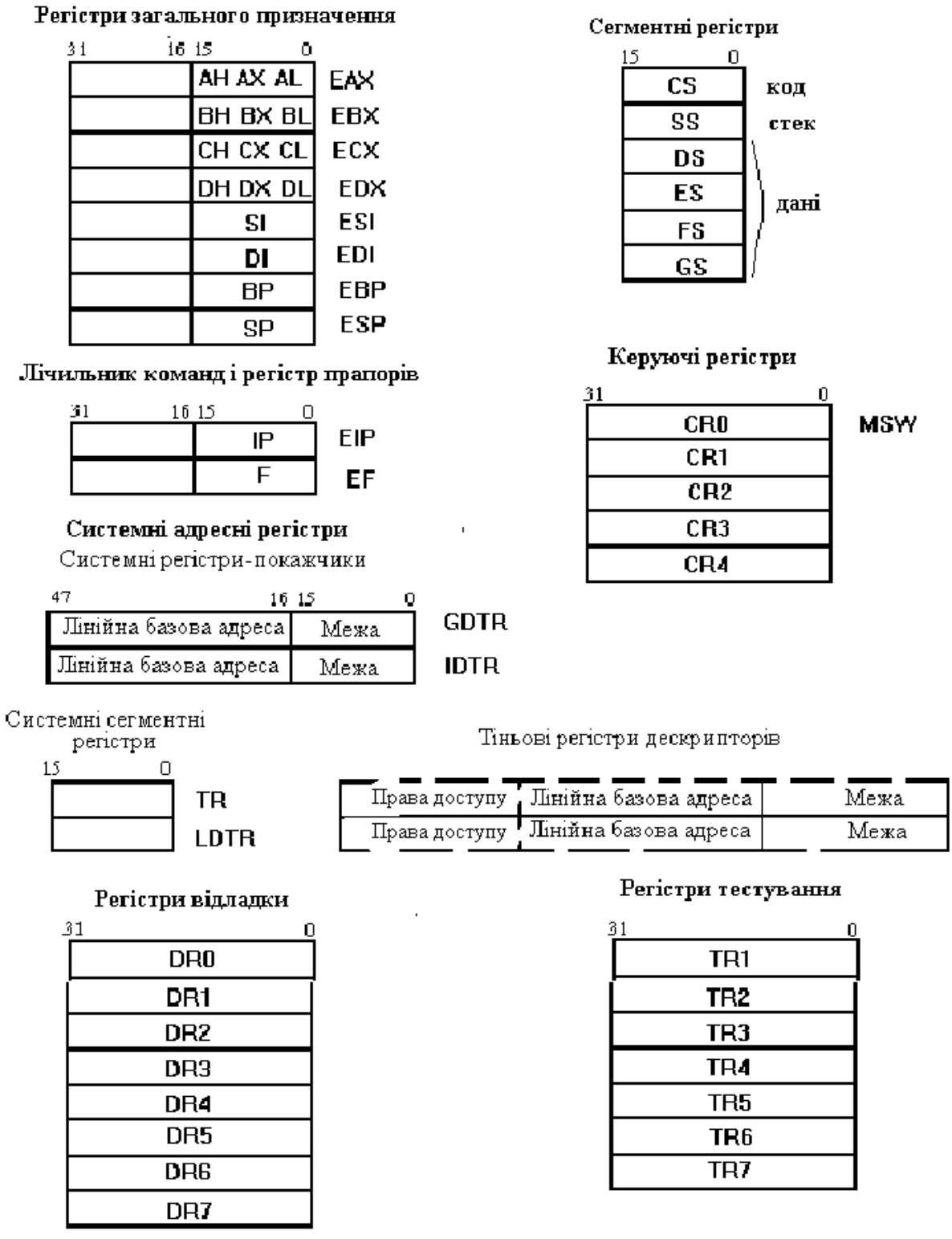


Рис. 4.8. Програмна модель 32-розрядного процесора

Регістри загального призначення містять всі регістри даних і регістри-вказівники МП i8086 і i80286 і стільки же додаткових 32-розрядних регістрів.

У позначенні 32-розрядних регістрів використовується початкова літера **E** (Expanded - розширений).

Лічильник команд EIP містить зміщення наступної виконуваної команди в сегменті кодів. При 16-розрядних адресах використовуються молодші 16 біт (IP).

Регістр прапорців EF поширений до 32 біт. Молодші 16 біт регістру EF створюють регістр прапорців F 16-розрядного процесора. У регістр EF додані нові прапорці:

- ID (Identification Flag) - прапорець дозволу команди ідентифікації CPUID (Pentium+ і деякі процесори типу 486);
- VIP (Virtual Interrupt Pending) - віртуальний запит переривання (Pentium+);
- VIF (Virtual Interrupt Flag) - віртуальна версія прапорця дозволу переривання IF для багатозадачних систем (Pentium+);
- AC (Alignment Check) - прапорець контролю вирівнювання. Використовується тільки на рівні привілеїв 3. Якщо AC=1 і AM=1 (AM - біт у регістрі керування CR0), то у разі звернення до операнда, не вирівняному за відповідною межею (2, 4, 8 байт), відбудеться виняток 17 (i486+);
- VM (Virtual 8086 Mode) - у захищеному режимі вмикає режим віртуального процесора 8086. Спроба використання привілейованих команд в цьому випадку викличе виняток 13;
- RF (Resume Flag) - прапорець поновлення. У режимі відлагодження одиничне значення RF дозволяє здійснити рестарт команди після особливого випадку відлагодження.

Сегментні регістри. Окрім сегментних регістрів МП i8086 і i80286 (DS, CS, SS, ES), програмна модель містить два додаткових сегментних регістри даних: FS і GS. З кожним з шести сегментних регістрів пов'язані тіньові регістри дескрипторів. В тіньові регістри у захищеному режимі

переписуються 32-розрядна базова адреса сегмента, 20-розрядна межа і атрибути (права доступу) з дескрипторних таблиць.

Регістри керування CR0-CR3 (Control Register) зберігають ознаки стану процесора, спільні для всіх задач. Молодші 4 біти регістра CR0 містять біти регістра MSW МП i80286 і деякі інші біти керування. Регістр CR1 зарезервований; CR2 зберігає 32-бітову лінійну адресу, за якою була отримана відмова сторінки пам'яті; CR3 в старших 20-ти розрядах зберігає фізичну базову адресу таблиці каталогу сторінок і біти керування кеш-пам’яттю. Регістр CR4 (Pentium+) містить біти дозволів архітектурних розширень МП.

Системні адресні регістри. Системні вказівники - регістри глобальної дескрипторной таблиці GDTR та таблиці переривань IDTR - зберігають відповідно 32-розрядні базові адреси і 16-розрядні межі таблиць. Системні сегментні регістри задач TR і локальної дескрипторной таблиці LDTR є 16-розрядними селекторами. Їм відповідають тіньові регістри дескрипторів, які містять 32-розрядну базову адресу сегмента, 20-розрядну межу і права доступу.

Регістри налагодження DR0-DR3 (Debug Register) зберігають 32-бітові адреси точок зупину у режимі налагодження, DR4-DR5 зарезервовані і не використовуються; DR6 - відображує стан контрольної точки; DR7 - керує розміщенням у програмі контрольних точок.

Регістри тестування TR (Test Register) входять до групи модельно-специфічних регістрів, їхній склад і кількість залежить від типу процесора: в МП 386 використовувались два регістри: TR6 і TR7, у Pentium - 12 - TR1-TR12. Ця група регістрів зберігає результати тестування МП і кеш-пам’яті.

Сегментна організація пам'яті

У 32-розрядних МП розрізняють три адресних простори пам’яті – логічний, лінійний і фізичний. Логічна адреса (або віртуальна) складається з

селектора і зміщення ЕА. Лінійна адреса утворюється додаванням базової адреси сегмента з ефективною адресою. Фізична адреса пам'яті утворюється після перетворення лінійної адреси блоком сторінкової переадресації.

Організація пам'яті залежить від режиму роботи МП. У ***реальному*** і ***віртуальному*** режимах i8086 адресація пам'яті така ж, як у МП i8086. У ***захищеному*** режимі використовується сегментна і сторінкова організація пам'яті. Сегментна організація використовується на прикладному рівні, а сторінкова - на системному. Формування адреси комірки пам'яті у захищеному режимі подане на рис. 4.9.

Рис. 4.9. Формування адреси комірки пам’яті у захищеному режимі

Блок сегментації перетворює простір логічних адрес на простір лінійних адрес. Вихідними даними для блока сегментації є зміщення ЕА у сегменті і сегментний регістр, які задаються в команді. Вміст сегментного регістра у захищеному режимі є селектором. Він містить інформацію про тип дескрипторної таблиці (глобальної або локальної) та індекс дескриптора (див. рис. 4.1). Індекс дескриптора є номером дескриптора у таблиці.

