

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний технічний університет
імені Івана Пулюя

Кафедра автоматизації
технологічних
процесів і виробництв

МЕТОДИЧНІ ВКАЗІВКИ

для виконання
курсowego проекту
з дисципліни

«Електроніка і мікропроцесорна техніка»

**“Розробка мікропроцесорної системи керування на основі
мікропроцесорного комплекту КР580 та на основі ОМЕОМ
К1816ВЕ51 (80С51)”**

Тернопіль 2016

Розробка мікропроцесорної системи керування на основі мікропроцесорного комплекту КР580 та на основі ОМЕОМ К1816ВЕ51/ Методичні вказівки для виконання курсового проекту з дисципліни "Електроніка і мікропроцесорна техніка" //Медвідь В.Р, Пісьціо В.П., Микулик П.М. Тернопіль: ТНТУ - 2016. 108 с.

Методичні вказівки для курсового проектування "Проектування мікропроцесорних систем керування на основі мікропроцесорного комплекту КР580 та на базі ОМЕОМ К1816ВЕ51" розроблено у відповідності з навчальною програмою курсу "Електроніка і мікропроцесорна техніка".

Укладачі Медвідь В. Р., доц., к.т.н., Пісьціо В. П., асистент, Микулик П.М., асистент.

Розглянуто й затверджено на засіданні кафедри автоматизації технологічних процесів та виробництв (протокол № ____ від _____ 2016 р.)

ЗМІСТ

I. Проектування мікропроцесорних систем керування на основі мікропроцесорного комплекту KP580 та на базі ОМЕОМ KP1816BE51.	
Технічне завдання	5
Вступ.....	8
1. Проектування мікропроцесорної системи на основі мікропроцесорного комплекту KP580	8
1.2. Проектування системної магістралі мікропроцесорної системи керування на основі МП-комплекту KP580	
1.2.1. Структура системної магістралі на основі МП - комплекту KP580.....	13
1.2.2. Організація пам'яті	13
1.2.2.1. Визначення об'ємів пам'яті	16
1.2.2.2. Карта пам'яті.....	16
1.2.2.3. Дешифрація мікросхем пам'яті	17
1.2.3. Вибір пристрою вводу/виводу	19
1.2.4. Організація клавіатури мікропроцесорної системи	21
1.2.5. Організація пристрою відображення інформації	21
1.3. Технічні засоби для узгодження сигналів мікропроцесорної системи з об'єктами керування	
1.3.1. Аналогові та дискретні давачі	23
1.3.1.1. Давачі (вимірювальні перетворювачі) температури	23
1.3.1.2. Давачі (вимірювальні перетворювачі) тиску і перепаду тисків	24
1.3.1.3. Давачі (вимірювальні перетворювачі) вологості	24
1.3.1.4. Давачі (вимірювальні перетворювачі) витрат	25
1.3.1.5. Давачі (вимірювальні перетворювачі) рівня рідини	25
1.3.1.6. Давачі (вимірювальні перетворювачі) якості повітря	25
1.3.2. Вивід дискретних сигналів	25
1.3.3. Ввід дискретних сигналів	27
1.3.4. Ввід аналогових сигналів	29
1.3.5. Вивід аналогових сигналів	31
1.3.6. Керування кроковим двигуном	33
1.4. Програмування МП KP580BM80A (i8080)	
1.4.1. Програмування роботи ППІ KP580BV55	35
1.4.2. Приклад програми для вводу даних з клавіатурної матриці 5x5	37
1.4.2.1. Процедура сканування клавіатури	37
1.4.2.2. Усунення деренчання контактів	38
1.4.2.3. Чекання звільнення клавіші	38
1.4.2.4. Процедура ідентифікації натисненої клавіші	38
1.4.2.5. Процедура опитування стану клавіатури	38
1.4.3. Опитування двійкового давача. Чекання події	39
1.4.4. Вивід символу на семисегментний індикатор (ССІ)	40
1.5. Розрахунок електричних і часових параметрів МП - пристроїв	
1.5.1. Розрахунок фільтрів по живленню	41
1.5.2. Розрахунок споживаної потужності та деяких елементів схеми	41
1.5.3. Розрахунок блоку живлення	42
1.5.3.1. Розрахунок випрямляча	43
2. Проектування системи керування на основі омеом K1816BE51 (80C51)	
2.1. Однокристальні мікро-ЕОМ сімейства МК51 (MCS51). Основні характеристики	45
2.2. Архітектура ОМЕОМ 80C51	45

2.3. Використання зовнішньої пам'яті і розширеного вводу/виводу	49
2.3.1. МК - системи з зовнішньою пам'яттю програм	50
2.3.2. МК-система з розширеним вводом/виводом	52
2.3.3. Під'єднання клавіатури та дисплею	54
2.3.4. Під'єднання ССІ до МК51	55
2.4. Приклад побудови практичних схем на основі МК51	
2.4.1. Вимірювання температури термopарою	56
2.4.2. Вимірювання компонентів електричних схем через вимірювання частоти	57
2.4.3. Керування швидкістю обертання двигуна постійного струму	58
2.4.4. Ввід даних від цифрових здавачів	59
2.4.5. Перетворення паралельного коду в послідовний	59
2.4.6. Вимірювання аналогового сигналу з використанням МК51	60
2.4.7. Вимірювання аналогових сигналів від декількох джерел	61
2.4.8. Формування вихідного аналогового сигналу пристроєм на основі МК51	61
2.5. Програмування контролера МК51	
2.5.1. Ввід інформації з клавіатури	62
2.5.2. Опитування двійкового давача. Чекання події	63
2.5.3. Опитування імпульсного давача	63
2.5.4. Усунення "деренчання" контакту	63
2.5.5. Підрахунок кількості імпульсів	63
2.5.6. Програмна реалізація процедури чекання заданого коду	64
2.5.7. Організація часової затримки тривалістю 50 мс в МК51	64
2.5.8. Передача і прийом даних через послідовний порт	65
2.5.9. Передача символного рядка через послідовний порт	66
2.5.10. Реалізація клавіатури (ще один варіант)	66
II. Проектування електричних схем адресних дешифраторів на основі елементів логіки	
Технічне завдання	68
1. Вступ	68
2. Проектування адресних дешифраторів на основі елементів логіки	69
ЛІТЕРАТУРА	76
Додаток 1	78
Додаток 2	99

I. ПРОЕКТУВАННЯ МІКРОПРОЦЕСОРНИХ СИСТЕМ КЕРУВАННЯ НА ОСНОВІ МІКРОПРОЦЕСОРНОГО КОМПЛЕКТУ КР580 ТА НА БАЗІ ОМЕОМ КР1816ВЕ51

Технічне завдання

Метою курсового проекту є поглиблення знань апаратних принципів побудови пристроїв мікропроцесорної техніки та набуття практичних навичок з розробки мікропроцесорних систем та їх програмного забезпечення.

Вхідні дані

Завданням курсового проекту є розробка мікропроцесорного пристрою, що включає:

- 8 - розрядний мікропроцесор або мікроконтролер відповідно до завдання,
- зовнішні ОЗП та ПЗП,
- паралельні інтерфейси вводу/виводу,
- пристрій вводу даних (клавіатура),
- пристрій індикації (n-розрядне цифрове табло на основі семисегментних світлодіодних індикаторів),
- аналогові та цифрові давачі,
- виконавчі пристрої.

Елементна база для виконання проекту вказується в табл. 1 відповідно до заданого викладачем варіанту.

Пояснювальна записка повинна містити:

- 1) структурну схему пристрою з поясненнями;
- 2) карту розподілу адресного простору пристрою;
- 3) обґрунтований вибір апаратних засобів, у тому числі ВІС пам'яті і периферійних модулів, та їхній опис;
- 4) функціональну схему блоку пам'яті, розподіл адресного простору та синтез адресного селектора для підключення зовнішньої пам'яті;
- 5) функціональну схему блоку вводу аналогових та цифрових даних;
- 6) функціональну схему під'єднання виконавчих пристроїв;
- 7) функціональну схему блоку індикації на основі семисегментних індикаторів;
- 8) функціональну схему блоку інтерфейсу між процесором, блоком вводу аналогових та цифрових даних і блоком індикації;
- 9) фрагменти програмного забезпечення, що ініціалізує розроблений пристрій і здійснює типові операції вводу-виводу (читання клавіатури і портів вводу, вивід інформації на індикатори і в порти виводу) з поясненнями (використовувані режими роботи периферійних ВІС, послідовність програмування і т.і.).

Адресація має відповідати розподілу адресного простору розробленого пристрою;

- 10) розрахунок споживаної пристроєм потужності;
- 11) розрахунок стабілізованого блоку живлення;
- 12) функціональну та електричну принципову схеми розробленого пристрою, виконану відповідно до вимог конструкторської документації, а також перелік елементів, які входять до складу пристрою.

