

ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА ЗВ'ЯЗКУ

Кафедра транспортного зв'язку

МЕТОДИЧНІ ВКАЗІВКИ

**до практичних занять і контрольних робіт
з дисципліни**

«МІКРОПРОЦЕСОРНА ТЕХНІКА»

Харків – 2016

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри транспортного зв'язку 15 березня 2016 р., протокол № 10.

У методичних вказівках викладено основні питання, які необхідно вирішити при розв'язанні практичних задач, пов'язаних з принципом побудови внутрішньої пам'яті, функціонуванням і програмуванням мікропроцесора, а також з арифметичними основами мікропроцесорних систем.

Розроблено для студентів всіх форм навчання і напрямів підготовки факультету АТЗ та слухачів ІППК.

Укладачі:

доценти І.В. Ковтун,
Н.А. Корольова

Рецензент

доц. Л.А. Клименко

МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять і контрольних робіт
з дисципліни

«МІКРОПРОЦЕСОРНА ТЕХНІКА»

Відповідальний за випуск Корольова Н.А.

Редактор Третьякова К.А.

Підписано до друку 21.04.16 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 2,0. Тираж 50. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.

Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

ЗМІСТ

Вступ.....	4
1 Арифметичні основи мікропроцесорних систем.....	5
2 Побудова внутрішньої пам'яті мікропроцесорної системи, що складається з ПЗП і статичного ОЗП.....	21
3 Вивчення функціонування і програмування БІС КР580ВВ55.....	32
4 Програмування мікропроцесора КР580ВМ80.....	
Складання простих програм та викладення на мові Assembler.....	43
Список літератури.....	48
Додатки.....	49

ВСТУП

Мікропроцесорна техніка отримала широке застосування у системах управління технологічним і контрольно – випробувальним обладнанням, транспортними засобами, побутовим обладнанням. Невеликі розміри, маса та енергоємність мікропроцесорів (МП) дозволяють вбудовувати їх безпосередньо в об'єкт управління. На базі мікропроцесорів створюються будь-які типи мікроЕОМ, контролерів, програматорів та інших пристроїв автоматики й обчислювальної техніки.

Досягнення мікропроцесорної техніки широко використовують практично у всіх сферах діяльності. У зв'язку з цим знання мікропроцесорної техніки сучасними спеціалістами стають обов'язковою умовою успішної діяльності. Для того, щоб опрацювати знання необхідно, окрім засвоєння теорії, навчитися вирішувати конкретні завдання, які пов'язані з вибором тієї чи іншої схеми мікропроцесорної системи, зрозуміти фізичну сутність явищ і процесів, що проходять у мікропроцесорних системах, засвоїти елементарні прийоми програмування.

У методичних вказівках розглянуто загальні відомості про арифметичні основи мікропроцесорних систем, побудову внутрішньої пам'яті мікропроцесорної системи, що складається з поступово запам'ятовуючого пристрою (ПЗП) і статичного оперативного запам'ятовуючого пристрою (ОЗП), програмування мікропроцесора КР580ВМ80, складання простих програм та викладення на мові Assembler.

Методичні вказівки до кожного практичного заняття містять варіанти завдань для вирішення в аудиторний та позааудиторний час. Крім того, ці завдання можуть бути використовувані у контрольній роботі для студентів заочної форми навчання (конкретне завдання).

Практичне заняття 1

Системи числення

Мета роботи – опанувати правила перетворення чисел та отримати практичні навички при розв’язанні задач з використанням різних систем числення.

Загальні відомості

Система числення – символічний метод запису чисел, подання чисел за допомогою заданого набору спеціальних письмових знаків. Усі системи числення діляться на дві групи: позиційні і непозиційні.

У *непозиційних системах* числення значення цифри (вага, тобто внесок, який вона робить у значення числа) не залежить від її позиції у записі числа.

У *позиційних системах* значення цифри (вага) залежить від її розташування у числі. Позиційні системи зручні тим, що вони дозволяють записувати будь-які числа за допомогою порівняно невеликої кількості знаків. Перевагою позиційних систем є простота і легкість виконання арифметичних операцій над числами, записаними в цих системах.

Кожна позиційна система характеризується певним алфавітом цифр та основою. Основа позиційної системи числення – кількість різних знаків і символів, які використовуються для зображення цифр у даній системі числення. Значення будь-якого числа визначається не тільки розрядністю (номером позиції), але також «ваговим» значенням та алфавітом системи числення. Будь-яка позиційна система може бути подана поліномом

$$d = a_n \cdot p^n + a_{n-1} \cdot p^{n-1} + \dots + a_1 \cdot p^1 + a_0 \cdot p^0, \quad (1.1)$$

де a – алфавіт системи числення;

p – основа системи числення;

n – вага розряду.

Існують такі позиційні системи числення:

- *десятькова система* має алфавіт з десяти символів (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), основою системи є 10;

- *двійкова системи* має алфавіт з двох символів (0, 1), основою системи є 2;

- *вісімкова система* має алфавіт з восьми символів (0, 1, 2, 3, 4, 5, 6, 7), основа системи дорівнює 8;

- *шістнадцяткова система* має алфавіт з шістнадцяти символів (0, 1, 2, 3, ..., 8, 9, A, B, C, D, E, F), основа системи дорівнює 16.

Перетворення з десяткової системи числення у двійкову, вісімкову, шістнадцяткову

Метод поділу. Для перетворення цілого числа з десяткової системи числення у будь-яку іншу позиційну систему необхідно розділити десяткове число на основу нової системи числення, потім отриману частку знову розділити на основу нової системи числення і так до тих пір, поки в частці не залишиться число менше за основу нової системи числення.

Число в новій системі числення запишеться у вигляді залишків від ділення, починаючи з останньої частки. Тобто перший залишок дає молодшу цифру, а останній – старшу.

Метод множення. Даний метод застосовується для перетворення десяткових дробів, зокрема для чисел менших одиницю. При цьому число множиться на основу числення, якщо результат ≥ 1 , то в старший розряд записується це число, якщо ні, то нуль. Множимо на основу числення тільки дробову частину результату і повторюємо процедуру далі до отримання потрібного ступеня точності або до обнулення результату.

Перетворення з двійкової, вісімкової, шістнадцяткової систем числення у десяткову

Для перетворення числа із системи числення з основою p у десяткову систему числення необхідно скористатися формулою (1.1) і кожній позиції числа присвоїти певну вагу. Потім значення ваги позиції перемножується на коефіцієнт, що займає цю позицію. Результати операцій множення, виконаних для всіх позицій числа, додаються один до одного.

Перетворення з двійкової системи числення у вісімкову і шістнадцяткову та навпаки

Для перетворення з двійкової системи числення у вісімкову необхідно згрупувати (починаючи з молодшого розряду) по три біти (тріади), далі кожну групу записати однією вісімковою цифрою (таблиця 1.1).

Таблиця 1.1 – Перетворення двійкових тріад у вісімкові цифри

Двійкові тріади	000	001	010	011	100	101	110	111
Вісімкові цифри	0	1	2	3	4	5	6	7

При необхідності старші розряди двійкового числа треба доповнювати нулями до 3-х розрядів (тріад) у двійковому коді.

Для перетворення з вісімкової системи числення у двійкову необхідно кожну цифру вихідного числа записати у вигляді еквівалентного трибітного двійкового числа (див. таблицю 1.1).

Для перетворення з двійкової системи числення у шістнадцяткову необхідно згрупувати (починаючи з молодшого розряду) по чотири біти (тетради), далі кожну групу записати однією шістнадцятковою цифрою (таблиця 1.2).

Таблиця 1.2 – Перетворення двійкових тетрад у шістнадцяткові цифри

Двійкові тетради	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Шістнадцяткові цифри	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Для перетворення з шістнадцяткової системи числення у двійкову необхідно кожну цифру вихідного числа записати у вигляді еквівалентного чотирибітного двійкового числа (див. таблицю 1.2).

Перетворення з вісімкової системи числення у шістнадцяткову і навпаки відбувається за допомогою двійкового коду. Для перетворення вісімкового числа в шістнадцяткову систему числення спочатку число перетворюють у двійкову



систему, потім, розбиваючи на тетради, починаючи з молодшого біта, перетворюють у шістнадцяткову за допомогою таблиці 1.2. Для перетворення числа з шістнадцяткової системи у вісімкову його перетворюють у двійкову систему, потім розбивають на тріади, починаючи з молодшого біта, і замінюють тріади відповідними еквівалентами у вісімковій системі (таблиця 1.1).

Приклади перетворення у позиційних системах числення наведено у додатку А, таблицю перетворення позиційних систем числення – у додатку Б.

Арифметичні дії у різних системах числення

Арифметичні операції в усіх позиційних системах числення виконуються за тими ж відомими правилами, з якими працюємо в десятковій системі.

Арифметика у двійковій системі числення заснована на використанні правил додавання, віднімання і множення.

<i>Правило</i> <i>додавання</i>	<i>Правило</i> <i>віднімання</i>	<i>Правило</i> <i>множення</i>
$0 + 0 = 0$	$0 - 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 \cdot 1 = 0$
$1 + 0 = 1$	$1 - 1 = 0$	$1 \cdot 0 = 0$
$1 + 1 = (1)0$	$(1)0 - 1 = 1$	$1 \cdot 1 = 1$
		
<i>перенесення одиниці</i> <i>до старшого розряду</i>	<i>перенесення одиниці</i> <i>зі старшого розряду</i>	

Двійкове додавання виконується за тими ж правилами, що і в десятковій системі числення, тобто порозрядно, але з тією лише різницею, що перенесення одиниці в старший розряд проводиться після того, як сума досягне не десяти, а двох (10_2).

Примітка. При додаванні кількох доданків стежити за одиницями перенесення в старші розряди, тому що ці одиниці можуть переходити не тільки в сусідні старші розряди, але і вище.

При відніманні двійкових чисел, якщо віднімається 0 – 1, то в даному випадку займається 1 зі старшого розряду. Ця зайнята одиниця зі старшого розряду переходить у молодший як дві одиниці (тобто старший розряд подається двійкою більшого степеня) $2 - 1 = 1$. Відповідь записуємо 1.

При множенні у двійковій системі числення двох n -розрядних чисел отримуємо 2^n -розрядний добуток. Множення виконується за допомогою операцій зсуву і додавання.

У вісімковій та у шістнадцятковій системі числення всі операції проводяться за тими ж правилами, за якими ці дії виконуються у десятковій системі. Додавання виконується порозрядно, починаючи з молодших розрядів. Кожний символ перетворюється у десяткову систему числення, потім виконується додавання, а результат обернено переводиться назад у вісімкову або шістнадцяткову систему.

При виконанні операцій додавання і віднімання зручно використовувати вісімкову (шістнадцяткову) таблицю додавання, при виконанні операції множення – таблицю множення (додаток В).

Приклади розв'язання задач

1 Додати два числа:

а) $10000000100_{(2)} + 111000010_{(2)} = 10111000110_{(2)}$;

б) $223,2_{(8)} + 427,54_{(8)} = 652,74_{(8)}$;

в) $3В3,6_{(16)} + 38В,4_{(16)} = 73Е,А_{(16)}$.

$$\begin{array}{r} 10000000100 \\ + \quad 111000010 \\ \hline 10111000110 \end{array}; \quad \begin{array}{r} 223,20 \\ + \quad 427,54 \\ \hline 652,74 \end{array}; \quad \begin{array}{r} 3В3,6 \\ + \quad 38В,4 \\ \hline 73Е,А \end{array}$$

Виконуємо перевірку результатів розрахунків за допомогою переведення кожного числа в десяткову систему числення. Зрозуміло, що результат повинен співпадати.

а) $10000000100_{(2)} = 1 \times 2^{10} + 1 \times 2^2 = 1024 + 4 = 1028_{(10)}$;

$111000010_{(2)} = 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^1 = 256 + 128 + 64 + 2 = 450_{(10)}$;

$10111000110_{(2)} = 1 \times 2^{10} + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^2 + 1 \times 2^1 =$

$$=1024+256+128+64+4+2=1478_{(10)};$$

$$1028_{(10)}+ 450_{(10)}= 1478_{(10)}.$$

Результати збігаються, тож обчислення виконано правильно.

$$\begin{aligned} \text{б) } 223,2_{(8)} &= 2 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 + 2 \times 8^{-1} = 128 + 16 + 3 + 0,25 = 147,25_{(10)}; \\ 427,54_{(8)} &= 4 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 5 \times 8^{-1} + 4 \times 8^{-2} = \\ &= 256 + 16 + 7 + 0,625 + 0,0625 = 279,6875_{(10)}; \\ 652,74_{(8)} &= 6 \times 8^2 + 5 \times 8^1 + 2 \times 8^0 + 7 \times 8^{-1} + 4 \times 8^{-2} = \\ &= 384 + 40 + 2 + 0,875 + 0,0625 = 426,9375_{(10)}; \\ 147,25_{(10)} + 279,6875_{(10)} &= 426,9375_{(10)}. \end{aligned}$$

Результати збігаються, тож обчислення виконано правильно.

$$\begin{aligned} \text{в) } 3\text{В}3,6_{(16)} &= 3 \times 16^2 + 11 \times 16^1 + 3 \times 16^0 + 6 \times 16^{-1} = \\ &= 768 + 176 + 3 + 0,375 = 947,375_{(10)}; \\ 38\text{В},4_{(16)} &= 3 \times 16^2 + 8 \times 16^1 + 11 \times 16^0 + 4 \times 16^{-1} = \\ &= 768 + 128 + 3 + 11 + 0,25 = 907,25_{(10)}; \\ 73\text{E},\text{A}_{(16)} &= 7 \times 16^2 + 3 \times 16^1 + 14 \times 16^0 + 10 \times 16^{-1} = \\ &= 1792 + 48 + 3 + 14 + 0,625 = 1854,625_{(10)}; \\ 947,375_{(10)} + 907,25_{(10)} &= 1854,625_{(10)}. \end{aligned}$$

Результати збігаються, тож обчислення виконано правильно.