Дескриптор містить базову адресу сегмента. Лінійна адреса утворюється шляхом додавання базової та ефективної адрес згідно рис. 4.2. Блок сторінкової переадресації утворює фізичну адресу пам'яті. При вимкненому блоці лінійна адреса збігається з фізичною. Блок обчислення ефективної адреси обчислює адресу-зміщення операнда у сегменті згідно з одним із наступних типів адресації, наведених в табл. 4.3. Алгоритм обчислення адреси комірки пам’яті для різних типів адресації наведено на рис. 4.10.

Регістри загального призначення МП можуть виконувати функції наступних регістрів: базового Base, індексного Index, масштабуючого множника Scale і зміщення Disp. У табл. 4.4 подано використання цих регістрів в залежності від режимів адресації: 16-бітової або 32-бітової. Масштабовані типи адресації можливі тільки в 32-бітовому режимі адресації.

Таблиця 4.3. Типи адресації у 32-розрядних процесорах

Тип адресації	**Обчислення ЕА**
Регістрова	ЕА= вмісту РЗП
Пряма	EA=Disp
Непряма регістрова	EA=Base
Базова	EA=Base+Disp
Індексна	EA=Index+Disp
Масштабована індексна	EA=Scale x Index+Disp
Базова індексна	EA=Base+ Index
Масштабована базова індексна	EA=Base+ Index x Scale
Базова індексна зі зміщенням	EA= Base+ Index +Disp
Масштабована базова індексна зі зміщенням	EA= Base+ Index x Scale+Disp

У реальному режимі за замовчуванням використовується 16-бітова адресація, але за допомогою префікса зміни розрядності адреси можна переключитися на 16-бітовий режим. У захищеному режимі тип адресації залежить від біта D у дескрипторі кодового сегмента (при D=0 використовується 16-бітова адресація, при D=1 - 32-бітова).

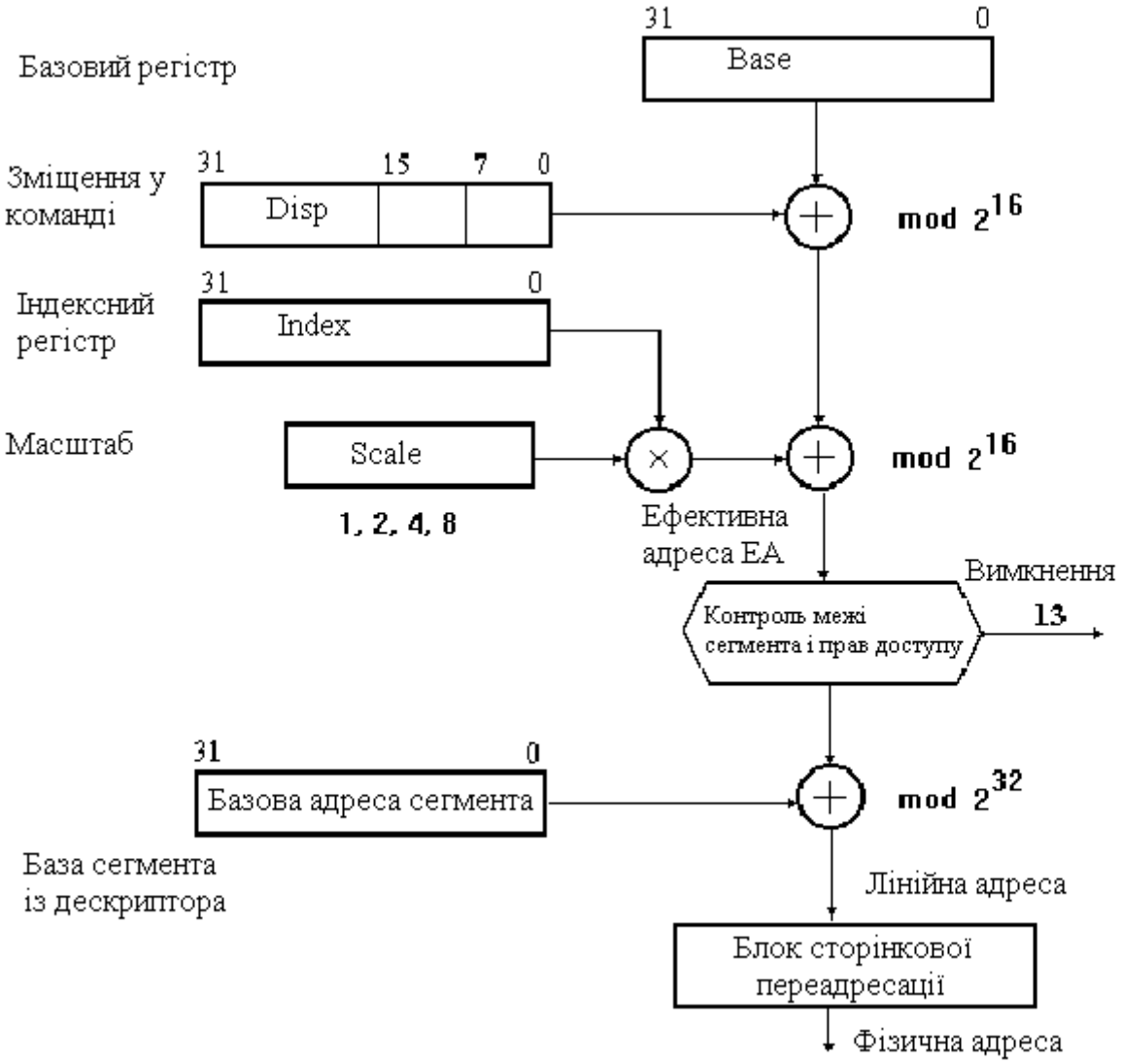


Рис. 4.10. Алгоритм обчислення адреси комірки пам’яті для різних типів адресації

Таблиця 4.4. Використання РЗП при обчисленні ефективної адреси

Компонент	**16-бітова адресація**	**32-бітова адресація**
Базовий регістр (Base)	BX або BP	Будь-який 32-бітовий РЗП
Індексний регістр (Index)	SI або DI	Будь-який 32-бітовий РЗП, крім ESP
Масштаб (Scale)	1	1; 2; 4 або 8
Зміщення (Disp)	0; 8 або 16 біт	0; 8 або 32 біт

Використання сегментних регістрів при адресації пам'яті визначається типом звернення до пам'яті і подане в табл. 4.5. Для деяких типів звернень допускається заміна сегментного регістра, що вводиться застосуванням

префіксів команд CS:, DS:, SS:, ES:, FS:, GS:, наприклад **ADD FS:[ESI],EAX** ; [FS:ESI]←[FS:ESI]+EAX.

Таблиця 4.5. Використання сегментних регістрів при адресації пам'яті

Тип звернення до пам'яті	Сегментний регістр		Зміщення
	за замовчанням	альтернативний	
Вибірка команд	CS	Немає	IP, EIP
Стекові операції	SS	Немає	SP, ESP
Адресація змінної	DS	CS, ES, SS, FS, GS	EA
Рядок-джерело	DS	CS, ES, SS, FS, GS	SI
Рядок-приймач	ES	Немає	DI
Використання BP, EBP або ESP як базового регістра	SS	CS, ES, DS, FS, GS	EA

Приклад 4.6. *Знайти значення фізичної адреси операнда в команді пересилання у регістр AL вмісту комірки пам'яті*

MOV AL, [BX+4*SI+1000H] ,

якщо базова адреса сегмента даних дорівнює 12456789H, а вміст регістрів BX=0120H, SI=1234H. Блок сторінкової переадресації відсутній.

Ефективну адресу комірки пам'яті знайдемо як

EA= 0120H+4*1234H+1000H=59F0H.

Виконавши операцію додавання 32-розрядної базової адреси з ефективною адресою EA, отримаємо лінійну адресу, яка збігається і з фізичною адресою:

12456789H+59F0H = 1245C179H.

Таким чином, фізична адреса дорівнює 1245C179H.

Формування базової адреси сегмента пояснюється рис. 4.11. Поле TI селектора сегмента визначає робочу дескрипторну таблицю (глобальну чи локальну), де знаходиться початкова адреса сегмента. Поле RPL визначає запрошений рівень привілею сегмента. Поле Index визначає зміщення від

початкової адреси таблиці. Зазначимо, що початкова адреса таблиці зберігається або у регістрі GDTR – для глобальної таблиці, або у тіньовому регістрі. В останньому випадку регістр LDTR є, в свою чергу, селектором і вказує, де в глобальній дескрипторній таблиці знаходиться інформація про початкову адресу локальної таблиці. Ця інформація переписується в тіньовий регістр.