Таблиця 1

Комплектація (елементна база) на курсове проектування		Номер варіанту завдань на курсове проектування																														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28			
Базовий мікропроцесорний пристрій:																																
- КР580ВМ80А		х	х		х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х		
- КР1816ВЕ51				х							х							х								х						
Об'єм, організація пам'яті:																																
- оперативної статичної (ОЗП)		2Кх8	2Кх8	1Кх8	3Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	2Кх8	2Кх8	1Кх8	2Кх8	4Кх8	2Кх8	4Кх8	4Кх8	4Кх8	4Кх8	2Кх8	1Кх8	2Кх8	2Кх8	1Кх8	1Кх8	4Кх8	2Кх8	4Кх8	2Кх8		
- постійної (ПЗП)		2Кх8	2Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8	1Кх8		
Пристрої вводу-виводу:																																
- КР580ВВ55		х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х		
- КР580ВВ51																																
Клавіатура з організацією матриці клавіш:		2х8	4х4	5х5	2х6	4х6	3х4	2х8	4х4	5х5	2х6	4х6	3х4	2х8	4х4	5х5	2х6	4х6	3х4	2х6	4х6	4х6	3х4	2х8	4х4	5х5	2х6	4х6	3х4	2х6		
Дисплей динамічний лінійний розрядністю:		3	4	5	4	5	3	3	4	4	5	4	4	5	4	4	5	4	5	5	4	4	6	5	4	3	4	3	4	5	4	
Давачі аналогові:																																
- терморезистор		х	х		х		х	х		х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	
- фоторезистор		х																														
Давачі цифрові:																																
- кінцевий,		х	х	х	х			х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	
- фотореле		х	х					х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	х	

Вступ

Мікропроцесорна система, яка представляє собою систему обробки інформації і керування, реалізовану на основі сучасної елементної бази і принципу програмного керування, складається з наступних пристроїв:

- процесора (МП), що виконує основні функції керування й обробки інформації;
- запам'ятовуючого пристрою (ЗП), призначеного для зберігання даних і програм,
- пристроїв вводу/виводу (ПВВ), що забезпечують взаємозв'язок із зовнішнім середовищем.

Ці пристрої об'єднані між собою внутрішньосистемним інтерфейсом і взаємодіють по адресному принципу - всі підпорядковані пристрої і їх складові частини мають неповторювані адреси.

В **однопроцесорних** системах всі функції обробки інформації та керування виконує один МП. У **багатопроцесорних** системах є кілька рівноправних процесорів або один центральний (головний) процесор, що виконує функції керування, і декілька підлеглих процесорів, орієнтованих на вирішення спеціалізованих завдань обробки інформації.

Мікропроцесор мікро-ЕОМ обробляє інформацію трьох типів - **дані, адреси й команди програми**. Над даними виконуються арифметичні й логічні операції процесором. Обробка адреси визначається способом зберігання і доступу до даних та команд, і також базується на виконанні арифметичних операцій. Обробка команд полягає в перетворенні коду команди в послідовність керуючих впливів (мікрооперацій) відповідно до алгоритму виконання команди. Кожна мікрооперація (або їхня сукупність - **мікрокоманда**) виконується у фіксовані інтервали часу (**такти**), а вся сукупність виконання команди протягом повного циклу (командного **циклу**) утворює мікропрограму. Під управлінням мікропрограми виконується обробка даних та адрес, а також керування іншими пристроями мікропроцесорної системи через внутрішньосистемний інтерфейс.

Для виконання перерахованих функцій процесор містить засоби обробки даних, обробки адреси, зберігання даних і адреси – оперативний запам'ятовуючий пристрій, засоби мікропрограмного керування, засоби синхронізації і керування режимами роботи процесора.

1. ПРЯКТУВАННЯ МІКРОПРОЦЕСОРНОЇ СИСТЕМИ НА ОСНОВІ МІКРОПРОЦЕСОРНОГО КОМПЛЕКТУ КР580

Мікропроцесорний комплект КР580 призначений для побудови 8-розрядних мікропроцесорних систем широкого призначення.

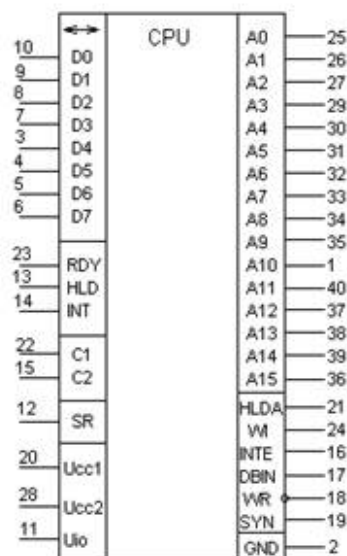


Рис. 1 Мікропроцесор КР580ВМ80А

Розглянемо мікропроцесорний комплект КР580, починаючи з мікропроцесора КР580ВМ80А, позначення якого зображене на рис. 1.

Мікропроцесор КР580ВМ80А – це 8-розрядний процесор, виконаний по nМОН-технології з джерелами живлення +5, +12, -5 В і споживаним струмом 60, 40 і 0.01 мА відповідно.

Він адресує 64 Кб адресний простір пам'яті і 256-байтний простір вводу/виводу, з частотою синхронізації до 2,5 МГц і мінімальним часом виконання команди 1,6 мкс.

Навантажувальна здатність виходів такого мікропроцесора – один ТТЛ вхід на одну вихідну лінію МП, тому виконують буферизацію шин.

Тактова частота 2 МГц. Фіксований набір команд, який налічує 78 команд.

Функціональне призначення виводів:

Номер	Позначення	Найменування	Назва	Тип	Стан
19	SYNC	Синхронізація	Ознака початку машинного циклу	Вихід	1
17	DBIN	Приєм даних	Ознака зчитування інформації	"-"	1
18	WR	Запис	Ознака запису інформації	"-"	0
23	RDY	Готовність	Сигнал готовності зовнішнього пристрою до обміну інформацією	Вхід	1
24	WI (WAIT)	Очікування	Ознака переходу в стан очікування готовності зовнішнього пристрою	Вихід	1
14	INT	Запит на переривання	Сигнал про необхідність обміну по перериванню	Вхід	1
16	INTE	Дозвіл переривання	Сигнал про готовність до обміну по перериванню	Вихід	1
13	HLD (HOLD)	Запит на захоплення	Сигнал про необхідність обміну за методом прямого доступу до пам'яті (ПДП)	Вхід	1
21	HLDA	Підтвердження захоплення	Ознака переходу до обміну за методом прямого доступу до пам'яті (ПДП)	Вихід	1
12	SR (RESET)	Скидання	Початкове встановлення (ініціалізація) МП	Вхід	1
3-10	D0-D7	Дані	Двонаправлена восьми-розрядна шина даних	Вхід	X
1,25-27 29-40	A0-A15	Адреси	Однонаправлена шістнадцятирозрядна адресна шина	Вихід	X

В данному мікропроцесорному комплекті системним генератором є мікросхема **КР580ГФ24** (рис. 2), виконана по біполярній технології, споживаний струм 15 і 12 мА від джерел +5, +12 В відповідно.

Сигнали синхронізації, які формуються задаючим генератором, стабілізовані кварцом з резонансною частотою до 27 МГц та утворюються подільником частоти з коефіцієнтом 9.

Для роботи генератора тактових імпульсів (ГТІ) передбачено увімкнення зовнішнього кварцового резонатора, а також LC- коливальної системи, які задають частоту генератора.

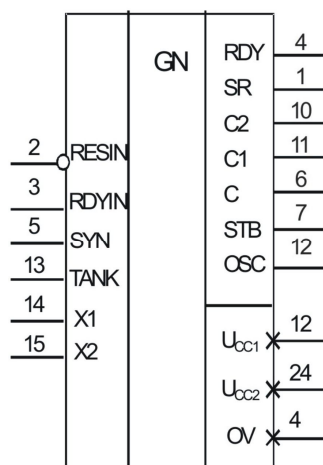


Рис. 2 Мікросхема КР580ГФ24

Призначення виводів мікросхеми:

SR	-	(вихід)	-	скидає в початковий стан МП-систему;
RESIN	-	(вхід)	-	вхід скидання в нуль генератора;
RDYIN	-	(вхід)	-	сигнал готовності;
RDY	-	(вихід)	-	вихідний сигнал готовності;
SYN	-	(вхід)	-	сигнал синхронізації;
C	-	(вихід)	-	тактовий сигнал, синхронізований з фазою сигналу C2;
STB	-	(вихід)	-	стробуючий сигнал стану;
C1, C2	-	(виходи)	-	тактові вихідні сигнали;
OSC	-	(вихід)	-	тактовий сигнал опорної частоти;
TANK	-	(вихід)	-	вивід для підключення коливального контуру;
X1, X2	-	(входи)	-	виводи для підключення кварцового резонатора;
Ucc1	-	(вхід)	-	напруга живлення +5 В;
Ucc2	-	(вхід)	-	напруга живлення +12 В;
GND	-	(вхід)	-	спільний вивід.

Призначення сигналів:

OSC – використовується для одночасної синхронізації декількох генераторів;

STB – формується при наявності на вході SYN високого рівня для занесення слова стану в системний контролер КР580ВК28;

SR – використовується для скидання МП та інших пристроїв, що входять до МП-системи;

TANK – використовується для під'єднання коливального контуру, який працює на вищих гармоніках кварцового резонатора, з метою стабілізації тактових сигналів опорної частоти.