2 Відняти число:

$$\text{а) } 1100000011,011_{(2)} - 101010111,1_{(2)} = 110101011,111_{(2)};$$

$$\text{б) } 1510,2_{(8)} - 1230,54_{(8)} = 257,44_{(8)};$$

$$\text{в) } 27\text{D},\text{D}8_{(16)} - 191,2_{(16)} = \text{E}\text{C},\text{B}8_{(16)}.$$

$$\begin{array}{r} \text{а) } \begin{array}{r} 1100000011,011 \\ - 101010111,1 \\ \hline 110101011,111 \end{array}; \quad \text{б) } \begin{array}{r} 1510,20 \\ - 1230,54 \\ \hline 257,44 \end{array}; \quad \text{в) } \begin{array}{r} 27\text{D},\text{D}8 \\ + 191,20 \\ \hline \text{E}\text{C},\text{B}8 \end{array} \end{array}$$

3 Виконати множення:

$$\text{а) } 100111_{(2)} \times 1000111_{(2)} = 101011010001_{(2)};$$

$$\text{б) } 1170,64_{(8)} \times 6,3_{(8)} = 57334,134_{(8)};$$

$$\text{в) } 61,\text{A}_{(16)} \times 40,\text{D}_{(16)} = 18\text{B}7,52_{(16)}.$$

	$\begin{array}{r} 100111 \\ \times 1000111 \\ \hline 100111 \\ 100111 \\ 100111 \\ \hline 101011010001 \end{array}$	$\begin{array}{r} 1170,64 \\ \times 46,3 \\ \hline 355\ 234 \\ 7324\ 70 \\ 47432\ 0 \\ \hline 57334,134 \end{array}$
a) +	б) +	в) +
$\begin{array}{r} 100111 \\ \times 1000111 \\ \hline 100111 \\ 100111 \\ 100111 \\ \hline 101011010001 \end{array}$	$\begin{array}{r} 1170,64 \\ \times 46,3 \\ \hline 355\ 234 \\ 7324\ 70 \\ 47432\ 0 \\ \hline 57334,134 \end{array}$	$\begin{array}{r} 61, A \\ \times 40, D \\ \hline 4F\ 52 \\ 18B7, 52 \end{array}$

Завдання

Арифметичні основи мікропроцесорних систем

Згідно з номером у журналі групи обрати варіант індивідуального завдання. Завдання складається з п'яти пунктів.

1 Перевести число із десяткової системи у двійкову, вісімкову та шістнадцяткову системи числення.

2 Перевести число в десяткову систему числення.

3 Виконати додавання чисел.

4 Виконати віднімання чисел.

5 Виконати множення чисел.

Примітка/. У завданні 1б) отримати 5 знаків після коми у двійковому подані, у завданнях 3 – 5 перевірити правильність обчислень за допомогою переведення початкових даних і результатів у десяткову систему числення.

Варіант 1

1 а) $666_{(10)}$; б) $153,25_{(10)}$.

2 а) $1100111011_{(2)}$; б) $10110101,1_{(2)}$; в) $671,24_{(8)}$; г) $41A,6_{(16)}$.

3 а) $10000011_{(2)} + 1000011_{(2)}$;

б) $1010010000_{(2)} + 1101111011_{(2)}$;

в) $110010,101_{(2)} + 1011010011,01_{(2)}$;

г) $356,5_{(8)} + 1757,04_{(8)}$;

д) $293,8_{(16)} + 3CC,98_{(16)}$.

4 а) $100111001_{(2)} - 110110_{(2)}$;

б) $1111001110_{(2)} - 111011010_{(2)}$;

в) $1101111011,01_{(2)} - 101000010,0111_{(2)}$;

- г) $2025,2_{(8)}-131,2_{(8)}$;
- д) $2D8,4_{(16)}-A3,B_{(16)}$.
- 5 а) $1100110_{(2)}*1011010_{(2)}$;
- б) $2001,6_{(8)}*125,2_{(8)}$;
- в) $2C,4_{(16)}*12,98_{(16)}$.

Вариант 2

- 1 а) $164_{(10)}$;
- б) $712,25_{(10)}$.
- 2 а) $1001110011_{(2)}$;
- б) $1111100111,01_{(2)}$;
- в) $413,41_{(8)}$;
- г) $118,8C_{(16)}$.
- 3 а) $1100001100_{(2)}+1100011001_{(2)}$;
- б) $110010001_{(2)}+1001101_{(2)}$;
- в) $11111111,001_{(2)}+111111110,0101_{(2)}$;
- г) $1443,1_{(8)}+242,44_{(8)}$;
- д) $2B4,C_{(16)}+EA,4_{(16)}$.
- 4 а) $1001101100_{(2)}-1000010111_{(2)}$;
- б) $1010001000_{(2)}-1000110001_{(2)}$;
- в) $1101100110,01_{(2)}-111000010,1011_{(2)}$;
- г) $1567,3_{(8)}-1125,5_{(8)}$;
- д) $416,3_{(16)}-255,3_{(16)}$.
- 5 а) $100001_{(2)}*1001010_{(2)}$;
- б) $1723,2_{(8)}*15,2_{(8)}$;
- в) $54,3_{(16)}*9,6_{(16)}$.

Вариант 3

- 1 а) $273_{(10)}$; б) $156,25_{(10)}$.
- 2 а) $1100000000_{(2)}$; б) $1011001101,00011_{(2)}$; в) $1017,2_{(8)}$;
- г) $111,B_{(16)}$.
- 3 а) $1110001000_{(2)}+110100100_{(2)}$;
- б) $1001001101_{(2)}+1111000_{(2)}$;
- в) $111100010,0101_{(2)}+1111111,01_{(2)}$;
- г) $573,04_{(8)}+1577,2_{(8)}$;
- д) $108,8_{(16)}+21B,9_{(16)}$.
- 4 а) $1010111001_{(2)}-1010001011_{(2)}$;
- б) $1110101011_{(2)}-100111000_{(2)}$;

- в) $1110111000,011_{(2)} - 111001101,001_{(2)}$;
 г) $1300,3_{(8)} - 464,2_{(8)}$;
 д) $37C,4_{(16)} - 1D0,2_{(16)}$.
 5 а) $1011010_{(2)} * 1000010_{(2)}$;
 б) $632,2_{(8)} * 141,34_{(8)}$;
 в) $2A,7_{(16)} * 18,8_{(16)}$.

Варіант 4

- 1 а) $105_{(10)}$; б) $377,5_{(10)}$.
 2 а) $1100001001_{(2)}$; б) $1111110110,01_{(2)}$; в) $112,04_{(8)}$;
 г) $334,A_{(16)}$.
 3 а) $101000011_{(2)} + 110101010_{(2)}$;
 б) $111010010_{(2)} + 1011011110_{(2)}$;
 в) $10011011,011_{(2)} + 1111100001,0011_{(2)}$;
 г) $1364,44_{(8)} + 1040,2_{(8)}$;
 д) $158,A_{(16)} + 34,C_{(16)}$.
 4 а) $1111111000_{(2)} - 100010011_{(2)}$;
 б) $1111101110_{(2)} - 11100110_{(2)}$;
 в) $1001100100,01_{(2)} - 10101001,1_{(2)}$;
 г) $1405,3_{(8)} - 346,5_{(8)}$; д) $3DD,4_{(16)} - 303,A_{(16)}$.
 5 а) $1011100_{(2)} * 1100100_{(2)}$;
 б) $347,2_{(8)} * 125,64_{(8)}$;
 в) $10,A8_{(16)} * 35,4_{(16)}$.

Варіант 5

- 1 а) $500_{(10)}$; б) $810,25_{(10)}$.
 2 а) $1101010001_{(2)}$; б) $1101110001,011011_{(2)}$; в) $1347,17_{(8)}$;
 г) $155,6C_{(16)}$.
 3 а) $1000101101_{(2)} + 1100000010_{(2)}$;
 б) $1111011010_{(2)} + 111001100_{(2)}$;
 в) $1001000011,1_{(2)} + 10001101,101_{(2)}$; г) $415,24_{(8)} + 1345,04_{(8)}$;
 д) $113,B_{(16)} + 65,8_{(16)}$.
 4 а) $1101111100_{(2)} - 100100010_{(2)}$;
 б) $1011010110_{(2)} - 1011001110_{(2)}$;
 в) $1111011110,1101_{(2)} - 1001110111,1_{(2)}$; г) $1333,2_{(8)} - 643,2_{(8)}$;
 д) $176,7_{(16)} - E5,4_{(16)}$.

- 5 а) $1101100_{(2)} * 1010011_{(2)}$;
 б) $516,54_{(8)} * 44,64_{(8)}$;
 в) $61,8_{(16)} * 48,9_{(16)}$.

Варіант 6

- 1 а) $218_{(10)}$; б) $176,25_{(10)}$.
 2 а) $111000100_{(2)}$; б) $10110011,01_{(2)}$; в) $1665,3_{(8)}$; г) $FA,7_{(16)}$.
 3 а) $11100000_{(2)} + 1100000000_{(2)}$;
 б) $110101101_{(2)} + 111111110_{(2)}$;
 в) $10011011,011_{(2)} + 1110110100,01_{(2)}$;
 г) $1041,2_{(8)} + 1141,1_{(8)}$;
 д) $3C6,8_{(16)} + B7,5_{(16)}$.
 4 а) $10110010_{(2)} - 1010001_{(2)}$; б) $1101000000_{(2)} - 10000000_{(2)}$;
 в) $1100101111,1101_{(2)} - 100111000,1_{(2)}$;
 г) $1621,44_{(8)} - 1064,5_{(8)}$;
 д) $1AC, B_{(16)} - BD, 7_{(16)}$.
 5 а) $1000000_{(2)} * 110110_{(2)}$; б) $714,34_{(8)} * 133,4_{(8)}$;
 в) $16, B_{(16)} * 2B, 6_{(16)}$.

Варіант 7

- 1 а) $306_{(10)}$; б) $218,5_{(10)}$.
 2 а) $1111000111_{(2)}$; б) $1001111010,010001_{(2)}$; в) $465,3_{(8)}$;
 г) $252,38_{(16)}$.
 3 а) $1000001101_{(2)} + 1100101000_{(2)}$;
 б) $1010011110_{(2)} + 10001000_{(2)}$;
 в) $1100111,00101_{(2)} + 101010110,011_{(2)}$; г) $520,4_{(8)} + 635,4_{(8)}$;
 д) $2DB, 6_{(16)} + 15E, 6_{(16)}$.
 4 а) $1101000101_{(2)} - 111111000_{(2)}$;
 б) $11110101_{(2)} - 110100_{(2)}$;
 в) $1011101011,001_{(2)} - 1011001000,01001_{(2)}$;
 г) $1034,4_{(8)} - 457,44_{(8)}$;
 д) $239, A_{(16)} - 9C, 4_{(16)}$.
 5. а) $1101101_{(2)} * 101010_{(2)}$; б) $310,2_{(8)} * 40,5_{(8)}$;
 в) $18,4_{(16)} * 35,4_{(16)}$.

Варіант 8

- 1 а) $167_{(10)}$; б) $607,5_{(10)}$.
- 2 а) $110010001_{(2)}$; б) $1110011100,111_{(2)}$; в) $704,6_{(8)}$;
г) $367,38_{(16)}$.
- 3 а) $10101100_{(2)}+111110010_{(2)}$;
б) $1000000010_{(2)}+110100101_{(2)}$;
в) $1110111010,10011_{(2)}+1011010011,001_{(2)}$;
г) $355,2_{(8)}+562,04_{(8)}$;
д) $1E5,18_{(16)}+3BA,78_{(16)}$.
- 4 а) $1010110010_{(2)}-1000000000_{(2)}$;
б) $1111100110_{(2)}-10101111_{(2)}$;
в) $1101001010,101_{(2)}-1100111000,011_{(2)}$;
г) $1134,54_{(8)}-231,2_{(8)}$;
д) $2DE,6_{(16)}-12A,4_{(16)}$.
- 5 а) $10101_{(2)}*11010_{(2)}$; б) $575,2_{(8)}*102,2_{(8)}$; в) $55,4_{(16)}*6,5_{(16)}$.

Варіант 9

- 1 а) $342_{(10)}$; б) $164,25_{(10)}$.
- 2 а) $1000110110_{(2)}$;
б) $1110010100,1011001_{(2)}$;
в) $666,16_{(8)}$;
г) $1C7,68_{(16)}$
- 3 а) $1101010000_{(2)}+1011101001_{(2)}$;
б) $100000101_{(2)}+1100001010_{(2)}$;
в) $1100100001,01001_{(2)}+1110111111,011_{(2)}$;
г) $242,2_{(8)}+1153,5_{(8)}$;
д) $84,8_{(16)}+27E,8_{(16)}$.
- 4 а) $1111110_{(2)}-1111011_{(2)}$;
б) $1111100000_{(2)}-111110011_{(2)}$;
в) $1111011111,1001_{(2)}-1010111100,01_{(2)}$;
г) $1241,34_{(8)}-1124,3_{(8)}$;
д) $15F,A_{(16)}-159,4_{(16)}$.
- 5 а) $1001010_{(2)}*1101111_{(2)}$;
б) $1616,3_{(8)}*61,3_{(8)}$;
в) $3A,38_{(16)}*64,4_{(16)}$.