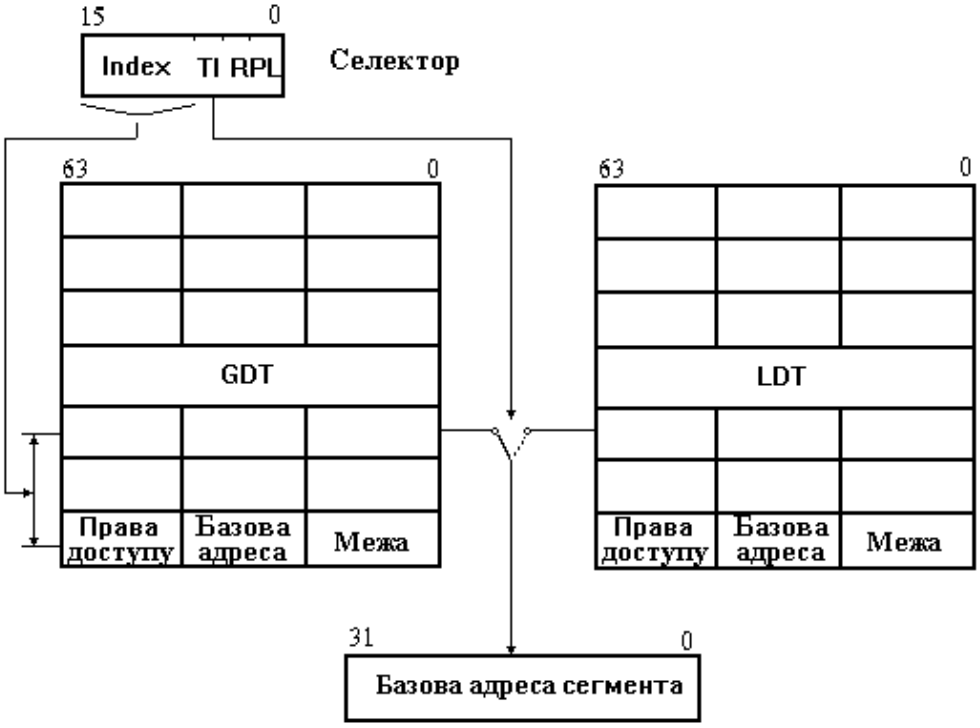


Рис. 4.11. Формування базової адреси сегмента

ЛЕКЦІЯ 16

Формат дескрипторів

Формат дескриптора для 32-розрядних процесорів наведений на рис. 4.12. Дескриптор МП i80286 містить 0 у бітах 63-48, а поля базової адреси і межі займають 24 і 16 біт відповідно. У 32-розрядному МП поле базової адреси займають другий, третій, четвертий та сьомий байти дескриптора. У ході виконання команди ці байти об'єднуються в одну 32-розрядну базову адресу.

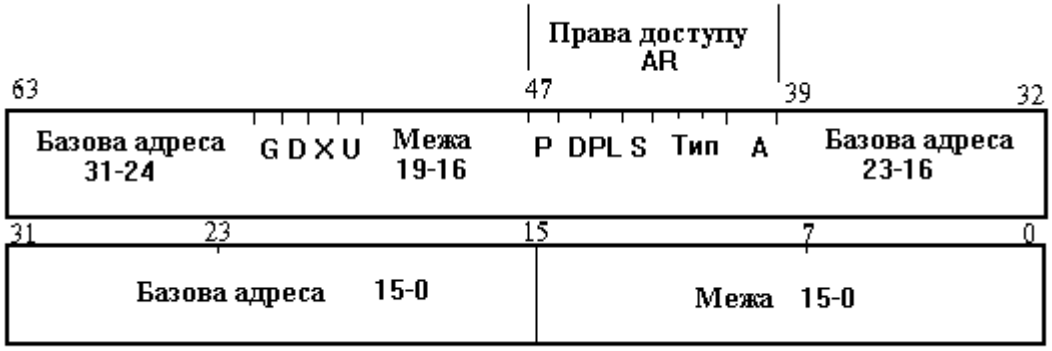


Рис. 4.12. Формат дескриптора для 32-розрядних процесорів

20-розрядне поле межі займає байти з номерами 0, 1 та молодші 4 біти 6-го байта дескриптора. Межа задає максимальне зміщення в сегменті або останню одиницю, що адресується в сегменті. При 20-розрядній межі максимальне значення елементів, що адресуються, 2^{20}. Однак оскільки елементом сегмента може бути не тільки байт, але і сторінка в 4 кбайт, сегмент може містити від одного байта до 4 Гбайт. Байт з номером 5 дескриптора AR (Access Rights) містить права доступу, зокрема, такі біти керування: P (Present) - біт присутності; DPL (Descripter Privilege Level) – поле рівня привілеїв сегмента; S (System) - системний біт; Type – поле типу сегмента; A (Accessed) - біт звернення.

Біт присутності P дорівнює 1, якщо сегмент знаходиться у фізичній пам'яті (ОЗП). У системі віртуальної пам'яті операційна система може передавати вміст деяких сегментів на диск, при цьому вона скидає біт P в нуль у дескрипторі цього сегмента. Якщо програма після цього знову звертається до сегмента, виникає особливий випадок відсутності сегмента. Операційна система шукає вільну область фізичної пам'яті (при цьому, можливо, відправляє на диск деякий інший сегмент), копіює вміст запрошеного сегмента з диска у пам'ять, записує в його дескриптор нову базову адресу, і здійснює рестарт команди, що викликала особливий випадок 11 відсутності сегмента. Описаний процес називається ***свопінгом*** (swapping), або ***довантаженням***;

Поле рівня привілеїв сегмента DPL містить 2 біт. Найвищому рівню привілею відповідає значення 0, найнижчому – значення 3.

Системний біт S має нульове значення (S=0) у дескрипторах сегмента кодів, системних сегментів для зберігання локальних таблиць дескрипторів, станів задач TSS (Task State Segment) і у дескрипторах, що називаються ***вентилями*** (Gate) або ***шлюзами***. В інших випадках S=1

Вентиль містить інформацію про логічну адресу входу до деякої системної програми і займає 8 байт. Формат вентилів наведено на рис. 4.13.

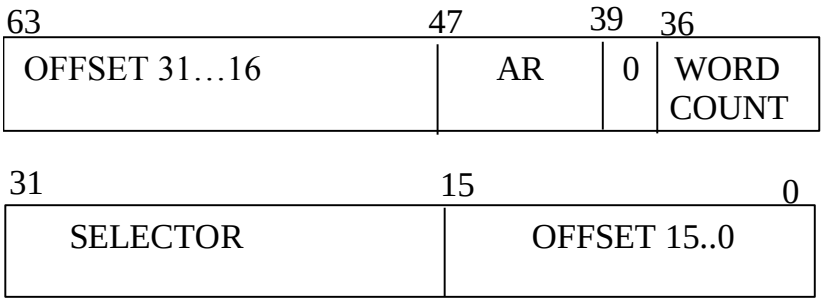


Рис. 4.13. Формат вентилів

Вентилі призначені для передачі керування і містять логічну адресу переходу у вигляді селектора SELECTOR і 32-розрядного зміщення OFFSET. ***Вентилі виклику*** використовуються для викликів процедур зі зміною рівня привілей; ***вентилі задач*** використовуються для переключення задач; ***вентилі переривань та вентилі пастки*** – для переходу до процедур обслуговування переривань, при цьому вентилі переривань забороняють переривання (скидають прапорець IF), а вентилі пастки – не забороняють.

Поле WORD COUNT (див. рис.4.13) у вентилях виклику визначає кількість слів, які копіються зі стеку однієї процедури у стек іншої процедури. Для інших вентилів поле WORD COUNT містить нульові значення.

Поле типу сегмента Type займає 3 розряди та визначає тип сегмента згідно з табл. 4.6.

Біт звернення A. У сегментах коду і даних A=0 означає, що до сегмента не було звернення. У системних об'єктах поле Type разом з бітом A визначає тип системного об'єкта згідно з табл. 4.6.