Мікросхеми **КР580ВА86/87** (рис. 3) - двонаправлені шинні формувачі з тристабільними виходами, виконані по біполярній технології, живляться від джерела +5 В і споживають струми 169 / 130 мА відповідно.

Канал А забезпечує струм навантаження 15 мА, Канал ВА забезпечує струм навантаження 32 мА. Напрямок передачі вибирається сигналом на вході Т. Мікросхема КР580ВА86 виконує передачу без інверсії, мікросхема КР580ВА87 – з інверсією.

Мікросхеми можуть використовуватися для організації двонаправленої шини даних.

Призначення виводів мікросхеми.

A0...A7	-	входи даних;
B0...B7	-	виходи даних;
OE	-	вхід дозволу передачі даних;
T	-	вхід напрямку передачі даних;
U_{cc}, GND	-	входи живлення.

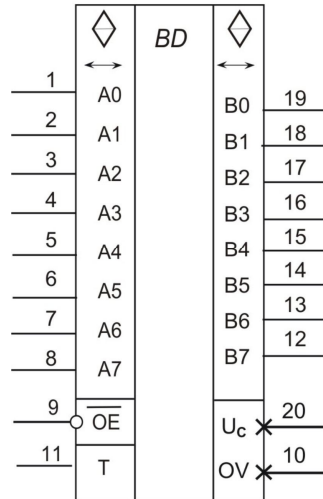


Рис. 3 Мікросхема KP580BA86A

Призначення сигналів.

При $T=0$ забезпечується передача даних з ліній B0...B7 до ліній A0...A7. При $T=1$ забезпечується передача даних з ліній A0...A7 до ліній B0...B7. І в першому, і в другому випадку $OE=0$. При $OE=1$ лінії B0...B7 шинного формувача переводяться в Z-стан.

Мікросхема KP580BA87 має в порівнянні із KP580BA86 інверсні виходи B0...B7 (нумерація виводів обох мікросхем однакова).

Мікросхеми **KP580IP82/83** (буферні регістри) (рис. 4) - це 8-розрядні регістри, виконані по біполярній технології, живляться від джерела +5 В і споживають струм 160 мА. Мікросхеми забезпечують струм навантаження 32 мА і можуть використовуватися у якості шинних формувачів **для побудови адресної шини.**

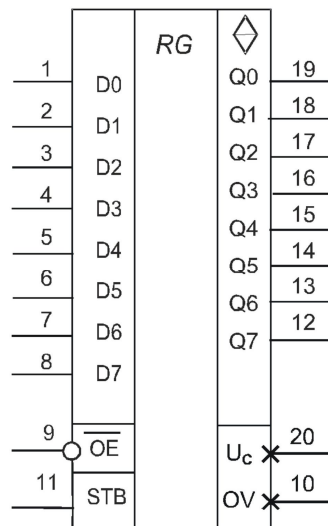


Рис. 4 Мікросхема KP580IP82A

Призначення виводів мікросхеми.

<i>D0...D7</i>	-	входи даних;
<i>Q0...Q7</i>	-	виходи даних;
<i>OE</i>	-	вхід дозволу передачі даних;
<i>STB</i>	-	вхід стробуючого сигналу;
<i>Vcc, GND</i>	-	входи живлення.

Призначення сигналів.

При $OE=0$ в момент подачі стробуючого сигналу на лінію *STB* вхідні дані записуються в регістр і з'являються на його виходах.

При $OE=1$ лінії *Q0...Q7* регістру переводяться в Z-стан.

Мікросхема КР580ИР83 має в порівнянні із КР580ИР82 інверсні виходи *Q0...Q7* (нумерація виводів обох мікросхем однакова).

Системний контролер КР580ВК28 (рис. 5) використовується разом із МП КР580ВМ80А для формування керуючих сигналів обміну *RD, WR, RDIO, WRIO, INTA* та як буферний регістр даних, через який відбувається обмін даними між МП та іншими пристроями.

Призначення виводів мікросхеми.

<i>STB</i>	-	(вхід)	-	стробуючий сигнал стану;
<i>HLDA</i>	-	(вхід)	-	сигнал підтвердження захоплення шини;
<i>TR</i>	-	(вхід)	-	видача інформації;
<i>RC</i>	-	(вхід)	-	прийом інформації;
<i>D0...D7</i>	-	(входи)/(виходи)	-	двонаправлений канал даних;
<i>DB0...DB7</i>	-	(входи)/(виходи)	-	двонаправлений канал даних системи;
<i>BUSEN</i>	-	(вхід)	-	вхід керування передачею даних та формуванням керуючих сигналів;
<i>INTA</i>	-	(вихід)	-	підтвердження запиту переривання;
<i>RD</i>	-	(вихід)	-	зчитування з пам'яті;
<i>RDIO</i>	-	(вихід)	-	зчитування з ПБВ;
<i>WR</i>	-	(вихід)	-	запис в пам'ять;
<i>WRIO</i>	-	(вихід)	-	запис в ПБВ;
<i>Vcc</i>	-	(вхід)	-	напруга +5 В;
<i>GND</i>	-	(вхід)	-	спільний вивід.

Призначення сигналів.

Контролер формує керуючі сигнали в залежності від **слова стану**, яке на початку кожного циклу поступає від МП по каналу даних *D0...D7*.

Лінія *DBIN* мікропроцесора з'єднується з лінією *RC* контролера, а лінія *WR* – з лінією *TR*.

BUSEN - асинхронний сигнал:

- при $BUSEN=0$ контролер передає або приймає дані та формує один з керуючих сигналів;
- при $BUSEN=1$ лінії контролера переводяться в Z-стан.

При роботі з МП КР580ВМ80 в циклі переривання мікропроцесора контролер формує три сигнали *INTA* для прийому трьох байтів команди *CALL* від контролера переривань КР580ВН59.

Під час дії сигналу *RC* буферна схема даних контролера формує єдиний вектор переривань за номером 7 і передає його в МП (код команди *RST7*), якщо в простих мікропроцесорних схемах (коли контролер переривань відсутній) вихід *INTA* під'єднати до напруги 12 В через резистор номіналом 1 КОм.

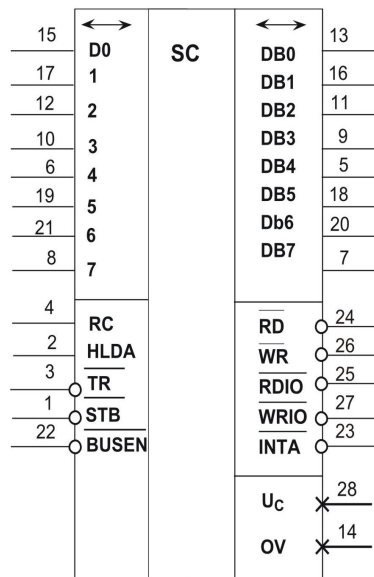


Рис. 5 Системний контролер KP580BK28

1.2. Проектування системної магістралі мікропроцесорної системи керування на основі МП-комплекту KP580

1.2.1. Структура системної магістралі на основі МП - комплекту KP580

Система керування на основі МП KP580BM80A, яка показана на рис. 6, а, включає:

- МП KP580BM80A, робота якого синхронізується імпульсами тактового генератора DD1 KP580ГФ24;
- регістра KP580ИP82, на якому побудована шина керування ШК;
- шинних формувачів KP580BA86, на яких побудовані шини ША та ШД.

Система керування на основі МП KP580BM80A, яка показана на рис. 6, б, включає:

- МП KP580BM80A, робота якого синхронізується імпульсами тактового генератора DD1 KP580ГФ24;
- системного контролера DD 3 KP580BK28, який формує системну шину даних ШД (DAT0...DAT7) та шину керування ШК (IACK, IORC, IOWC, MRDC, MWTC та HACK);
- регістрів DD4, DD5 KP580ИP82, на яких побудована адресна шина ША (ADR0...ADR15).