Варіант 10

- 1 а) $524_{(10)}$; б) $579,5_{(10)}$.
2 а) $101111111_{(2)}$;
б) $10011000,1101011_{(2)}$;
в) $140,22_{(8)}$;
г) $1DE,54_{(16)}$.
3 а) $1101010000_{(2)}+11100100_{(2)}$;
б) $100110111_{(2)}+101001000_{(2)}$;
в) $1111100100,11_{(2)}+1111101000,01_{(2)}$;
г) $1476,3_{(8)}+1011,1_{(8)}$;
д) $3E0,A_{(16)}+135,8_{(16)}$.
4 а) $1010010100_{(2)}-11101110_{(2)}$;
б) $10000001110_{(2)}-10011100_{(2)}$;
в) $1110100111,01_{(2)}-110000001,1_{(2)}$;
г) $1542,5_{(8)}-353,24_{(8)}$;
д) $3EB,8_{(16)}-3BA,8_{(16)}$.
5 а) $111000_{(2)}*100111_{(2)}$; б) $157,4_{(8)}*101,1_{(8)}$;
в) $19,7_{(16)}*58,78_{(16)}$.

Варіант 11

- 1 а) $113_{(10)}$; б) $535,1875_{(10)}$.
2 а) $11101000_{(2)}$; б) $1101101000,01_{(2)}$;
в) $1600,14_{(8)}$; г) $1E9,4_{(16)}$.
3 а) $1000111110_{(2)}+1011000101_{(2)}$;
б) $1001000_{(2)}+1101101001_{(2)}$;
в) $110110010,011_{(2)}+1000011111,0001_{(2)}$;
г) $620,2_{(8)}+1453,3_{(8)}$;
д) $348,1_{(16)}+234,4_{(16)}$.
4 а) $1100001010_{(2)}-10000011_{(2)}$;
б) $1101000001_{(2)}-10000010_{(2)}$;
в) $110010110,011_{(2)}-10010101,1101_{(2)}$;
г) $1520,5_{(8)}-400,2_{(8)}$;
д) $368,4_{(16)}-239,6_{(16)}$.
5 а) $1100110_{(2)}*110010_{(2)}$; б) $177,4_{(8)}*23,4_{(8)}$; в) $10,6_{(16)}*26,8_{(16)}$.

Варіант 12

- 1 а) $294_{(10)}$; б) $950,25_{(10)}$.
2 а) $10000011001_{(2)}$; б) $1101100,01_{(2)}$; в) $1053,2_{(8)}$; г) $200,6_{(16)}$.
3 а) $1000111110_{(2)}+10111111_{(2)}$;
б) $1111001_{(2)}+110100110_{(2)}$;
в) $1001110101,00011_{(2)}+1001001000,01_{(2)}$;
г) $104,4_{(8)}+1310,62_{(8)}$;
д) $2BD,3_{(16)}+EB,C_{(16)}$.
4 а) $11110111_{(2)}-11110100_{(2)}$;
б) $1001100111_{(2)}-101100111_{(2)}$;
в) $1100110111,001_{(2)}-1010001101,0011_{(2)}$;
г) $631,1_{(8)}-263,2_{(8)}$;
д) $262,8_{(16)}-1D6,88_{(16)}$.
5 а) $111101_{(2)}*1111_{(2)}$; б) $1751,2_{(8)}*77,24_{(8)}$; в) $40,4_{(16)}*54,6_{(16)}$.

Варіант 13

- 1 а) $617_{(10)}$; б) $412,25_{(10)}$.
2 а) $110111101_{(2)}$; б) $111001000,01_{(2)}$;
в) $1471,17_{(8)}$; г) $3EC,5_{(16)}$.
3 а) $1110100100_{(2)}+1010100111_{(2)}$;
б) $1100001100_{(2)}+1010000001_{(2)}$;
в) $1100111101,10101_{(2)}+1100011100,0011_{(2)}$;
г) $750,16_{(8)}+1345,34_{(8)}$;
д) $158,4_{(16)}+396,8_{(16)}$.
4 а) $10000000010_{(2)}-100000001_{(2)}$;
б) $1110111111_{(2)}-1010001_{(2)}$;
в) $1011001100,1_{(2)}-100100011,01_{(2)}$;
г) $1110,62_{(8)}-210,46_{(8)}$;
д) $1D8,D8_{(16)}-110,4_{(16)}$.
5 а) $11001_{(2)}*1011100_{(2)}$; б) $1440,4_{(8)}*17,6_{(8)}$;
в) $14,8_{(16)}*4A,3_{(16)}$.

Варіант 14

- 1 а) $1047_{(10)}$; б) $814,5_{(10)}$.
2 а) $1101100000_{(2)}$; б) $1011010101,1_{(2)}$;
в) $452,63_{(8)}$; г) $1E7,08_{(16)}$.

- 3 а) $1101100101_{(2)}+100010001_{(2)}$;
 б) $1100011_{(2)}+110111011_{(2)}$;
 в) $1010101001,01_{(2)}+10011110,11_{(2)}$;
 г) $1672,2_{(8)}+266,2_{(8)}$;
 д) $18B,A_{(16)}+2E9,2_{(16)}$.
- 4 а) $1110111011_{(2)}-100110111_{(2)}$;
 б) $1110000101_{(2)}-1001110_{(2)}$;
 в) $1011110100,0011_{(2)}-101001011,001_{(2)}$;
 г) $1560,22_{(8)}-1142,2_{(8)}$;
 д) $1A5,8_{(16)}-7D,A_{(16)}$.
- 5 а) $111100_{(2)}*111100_{(2)}$; б) $274,5_{(8)}*31,34_{(8)}$;
 в) $13,4_{(16)}*38,48_{(16)}$.

Вариант 15

- 1 а) $887_{(10)}$; б) $801,5_{(10)}$.
- 2 а) $1010100001_{(2)}$; б) $1011110000,100101_{(2)}$; в) $1034,34_{(8)}$;
 г) $72,6_{(16)}$.
- 3 а) $1010110101_{(2)}+101111001_{(2)}$;
 б) $1111100100_{(2)}+100110111_{(2)}$;
 в) $111111101,01_{(2)}+1100111100,01_{(2)}$;
 г) $106,14_{(8)}+322,5_{(8)}$;
 д) $156,98_{(16)}+D3,2_{(16)}$.
- 4 а) $1111100100_{(2)}-110101000_{(2)}$;
 б) $1110110100_{(2)}-1101010101_{(2)}$;
 в) $1100001,0101_{(2)}-1011010,101_{(2)}$;
 г) $537,24_{(8)}-510,3_{(8)}$;
 д) $392,B_{(16)}-149,5_{(16)}$.
- 5 а) $111100_{(2)}*1101001_{(2)}$; б) $1567,2_{(8)}*147,2_{(8)}$;
 в) $44,8_{(16)}*13,6_{(16)}$.

Вариант 16

- 1 а) $969_{(10)}$; б) $973,375_{(10)}$.
- 2 а) $10100010_{(2)}$; б) $110010010,101_{(2)}$; в) $605,02_{(8)}$; г) $3C8,8_{(16)}$.
- 3 а) $1111010100_{(2)}+10000000010_{(2)}$;
 б) $101001011_{(2)}+10000000010_{(2)}$;
 в) $1011101001,1_{(2)}+1110111,01_{(2)}$;
 г) $1053,34_{(8)}+1513,2_{(8)}$;
 д) $40A,E8_{(16)}+92,7_{(16)}$.

- 4 а) $1001100011_{(2)} - 111111110_{(2)}$;
 б) $1110001000_{(2)} - 1011110_{(2)}$;
 в) $10000010111,001_{(2)} - 1000010,01_{(2)}$;
 г) $553,2_{(8)} - 105,5_{(8)}$;
 д) $298,9_{(16)} - 67,4_{(16)}$.
- 5 а) $1110000_{(2)} * 1000101_{(2)}$; б) $436,2_{(8)} * 57,14_{(8)}$;
 в) $61,4_{(16)} * 1E, B8_{(16)}$.

Вариант 17

- 1 а) $163_{(10)}$; б) $694,375_{(10)}$.
- 2 а) $1001101001_{(2)}$; б) $1000001101,01_{(2)}$; в) $247,1_{(8)}$; г) $81,4_{(16)}$.
- 3 а) $1010111011_{(2)} + 11001000_{(2)}$;
 б) $1111101010_{(2)} + 1101100100_{(2)}$;
 в) $1100011100,1001_{(2)} + 10111100,1_{(2)}$;
 г) $1711,6_{(8)} + 1763,34_{(8)}$;
 д) $30A,4_{(16)} + 89,48_{(16)}$.
- 4 а) $111100101_{(2)} - 1101101_{(2)}$;
 б) $1001011100_{(2)} - 110110101_{(2)}$;
 в) $1110011001,1011_{(2)} - 1101101100,11_{(2)}$;
 г) $1617,4_{(8)} - 1442,6_{(8)}$;
 д) $36C,2_{(16)} - 38,5_{(16)}$.
- 5 а) $1100001_{(2)} * 1011100_{(2)}$; б) $104,54_{(8)} * 66,3_{(8)}$;
 в) $4D, A_{(16)} * 69,6_{(16)}$.

Вариант 18

- 1 а) $917_{(10)}$; б) $74,5_{(10)}$.
- 2 а) $1110011100_{(2)}$; б) $111110100,101_{(2)}$; в) $1446,62_{(8)}$;
 г) $9C, D_{(16)}$.
- 3 а) $11100101_{(2)} + 1110111111_{(2)}$;
 б) $1101111_{(2)} + 1000010_{(2)}$;
 в) $1000010100,011_{(2)} + 1111110111,011_{(2)}$;
 г) $1664,1_{(8)} + 501,3_{(8)}$;
 д) $1F0,6_{(16)} + 34,4_{(16)}$.
- 4 а) $1011110110_{(2)} - 1001011001_{(2)}$;
 б) $1101101110_{(2)} - 1000111000_{(2)}$;
 в) $1101110010,01_{(2)} - 111110110,01_{(2)}$;
 г) $1653,1_{(8)} - 415,6_{(8)}$;
 д) $1B9,4_{(16)} - 1B4,6_{(16)}$.

- 5 а) $1010000_{(2)} * 1101011_{(2)}$; б) $1605,14_{(8)} * 22,04_{(8)}$;
 в) $24,4_{(16)} * 5E,4_{(16)}$.

Вариант 19

- 1 а) $477_{(10)}$; б) $863,25_{(10)}$.
 2 а) $101011100_{(2)}$; б) $11100011,1_{(2)}$; в) $1762,7_{(8)}$; г) $1B5,6_{(16)}$.
 3 а) $1011010111_{(2)} + 1011110101_{(2)}$;
 б) $1110001001_{(2)} + 1110101011_{(2)}$;
 в) $1100011000,101_{(2)} + 10000010100,1_{(2)}$;
 г) $1742,4_{(8)} + 456,1_{(8)}$;
 д) $29E,3_{(16)} + D8,4_{(16)}$.
 4 а) $1000001000_{(2)} - 101110000_{(2)}$;
 б) $1111011010_{(2)} - 101001001_{(2)}$;
 в) $1101101,1011_{(2)} - 111110,001_{(2)}$;
 г) $1026,66_{(8)} - 124,2_{(8)}$;
 д) $3E0,2_{(16)} - 1EA,2_{(16)}$.
 5 а) $1101101_{(2)} * 100000_{(2)}$; б) $1355,5_{(8)} * 125,64_{(8)}$;
 в) $20,4_{(16)} * 2F,4_{(16)}$.

Вариант 20

- 1 а) $804_{(10)}$; б) $207,625_{(10)}$.
 2 а) $10010000_{(2)}$; б) $1110101100,1011_{(2)}$; в) $1164,36_{(8)}$;
 г) $1D5,C8_{(16)}$.
 3 а) $1100010100_{(2)} + 1100011010_{(2)}$;
 б) $1001001_{(2)} + 1100010001_{(2)}$;
 в) $1000110,101_{(2)} + 1010010001,001_{(2)}$;
 г) $433,4_{(8)} + 1774,2_{(8)}$;
 д) $F7,4_{(16)} + 178,4_{(16)}$.
 4 а) $10111110_{(2)} - 1100010_{(2)}$;
 б) $1111110000_{(2)} - 100111011_{(2)}$;
 в) $1011011100,011_{(2)} - 111011111,1_{(2)}$;
 г) $314,54_{(8)} - 77,14_{(8)}$;
 д) $233,68_{(16)} - DB,4_{(16)}$.
 5 а) $1110010_{(2)} * 1010111_{(2)}$;
 б) $242,2_{(8)} * 73,2_{(8)}$;
 в) $1D,A_{(16)} * 8,4_{(16)}$.

Практична робота 2

Пам'ять мікропроцесорних систем, її види

Мета роботи – опанувати побудову внутрішньої пам'яті мікропроцесорної системи, що складається із постійного запам'ятовуючого пристрою (ПЗП) і статичного оперативного запам'ятовуючого пристрою (ОЗП), якщо процесорна система працює у реальному режимі часу.