Таблиця 4.6. Типи сегментів і системних об'єктів

Type			A	Тип сегмента	Дозволені операції
Сегменти кодів і даних					
0	0	0	A	Даних	Тільки зчитування
0	0	1	A	Даних	Зчитування і запис
0	1	0	A	Стека	Тільки зчитування
0	1	1	A	Стека	Зчитування і запис
1	0	0	A	Коду	Тільки виконання
1	0	1	A	Коду	Виконання і зчитування
1	1	0	A	Підлеглий сегмент коду**	Тільки виконання
1	1	1	A	Підлеглий сегмент коду**	Виконання і зчитування
Системні сегменти***					
0	0	0	1	Доступний сегмент TSS стану задачі* i80286	
0	0	1	0	Таблиця локальних дескрипторів LDT	
0	0	1	1	Зайнятий сегмент TSS стану задачі i80286	
1	0	0	1	Доступний сегмент TSS стану задачі i386+	
1	0	1	0	Зарезервовано	
1	0	1	1	Зайнятий сегмент TSS стану задачі* i386+	
Вентилі***					
0	1	0	0	Вентиль виклику i80286 (Call Gate)	
0	1	0	1	Вентиль задачі i80286 (Task Gate)	
0	1	1	0	Вентиль переривання i80286 (Interrupt Gate)	
0	1	1	1	Вентиль пастки i80286 (Trap Gate)	
1	1	0	0	Вентиль виклику i386+ (Call Gate)	
1	1	0	1	Вентиль задачі i386+ (Task Gate)	
1	1	1	0	Вентиль переривання i386+ (Interrupt Gate)	
1	1	1	1	Вентиль пастки i386+ (Trap Gate)	

Примітки:
* На практиці такі сегменти стека не використовуються;
** підлеглі сегменти кодів див. у розділі 4.2.4.
*** інші стани полів Type та A не використовуються.

У старшій тетраді 6-го байта дескриптора знаходяться наступні біти керування:

- G (Granularity) - *біт гранулярності*. При G=0 одиницею пам'яті в сегменті є байт, при G=1 - сторінка довжиною 4 кбайт;

- D (Default size) - *біт розміру*. При D=0 операнди у пам'яті вважаються 16-розрядними, D=1 - 32-розрядними. Застосовується для сумісності з МП i80286;
- U (User) - *біт користувача*. Може бути встановлений або скинутий програмно.

Як видно з опису дескриптора, 32-розрядний МП дозволяє створення сегментів, в яких можуть виконуватися операції зчитування, зчитування/запису, виконання або виконання/зчитування. Для того, щоб створити характерні для МП i8086 сегменти, в яких виконуються одночасно всі перераховані операції, використовують перекриття сегментів пам'яті, тобто початкова адреса одного сегмента є адресою іншого.

Сторінкова організація пам'яті

Цей тип організації застосовується, як правило, у системах віртуальної пам'яті, що дозволяє програмісту використати більший простір адрес, ніж існуюча фізична пам'ять. Враховуючи властивість просторової локальності кодів і даних (близького розташування необхідних комірок пам'яті), доцільно оперувати не байтами, а деякими невеликими модулями пам'яті - сторінками. При сторінковому перетворенні весь лінійний адресний простір 32-розрядного МП ємністю 4Г байт розбивається на 2^{20} сторінок по 4К байт. Фізичний простір пам'яті мікропроцесорної системи також розбивається на сторінки, причому в фізичній пам'яті сторінок значно менше 2^{20}. Наприклад, при обсязі пам'яті 4М байт кількість фізичних сторінок (їх ще називають сторінковими кадрами або page frame) дорівнює 2^{10}. Відсутні у ВІС фізичної пам'яті сторінки зберігаються у зовнішній пам'яті (накопичувачі на твердому магнітному диску) і при необхідності завантажуються в фізичну пам’ять, тобто відбувається процес свопінгу. Прикладні програми можуть розпоряджатися усім простором віртуальної пам'яті - 4Г байт. Процес сторінкового перетворення адреси пояснюється рис. 4.14.

У процесі перетворення старші 20 біт 32-розрядної лінійної адреси замінюються іншим 20-розрядним значенням – номером фізичної сторінки згідно з механізмом перетворення адреси (рис. 4.14). Регістр керування CR3 PBDR (Page Directory Base Register) (див. розд. 4.2.2) містить фізичну базову адресу каталогу сторінок. ***Каталог сторінок*** знаходиться у фізичній пам'яті постійно і не бере участь у свопінгу. Він містить 1024 32-розрядних адреси PDE - (Page Directory Entry). Кожна з них є початковою адресою таблиць сторінок. ***Таблиця сторінок PTE*** (Page Table Entry) містить адреси сторінкових кадрів у фізичній пам'яті.

Фізична базова адреса каталогу сторінок формується із значення рядку таблиці сторінок PTE і 12 розрядів зміщення лінійної адреси.

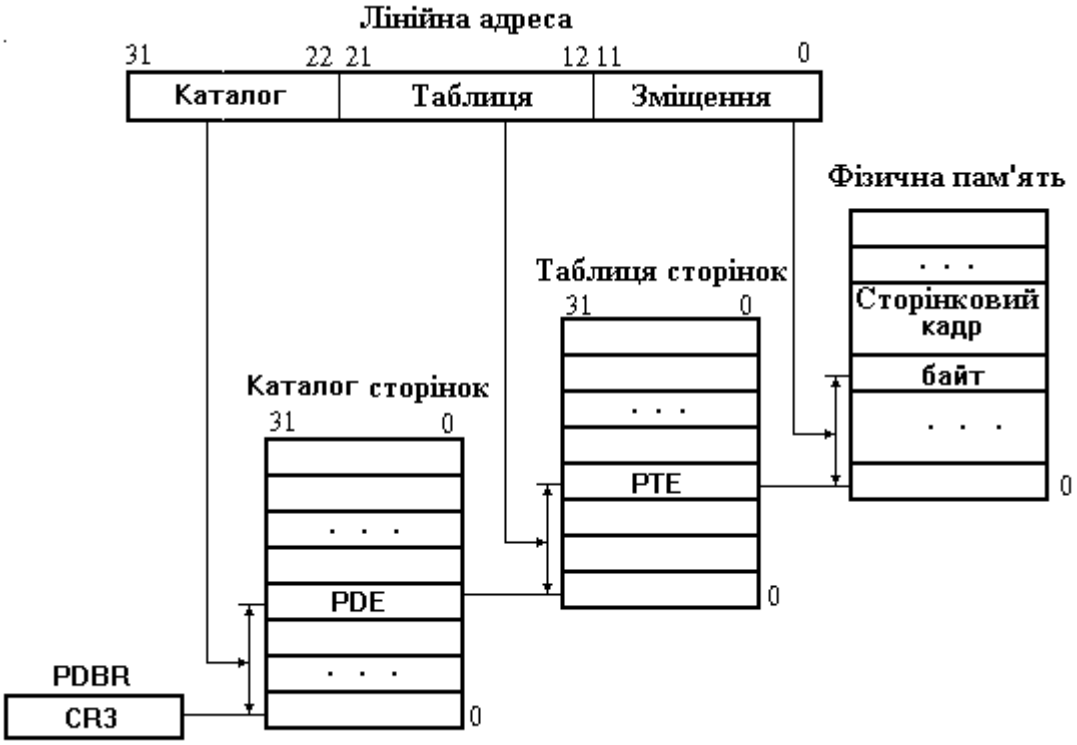


Рис. 4.14. Сторінкове перетворення адреси

Захист по привілеях

Система привілеїв призначена для запобігання недозволеним взаємодіям користувачів, несанкціонованому доступу до даних,

пошкодженню програм і даних. Частково ці задачі вирішуються організацією захищеного режиму пам'яті, частково - захистом по привілеях. 32-розрядні процесори підтримують 4 рівні привілеїв 0-3, причому рівень 0 є найбільш привілейованим. Рівень 0 звичайно присвоюється ядру операційної системи, рівень 1 - системним сервісам, рівень 2 - розширенням операційної системи і рівень 3 - прикладним програмам користувача. При виконанні програми контролюється, чи може програма:

- виконувати привілейовані команди;
- звертатися до даних інших програм;
- передавати керування іншій програмі командами передачі керування типу FAR.

До привілейованих команд належать команди, що змінюють сегментацію, впливають на механізм захисту, модифікують прапорець дозволу переривань IF. При спробі виконати ці команди на рівнях привілеїв 1, 2 або 3 генерується виняток 13.

Для контролю звернення програми до даних інших програм використовуються поля CPL (Current Privilege Level або Code Privilege Level - поточний рівень привілеїв; задається полем RPL селектора CS) і дескриптора даних DPL. Доступ до даних дозволяється при CPL≤ DPL.

Передача керування програм різноманітних рівнів привілеїв здійснюється за допомогою використання:

- підлеглих сегментів коду;
- дескрипторів вентилів викликів (шлюзів).

У підлеглих сегментах виконання команд можливе, якщо поточний рівень привілеїв (CPL) не нижче рівня привілеїв дескриптора (DPL) підлеглого сегмента, у непідлеглих – керування сегменту передається при CPL = DPL. Звичайно в підлеглих сегментах кодів розміщують бібліотеки, до яких можуть звертатися програми різноманітних рівнів привілеїв. Використання підлеглих кодових сегментів не змінює поточний рівень

привілеїв. Єдиним засобом зміни рівня привілею є використання вентилів викликів. Вентилі ідентифікують дозволені точки входу у кодові сегменти з більшим рівнем привілею. У дескрипторі вентиля задається повна адреса точки входу (селектор: зміщення) тієї процедури, якій передається керування.