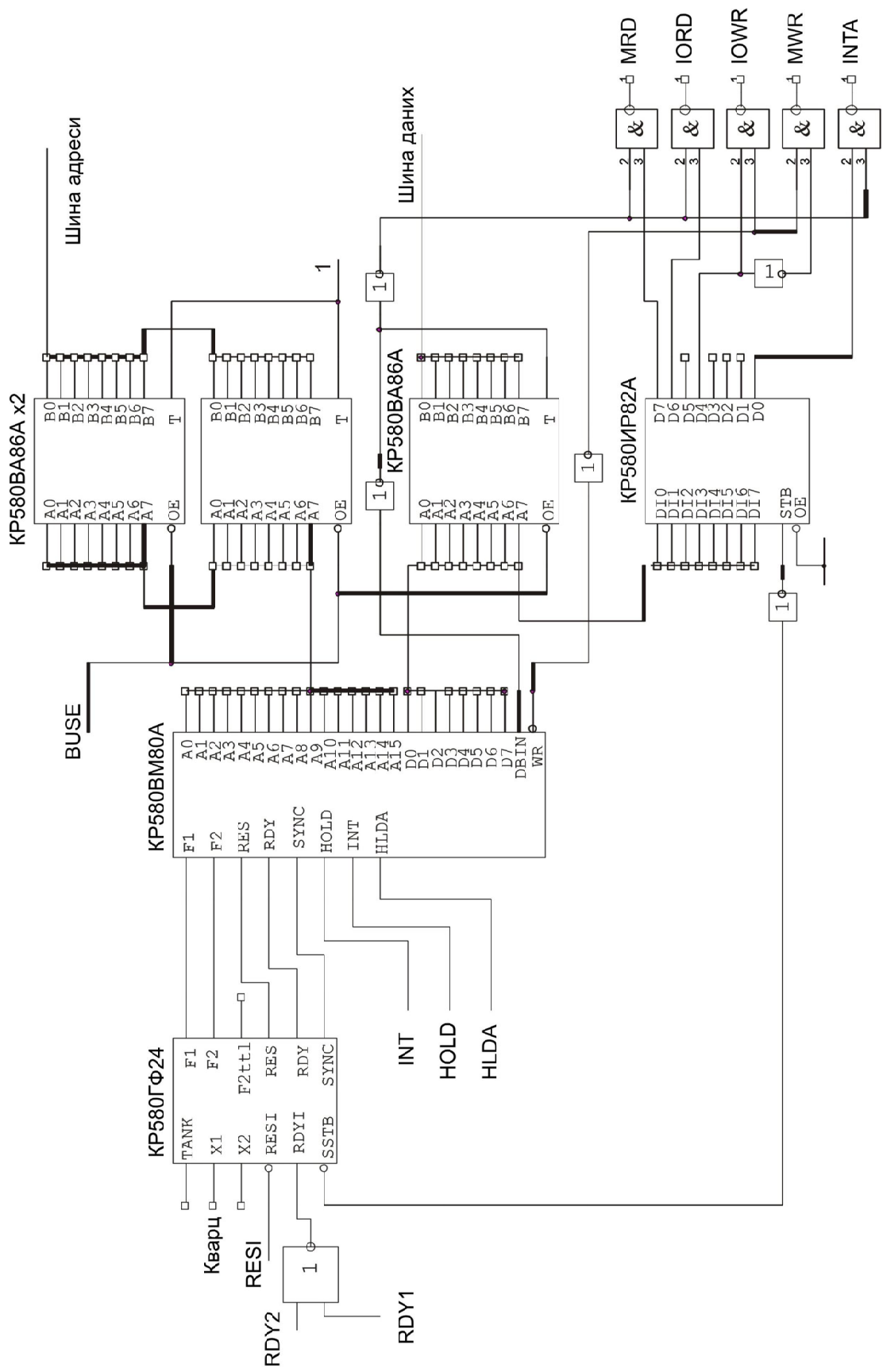
До системної магістралі, в свою чергу, під'єднуються інші пристрої мікропроцесорної системи – постійна та оперативна пам'ять, пристрої вводу-виводу (послідовні та паралельні), контролери переривання та ПДП і т.і.

1.2.2. Організація пам'яті МП системи

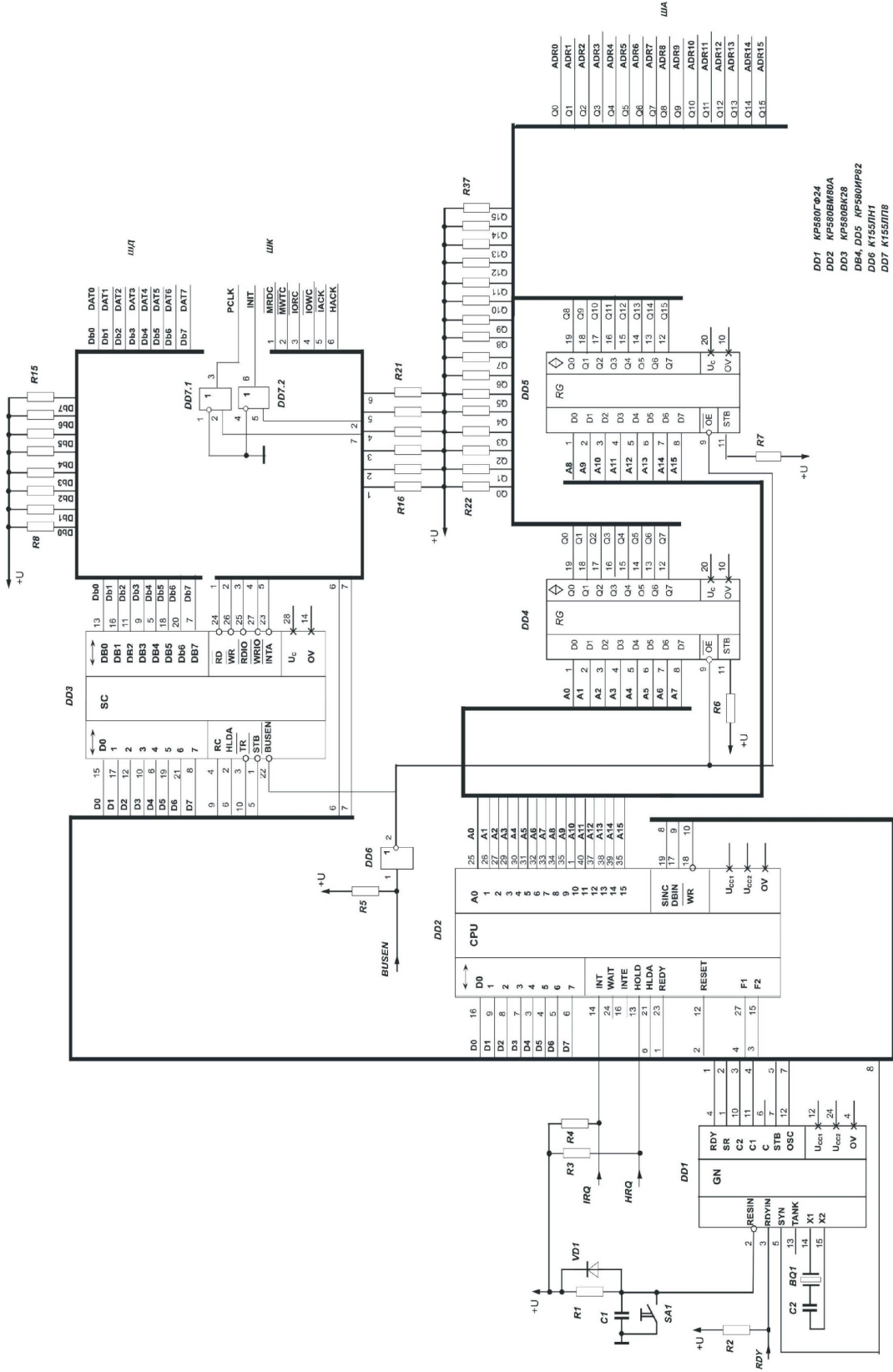
Запам'ятовуючі пристрої в МП-системі (МПС) діляться на:

1. Постійно запам'ятовуючий пристрій (ПЗП) - працює лише у режимі читання. Він використовується для збереження програм і даних, які потрібні для роботи системи і не підлягають зміні.

2. Оперативно запам'ятовуючий пристрій (ОЗП) - працює у двох режимах і зберігає поточну інформацію, яка введена у МПС. Цей пристрій працює у двох режимах, які залежать від виконання програм і введених даних. За допомогою цього пристрою змінюються необхідні параметри даних, записуються поточні результати для їх виводу або подальшої обробки і т.і.



a)



б)

Рис. 6 Структура системної магістралі на МП - комплекті КР580: а) з використанням шинних формувачів та регістра, б) з використанням системного контролера та регістра

1.2.2.1. Визначення об'ємів пам'яті

Вибирається, наприклад, пам'ять об'ємом 2 Кб ПЗП і 2 Кб ОЗП.

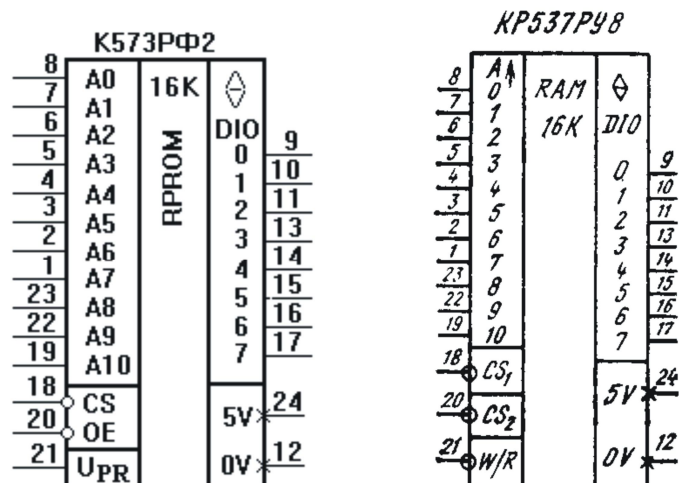


Рис. 7 Мікросхеми KP573PФ2 (ПЗП) і KP537PY8A (ОЗП)

Розподіл пам'яті проводиться з урахуванням того, що основна програма попередньо записується у ПЗП, а тому перше звертання МП здійснюється до нульової комірки ПЗП.