Загальні відомості

Система пам'яті є функціональною частиною мікропроцесорної системи, призначеної для запису, зберігання і видачі інформації. Технічні засоби, що реалізують функції пам'яті називаються пристроями, які запам'ятовують, або *запам'ятовуючими пристроями*.

Пристрої, що запам'ятовують (ЗП), є найважливішою складовою частиною будь-якого обчислювального пристрою, у тому числі і побудованого на мікропроцесорних великих інтегральних схемах (ВІС).

За функціональним призначенням ЗП, використовуваних в мікропроцесорних системах, можна розділити на такі групи:

- *надоперативні* ЗП – набір регістрів, вміст яких безпосередньо використовується при обробці інформації у мікропроцесорі;
- *кеш-пам'ять* допоміжна оперативна пам'ять, недоступна для програміста, служить для підвищення швидкодії МП;
- *оперативні* ЗП, що зберігають оперативну інформацію (операнди, частини програми), потрібну в процесі роботи;
- *постійні* ЗП, призначені для тривалого зберігання незмінної у процесі роботи мікроЕОМ інформації (програм, мікропрограм, констант);
- *напівпостійні* ЗП, який за виконуваними функціями повністю відповідають постійним ЗП, але відрізняються від останніх можливістю швидкої зміни інформації, що зберігається у них, при необхідності зміни програм або констант за обмежений час;

- *зовнішні ЗП* для зберігання великих об'ємів інформації, з невеликою питомою вартістю біта інформації, що зберігається;
- *буферні ЗП* для узгодження різних рівнів системи пам'яті мікроЕОМ між собою і зовнішніх пристроїв із системою пам'яті.

Запам'ятовуючі пристрої мікроЕОМ можуть бути реалізовані на основі тільки ВІС ОЗП. Проте втрата інформації при вимкненні живлення, вища питома вартість біта інформації, що зберігається, і ряд інших причин привели до широкого використання в мікроЕОМ постійних і напівпостійних ЗП.

Пристрої, що запам'ятовують, характеризуються рядом якісних показників.

1 *Ємність ЗП* визначається максимально можливою кількістю бітів інформації, що зберігається.

2 *Ширина вибірки* визначається кількістю інформації, записуваної ЗП або витягнутої з нього за одне звернення.

3 *Час звернення* визначається з моменту подачі у пристрій сигналу запису або читання до того моменту, коли закінчатся всі дії, пов'язані з виконуваною операцією, і пристрій буде готовий прийняти та реалізувати наступну операцію звернення до накопичувача інформації.

4 *Швидкість обміну інформацією між ЗП й іншими пристроями* є важливим параметром пристроїв, що запам'ятовують. Вона визначається кількістю бітів (байтів), передаваних в одиницю часу.

5 *Показник питомої вартості* застосовується для оцінки економічних характеристик ЗП. Він визначається відношенням його вартості до інформаційної ємності, тобто вартості біта інформації, що зберігається.

6 *Надійність*, а для систем спеціального призначення – *масогабаритні показники і споживана потужність* є для ЗП істотними характеристиками.

7 Одна з характеристик ЗП – здатність зберігати інформацію при вимкненні джерел живлення. У цьому випадку розрізняють *енергозалежну або незалежну пам'ять*.

Постійні запам'ятовуючі пристрої

Постійні запам'ятовуючі пристрої (ПЗП) у мікропроцесорних обчислювальних системах служать для зберігання програм та іншої незмінної інформації. Важлива перевага ПЗП в порівнянні з ОЗП – збереження інформації при вимиканні живлення. Вартість біта інформації, що зберігається у ПЗП, може бути майже на порядок нижче, ніж в ОЗП. Постійні ЗП можуть бути реалізовані на основі різних фізичних принципів та елементів і відрізняються способом занесення інформації, кратністю занесення, способом стирання.

В даний час застосовуються такі види ПЗП:

- програмовані на заводі-виготовнику або масочні ПЗП (МПЗП);
- програмовані користувачем;
- перепрограмовані ПЗП.

Перші два види ПЗП допускають тільки одноразове програмування, третій вид дозволяє змінювати інформацію, що зберігається в ньому, багато разів.

Програмовані масочні ПЗП програмуються їх виготовником, який за підготовленою користувачем інформацією робить фотошаблони, за допомогою яких заносить цю інформацію у процесі виробництва на кристал ПЗП. Цей спосіб найдешевший і призначений для великосерійного виробництва ПЗП.

Масочні ПЗП будуються на основі діодів, біполярних і МДП- транзисторів. У діодних ПЗП діоди включені в тих перетинах матриці, які відповідають запису «1», і відсутні в місцях, де повинен бути записаний «0». Оскільки діодна матриця є елементом з гальванічними зв'язками, то вихідні сигнали мають ту ж форму, що і входні. Таким чином, якщо на входи подаються напруги постійних рівнів, то і на виходах рівні будуть також постійними, тому відпадає необхідність у вихідному регістрі для зберігання інформації.

Масочні ПЗП на біполярних і МДП-транзисторах також будуються у вигляді матриць. Масочні ПЗП характеризуються великою надійністю, але при їх виготовленні виникає ряд незручностей для замовника і виготовника.

Велика номенклатура ПЗП і мала їх тиражність потребують від виготовника підвищених витрат на фотошаблони, що збільшує вартість ПЗП. Відсутня можливість оперативно змінювати інформацію в ПЗП без виготовлення нової ІС, що особливо незручно на етапі розроблення програм системи.

Програмовані користувачем ПЗП – більш універсальні і, отже, дорожчі прилади. Вони є матрицями біполярних приладів, зв'язки яких з адресними і розрядними шинами руйнуються при занесенні (на спеціальних програмуючих пристроях) відповідних кодових комбінацій. Ці пристрої виробляють напруги, необхідні і достатні для перепалювання плавких перемичок у вибраних елементах ПЗП. Можливість програмування користувачем зробила ПЗП цього типу надзвичайно зручними при розробці мікроЕОМ.

Найбільшого поширення набули ПЗП з *ультрафіолетовим (УФ) стиранням* серії К573, з плавкими перемичками серій К556 і К541, з електричним стиранням і записом інформації серій К558, К1601, К1609.

Перепрограмуючі ПЗП – це ПЗП із змінним вмістом. На затворах матриці МОП-транзисторів тривалий час можуть зберігатися заряди, які створюють заданий код. Всі перепрограмуючі ПЗП є МОП-приладами.

При необхідності в перепрограмуванні мікросхеми заздалегідь записану інформацію стирають УФ променем через прозоре кварцове віконце на поверхні корпусу мікросхеми. Час збереження інформації в мікросхемах ПЗП даного типу визначається якістю призатворного діелектрика і для сучасних мікросхем складає десять років і більше.

Мікросхеми ПЗП з електричним стиранням інформації популярні у розробників мікропроцесорної техніки завдяки можливості швидкого стирання і запису, великій допустимій кількості циклів перезапису інформації (10000 разів і більше). Проте вони достатньо дорогі і складні в порівнянні з мікросхемами ПЗП з УФ стиранням і тому поступаються останнім за ступенем використання у мікропроцесорній апаратурі. Основу комірки, що запам'ятовує, в ПЗП з електричним стиранням складає МОП-транзистор з плаваючим затвором, такий же, як і в ПЗП з УФ стиранням. Але в

мікросхемах даного типу технологічними методами забезпечено можливість зворотного тунелювання, тобто відбору електронів з плаваючого затвора, що дозволяє вибірково стирати занесену інформацію.

Оперативні запам'ятовуючі пристрої

За принципом зберігання інформації напівпровідникові оперативні запам'ятовуючі пристрої (ОЗП) поділяються на *динамічні і статичні*.

Динамічні пристрої, що запам'ятовують, будуються на основі елемента, що запам'ятовує, зберігають свій стан тільки певний проміжок часу і тому вимагають періодичного відновлення. Елементом динамічних напівпровідникових ЗП служить конденсатор, в якому інформація зберігається у формі наявності або відсутності заряду. Через витoki поступово зменшується заряд на конденсаторі, що запам'ятовує; для відновлення заряду конденсатор, що запам'ятовує, періодично вмикають до джерела живлення.

Статичні пристрої, що запам'ятовують, є найбільш поширеним видом пам'яті мікропроцесорних систем. Більшість статичних пристроїв, що запам'ятовують, реалізуються на основі МОП-технології, але існують і статичні ЗП на біполярних схемах.

Елемент пам'яті статичного ЗП є звичайним тригером. Він може бути встановлений або в стан «1», або в стан «0». Якщо тригер встановлений в «1», то цей стан зберігається до тих пір, поки не буде проведене скидання тригера або не буде вимкнене живлення.

Приклад вирішення завдання

Припустимо, адреси, що покриваються ПЗП і ОЗП:

ПЗП від 00000H до 03FFFFH;
ОЗП від 0C0000H до 0FFFFFFH.

Шина адреси є 20 – розрядною, отже адрес пам'яті записується на п'яти 16-кових розрядах (один 16-ковий розряд

подається на 4 двійкових). В адресах ОЗП в цьому випадку перше значення «0» пишеться для визначення С і F, як символів числа, а не букв.

Визначення ємності ПЗП і ОЗП

За отриманим діапазоном адрес визначимо ємність ПЗП і ОЗП. Знайдемо кількість розрядів, що змінюються, і запишемо адресу в двійковому коді.

ПЗП

Початкова адреса: 00000000000000000000 в.

Кінцева адреса: 00000011111111111111 в.

Змінилися 14 розрядів, отже, ємність ПЗП – 2^{14} .

Для 8-розрядної шини даних ємність ПЗП – $2^{14} \times 8$:

$$2^{14} = 2^{10} \times 2^4,$$

$$2^{10} = 1\text{К} - \text{кілобіт},$$

таким чином, ємність ПЗП дорівнює $16\text{К} \times 8$.

ОЗП

Початкова адреса: 11000000000000000000 в.

Кінцева адреса: 11111111111111111111 в.

Змінилися 18 розрядів.

Для 8-розрядної шини даних ємність ОЗП дорівнює

$$2^{18} \times 8,$$

$$2^{18} = 2^{10} \times 2^8 = 256\text{К},$$

таким чином, ємність ОЗП дорівнює $256\text{К} \times 8$.

Для зображення схеми необхідно визначити:

- ємності мікросхем ОЗП і ПЗП (за таблицею 2.2);
- структури ОЗП і ПЗП (кількість мікросхем, спосіб з'єднання);
- загальну структуру пам'яті.

Як приклад ПЗП має ємність $16\text{К} \times 1$ (рисунок 2.1), а ОЗП – $128\text{К} \times 8$ (рисунок 2.2).

Таким чином, схема ПЗП має 14 адресних входів $16\text{К} = 2^{14}$, один вхід / вихід даних і вхід CS (вибірки кристала).

Схема ОЗП має 17 адресних входів $128\text{K} = 2^{17}$, 8 входів/виходів даних, входи CS і \overline{W}/R .

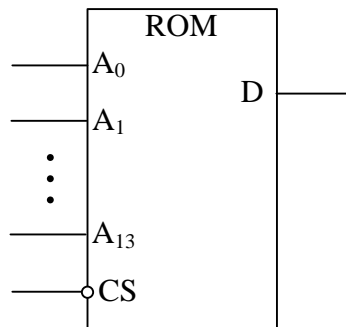


Рисунок 2.1 – Мікросхема ПЗП 16К×1

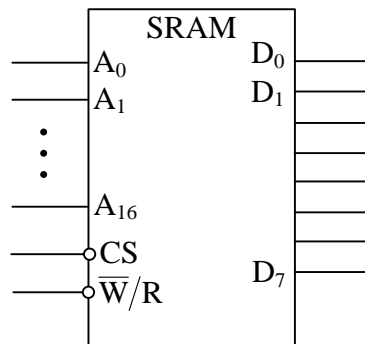


Рисунок 2.2 – Мікросхема ОЗП 128К×8

Визначення ПЗП і ОЗП

Структура ПЗП

Ємність ПЗП – 16К×8. Ємність мікросхеми – 16К×1, отже, для отримання потрібної ємності ПЗП необхідно об'єднати паралельно по адресних входах і входу CS 8 мікросхем, кожна з яких забезпечить один розряд шини даних (рисунок 2.3).

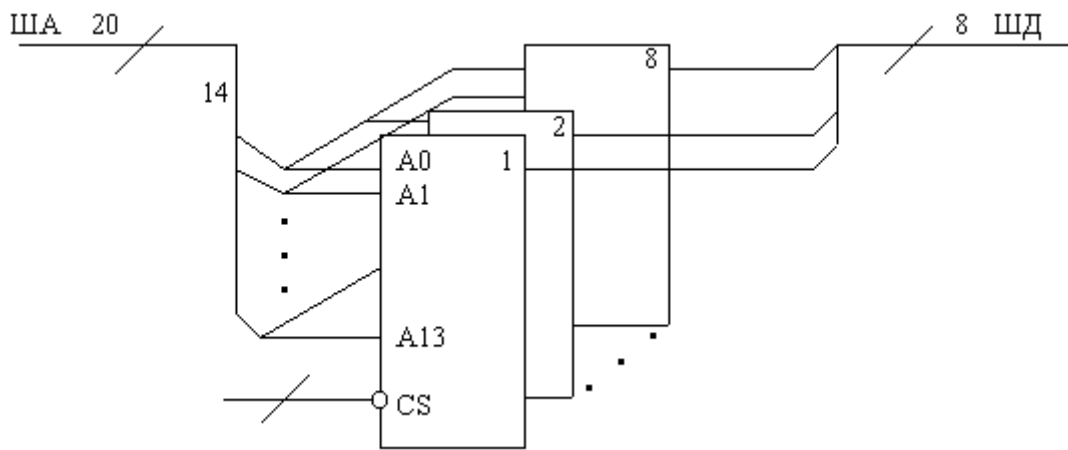


Рисунок 2.3 – Структура ПЗП 16К×8

Структура ОЗП

Ємність ОЗП – 256К×8, ємність мікросхеми – 128К×8, отже, для побудови такого ОЗП необхідні 2 мікросхеми.