Переключення задач

У багатозадачних системах і системах з великою кількістю користувачів МП виконує деяку частину команд однієї задачі (програм), після цього переключається на виконання іншої задачі і так далі, доки знову повертається до першої задачі. Для підтримки багатозадачного режиму в МП є наступні засоби:

- сегмент стану задачі TSS;
- дескриптор сегмента стану задачі;
- регістр задачі TR;
- вентиль задачі.

Дескриптор сегмента стану задачі вказує на сегмент, що містить повний стан задачі, а вентиль задачі містить селектор, що вказує на дескриптор TSS. Регістр TR є селектором сегмента TSS поточної задачі. Кожна задача має свій сегмент стану. У сегменті TSS міститься інформація про стан процесора на час переключення задач – вміст майже всіх регістрів МП, включаючи регістр прапорців, роздільні вказівники стеків для рівнів привілеїв 0, 1, 2 і посилання на селектор TSS задачі, що викликала дану задачу.

Переключення задач здійснюється або за командами міжсегментних переходів JMP FAR чи викликів підпрограм CALL FAR, або за апаратними чи програмними перериваннями і винятками. У першому випадку програма повинна посилатися на сегмент стану задачі TSS або на дескриптори вентиля задачі в GDT(LDT). В другому випадку відповідний перериванню дескриптор в таблиці переривань IDT має бути дескриптором вентиля задачі.

При передачі керування викликаній задачі за командою IRET перевіряється прапорець вкладеної задачі NT (Nested Task). При NT=0

команда IRET працює у звичайному режимі, залишаючись у поточній задачі. При NT=1 команда IRET виконує переключення на попередню задачу.

ЛЕКЦІЯ 18

Особливості архітектури мікропроцесорів Pentium

Мікропроцесор Pentium являє собою високопродуктивний 32-розрядний процесор з внутрішньою 64-розрядною шиною даних. Процесор є продовженням розробок процесорів i80x86 і програмно-сумісний з ними, але має ряд особливостей. У МП Pentium вперше застосована 0,8 мкм BiCMOS-технологія. BiCMOS-технологія об'єднує переваги двох технологій - швидкодію біполярної і мале енергоспоживання CMOS. Використання субмікронної технології дозволило збільшити кількість транзисторів до 3,1 млн. Для порівняння: процесор 8086 містить 29 тис. транзисторів, а найближчий до Pentium процесор i486 – 1,2 млн. транзисторів. Збільшення кількості транзисторів більше, ніж у два рази, дозволило розмістити у одній мікросхемі компоненти, що раніше розташовувались в інших мікросхемах. Це зменшило час доступу і збільшило продуктивність процесора. Висока тактова частота, суперскалярна архітектура, розділена кеш-пам'ять для програм і даних та інші вдосконалення дозволили досягти більшої продуктивності та сумісності з програмним забезпеченням, що було розроблено для мікропроцесорів фірми Intel. МП Pentium дозволяє використовувати такі операційні системи, як UNIX, Window-NT, OS/2, Solaris і NEXTstep. Особливості архітектури наступні.

Структурна схема та характеристики. Узагальнена структурна схема мікропроцесора Pentium (рис. 4.15) містить:

- ШІ - 64-бітовий шинний інтерфейс;
- два 32-розрядних цілочислових АЛП;
- кеш-пам'ять команд;

- кеш-пам'ять даних;
- РЗП;
- буфери вибірки з випередженням (БВВ);
- блок передбачення адреси переходу (БПАП);
- блок конвеєрних обчислень з плаваючою комою (БКОПК).

Шинний інтерфейс призначений для сопряження внутрішньої шини процесора із зовнішньою шиною.

Розширена 64-разрядна шина даних. Завдяки цьому МП Pentium підтримує декілька типів циклів шини, включаючи і пакетний режим, при якому частина даних з 256 біт передається у кеш-пам'ять даних за один цикл. Це істотно підвищує швидкість передачі порівняно з процесором i486 DX. Наприклад, при частоті шини 66 МГц Pentium має швидкість передачі 528М байт/сек, i486 DX при частоті шини 50 МГц має швидкість передачі 160М байт/сек.

Розширена шина даних забезпечує конвеєризацію циклів шини, що веде до збільшення пропускної здатності шини та дозволяє другому циклу починатися раніше, ніж завершився перший.

Суперскалярна архітектура. Термін “суперскалярна” означає мікропроцесорну архітектуру, що містить більше, ніж один обчислювальний блок. Процесор Pentium має два конвеєри, що можуть виконувати дві команди одночасно - U-конвеєр з повним набором і V-конвеєр з обмеженим набором команд. На рис. 4.15 конвеєри спрощено подані двома цілочисловими *АЛП*, *РЗП* і *БВВ*. Як і у випадку єдиного конвеєра процесора i486, подвійний конвеєр процесора Pentium виконує цілочислові команди у 5 етапів (рис. 4.16):

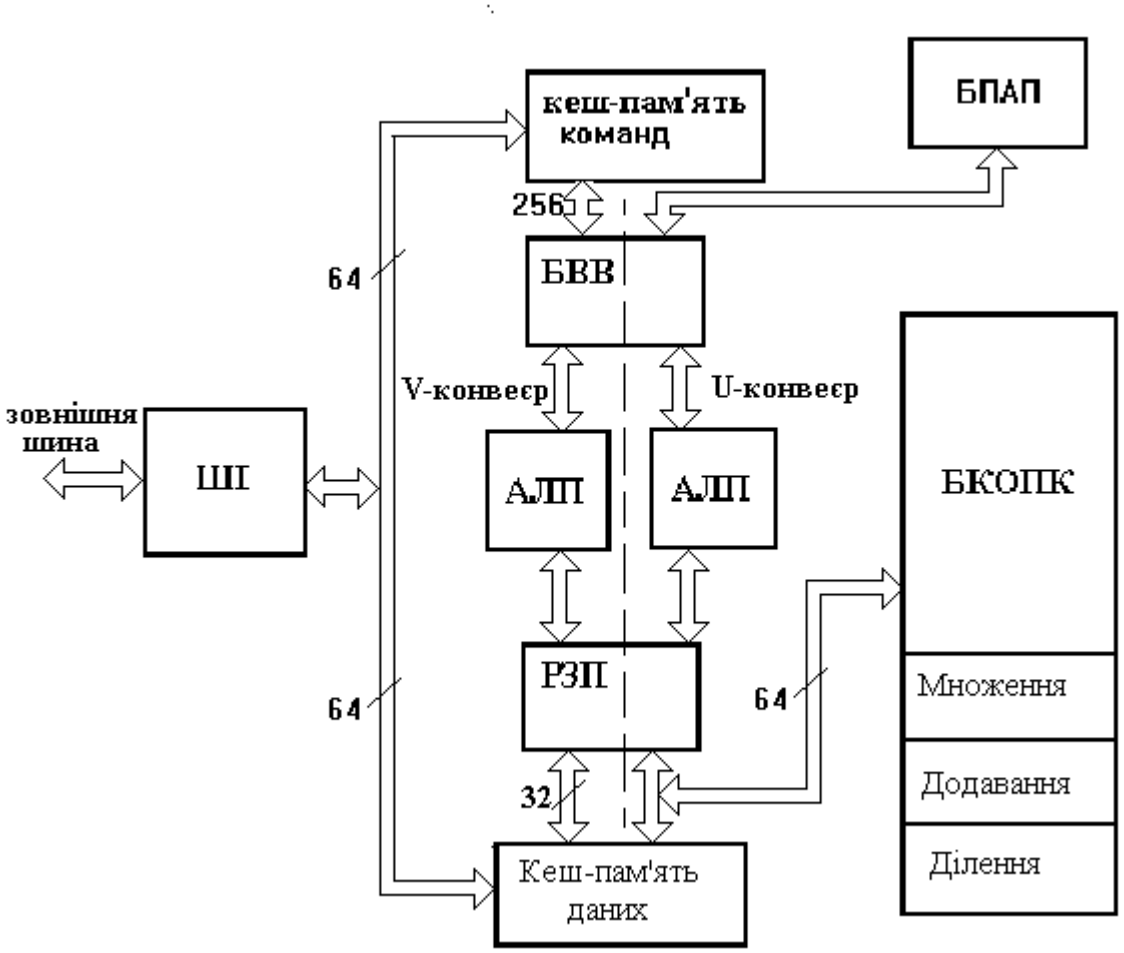


Рис. 4.16. Узагальнена структурна схема мікропроцесора Pentium

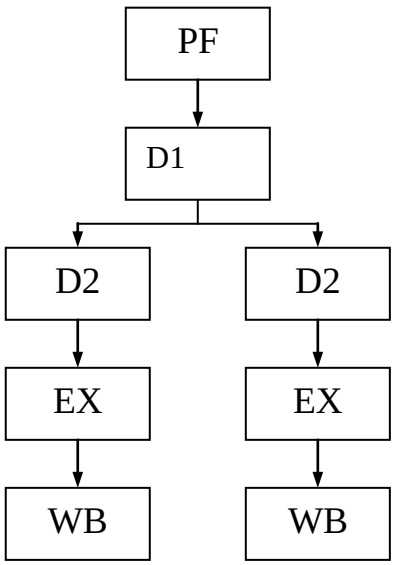


Рис. 4.17. Етапи виконання цілочислової команди у процесорі Pentium

- вибірка з випередженням команди з пам'яті (передвибірка) PF (PreFetch);
- декодування команди (стадія 1) D1;
- декодування команди (стадія 2) D2;
- виконання команди EX;
- запам’ятовування результату у буфері відкладеного запису WB