В якості ЗП пристроїв вибираються, наприклад, мікросхеми типу:

- KP573PФ2 (ПЗП): час вибірки становить 450 нс; потужність 580 мВт при напрузі +5В);
- KP537PY8A (ОЗП): час доступу 220 нс; споживана потужність 6 мВт при напрузі живлення 5В (рис. 7).

A0-A10 - вхідні адреси комірок пам'яті, які з'єднуються із системною ША МПС;

D0-D7 - вихідні виводи, на які виставляються дані, які зберігаються у запам'ятовуючому пристрої, з'єднуються з шиною даних МПС;

CS - вхідний вивід, за яким визначається звертання до ПЗП чи ОЗП;

CE - вхідний вивід, який визначає режим роботи запам'ятовуючого пристрою (читання або запис).

Сімейство	Технологія	Тип ВІС	Організація
K537	КМОН	KP537PY8A	2Кx8
K573	пМОН	K573PФ2	2Кx8

1.2.2.2. Карта пам'яті

Оскільки, кожен пристрій пам'яті має 2 Кб, це означає, що кожен пристрій буде займати 2048 адрес. Ці пристрої підключаються до шини адрес паралельно. Їх адресний простір можна забезпечити за допомогою одинадцяти ліній (A0 – A10).

Першу частину адрес відведено для ПЗП, а не для ОЗП, оскільки в перший момент роботи мікропроцесора відбувається його ініціалізація, в ході якої вміст усіх регістрів скидається в нуль, в тому числі і його програмний лічильник (PC), у зв'язку з чим мікропроцесор звертається до першої команди за адресою 0000H.

Адреса	Адреса	№ корпуса	Тип/типономінал
0000000000000000B	0000H	№ 1	ПЗП/K573PФ2
....		
0000011111111111B	07FFH	-	Вільні адреси
0000100000000000B	0800H		
....		
0000111111111111B	0FFFH		

0001000000000000B 0001011111111111B	1000H 17FFH	№ 2	ОЗП/К537РУ10
0001100000000000B 1111111111111111B	1800H FFFFH		

1.2.2.3. Дешифрація мікросхем пам'яті

Ні режим роботи, ні вибір комірки пам'яті не визначають вид ЗП. Тому, у цих пристроях передбачений вивід CS, який визначає активний стан ОЗП або ПЗП.

Для визначення виду ЗП потрібно вибрати лінію старших адресних розрядів МП, наприклад, A12, та з'єднати її з входом CS (активний рівень – логічний «0») мікросхеми пам'яті.

Оскільки для вибору ЗП використовується одна лінія, то для почергового звертання до мікросхем пам'яті між ОЗП і ПЗП ставиться інвертор (рис. 8).

Якщо стан лінії A12 відповідає логічний "0", то звертання відбудеться до ПЗП, а для стану лінії A12 логічна "1" – до ОЗП.

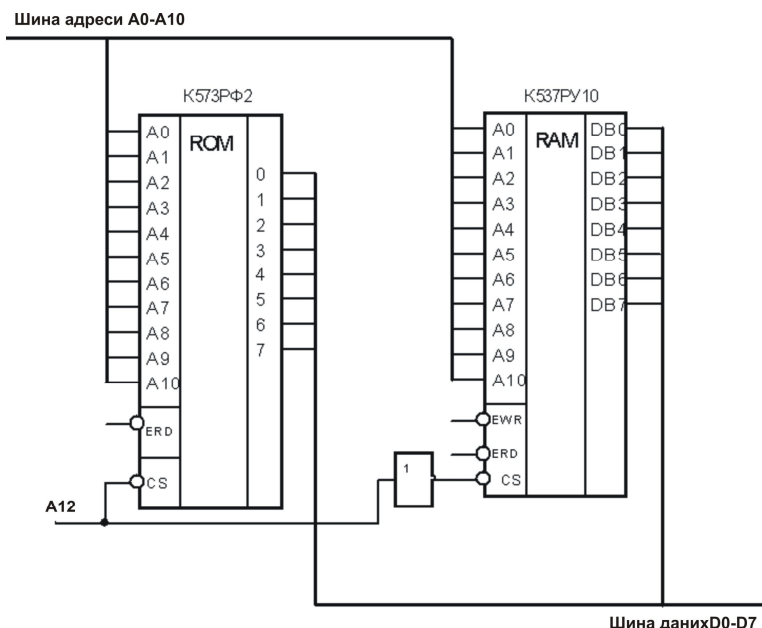


Рис. 8 Підключення мікросхеми КР573РФ2 (ПЗП) і КР537РУ8А (ОЗП)

Для поділу адресного простору мікропроцесора між мікросхемами пам'яті, якщо їх більше, ніж дві, а також між пам'яттю та пристроями вводу-виводу (ПВВ) використовується системний дешифратор адреси ДА (рис. 9) DD3 (К155 ИД7).

В схемах, що містять контролер ПДП, використовують окремо два дешифратори: дешифратор для мікросхем пам'яті та дешифратор ПВВ, причому, останній додатково стробується сигналом дозволу доступу від контролера ПДП.

Типовий блок пам'яті на основі мікросхем статичної оперативної пам'яті К537РУ8А та однократно програмованих мікросхем постійної пам'яті КР556РТ7 показаний на рис. 9. Вибір оперативної або постійної пам'яті забезпечується дешифратором адреси DD3. Обмін даними між пам'яттю та системною шиною даних забезпечується шинним формувачем DD7.

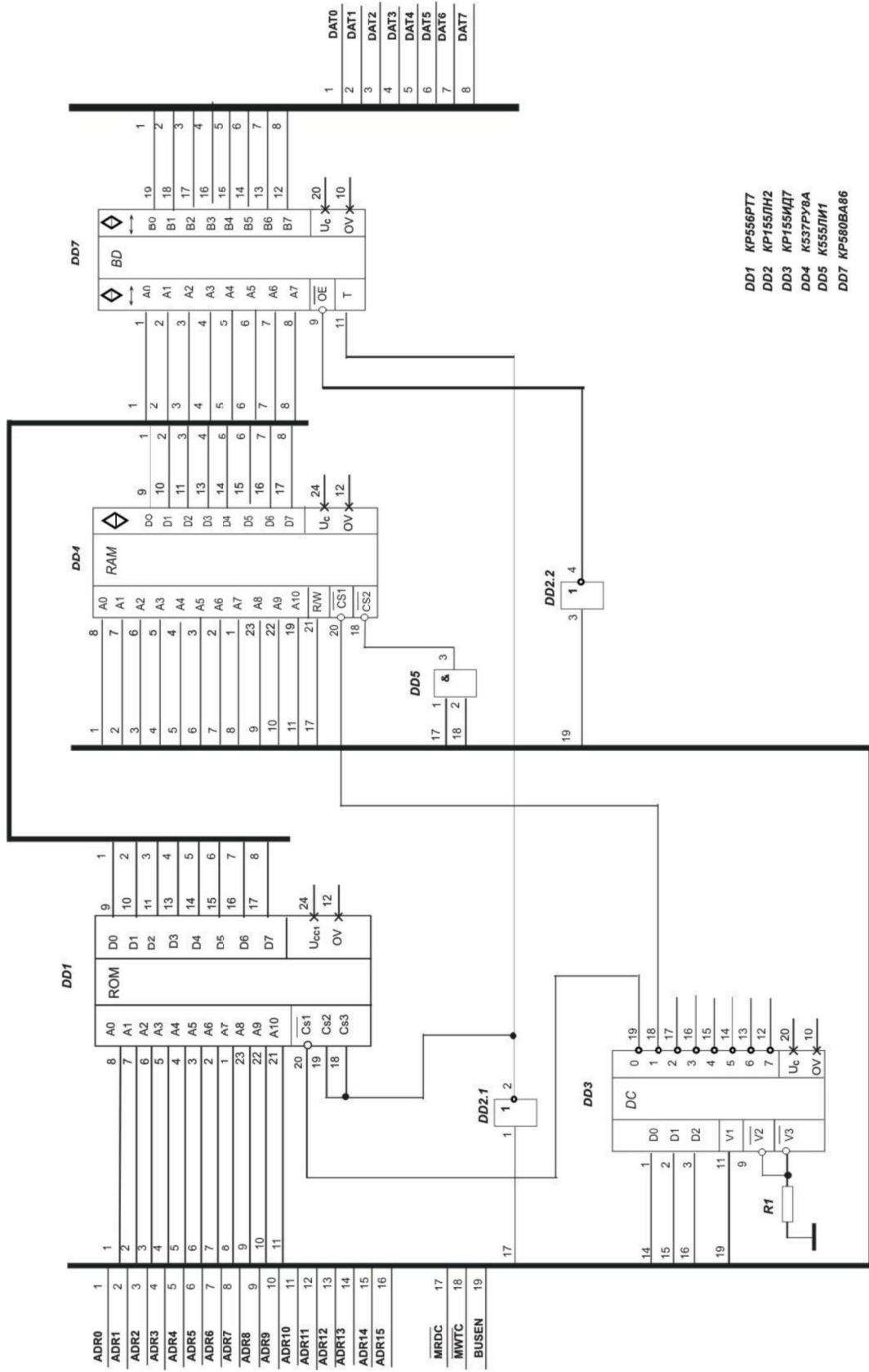


Рис. 9 Організація пам'яті МП - системи

1.2.3. Вибір пристрою вводу/виводу

За допомогою пристрою вводу/виводу у МПС вводиться інформація від зовнішніх пристроїв, а також виводиться на контрольні пристрої.