Для побудови ОЗП звернемося до адрес, на яких працює ця пам'ять. За завданням змінюються 18 молодших розрядів $A_0 - A_{17}$. Для кожної мікросхеми можуть змінюватися лише 17 адрес $A_0 - A_{16}$. Таким чином, старший розряд визначає напрямок на мікросхему. Якщо $A_{17} = 0$, буде задіяна перша мікросхема ОЗП, якщо 1 – друга (рисунок 2.4).

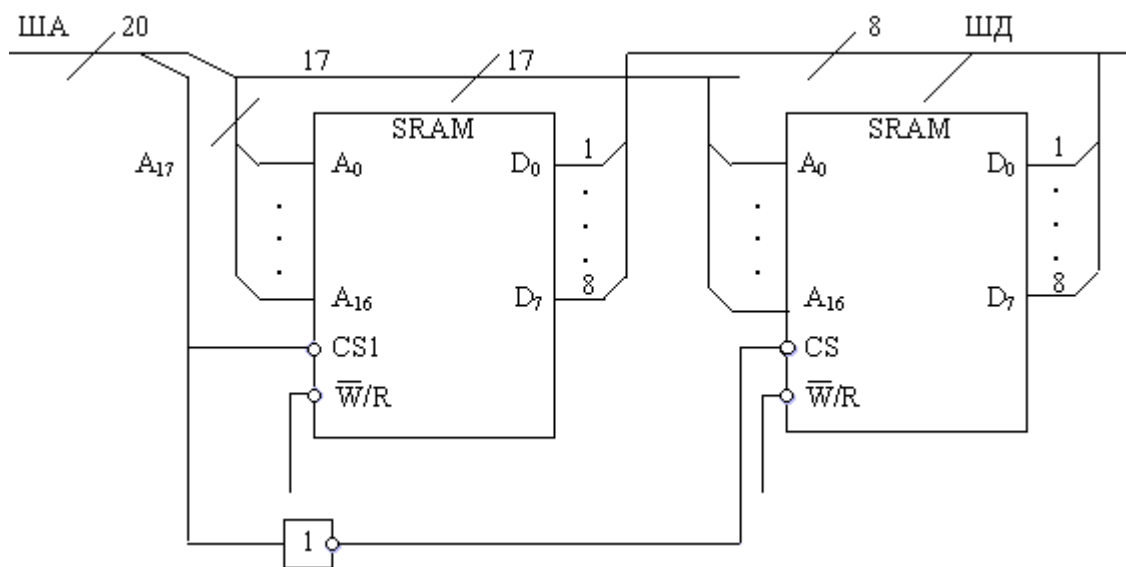


Рисунок 2.4 – Структура ОЗП 256К×8

Загальна структура пам'яті наведена на рисунку 2.5.

За завданням початкові адреси ПЗП та ОЗП такі:

ПЗП (00000Н - 00000000000000000000В),

ОЗП (0С0000Н - 11000000000000000000В).

ПЗП працює за станом розрядів $A_{19}, A_{18} - 00$, а за станом $A_{19}, A_{18} - 11$ працює ОЗП. За допомогою найпростішої логіки можна побудувати дешифратор напрямків ПЗП / ОЗП (рисунок 2.6).

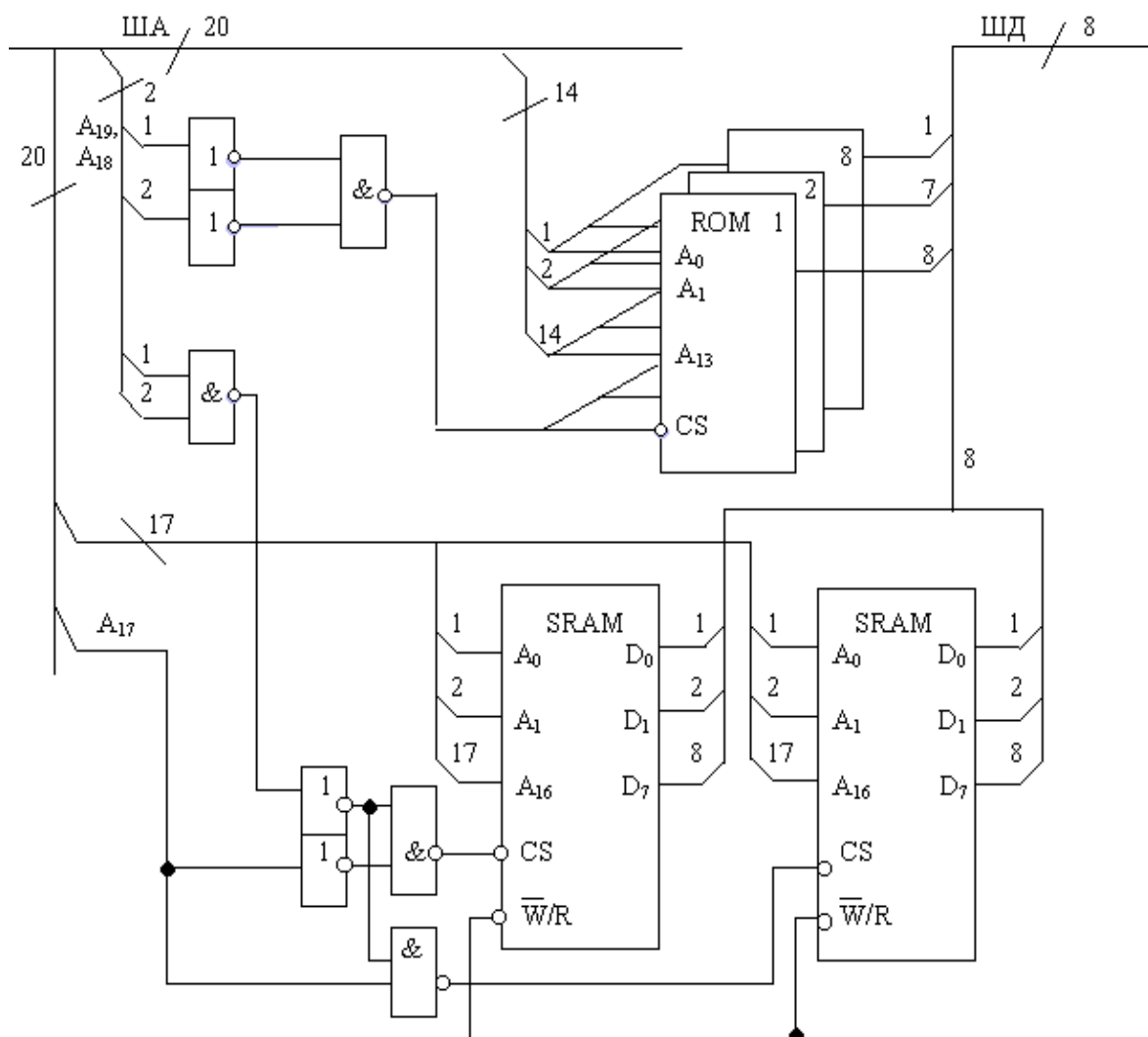


Рисунок 2.5 – Загальна структура проектованої пам'яті

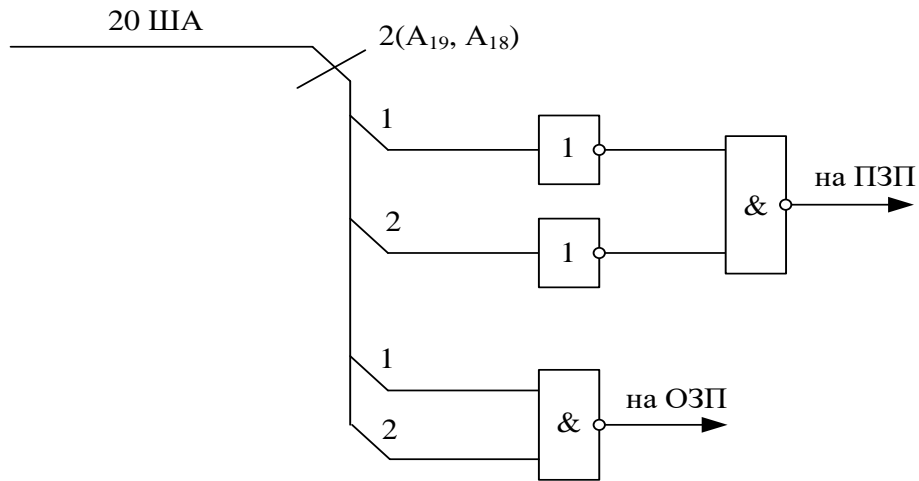


Рисунок 2.6 – Дешифратор напрямків ПЗП-ОЗП

Завдання

Побудова внутрішньої пам'яті мікропроцесорної системи, що складається з ПЗП і статичного ОЗП

Побудувати внутрішню пам'ять мікропроцесорної системи, що складається з ПЗП і статичного ОЗП. Процесорна система працює в реальному режимі.

Розрядність ША - 20, ШД - 8.

Адреси, що використовуються просторами ПЗП і ОЗП, і ємність мікросхеми вибрати з таблиць 2.1, 2.2.

Вибір варіанта

Адреси, що використовуються ПЗП (таблиця 2.1), визначаються за передостанньою цифрою номера залікової книжки, а ОЗП – за останньою.

Ємність мікросхеми ПЗП (таблиця 2.2) визначається за останньою цифрою номера залікової книжки, ОЗП – за передостанньою.

Таблиця 2.1 – Індивідуальні завдання

Номер	Адреси ПЗП	Номер	Адреси ОЗП
1	00000H÷0FFFFH	1	20000H÷3FFFFH
2	00000H÷1FFFFH	2	40000H÷5FFFFH
3	00000H÷07FFFH	3	60000H÷7FFFFH
4	00000H÷1FFFFH	4	80000H÷9FFFFH
5	00000H÷03FFFH	5	0C0000H÷0DFFFFH
6	00000H÷0FFFFH	6	0A0000H÷0BFFFFH
7	00000H÷7FFFH	7	0E0000H÷FFFFFFH
8	00000H÷1FFFFH	8	40000H÷7FFFFH
9	00000H÷07FFFH	9	80000H÷BFFFFH
10	00000H÷03FFFH	10	0C0000H÷0FFFFFFH

Таблиця 2.2 – Індивідуальні завдання

Номер	Мікросхема ПЗП	Номер	Мікросхема ОЗП
1	32К×1	1	64К×1
2	64К×4	2	128К×4
3	64К×1	3	64К×8
4	16К×4	4	64К×4
5	32К×4	5	32К×4
6	64К×8	6	128К×1
7	32К×8	7	32К×8
8	16К×8	8	256К×1
9	32К×1	9	32К×1

Практична робота 3

Вивчення функціонування і програмування ВІС КР580ВВ55

Мета роботи – опанувати побудову конфігурацій, програмування паралельного програмованого інтерфейсу (ППІ) згідно з різними режимами роботи ППІ за допомогою керуючого слова.

Загальні відомості

Для організації обміну інформацією у паралельному коді між шиною даних мікропроцесорної системи і пристроями введення - виведення (ПВВ) використовується ВІС КР580ВВ55 – паралельний програмований інтерфейс (ППІ).

ПВВ підключається до каналів (портів) ВІС ППІ. У структуру ППІ входять 3 (три) восьмирозрядні канали – КА, КВ, та КС. КС складається з двох чотирирозрядних каналів – КС1 (біти С0 – С3) та КС2 (біти С4 – С7).

ВІС може працювати в трьох режимах:

РЕЖИМ «0» – синхронна програмно – керуюча передача даних через три (3) незалежні восьмирозрядні канали КА, КВ і КС або через два (2) восьмирозрядні канали КА і КВ, а також через два (2) чотирирозрядні канали КС1 и КС2. Всі канали незалежно один від одного можна налаштувати на введення або виведення.

РЕЖИМ «1» – асинхронне **однонаправлене** введення - виведення через два канали КА і КВ під керуванням трьох сигналів квитирування, переданих через канал С (КС). Керуючі сигнали називаються сигналами квитирування, оскільки це сигнали – квитанції, які оповіщають про певну подію.

РЕЖИМ «2» – асинхронне двонаправлене введення-виведення тільки через канал А під керуванням п'яти сигналів квитирування, переданих через канал С (КС).

Канал А може працювати у режимах «0», «1», «2», канал В – «0», «1», канал С – «0».

Програмування ППШ

Оскільки ВІС є програмованою, то перед роботою до неї завантажуються керуюче слово (КС) певного формату в регістр керуючого слова (РКС). Формат КС приведено на рисунку 3.1.