Перший етап виконується блоком *БВВ*, який має чотири 32-розрядних буфери. Дві незалежні пари буферів вибірки працюють сумісно з *БПАП*, який передбачає, буде перехід чи ні. Якщо перехід не передбачається, продовжується вибірка. Якщо передбачається перехід, то дозволяється робота іншого буфера, і він починає передвибірку з точки переходу. Якщо передбачений перехід не здійснився, конвеєри команд очищуються, і передвибірка починається знову. На другій стадії декодування команди відбувається формування адрес операндів пам’яті.

Кожний конвеєр має свій 64-розрядний буфер відкладеного запису, які можуть заповнюватися за 1 такт, наприклад, при одночасних кеш-промахах запису на обох конвеєрах. Ніякі запити на читання не порушують порядку запитів на запис, які вже знаходяться у буфері. Pentium підтримує строгий порядок запису.

Високопродуктивний математичний співпроцесор БКОПК містить 8-тактовий конвеєр і апаратні засоби реалізації арифметичних операцій – множення, додавання, ділення. Більша частина команд операцій з плаваючою комою можуть виконуватись в одному цілочисловому конвеєрі, після чого вони надходять у конвеєр обчислень з плаваючою комою. Продуктивність вбудованого арифметичного співпроцесора Pentium переважає продуктивність математичного співпроцесора FPU-486 (Floating-Point Unit) у 2-10 разів.

Використання подвійного конвеєра дозволяє декільком командам знаходитися в різних стадіях виконання і додатково збільшити продуктивність МП шляхом повного заповнення конвеєрів командами. У процесорі Pentium використовується апаратне виконання команд, що також підвищує продуктивність процесора.

Розділені кеш-пам'яті команд і даних. МП Pentium має розділені кеш-пам'яті команд і даних. Це дозволяє уникнути конфліктів між процесом вибірки для однієї команди і доступом до даних для іншої, які можуть виникати, наприклад, у процесорі i486. При реалізації розділеного кешування для команд і даних обидві команди можуть виконуватися одночасно. Ємність кеш-пам'яті команд та кеш-пам'яті даних у процесорі Pentium однакова і складає 8К байт. Кеш-пам'ять команд і даних виконана за схемою двоканальної асоціативної кеш-пам'яті (див. розд. 5.5). Кеш-пам'ять даних має два інтерфейси, по одному для кожного з конвеєрів, що дозволяє забезпечувати даними дві окремі команди протягом одного машинного циклу.

Кеш-пам'ять даних працює з відкладеним (до звільнення зовнішньої шини) записом і налагоджується у режим наскрізного або зворотного запису. В останньому випадку дані зчитуються з кеш-пам'яті, а після цього записуються в основну пам'ять. Такий спосіб кешування дозволяє збільшити продуктивність порівняно з простим кешуванням з безпосереднім записом, при якому процесор записує дані одночасно в кеш-пам'ять і основну пам'ять. Кеш-пам'ять даних підтримує протокол MESI. Цей протокол забезпечує роботу з урахуванням можливості звертання до кеш-пам'яті даних процесора, що розглядається, з боку інших процесорів. Назва протоколу MESI складається з назв станів рядка кеш-пам'яті: M (Modified), E (Exclusive), S (Shared), I (Invalid). Стани рядка кеш-пам'яті визначаються таким чином:

M-стан - рядок наявний тільки у кеш-пам'яті процесора, що розглядається. Рядок є модифікованим, тобто відрізняється від вмісту основної пам'яті. Запис у нього можливий без генерації зовнішнього (по відношенню до локальної шини) циклу звернення.

E-стан - рядок наявний тільки у кеш-пам'яті процесора, що розглядається, але він є немодифікованим. Запис у рядок можливий без генерації зовнішнього циклу звернення. При запису у рядок він перейде у ***M***-стан;

S-стан - рядок наявний у кеш-пам’яті процесора, що розглядається, та потенційно може бути наявним у кеш-пам'яті інших процесорів. Його читання можливе без генерації зовнішнього циклу, а запис повинен супроводжуватися наскрізним записом у основну пам'ять, що спричинить анулювання відповідних рядків у кеш-пам'яті інших процесорів;

I-стан - рядок відсутній у кеш-пам'яті, його читання з основної пам’яті може привести до генерації циклу заповнення рядка кеш-пам’яті. Запис у рядок кеш-пам’яті буде наскрізний з використанням зовнішньої шини.

Підтримка мультипроцесорного режиму роботи. Архітектура Pentium дозволяє працювати двом і більше Pentium-процесорам у мультипроцесорних системах. Реалізовано інтерфейс побудови двопроцесорних систем із симетричною архітектурою (починаючи з другого покоління Pentium).

Засоби завдання розміру сторінки пам'яті. Pentium процесор має опцію (спеціальний біт керування) для вибору розміру сторінок пам'яті - традиційну (4К байт) і розширену (4М байт). Збільшення розміру сторінки доцільне при використанні громіздких графічних додатків.

Засоби виявлення помилок і тестування за допомогою функціональної надмірності. З метою підвищення надійності у процесорі Pentium передбачено внутрішнє виявлення помилок внутрішніх пристроїв (внутрішній контроль паритета) та зовнішнього шинного інтерфейса,

контроль паритета шини адреси та тестування за допомогою функціональної надмірності. ***Внутрішнє визначення помилок*** полягає у доповненні кодів команд і даних бітом парності, що дозволяє визначати помилки таким чином, що це залишається непомітним як для системи, так і для користувача. ***Тестування за допомогою функціональної надмірності*** використовується у програмних додатках, особливо критичних до достовірності результатів. Тестування за допомогою функціональної надмірності базується на роботі двох Pentium-процесорів у конфігурації ***основний/контролюючий (master/checker)***. У такій конфігурації основний процесор працює у звичайному однопроцесорному режимі. Контролюючий процесор виконує ті ж самі операції, але не керує шиною, і порівнює вихідні сигнали основного процесора з тими сигналами, які він генерує сам. У випадку розбіжності отриманих результатів виробляється сигнал помилки, який може оброблятися системою як переривання. Такий спосіб дозволяє викривати більше 99% помилок. Крім того, засоби тестування передбачають можливість виконання вбудованого теста BIST (Built In Self Test), що забезпечує виявлення помилок мнемокодів, програмовних логічних матриць, тестування кеш-пам'яті команд і даних, адресних буферів та ПЗП. В цілому самотестування охоплює більш ніж 70% вузлів процесора. Всі процесори мають стандартний тестовий порт IEEE 1149.1 для самотестування за допомогою стандартного інтерфейсу JTAG.

До особливостей процесорів Pentium відносяться:

- введення декількох нових команд, у тому числі розпізнавання моделі процесора;
- введення засобів керування енергоспоживанням;
- застосування конвеєрної адресації шинних циклів;
- скорочений час (кількість тактів) виконання команд;
- трасування команд і моніторинг продуктивності;

- розширення можливостей віртуального режиму - впровадження віртуалізації прапорця переривань;

Реалізовано нові додаткові засоби відлагодження:

- зондовий режим (Probe Mode), що забезпечує доступ до внутрішніх регістрів, пристроїв введення/виведення і системної пам'яті процесора. Цей режим дозволяє перевіряти і змінювати стан процесора за допомогою засобів відлагодження програм з можливостями, подібними до можливостей внутрішньосхемних емуляторів;
- розширення відлагодження DE (Debug Extensions), які дозволяють ставити контрольні точки за адресами введення/виведення;
- внутрішні лічильники, які використовуються для поточного контролю продуктивності та обліку кількості подій;
- покрокове виконання за допомогою команди CPUID.