Для МПС зовнішніми пристроями можуть бути:

- сегментні індикатори;
- клавіші клавіатури;
- ЦАП або АЦП і т.і.

Узгодженість зовнішніх пристроїв із внутрішніми визначається як навантажувальною здатністю пристрою вводу/виводу, так і кількістю вихідних ліній МПС та кількістю вхідних ліній зовнішніх пристроїв.

Програмований паралельний інтерфейс KP580BB55

Складовими частинами інтерфейсу можуть бути шинні формувачі (ШФ), багаторежимні буферні регістри (ББР) або програмовані периферійні адаптери (ППА), наприклад, типу **KP580BB55** (рис. 10).

Графічне позначення виводів ППА типу KP580BB55:

D0 - D7 - вхідні виводи ППА, з'єднуються з системною ШД МП-системи (D0 - молодший розряд);

PA0 - PA7 - вихідна шина даних каналу А;

PB0 - PB7 - вихідна шина даних каналу В;

PC0 - PC3 - вихідна шина даних каналу С (молодші розряди);

PC4 - PC7 - вихідна шина даних каналу С (старші розряди);

CS - вибір пристрою;

A0, A1 - адресні входи, за якими проводиться вибір каналу, і які з'єднуються з молодшими розрядами системної ША;

RD - сигнал, за яким проводиться встановлення даних внутрішнього регістру на ШД;

WR - вхідний сигнал, який дає дозвіл на запис даних у ППА;

SR – скидання, сигнал, по якому ВІС переходить у початковий стан.

Коли на вході CS є сигнал низького рівня, в залежності від того, який з сигналів системного контролера IORD чи IOWR є активним, відбудеться запис в ППА чи зчитування з нього. Комбінація сигналів на входах A0 та A1 визначає канал, з яким відбудеться обмін інформацією.

Перед початком роботи ППА треба запрограмувати, тобто задати режими роботи кожного каналу. Програмування відбувається шляхом одноразового запису в регістр керуючого слова (РКС) відповідного байту.

Вибраний тип ППА має три режими роботи:

режим "0" - звичайний ввід/ вивід;

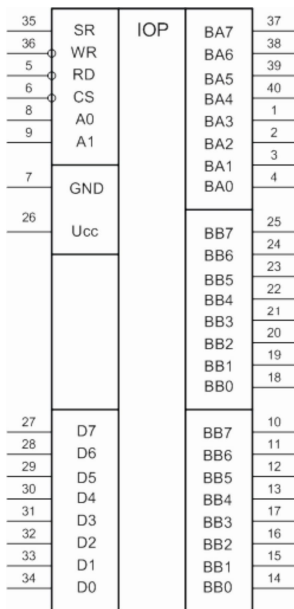
режим "1" - стробований ввід/ вивід;

режим "2" - двонаправлена магістраль.

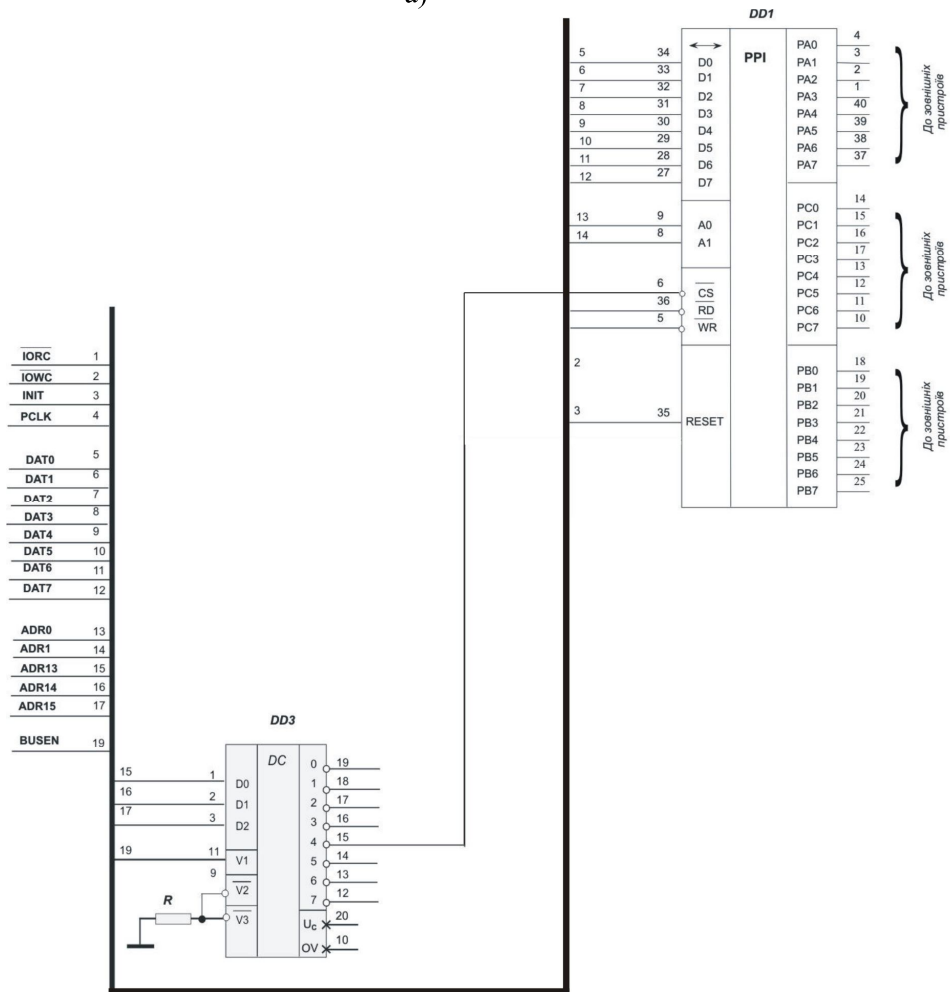
Найбільш зручною є робота ППА в режимі 0, в режимі вводу сигналів, які поволі змінюються. При виводі в цьому режимі на лініях каналів можна програмно формувати різні сигнали, в тому числі керувати роботою індикаторів.

Як правило, канали А, В та молодший півканал С програмуються на вивід в режимі 0, старший півканал С- на ввід.

Оскільки, максимальна навантажувальна здатність виходів K580BB55 становить 2 мА, а струм сегментів знаходиться в межах 10-15 мА, між виходами ППА і виводами індикаторів включають буферні елементи з відкритими колектором – інвертори 155ЛН2, 155ЛН3 або повторювачі 155ЛП9. Опори між виходами інверторів і входами індикаторів обмежують середній струм через сегмент на рівні 10 мА.



а)



б)

Рис. 10 Програмований паралельний адаптер (ППА) KP580BB55 (8255): а) позначення на електричній схемі, б) під'єднання до системної магістралі МП KP580BM80A

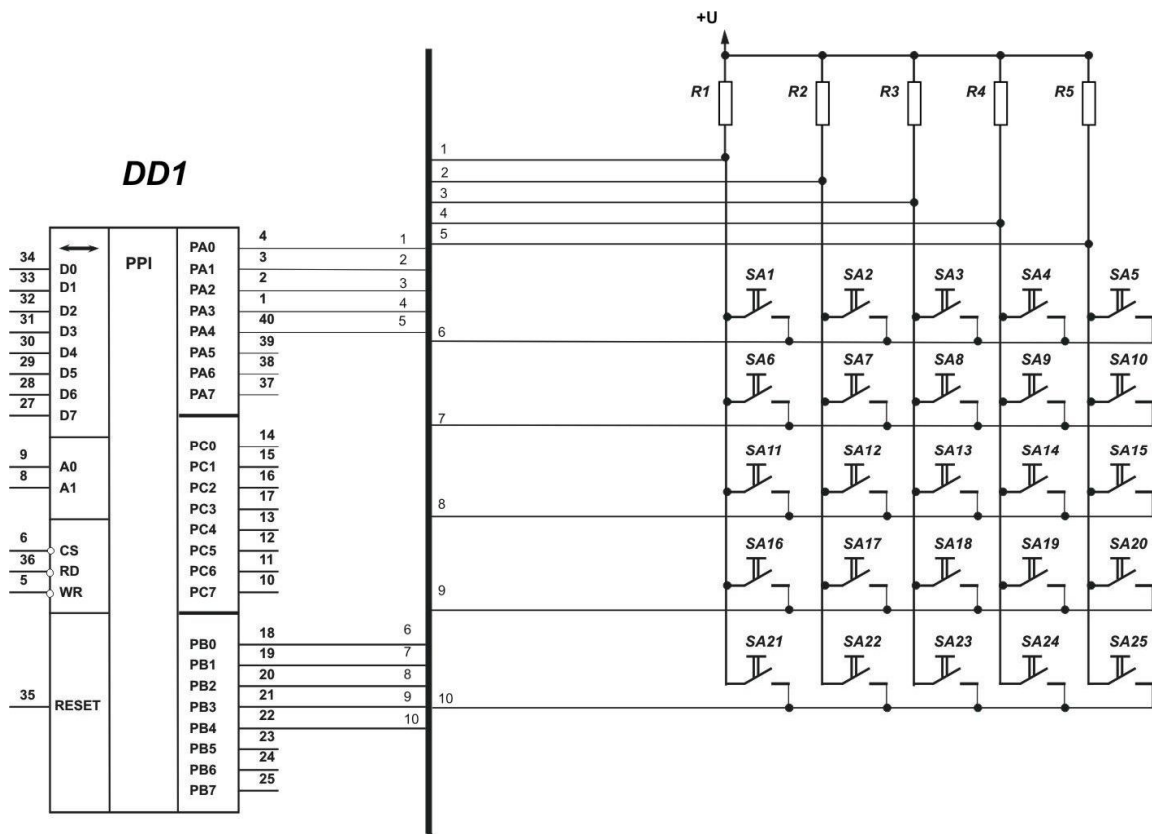
1.2.4. Організація клавіатури мікропроцесорної системи