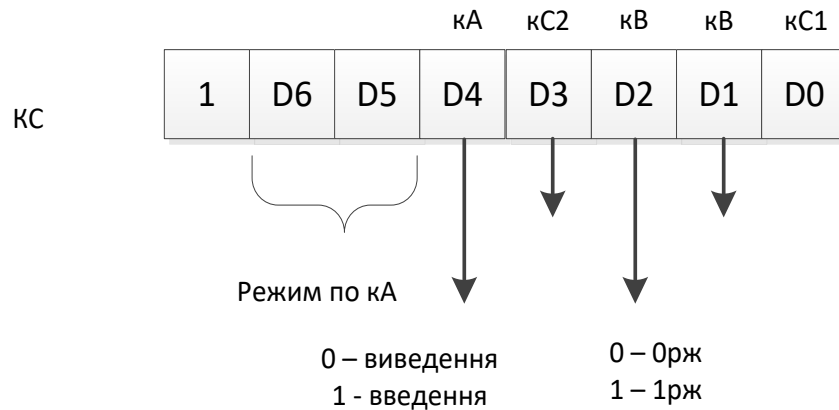


Рисунок 3.1 – Формат керуючого слова (КС)

КС визначає:

- 1) режими роботи каналів КА, КВ та КС1 і КС2;
- 2) напрямок передачі через канали: **0** – виведення; **1** – введення.

Розряд $D7=1$ – це ознака КС.

Розряди $D6D5$ визначають режим роботи каналу А (КА);

$D6D5 = 00$ – режим «0»;

$D6D5 = 01$ – режим «1»;

$D6D5 = x1$ – режим «2».

Розряд $D4$ визначає напрямок передачі через восьмибітовий канал А (КА):

0 – виведення; **1** – введення.

Розряд $D3$ визначає напрямок передачі через чотирибітовий канал С2 (КС2):

0 – виведення; **1** – введення.

Розряд $D2$ визначає режим роботи восьмибітового каналу В (КВ):

$D2 = 0$ – режим «0».

Розряд $D1$ визначає напрямок передачі через восьмибітовий канал В (КВ):

0 – виведення; **1** – введення.

Для програмування ППІ необхідно завантажити керуюче слово (КС) у регістр керуючого слова (РКС).

Усі регістри ППІ мають такі адреси:

	7	6	5	4	3	2	1	0
Регістр каналу А								
Регістр каналу В								
Регістр каналу С								
Регістр керуючого слова (РКС)								

Програма ініціалізації матиме вигляд:

DI ; заборона переривання
MVI A, КС ; керуюче слово (КС) → Акумулятор
OUT xxxxxx11 ; КС → РКС ППІ
EI ; дозвіл переривання

Функціонування ППІ у режимі «0»

Це синхронна програмно – керована передача даних через канали А, В і С (КА, КВ і КС) без сигналів керування.

На рисунку 3.2 приведено конфігурацію виводів ППІ при роботі усіх каналів у режимі «0». КА – введення, КС2 – виведення, КС1 – введення, КВ – виведення інформації.

Керуюче слово буде мати вигляд (рисунок 3.2)

$$КС = 10010001_2 = 91_{16}.$$

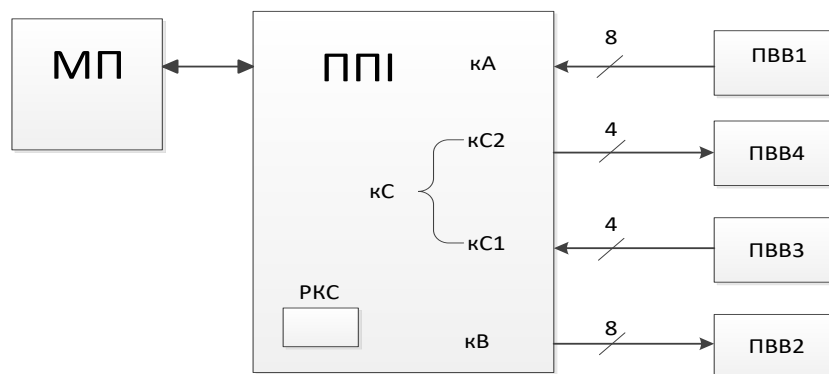


Рисунок 3.2 – Конфігурація виводів ППІ

Функціонування ППІ у режимі «1»

Це асинхронне однонаправлене введення/виведення через канали А і В під керуванням трьох сигналів квитирування, які передаються через КС.

Сигнали квитирування:

«STB...» («Строб...») – цей сигнал виробляє пристрій - джерело переданого байта і передає на пристрій - приймач. Напрямок строба збігається з напрямком переданого байта.

«ASK...» («Підтвердження...») – даний сигнал виробляє пристрій - приймач до пристрою - джерела, інформуючи про отримання байта.

«IRQ...» («Запит переривання...») – такий сигнал виробляє ППІ до МП або до контролера переривань. За цим сигналом МП ініціює видачу нового байта в канал ППІ при виході інформації або введення байта з каналу при введенні інформації.

При введенні інформації:

ПВВ – пристрій – джерело,

ППІ – пристрій – приймач.

Введення інформації через канал А супроводжується такими сигналами квитирування:

- сигнал «STB RC по КА» («Строб прийому по КА») виробляє ПВВ при видачі байта в канал А і передає в ППІ у канал С2 (на виводі С4);

- сигнал «ASK RC по КА» («Підтвердження прийому по каналу А») формує ППІ при отриманні байта (на виводі С5) і передає до ПВВ.

Якщо переривання дозволені, то ППІ формує сигнал «IRQ по КА» («Запит переривання по каналу А») на виводі С3 і передає його до мікропроцесора або контролера переривань.

Вільні виводи С7 і С6 каналу С2 (КС2) можна налаштувати на введення або виведення бітом D3 у керуючому слові.

На рисунку 3.3 наведено керуюче слово і конфігурацію виводів ППІ при **введенні інформації через канал А в режимі «1»**.

У керуючому слові «КС» запишеться таке:

D6D5 = 0 12, бо КА працює у режимі «1»,

D4 = 1, бо КА налаштований на введення інформації.

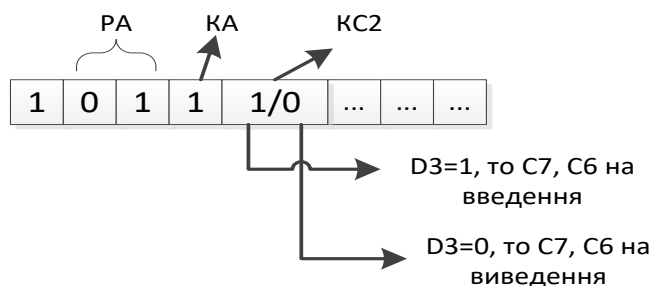
Якщо $D3 = 1$, то вільні виводи каналу C2 (C7 і C6) будуть налаштовані на введення інформації; якщо $D3 = 0$, тоді виводи C7 і C6 будуть налаштовані на вивід.

Одночасно канал В (КВ) може працювати у режимі «0» або в режимі «1».

Біти D2, D1, D0 керуючого слова у даному прикладі не визначені, оскільки режим роботи каналу В (режим «0» або режим «1») і напрямок передачі по каналах КВ і КС1 (введення або виведення) не задані.

Якщо КВ працює в режимі «1», то у керуючому слові (КС) біт D0 – байдужий, оскільки всі виводи КС1 використовуються під сигнали квитирування (вільних виводів немає).

Керуюче слово:



Конфігурація виводів:

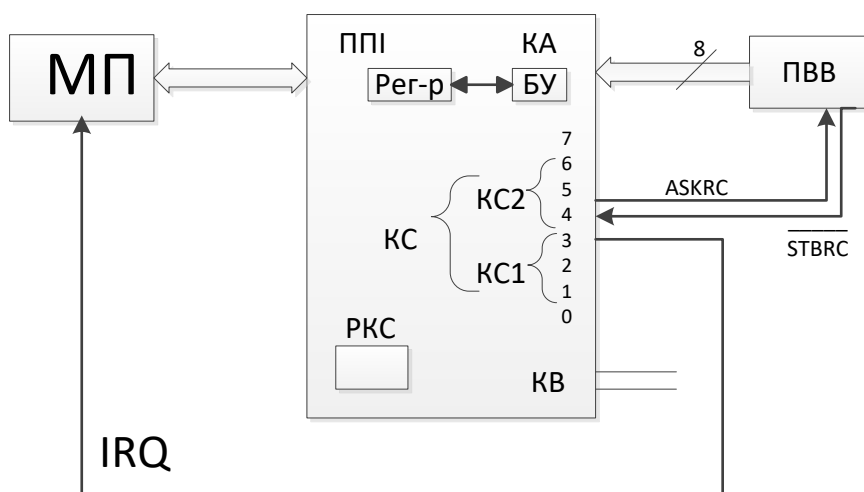


Рисунок 3.3 – Конфігурація виводів ПІІ при введенні інформації через канал А в режимі «1», режим роботи каналу В і напрямок передачі КВ та по вільних виводах каналу С1(C0-C2), які не задано

Введення інформації через канал В (КВ) у режимі «1» супроводжується аналогічними сигналами, які формуються на виводах каналу С1 (КС1).

«STB RC по КВ» («Строб прийому по КВ») виробляється ПБВ, підключеного до каналу В (КВ) і приймається ППІ на вивід С2;

«ASK RC по КВ» («Підтвердження прийому по КВ») формується ППІ на виводі С1 і передається до ПБВ;

«IRQ по каналу В» («Запит переривання по каналу В») формується на виводі С0 ППІ та передається до мікропроцесор або контролер переривань.

При виведенні інформації:

ППІ – пристрій – джерело,

ПБВ – пристрій – приймач.

Виведення інформації через канал А в режимі «1» супроводжується такими сигналами квитирування:

- сигнал «STB WR по КА» («Строб запису по КА») формується ППІ при видачі байта з каналу А на виході С7 (канал С2) і передається ПБВ;

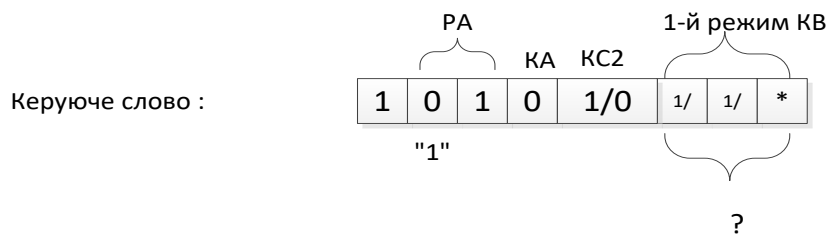
- сигнал «SAK WR по КА» («Підтвердження запису по КА») формує ПБВ при отриманні байта і передається до ППІ на вивід С6 (канал С2);

- сигнал «IQR по каналу А» («Запит переривання по каналу А») формується ППІ на виводі С3 каналу С1 і передається до мікропроцесора або контролера переривань, якщо переривання дозволені.

Вільні виводи каналу С2 (КС2) – С5 і С4 – можна налаштувати бітом D3 керуючого слова на введення (D3 = 1) або на виведення (D = 0).

Одночасно канал В (КВ) може працювати у режимі «0» або «1».

На рисунку 3.4 приведено керуюче слово і конфігурації виводів ППІ при **виведенні інформації через канал А у режим «1»**.



Конфігурація виводів:

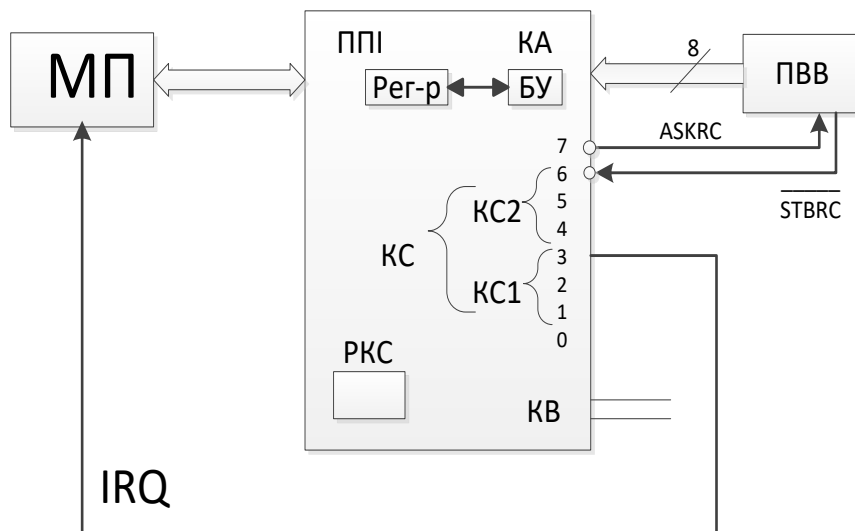


Рисунок 3.4 – Конфігурація виводів ППІ при виведенні інформації через канал А у режимі «1», режим роботи В та напрямок передачі через КВ і по вільних виводах С1 (С0-С2), які задано

Виведення інформації по каналу В (КВ) в режимі «1» супроводжується аналогічними сигналами, які формуються на виводах каналу С1 (КС1):

«STB WR по КВ» («Строб запису по КВ») виробляється ППІ на виході С1 при видачі байта і передається в ПВВ, підключений до каналу В (КВ);

«ASK WR по КВ» («Підтвердження запису по КВ») формується ПВВ і передається на вивід С2 каналу С1 ППІ;

«IRQ по KB» («Запит переривання по каналу В») формується ППІ на виводі C0 каналу C1 та передається мікропроцесору або контролеру переривань.

Якщо KB працює у режимі «1», то в керуючому слові (КС) біт D0 – байдужий, бо всі виводи КС1 (C0 – C2) використовуються під сигнали квитирування (вільних виводів немає).