Розширення архітектури. Додатково до базової архітектури 32-розрядних процесорів (див. розд. 4.2) Pentium має набір регістрів, специфічних для моделі MSR (Model Specific Registers). Склад регістрів MSR може бути різним у різних моделях МП (Pentium і Pentium Pro), що призводить до їхньої можливої несумісності. Програмне забезпечення, що використовує регістри MSR, повинне спиратися на відомості про процесор, отримані за допомогою команди CPUID.

До складу регістрів MSR входять:

- тестові регістри TR1-TR12 (див. розд. 4.2.1);
- засоби моніторингу продуктивності;
- регістри-фіксатори адреси і даних циклу, що викликав спрацювання контролю машинної помилки.

Тестові регістри дозволяють керувати більшістю функціональних вузлів процесора, забезпечуючи можливість докладного тестування їхньої працездатності. За допомогою бітів регістра TR12 можна заборонити нові архітектурні властивості (передбачення і трасування розгалужень, паралельне виконання команд), а також роботу кеш-пам'яті.

Засоби моніторингу продуктивності дозволяють оптимізувати апаратне та програмне забезпечення завдяки виявленню у програмному коді потенційно “вузьких місць”. Розробник може спостерігати та підраховувати такти внутрішніх процесорних подій, які впливають на продуктивність операцій читання і запису, вдалі та невдалі звернення до кеш-пам’яті, переривання, використання шини. Це дозволяє оцінювати ефективність програмного кода і здійснювати детальне налагодження програмних додатків або систем для отримання максимальної продуктивності. Засоби моніторингу продуктивності містять таймер реального часу і лічильники подій. Таймер TSC (Time Stamp Counter) виконаний на базі 64-розрядного лічильника, вміст якого інкрементується з кожним тактом роботи ядра процесора. Для читання його вмісту призначена команда RDTSC. 40-разрядні лічильники подій CTR0, CTR1 програмуються на підрахування подій різних класів, пов'язаних із шинними операціями, виконанням команд, роботою конвеєрів, кеш-памяті, контролем точок зупину, тощо. 6-бітові поля типів подій дозволяють кожному з лічильників незалежно підраховувати події із великого списку. Стан лічильників може бути попередньо встановлений і зчитаний програмно. Крім того, існують зовнішні лінії PM[1:0], які програмуються на вказання фактів спрацьовування або переповнення відповідних лічильників. Оскільки ці сигнали можуть змінювати своє значення з частотою, що не перевищує частоту системної шини, через внутрішнє множення частоти кожна поява цих сигналів може означати і декілька (до значення коефіцієнта множення) фактів спрацювання лічильників.

Назва *регістрів-фіксаторів адреси і даних циклу, що викликав спрацювання контролю машинної помилки*, вказує на їх можливу несумісність для різних класів (Pentium та Pentium Pro) або навіть для різних моделей процесорів. Програма, що їх використовує, повинна спиратися на відомості про процесор, отримані за командою CPUID.

Процесори Pentium мають можливість зниження енергоспоживання у неробочому режимі. За сигналом STOPCLK# процесор вивантажує буфери відкладеного запису і входить у режим Stop Grant, у якому припиняється тактування більшості вузлів процесора, що спричинює зниження споживання приблизно в 10 разів. У цьому стані МП припиняє виконання команд і не обслуговує переривання, однак продовжує спостереження за шиною даних. З цього стану процесор виходить після зняття сигналу STOPCLK#. Керування сигналом STOPCLK# разом із використанням режиму SMM реалізує механізм *розширеного керування живленням APM (Advanced Power Management)*. Для уповільнення процесора із пропорційним зниженням споживаної потужності сигнал STOPCLK# має бути періодичним імпульсним. Скважність імпульсів визначає коефіцієнт простою процесора і його продуктивність.

У стан зниженого споживання Auto HALT PowerDown процесор переходить при виконанні команди HALT. У цьому стані процесор реагує на всі переривання і також продовжує спостереження за шиною.

У режимі припинення зовнішньої синхронізації процесор споживає мінімальну потужність, але не виконує ніяких функцій. Наступна подача сигналу синхронізації повинна супроводжуватися сигналом апаратного скидання RESET.

Контрольні питання

1. Назвіть складові частини мікропроцесорного комплекту.
2. За якими класифікаційними ознаками поділяються мікропроцесори і мікропроцесорні комплекти?
3. На які задачі орієнтовані спеціалізовані мікропроцесори?
4. Які переваги і недоліки мають секційні мікропроцесори порівняно з однокристальними?
5. Що означає біт у двійковій системі?
6. Перетворіть на десятковий код наступні двійкові числа:

а) 0001; б) 0101; в) 1000; г) 1011; д) 1111; е) 0111.

7. Перетворіть на двійковий код наступні десяткові числа з точністю 2^{-6}:

а) 23,55; б) 39,41.

8. Виконайте додавання чисел 10110011_2 та 11110010_2 з перевіркою результату у десятковій системі.
9. Виконайте віднімання чисел 132_{10} та 77_{10} у двійковому вигляді.
10. Виконайте множення чисел 110110_2 та 111_2 з перевіркою результату у десятковій системі.
11. Вкажіть розрядність результату множення чисел 11011100_2 та 11101111_2.
12. Перетворіть на шістнадцятковий еквівалент числа: а) 1001_2; б) 1100_2; в) 01111110_2; г) 11011011_2.
13. Перетворіть на двійково-десятковий еквівалент наступні десяткові числа: а) 39_{10}, б) 65_{10}; в) 40_{10}, г) 17_{10}.
14. Перетворіть на десятковий еквівалент наступні двійково-десяткові числа: а) $1000\ 0000_{2\text{-}10}$; б) $0000\ 0001_{2\text{-}10}$; в) $1001\ 0010_{2\text{-}10}$; г) $0111\ 0110_{2\text{-}10}$.
15. Назвіть призначення і складові частини системної шини.
16. Вкажіть принципи передачі інформації по шинах: 1) адреси; 2) даних; 3) керування.

17. Назвіть та дайте характеристику принципам побудови мікропроцесорних систем.
18. Наведіть типову структуру мікропроцесорної системи та поясніть призначення функціональних модулів.
19. Поясніть призначення входу керування третім станом
20. Вкажіть характерні особливості мінімального та максимального режимів роботи МП 8086.
21. Вкажіть існуючі формати даних у МП і8086.
22. Наведіть приклади упакованого і розпакованого двійково-десяткових чисел.
23. Яким чином в МП подаються від'ємні числа?
24. Поясніть принцип конвеєрної архітектури.
25. Вкажіть функції операційного пристрою та шинного інтерфейсу.
26. Яким чином визначаються тип, початок і кінець циклу шини за допомогою ліній стану?
27. Які блоки МП беруть участь у формуванні 20-розрядної фізичної адреси?
28. Обчисліть 20-розрядну фізична адресу DS:SI, якщо DS=1234H, SI=5678H.
29. Підберіть дві пари 16-розрядних логічних адрес, які є еквівалентними фізичній адресі 12008H.
30. Які групи регістрів входять до програмної моделі МП?
31. Які сегментні регістри за замовчуванням адресують початок сегментів кодів, стека, даних?
32. Вкажіть призначення регістра прапорців.
33. Наведіть приклад виконання команди, при якій встановлюється прапорець знака.
34. Наведіть приклад виконання команди, при якій встановлюється прапорець паритету.
35. Наведіть приклад виконання команди, при якій встановлюється прапорець нульового результату.

36. Наведіть приклади команд введення/виведення з прямою адресацією 8-розрядного, 16-розрядного портів.
37. Які існують типи адресації операндів у пам’яті?
38. Яким чином обчислюється ефективна адреса операнда при різних типах адресації?
39. Назвіть та охарактеризуйте існуючі типи циклів шини.
40. Дайте визначення вектора переривань і карти векторів переривань.
41. Які дії виконує МП при переході на підпрограму обробки переривань?
42. Назвіть та охарактеризуйте існуючі типи переривань у МП і8086
43. Для чого призначений ПЗП?
44. Дайте визначення комірки пам’яті.
45. Назвіть типи ПЗП.
46. Поясніть поняття банку пам’яті.
47. Що являє собою елемент пам’яті статичного ОЗП?
48. Які особливості має побудова модулів оперативної пам'яті для мікропроцесорних систем на базі 16-розрядних процесорів?
49. Які сигнали використовуються для вибору банків пам’яті ОЗП?
50. Назвіть чотири можливих випадки звернення до пам'яті в 16-розрядних процесорах?
51. Що таке маршрутизація байта?
52. Які рекомендації можна дати щодо розташування даних у стеку?
53. Яким чином забезпечується зберігання інформації, що надходить від ПВВ?
54. Назвіть способи адресування портів введення/виведення.
55. Наведіть порівняльну характеристику існуючих видів обміну.
56. Які існують типи програмного обміну?
57. Наведіть структурну схему обміну за стробом готовності.
58. Наведіть структурну схему обміну за перериванням.
59. Наведіть структурну схему обміну в режимі ПДП.