На рис. 11 показано варіант реалізації клавіатурного пульта вводу інформації на основі матриці ключів 5x5. Ключі клавішних матриць під'єднані безпосередньо до портів А та В паралельного програмованого інтерфейсу KP580BB55 (DD1). Ключі розташовані таким чином: один вивід кожного нормально розімкнутого ключа з'єднаний з однією із вертикальних ліній матриці, а інший – з однією із горизонтальних ліній.

Горизонтальні лінії матриці є її вхідними лініями, вертикальні – вихідними. В кожен момент часу лише одна з горизонтальних ліній повинна мати рівень логічного "0", всі інші – рівень логічної "1". Зокрема, для організації клавіатури використовуються старші розряди портів А та В.

В даному випадку, при натисканні однієї з шістнадцяти клавіш будуть замикатися два провідники. Програмним способом проводиться пошук точки замикання, посылаючи на старші розряди порту А сигнали та аналізуючи сигнали, що з'являються на старших розрядах порту В.

Для спряження клавіатури з мікропроцесором використовуються вісім ліній D0...D7.



DD1 KP580 BB55

Рис. 11 Організація клавіатури 5x5 на базі ППІ KP580BB55

1.2.5. Організація пристрою відображення інформації

На рис. 12 зображено один із можливих варіантів організації цифрового табло для відображення чотирьох десяткових цифр за допомогою світлодіодних семисегментних індикаторів АЛС324А. Для виводу даних на індикатори у вигляді двійково-десяткового коду цифри (виводи С0...С3) та для визначення розряду цифри (виводи С4...С5) використовується програмований паралельний інтерфейс KP580BB55, для перетворення двійково-десяткового коду в семисегментний код використовується перетворювач, що містить регістр пам'яті К176ИД2.

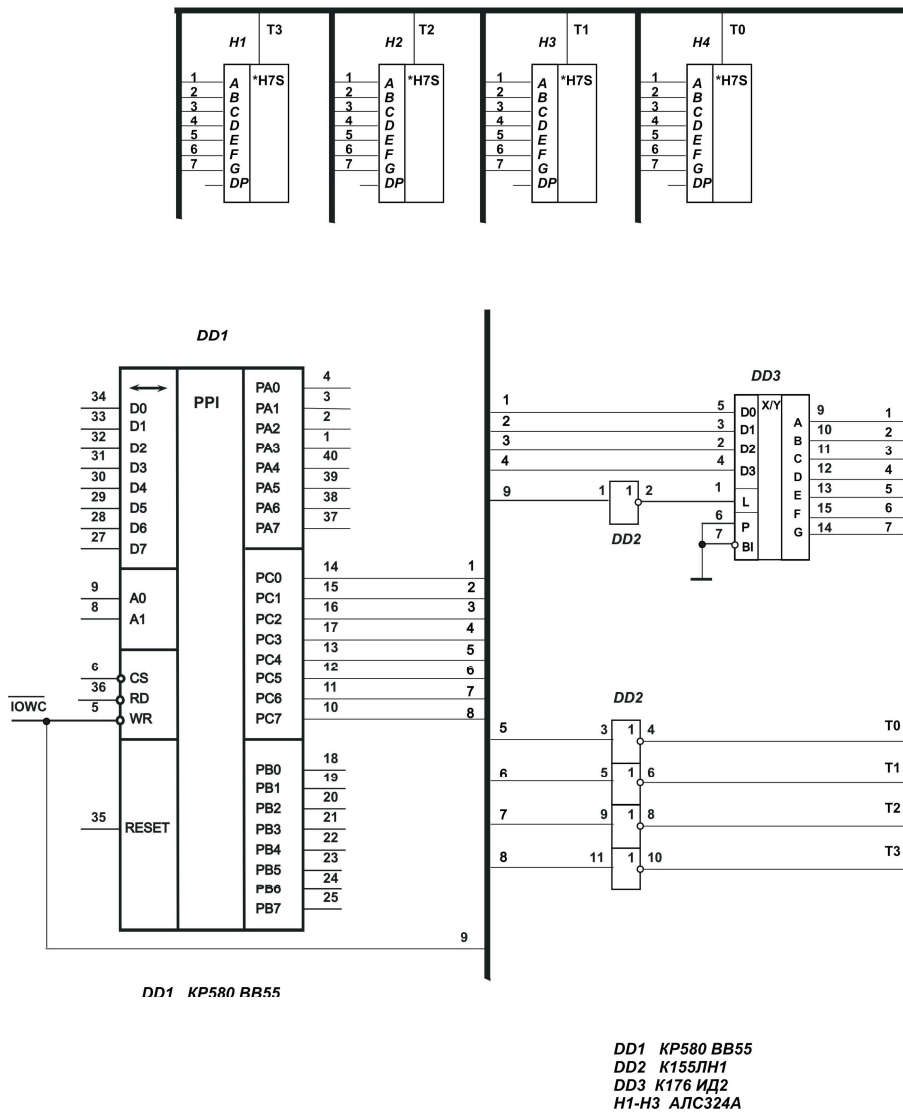
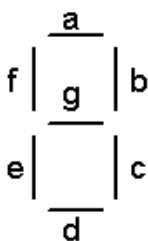


Рис. 12 Організація цифрового табло для відображення чотирьох десяткових цифр на основі світлодіодних семисегментних індикаторів АЛС324А

В якості контролера дисплею може використовуватися також програмований інтерфейс **КР580ВВ79**, дисплейна частина якого забезпечує вивід інформації по двох чотирирозрядних каналах у вигляді двійкового коду на 8- та 16-розрядні цифрові або алфавітно-цифрові дисплеї.

Інформація на дисплей може виводитися двома способами: зліва направо без зсуву та справа наліво із зсувом.

Для запалювання сегменту індикатора на нього потрібно подати рівень логічного нуля, що відповідає індикаторам типів АЛС32Б, АЛС342Б. На індикаторах буде виводитися одна з трьох цифр: 1, 2, 4. На основі цього нижче в таблиці наведений стан рівнів на виходах індикатора а-g, в залежності від цифри.



Цифра	a	b	c	d	e	f	G	Число
1	1	0	0	1	1	1	1	4FH
2	0	0	1	0	0	1	0	12H
4	1	0	0	1	1	0	0	4CH

З даної таблиці видно, що сегмент “b” запалений завжди, тому його можна заземлити. Сегменти “d” і “e” завжди мають один рівень, тому їх можна об'єднати і керувати ними одним сигналом. Тому для управління одним індикатором потрібно 5 ліній. Для управління двома індикаторами в статичному режимі необхідно 10 ліній.

1.3. Технічні засоби для узгодження сигналів мікропроцесорної системи з об'єктами керування

1.3.1. Аналогові та дискретні давачі

Давачі розрізняються типом вимірюваного параметра, способом вимірювання, конструкцією, діапазоном вимірювання, видом вихідного сигналу і т.д.

Давач — вимірювальний пристрій у вигляді конструктивної сукупності одного або декількох вимірювальних перетворювачів величини, що вимірюється і контролюється, у вихідний сигнал для використання в системах керування і має нормовані метрологічні характеристики.

Будь-який давач складається з чутливого елементу та перетворювальної системи. Іноді сам чутливий елемент є одночасно і перетворювальною системою. Чутливий елемент завжди безпосередньо пов'язаний з тим середовищем, параметр якого він вимірює. У теорії вимірювань для нього прийнято назву **вимірювальний перетворювач**.

Аналогові вимірювальні давачі - первинні перетворювачі

Такий тип давачів застосовується в системах безперервного вимірювання і регулювання. Принцип дії цих давачів полягає в тому, що при зміні параметру відбувається відповідна зміна його вихідного сигналу.

Дискретні вимірювальні давачі - сигналізатори (давачі-реле)

Такий тип давачів застосовується, коли необхідно відстежити конкретне значення вимірюваного параметра для яких або подальших дій. Ці давачі встановлюються там, де відсутня необхідність отримання всіх значень вимірюваного параметра. Якщо параметр досяг якого-небудь конкретного обмежувального значення, вимірювальна система видає сигнал (зміна рівня напруги, імпульс і т.п.).