Функціонування ППІ у режимі «2»

Це двонаправлене введення – виведення тільки через КА під керуванням п'яти сигналів квитирування, які передаються через КС.

Три (3) сигнали керують введенням («STB RC по КА», «ASK RC по КА», «IRQ по КА») і три (3) сигнали керують виводом («STB WR по КА», «ASK WR по КА», «IRQ по КА»). Сигнал «IRQ по КА» однаковий для введення та для виведення. Формування і напрямок передачі усіх сигналів було розглянуто раніше.

Режим «1» відрізняється від режиму «2» таким:

- у режимі «1» напрямок передачі по каналу А задається бітом D4 керуючого слова;
- у режимі «2» канал А (КА) – двонаправлений, а конкретний напрямок передачі по КА задається системними керуючими сигналами.

\overline{WRITE} $\overline{ЗАПИС}$ і \overline{READ} $\overline{ЧИТАННЯ}$:

якщо $\overline{WR} = 0$, $\overline{RD} = 1$ – канал А налаштований на виведення інформації,

якщо $\overline{WR} = 1$, $\overline{RD} = 0$ – канал А налаштований на введення інформації.

Особливість формату керуючого слова в режимі «2»

На рисунку 3.5 зображено конфігурацію виводів ППІ при роботі КА в режимі «2», а також приведено формат керуючого слова в цьому режимі.

Бит D4 керуючого слова, визначальний напрямок передачі через канал А, - байдужий (вказано зірочкою *), оскільки канал А у режимі «2» є двонаправленим, і може працювати як на введення, так і на виведення.

Бит D3 у керуючому слові, визначає напрямок передачі через канал С2, також байдужий, бо всі виводи КС2 (С7,С6,С5 и С4) зайняті під сигналом квитирування.

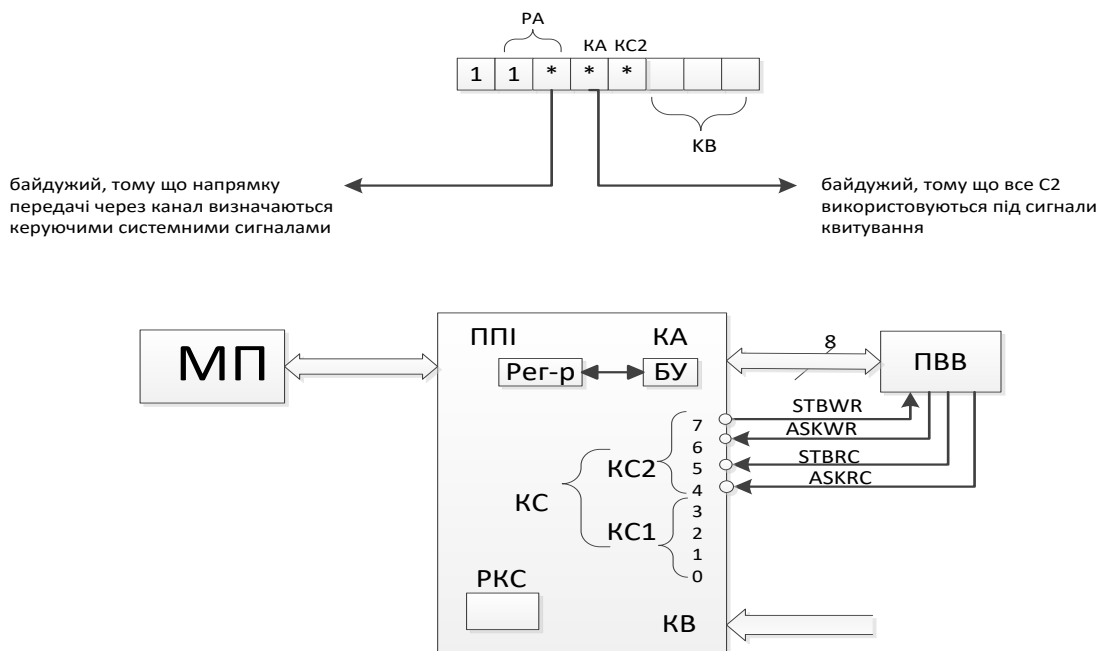


Рисунок 3.5 – Конфігурація виводів ППІ при роботі каналу А у режимі «2», робота каналу В і каналу С1, які не визначено

При роботі каналу А у режимі «2» канал В може бути запрограмовано на роботу у режимі «0» або «1».

Приклад вирішення завдання

Запрограмувати ВІС програмованого паралельного інтерфейсу КР580ВВ55 (ППІ) на введення інформації через канал А у режимі «0» та на виведення інформації через канал В у режимі «1». Зобразити конфігурацію виводів ППІ після програмування. Вільні виводи каналу С (КС) налаштувати на виведення інформації.

Рішення

1 Аналіз роботи каналів ВІС КР580ВВ55.

Канал А працює у режимі «0» на введення інформації. Режим «0» - це синхронна програмно - керована передача даних через канал А без сигналів квитирування (управління, керування).

Тому у керуючому слові:

- $D6D5=00$, бо КА працює у режимі «0»;
- $D4=1$, бо через КА організується введення інформації.

Канал В слід налаштувати на виведення інформації у режимі «1». Режим «1» - це асинхронне однонаправлене введення/виведення інформації через канал В під керуванням трьох сигналів квитирування, переданих через канал С1 (КС1).

Тому у керуючому слові:

- $D2=1$, бо КВ працює у режимі «1»;
- $D1=0$, бо через КВ організується виведення інформації.

Канал С працює у режимі «0» на виведення інформації. Режим «0» - це синхронна програмно - керована передача даних через канал С без сигналів квитирування (управління, керування).

Тому у керуючому слові:

- $D3=0$, бо КС2 працює на виведення інформації;
- $D0=0$, бо КС1 працює на виведення інформації.

2 Формування керуючого слова (КС).

D7	D6	D5	D4	D3	D2	D1	D0
	Режим роботи каналу А «0»		КА – введення	КС2 – виведення	Режим КВ «1»	КВ – виведення	КС1 виведення
1	0	0	1	0	1	0	0

$$КС = 10010100_2 = 94_{16}.$$

3 Конфігурація виводів ППІ.

Конфігурацію виводів ППІ для даного прикладу приведено на рисунку 3.5.

Пристрій виводу ПВВ3 підключається до КВ. Канал В налаштований на виведення інформації у режимі «1». Виведення у режимі «1» супроводжується трьома (3) сигналами квитирування, що передаються по каналу С1 (КС1):

- «STB WR по KB» - від ППІ (вивід C1) до ПБВ3;
- «ASK WR по KB» - від ПБВ3 до ППІ на вивід C2;
- «IRQ по KB» - від ППІ (вивід C0) до мікропроцесора (МП).

Вільні виводи каналу C(C7, C6, C5, C4 и C3) налаштовано на вивід, тому до них підключається пристрій виведення інформації ПБВ2 (рисунок 3.6).

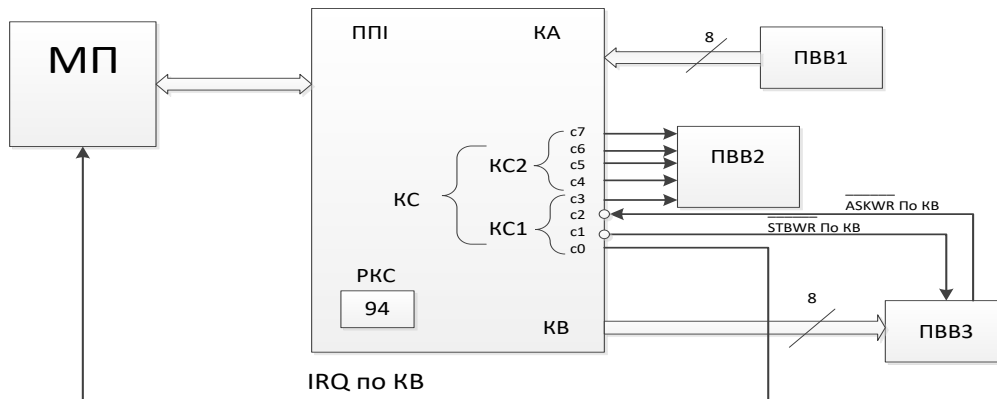


Рисунок 3.6 – Конфігурація виводів ВІС програмованого паралельного інтерфейсу (ППІ)

Завдання 3

Вивчення функціонування та програмування ВІС КР580ВВ55

Згідно з номером у журналі групи обрати у таблиці 3.1 варіант індивідуального завдання. Необхідно задати режим роботи ППІ за допомогою керуючого слова. Зобразити конфігурацію виводів ППІ після програмування.

Таблиця 3.1 – Індивідуальні завдання

Но- мер	Задати режим роботи ППІ за допомогою керуючого слова:					
	Канал А		Канал В		Канал С	
	Ввід у режимі	Вивід у режимі	Ввід у режимі	Вивід у режимі	Ввід у режимі	Вивід у режимі
1	1		1			+
2		1		1	+	
3	1			1	+	
4		1	1			+
5	2		0			+
6	2			0	+	
7	2			1	+	
8	2		1			+
9	1			0	+	
10		1	0			+
11	1		1		+	
12		1		1		+
13	1			1		+
14		1	1		+	
15	2		0		+	
16	2			0		+
17	2			1		+
18	2		1		+	
19	1			0		+
20		1	0		+	

Практична робота 4

Програмування мікропроцесора КР580ВМ80. Складання простих програм і викладення їх на мові Assembler

Мета роботи - ознайомлення з типами, форматами команд та етапами їх виконання, вивчення загальних команд даних, дослідження простих програм на мові Assembler.

Всі команди мікропроцесора наведено у додатку Е. Сутність кожної команди зазначено після крапки з комою. Умовні позначення розшифровано в нижній частині таблиці в розділі «Умовні позначки». Коди команд наведено у додатку Е.

Лінійні програми характеризуються послідовним записом виконання команд, що вміщуються у цих програмах.

Приклад вирішення завдання

Приклад 1. Скласти програму додавання двох шістнадцяткових чисел 4E і 2F. Результат розмістити в комірці пам'яті з адресою 23FEH.

Оскільки операнди задані в явному вигляді, необхідно помістити їх в операційні реєстри загального призначення. При цьому один з операндів перед виконанням додавання обов'язково повинен бути розташованим в акумуляторі А. Результат додавання також буде в акумуляторі. Користуючись командою мікропроцесора виконуємо пересилання вмісту акумулятора в комірку пам'яті з адресою 23FEH.

З урахуванням цього програма прийме вигляд:

MVI A, 4E; – завантаження числа 4E у реєстр А;

MVI C, 2F; – завантаження числа 2F у реєстр С;

ADD C; – додавання вмісту реєстра С до вмісту реєстра А, результат у реєстрі А;

STA 23FE; – пересилання вмісту реєстра А в комірку пам'яті з адресою 23FEH;

HLT; – останов.

Користуючись кодами команд, наведених у додатку Е виконуємо асемблювання програми і розмістимо її в пам'яті мікро-ЕОМ, починаючи з адреси 2150H (таблиця 4.1).

Таблиця 4.1 – Програма на мові Assembler

Адреса	Код операції	Команда	Коментар
2150	3E	MVI A, 4E	завантаження числа 4E у реєстр А
2151	4E		
2152	0E	MVI C, 2F	завантаження числа 2F у реєстр С
2153	2F		
2154	81	ADD C	додавання вмістів реєстрів С і А
2155	32	STA 23FE	запис результату в комірку пам'яті з адресою 23FEH
2156	FE		
2157	23		
2158	76	HLT	останов

Приклад 2. Скласти вміст двох комірок пам'яті з адресами 2200H і 2210H, з отриманого числа відняти число 1AH, результат проінвертувати і додати до нього одиницю, після чого результат переслати в комірку пам'яті з адресою 2300.

Вихідні дані: в комірку 2200H за допомогою команди S завантажити число 4DH, а в комірку 2210H – число 4AH.

1-й варіант вирішення завдання:

LDA 2200 – завантаження вмісту комірки пам'яті з адресою 2200H в акумулятор;

MOV B,A – пересилання його в регістр B;

LDA 2210 – завантаження вмісту комірки пам'яті з адресою 2210H в акумулятор;

ADD – додавання до вмісту акумулятора вмісту регістра B;

CMA – інвертування акумулятора (заміна одиниць нулями, а нулів одиницями);

ADI 01 – додавання одиниці до вмісту акумулятора;

STA 2300 – пересилання результату в комірку з адресою 2300H;

HLT – зупинення роботи.

2-й варіант вирішення завдання (з використанням адресації через реєстрову пару HL):

LXI H, 2200 – завантаження в реєстрову пару HL адреси першого операнда;

LDA 2210 – завантаження в акумулятор другого операнда;

ADD M – додавання до вмісту акумулятора вмісту комірки пам'яті, адреса якої знаходиться в реєстровій парі HL (тобто 2200H);

SUI 1A – вирахування із вмісту акумулятора числа 1AH;

CM Aінвертування акумулятора;

INR A – інкремент (збільшення на одиницю) вмісту акумулятора;

STA 2300 – пересилання результату в комірку пам'яті з адресою 2300H;

HLT – зупинення роботи.