60. Яким чином відбувається запам'ятовування вмісту акумулятора, РЗП, програмного лічильника та прапорців при обміні за перериванням?
61. У яких випадках доцільно застосування прямого доступу до пам'яті?
62. Вкажіть призначення ВІС програмовного паралельного інтерфейсу КР580ВВ55.
63. Опишіть режими роботи програмовного паралельного інтерфейсу.
64. Назвіть можливі комбінації ввімкнення портів ВІС КР580ВВ55.
65. Які порти паралельного інтерфейсу можуть працювати у всіх можливих режимах?
66. Запишіть керуюче слово для роботи ППІ у режимі 0 при налагодженні портів А і В на виведення, порту С – на введення.
67. Поясніть принцип скидання/встановлення розрядів порту С.
68. Які функції програмовного таймера у мікропроцесорній системі?
69. З яких блоків складається таймер? Що входить до складу одного лічильника?
70. Опишіть режими роботи таймера.
71. Чим відрізняється дія сигналу GATE в режимах 0 та 1?
72. Які нові функції має ВІС КР1810ВИ54 порівняно з ВІС КР580ВИ53?
73. Назвіть можливі комбінації роботи каналів таймера.
74. Назвіть приклади використання таймера мікропроцесорної системи.
75. Запишіть керуючі слова для роботи таймерів у режимах 0, 2, 5.
76. Яким способом можна прочитати вміст внутрішніх регістрів таймера ?
77. У чому полягає відмінність між 4 і 5 режимами ?
78. У якому режимі таймер працює як подільник частоти ?
79. Яке максимальне число можна завантажити у внутрішні регістри таймера?
80. Для чого потрібні лінії А0, А1 таймера ?
81. Вкажіть призначення ВІС програмовного інтерфейсу клавіатури та індикації.

82. Назвіть максимальну кількість клавіш, яка може бути під’єднана до ВІС програмовного інтерфейсу клавіатури та індикації КР580ВВ79.
83. Поясніть особливості режиму опитування матриці клавіатури.
84. Поясніть особливості режиму опитування матриці датчиків.
85. Поясніть особливості режиму введення за стробом.
86. Вкажіть призначення регістрів, які входять до програмної моделі 32-розрядного процесора.
87. 2. Вкажіть призначення прапорців, які входять до програмній моделі 32-розрядного процесора.
88. Вкажіть призначення регістрів керування та регістрів тестування.
89. Вкажіть призначення та існуючі типи дескрипторних таблиць.
90. Як значення системного біта визначає тип дескриптора?
91. Дайте визначення процесу свопінгу і поясніть, як він відбувається.
92. Поясніть принцип сторінкової організації пам’яті.
93. Поясніть необхідність та принцип функціонування механізму захисту по привілеях.
94. За яких умов дозволяється зчитувати (записувати) дані певного сегмента?
95. Яка інформація міститься у вентилях викликів?
96. Поясніть особливості багатозадачного режиму роботи.
97. Вкажіть призначення регістра TR.
98. Які операційні системи можуть використовуватися у МП Pentium?
99. Назвіть призначення основних блоків структурної схеми процесора Pentium.
100. Поясніть роботу блока передбачення адреси переходу на прикладі виконання команди JC LABEL.
101. Який ефект має розділення кеш-пам’яті на кеш-пам'ять команд і кеш-пам'ять даних?
102. Які засоби виявлення помилок має процесор Pentium?

103. В чому полягає принцип тестування за допомогою функціональної надмірності?
104. Які функції тестування має процесор Pentium?
105. Вкажіть призначення та можливості засобів моніторингу продуктивності.
106. Які особливості регістрів-фіксаторів необхідно враховувати при розробці програмного забезпечення?
107. Вкажіть призначення та принцип режиму MMX.
108. Вкажіть призначення та принцип режиму SMM.
109. Як здійснюється переключення у режим зменшеного енергоспоживання і вихід з нього?

НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ

ОСНОВНА ЛІТЕРАТУРА

1. http://www.kaf-pe.ntu-kpi.kiev.ua/ Жуйков В.Я., Терещенко Т.О., Петергеря Ю.С. і ін «Мікропроцесори і мікроконтролери» - Електронний підручник
2. Мікропроцесорна техніка. Друге видання. Доповнене./ Ю.І. Якименко, Т.О. Терещенко, Є.І. Сокол, В.Я. Жуйков, Ю.С. Петергеря. За ред. Т.О. Терещенко. – Київ, 2004. – 440 с
3. Схемотехника электронных систем. Том 3. Микропроцессоры и микроконтроллеры / Бойко В.І., Гуржій А.М., Жуйков В.Я., Зорі А.А., Співак В.М., Терещенко Т.О, Петергеря Ю.С. - СПб.: БХВ Петербург, 2004. – 464 с.
4. Схемотехніка електронних систем. Том 3. Мікропроцесори та мікроконтролери / Бойко В.І., Гуржій А.М., Жуйков В.Я., Зорі А.А., Петергеря Ю.С., Співак В.М., Терещенко Т.О, Якименко Ю.І. - К.: Вища школа, 2004. – 399 с.:іл.
5. Мікропроцесорна техніка: Підручник / Ю.І. Якименко, Т.О. Терещенко, Є.І. Сокол, В.Я. Жуйков, Ю.С. Петергеря / За ред. Т.О. Терещенко. – К.: Видавництво “Політехнік”, 2002. – 439 с .
6. Микропроцессорный комплект К1810. Справочная книга. /Под ред. Казаринова Ю.М. - М., Высшая школа, 1990.
7. Питер Абель. Язык Асемблера для IBM PC и программирование. М., 1992.
8. Гук М. Процессоры Intel от 8086 до Pentium II. Санкт-Петербург,Питер Паблишинг, 1997.

ДОДАТКОВА ЛІТЕРАТУРА

1. Жуйков В.Я., Терещенко Т.А., Петергеря Ю.С. Методичні вказівки до вивчення курсу “Мікропроцесорна техніка”, розділ “Побудова модулів пам‘яті мікропроцесорних систем” для студентів спеціальності 7.090803 – “Електронні системи” всіх форм навчання – К.: НТУУ “КПІ”, 1999 – 32 с.
2. Терещенко Т.О., Петергеря Ю.С. Методичні вказівки до вивчення курсу “Мікропроцесорна техніка”, розділ “Архітектура одно кристальних мікропроцесорів” для студентів спеціальності 7.090803 – “Електронні системи” усіх форм навчання – К.: НТУУ “КПІ”, 1999 – 58 с.
3. Терещенко Т.О., Петергеря Ю.С. Методичні вказівки до виконання курсових робот з курсу “Мікропроцесорна техніка” для студентів спеціальності 7.090803 – “Електронні системи” усіх форм навчання – К.: НТУУ “КПІ”, 2000 – 62 с.
4. Співак В.М. Швайченко В.Б. Терещенко Т.О. Методичні рекомендації до самостійних робот та курсового проекту з дисциплін “Електромеханіка”, “Електропривод та теорія автоматичного регулювання для студентів напряму підготовки 6.0912 “Акустотехніка” та 6.0924 “Телекомунікації” всіх форм навчання – К.: НТУУ “КПІ”, 2000 – 78 с.
5. Мікропроцесорна техніка: Метод. вказівки до виконання лабораторних робіт для студ. спец. 7.090803 всіх форм навчання / Уклад.: Т.О.Терещенко, О.В.Хоменко, Л.М.Батрак. – К.ІВЦ “Видавництво Політехніка”, 2002. – 44с.
6. Терещенко Т.О., Москаленко Є.В., Петергеря Ю.С. Методичні вказівки до вивчення курсу “Мікропроцесорна техніка”, розділ “Сигнальні і медійні процесори” для студентів спеціальності 7.090803 – “Електронні системи” усіх форм навчання – К.: НТУУ “КПІ”, 2002 – 68 с.
7. Методичні вказівки до вивчення курсу “Мікропроцесорна техніка”: Розділ “Програмування мовою асемблера” для студ. спец. 7.090803 всіх форм навчання / Уклад.: Т.О.Терещенко, О.В.Хоменко, Л.М.Батрак. – К.ІВЦ “Видавництво Політехніка”, 2001. – 64с.

8. Терещенко Т.О., Петергеря Ю.С. Методичні вказівки до виконання курсових робот з курсу “Мікропроцесорні пристрої управління та обробки інформації” для студентів спеціальності 7.090803 – “Електронні системи” усіх форм навчання – К.: НТУУ “КПІ”, 2000 – 56 с.