Завдання 4

Програмування мікропроцесора KP580BM80. Складання простих програм і викладення їх на мові Assembler

Згідно з номером у журналі групи обрати у таблиці 4.1 варіант індивідуального завдання. Необхідно скласти програму, виконати її на мові Assembler та подати у вигляді таблиці. Виконати вручну розрахунки завдання у двійковій формі. Результат перевести в 16-кову форму. Вміст комірок пам'яті, що використовуються у програмі, приймати довільно.

Таблиця 4.1 – Індивідуальні завдання

Варіант	Індивідуальне завдання
1	2
1	Вміст комірки пам'яті 2200H скласти з числом 2FH і отриманий результат відняти від вмісту комірки 2201H
2	Скласти вміст комірок пам'яті 2200H і 2201H та від отриманого результату відняти число 1BH
3	Від вмісту комірки 2200H відняти число 01 і до отриманого результату додати одиницю
4	Проінвертувати вміст комірки 2200H і до отриманого результату додати одиницю, після чого отримане число скласти з вмістом комірки 2201H
5	Скласти число 4FH із проінвертованим вмістом комірки пам'яті 2200H і від отриманої суми відняти одиницю
6	Проінвертувати вміст комірок пам'яті 2200H і 2201H після чого скласти їх вміст
7	Від вмісту комірки пам'яті 2200H відняти вміст комірки 2201H, результат проінвертувати
8	Вміст комірки 2200H переслати в комірку 2201H, потім проінвертувати суму вмісту цих комірок
9	Скласти два числа, розташовані в комірках 2200H і 2201H, до результату додати число 03
10	Проінвертувати вміст комірки 2201H, додати до результату 1 і скласти результат з числом 0H

Продовження таблиці 4.1

1	2
11	До вмісту комірки з адресою 2200H додати одиницю, проінвертувати результат і відняти від нього число 1BH
12	Скласти проінвертований вміст комірок пам'яті 2200H і 2201H, до результату додати число 1CH
13	Від вмісту комірки пам'яті 2200H відняти одиницю, результат проінвертувати і скласти з числом 2DH
14	До проінвертованого вмісту комірки 2200H додати 1 і від результату відняти вміст комірки 2201H
15	Від числа FFH відняти проінвертований вміст комірки 2201H, результат проінвертувати
16	Вміст комірки пам'яті з адресою 2200H скласти з різницею вмісту комірки пам'яті з адресою 2201H і числа 41H, отриманий результат проінвертувати
17	Скласти вміст комірки пам'яті з адресами 2200H і 2201H, від отриманого результату відняти число 1BH
18	Скласти два числа, розташованих у комірках пам'яті з адресами 2200H і 2201H, відняти від результату число 0AH, після чого проінвертувати отриманий результат
19	Від числа FFH відняти число 05, від результату відняти проінвертований вміст комірки 2201H
20	Від суми числа OFH і проінвертованого вмісту комірки 2201H відняти число 4EH

Список літератури

1 Мікропроцесорна техніка: Підручник / Ю.І. Якименко, Т.О. Терещенко, Є.І. Сокол [та ін.] за ред. Т.О. Терещенко. – К.: Видавництво "Політехнік", 2003. – 440 с.

2 Дибкова Л.М. Інформатика та комп'ютерна техніка: Посібник. – К.: Видавничий центр "Академія", 2002. – 692 с.

3 Руденко В.Д., Макарчук О.М., Патланжоглу М.О. Практичний курс інформатики / За ред. Мадзігона В.М. – К.: Фенікс, 1999. – 304 с.

4 Андреева Е.В. Системы исчисления и компьютерная арифметика / Е.В. Андреева, И.Н. Фалина. – М.: Лаборатория Базовых Знаний, 2000. – 248 с.

5 Фатеева Н.М. Арифметические и логические основы компьютера: Учеб.-метод. указания / Н.М. Фатеева, О.А. Возилкина, Н. В. Тумбаева. – Барнаул: Изд-во АГАУ, 2008. – 53 с.

ДОДАТКИ

Додаток А

Таблиця А.1 – Перетворення у позиційних системах числення

1	10→2 $\begin{array}{r l} 37 & 2 \\ \hline 36 & \boxed{18} \quad 2 \\ \underline{1} & \boxed{18} \quad \boxed{9} \quad 2 \\ & \quad \underline{0} \quad \boxed{8} \quad \boxed{4} \quad 2 \\ & & \quad \underline{1} \quad \boxed{4} \quad \boxed{2} \quad 2 \\ & & & \underline{0} \quad \boxed{2} \quad \underline{1} \\ & & & & \underline{0} \end{array}$ $37_{(10)}=100101_{(2)}$	5	2→10 $\begin{matrix} 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{matrix}_{(2)}=1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 37$
		6	2→16 $\underbrace{100}_{2} \underbrace{101}_{5}_{(2)} = 25_{(16)}$
2	10→8 $\begin{array}{r l} 37 & 8 \\ \hline 32 & \underline{4} \\ \hline & 5 \end{array}$ $37_{(10)}=45_{(8)}$	7	8→2 $\begin{matrix} \swarrow & \searrow \\ 41_{(8)} = 100101_{(2)} \\ 100 & 101 \end{matrix}$
		8	16→2 $\begin{matrix} \swarrow & \searrow \\ 25_{(16)} = 00100101_{(2)} \\ 0010 & 0101 \end{matrix}$
3	10→16 $\begin{array}{r l} 37 & 16 \\ \hline 32 & \underline{2} \\ \hline & 5 \end{array}$ $37_{(10)}=25_{(16)}$	9	8→10 $\begin{matrix} 1 & 0 \\ 4 & 5 \end{matrix}_{(8)} = 4 \cdot 8^1 + 5 \cdot 8^0 = 37_{(10)}$
			16→10 $\begin{matrix} 1 & 0 \\ 2 & 5 \end{matrix}_{(16)} = 2 \cdot 16^1 + 5 \cdot 16^0 = 37_{(10)}$
4	2→8 $\underbrace{100}_{4} \underbrace{101}_{5}_{(2)} = 45_{(8)}$		16→8 $25_{(16)} = 00100101$ $\begin{matrix} \swarrow & \searrow \\ 25_{(16)} = 00100101_{(2)} = 45_{(8)} \\ 0010 & 0101 & 0 & 4 & 5 \end{matrix}$
			8→16 $\begin{matrix} \swarrow & \searrow \\ 41_{(8)} = 100101_{(2)} = 25_{(16)} \\ 100 & 101 & 2 & 5 \end{matrix}$

Додаток Б

Таблиця Б.1 – Перетворення позиційних систем числення

Десяткова система числення	Двійкова система числення	Вісімкова система числення	Шістнадцяткова система числення
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	<i>A</i>
11	1011	13	<i>B</i>
12	1100	14	<i>C</i>
13	1101	15	<i>D</i>
14	1110	16	<i>E</i>
15	1111	17	<i>F</i>

Додаток В

Таблиця В.1 – Додавання у вісімковій системі числення

+	0	1	2	3	4	5	6	7	10
0	0	1	2	3	4	5	6	7	10
1	1	2	3	4	5	6	7	10	11
2	2	3	4	5	6	7	10	11	12
3	3	4	5	6	7	10	11	12	13
4	4	5	6	7	10	11	12	13	14
5	5	6	7	10	11	12	13	14	15
6	6	7	10	11	12	13	14	15	16
7	7	10	11	12	13	14	15	16	17
10	10	11	12	13	14	15	16	17	20

Таблиця В.2 – Додавання у шістнадцятковій системі числення

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Таблиця В.3 – Множення у вісімковій системі числення

·	0	1	2	3	4	5	6	7	10
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	10
2	0	2	4	6	10	12	14	16	20
3	0	3	6	11	14	17	22	25	30
4	0	4	10	14	20	24	30	34	40
5	0	5	12	17	24	31	36	43	50
6	0	6	14	22	30	36	44	52	60
7	0	7	16	25	34	43	52	61	70
10	0	10	20	30	40	50	60	70	100

Таблиця В.4 – Множення у шістнадцятковій системі числення

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Додаток Г

Таблиця Г.1 – Степені числа 2 у десятковій системі числення

$2^0 = 1$	$2^7 = 128$	$2^{14} = 16\,384$
$2^1 = 2$	$2^8 = 256$	$2^{15} = 32\,768$
$2^2 = 4$	$2^9 = 512$	$2^{16} = 65\,536$
$2^3 = 8$	$2^{10} = 1\,024$	$2^{17} = 131\,072$
$2^4 = 16$	$2^{11} = 2\,048$	$2^{18} = 262\,144$
$2^5 = 32$	$2^{12} = 4\,096$	$2^{19} = 524\,288$
$2^6 = 64$	$2^{13} = 8\,192$	$2^{20} = 1\,048\,576$

Таблиця Г.2 – Степені числа 16 у шістнадцятковій системі числення

n (ступінь)	0	1	2	3	4	5	6
16^n	1	16	256	4096	65536	1048576	16777216

Таблиця Г.3 – Степені числа 8 у вісімковій системі числення

n (ступінь)	0	1	2	3	4	5	6
8^n	1	8	64	512	4096	32768	262144

Додаток Д

Таблиця Д.1 – Система команд мікропроцесора

Однобайтові пересилання		Двобайтові пересилання	
MOV R1,R;	R→R1	SPHL;	HL→SP
MVI R,D8;	D8→R	LXI YZ,D16;	D16→YZ
STA adr;	A→M(adr)	SHLD adr;	HL→M(adr); M(adr+1)
LDA adr;	M(adr)→A	LHLD adr;	M(adr)→L, M(adr+1)→H
STA YZ ⁺ ;	A→M(YZ ⁺)	PUSH YZ ⁺⁺ ;	YZ ⁺⁺ →M(SP-1); M(SP-2)
LDAX YZ ⁺ ;	M(YZ ⁺)→A	PUSH PSW;	A→M(SP-1); F→M(SP-2); SP-2→SP
		POP YZ ⁺⁺ ;	M(SP), M(SP+1)→YZ ⁺⁺
		POP PSW;	M(SP)→F; M(SP+1)→A; SP+2→SP
Команди вводу та виводу		Обмін байтами	
IN N	(N)→A	XCHG;	HL↔DE
OUT N;	A→(N)	XTHL;	H↔M(SP+1),L↔M(SP)
Арифметичні і логічні операції з одним операндом			
CMC"; C→C	CMA; $\bar{A} \rightarrow A$	INR""R; R+1→R	INX YZ; YZ+1→YZ
CTS"; 1→C	DAA'; Десяткова корекція	DCR""R; R-1→R	DCX YZ; YZ-1→YZ
Арифметичні і логічні операції з двома операндами			
ADD' R; A+R→A	ADI' D8; A+D8→A	CPI' D8; Встановлення ознак відпо- CMP' R; відно до A-D8 або A-R	
ADC' R; A+(R+C)→A	ACI' D8; A+(D8+C)→A		
SUB' R; A-R→A	SUI' D8; A-D8→A	16-бітові операції	
SBB' R; A-(R+C)→A	SBI' D8; A-(D8+C)→A		
ANA' R; A∩R→A	ANI' D8; A∩D8→A	DAD" YZ; HL+YZ→HL	
ORA' R; A∪R→A	ORI' D8; A∪D8→A		
XRA' R; A⊕R→A	XRI' D8; A⊕D8→A		
Команди зсуву вмісту акумулятора		Команди передачі керування	
RLC"; Зсув вліво		PCHL;	HL→PC
RAL"; Зсув вліво через біт ознаки C		JMP adr;	adr→PC
RRC"; Зсув вправо		J-CON adr;	adr→PC
RAR"; Зсув вправо через біт ознаки C			
Спеціальні команди		Команди виклику та повертання з підпрограми	
EI; Дозвіл переривання	CALL adr;	PC→M(SP-1)M(SP-2)	
DI; Заборона переривання	C-CON adr;	adr→PC	
HLT; Останов	RST X;	PC→M(SP-1)M(SP-2)	
NOP; Холоста операція	де X=0,1,...,7	adr→PC	
Формат регістра F		Для кожного X adr відповідно дорівнює 0H; 8H; 10H; 18H; 20H; 28H; 30H; 38H;	
D7 D6 D5 D4 D3 D2 D1 D0	RET;	M(SP)M(SP+1)→PC	
S Z 0 AC 0 H 1 C	R-CON;	SP+2→SP	

Умовні позначки:

- ' – команда діє на всі ознаки регістра F;
- " – команда діє тільки на ознаку наявності перенесення із старшого розряду (C);
- "" – команда діє на всі ознаки регістра F, крім ознаки C;
- R, R1 – вміст регістрів A, B, C, D, E, H, L або комірки пам'яті M(HL);

YZ – вміст регістрової пари BC, DE, HL або регістра SP (YZ у мнемоніці замінюється на B, D, H або SP);

YZ⁺ – вміст регістрової пари BC або DE (YZ⁺ у мнемоніці замінюється на B або D);

YZ⁺⁺ – вміст регістрової пари BC, DE, HL або PSW (YZ⁺⁺ у мнемоніці замінюється на B, D, H або PSW);

SP – вміст вказівника стеку перед виконанням команди;

D8 – 8-розрядний операнд D (вміст другого байта команди);

(N) – вміст порту вводу або виводу з номером N (N=0; 1; ...; 255);

D16 – 16-розрядний операнд (другий та третій байти команди);

adr - 16-розрядна адреса в трибайтовій команді;

M() – вміст комірки пам'яті (адреса вказується у скобках);

-CON – частина мнемоніки команди, яка визначає умови виконання команди (-CON у мнемоніці замінюється на NZ, Z, NC, C, PO, PE, P або M)