Дискретними (бінарними) є **давачі на основі контакту**, вони мають два положення: замкнутий - розімкнений або 1 і 0. Контактми «уловлюють» кілька точок: початок руху, точку зниження швидкості, кінець руху.

Для виявлення положення точок у просторі і застосовуються саме дискретні давачі. Але сам по собі механічний контакт ненадійний. Тому механічні контакти замінюються безконтактними давачами. Найпростіший варіант - це геркони: магніт наблизився, контакт замкнувся. Точність спрацювання геркону залишає бажати кращого, застосовувати такі давачі можливо тільки для визначення положення (наприклад, дверей).

Більш складним і точним варіантом слід вважати різні **безконтактні давачі (ББК)**. Якщо металевий прапорець увійшов в проріз, то давач спрацював. Точність спрацювання (диференціал ходу) таких давачів становить 3 міліметри.

1.3.1.1. Давачі (вимірювальні перетворювачі) температури

Для того, щоб вибрати тип вимірювача, потрібно знати діапазон вимірюваних температур. Для води він може бути в діапазоні (0 ... 150) °С, повітря - (-50 ... + 50) °С.

Тому, в якості **аналогових давачів температури** можна застосувати так звані **термоперетворювачі опору**. Фізичний сенс їх роботи заснований на зміні опору чутливого елементу (наприклад, металу) в залежності від температури середовища, в яке він занурений.

Пропускаючи через цей елемент електричний струм, можемо отримати залежність зміни напруги від температури.

Раніше в якості такого металу застосовувалася мідь. Були мідні чутливі елементи з опором 50 Ом або 100 Ом при 0°С. Їх недоліком було те, що при значних довжинах

проводів, які з'єднували їх з вторинними пристроями, опір цих проводів були порівнювані, а то і більші, ніж опір самих давачів. Це вносило похибки в вимірювання, які необхідно було компенсувати.

Зараз від цієї проблеми відійшли, застосовуючи метали, що мають 500 Ом або 1000 Ом при 0°C. Це платина (Pt) і нікель (Ni). Тому сьогодні в інженерних системах найчастіше застосовуються давачі типу Pt 1000.

У вимірювальних системах використовуються **заглибні** термоперетворювачі опору, чутливий елемент яких безпосередньо занурений у вимірюване середовище, і **накладні**, які вимірюють температуру поверхні, припускаючи, що вона приблизно дорівнює температурі самого середовища.

В якості аналогових давачів температури також використовуються **термо-електричні перетворювачі (термопари)**. В термопарах застосовується властивість вимірювання термо-ерс, яка виникає на спайі різнорідних металів при зміні температури.

Термопари виготовляються з платино-родієвого (до 1700° C), хромель-алюмелевого (до 1100° C) або хромель-копелевого (до 700° C) сплавів.

При виготовленні термо-електричних перетворювачів широко використовується так званий термопарний кабель. Він являє собою гнучку металеву трубку з розміщеними всередині неї однієї або двох пар термоелектродів. Простір навколо термоелектродів заповнений ущільненою мінеральною ізоляцією. Термоелектроди термопарного кабелю з боку робочого торця з'єднуються між собою лазерним зварюванням, утворюючи робочий спай, який заглушений привареною сталевною пробкою.

В якості **дискретних давачів температури** найчастіше застосовуються **манометричні термометри**. Це пристрої, в яких чутливим елементом є термобалон, який з'єднаний капілярною трубкою з сильфоном. При зміні температури термобалону змінюється тиск в системі і сильфон переміщує механізм, який закінчується контактними пристроями. Часто можна почути назву таких давачів - **сигналізатори температури або термостати**.

1.3.1.2. Давачі (вимірювальні перетворювачі) тиску і перепаду тисків

Давачі тиску, як і давачі температури, бувають аналогові й дискретні.

Раніше в якості **аналогових давачів** використовувалися **мембранні та сильфонні давачі**, принцип дії яких був заснований на механічному переміщенні (стисненні або розширенні) даних чутливих елементів при зміні тиску середовища. Далі ці чутливі елементи мали шток, що переміщається в магнітному полі і міняюча величину, наприклад, магнітної індукції.

Зараз в якості чутливих елементів все частіше застосовують **тензорезистори**. При стисненні або розширенні такого резистора змінюється його опір. А далі, так само, як і в термоперетворювачі опору, даний резистор включається в електричну схему.

Дискретні давачі тиску розраховані на необхідність фіксації певного значення тиску чи перепаду тиску. Для цього застосовуються електроконтактні манометри і дифманометри, в якості чутливих елементів яких застосовані трубчасті пружини і мембрани.

1.3.1.3. Давачі (вимірювальні перетворювачі) вологості

Майже всі сучасні **аналогові давачі вологості** мають ємнісний чутливий елемент. Їх принцип роботи заснований на зміні ємності чутливого елемента при зміні вологості. Далі цей чутливий елемент включено у вимірювальну схему вторинного приладу.

Досить часто зустрічаються **суміщені аналогові давачі вологості і температури**. Таким чином, в точці відбору, де вимагається вимірювання цих двох параметрів, встановлюється один прилад замість двох. Такий давач має два незалежні вихідні сигнали - один по вологості, інший - по температурі.

Дискретні давачі вологості відрізняються від аналогових наявністю контактної групи, яка спрацьовує тільки при заданому значенні. Такі давачі також називають **гігростатами**.

1.3.1.4. Давачі (вимірювальні перетворювачі) витрат

Використовують **ультразвукові, індукційні, вихрові аналогові витратоміри** й витратоміри, в яких вимірювання витрати виконується шляхом вимірювання різниці тисків на так званому звужуючому пристрої з подальшою обробкою значень параметру за відповідною формулою. Зокрема, такі витратоміри входять разом з термоперетворювачами опору в комплект **теплотлічильників**.

Дискретні давачі витрат можуть бути виконані у вигляді крильчатки, що обертається в потоці рідини. З такого давача сигнал видається у вигляді імпульсу при здійсненні кожного повного її обороту. Підраховуючи ці імпульси, можна організувати облік кількості рідини, що пройшла через прилад за певний час.

Сигналізатори також можуть бути виконані у вигляді заслінки, встановленої поперек потоку. Чим більше витрата, тим більший тиск потоку на її поверхню і більший кут її відхилення від вертикального положення. При певному куті відхилення спрацьовує контакт і видається сигнал про наявність витрати. Такі сигналізатори витрати часто називають **реле протоки**.

1.3.1.5. Давачі (вимірювальні перетворювачі) рівня рідини

Аналогові давачі рівня - це ті ж давачі диференціального тиску, тому що будь-який стовп рідини створює різницю тисків між верхнім і нижнім рівнем.

Дискретні давачі рівня (фактично це датчики наявності рідини) побудовані на принципі електропровідності рідини і складаються з, як мінімум, двох електродів, через які проходить електричний струм. При зануренні їх у воду утворюється замкнуте електричне коло. Зокрема, такі системи застосовуються в дренажних приямках для вимірювання наявності рівня в них води.

1.3.1.6. Давачі (вимірювальні перетворювачі) якості повітря

У системах вентиляції та кондиціонування повітря часто підтримуються не тільки температура і вологість повітря, але і його якість, тобто кількість вуглекислого газу, озону, таких домішок як, сигаретний дим, запах поту, алкоголю, вихлопних газів і т.д.

Для вимірювання цих параметрів застосовуються:

- давачі, які визначають наявність у повітрі якихось окремих речовин, наприклад вуглекислого газу,
- давачі, які аналізують якість повітря по комплексу присутніх у ньому домішок.

Такі давачі також можуть бути аналоговими і дискретними.

Для уніфікації вихідних сигналів прийнятий ряд стандартних сигналів по струму - сигнали (0-5), (0-20) і (4-20) мА і напрузі - сигнал (0-10) В. Перетворення в стандартний сигнал може виконуватися перетворювачем, вбудованим в давач, або окремим перетворюючим пристроєм.

Деякі типи давачів можуть бути як з такими вбудованими перетворювачами, так і без них. Інші - тільки з вбудованими перетворювачами.

1.3.2. Вивід дискретних сигналів

Технічні засоби виводу дискретних сигналів від мікропроцесорної системи призначені для передачі дискретних керуючих впливів на виконавчі пристрої об'єктів автоматизації. Функцію таких об'єктів можуть виконувати соленоїди, електро- та гідроклапани, електродвигуни постійного та змінного струму, індикатори сигналізації, реле та пускачі і т.п.

Деякі схеми управління виконавчими механізмами за допомогою вихідних формувачів показані на рис. 13. Найбільш простим формувачем є мікросхеми з підвищеною навантажувальною здатністю, що мають виходи з відкритим колектором або емітером (рис. 13, а, б, DD1 K155ЛН1), навантаженням яких можуть слугувати оптоелектронні пристрої, світлодіоди або лампи розжарювання, малопотужні герконові або поляризовані реле, входи мікросхем зовнішніх пристроїв і т.п.

Для керування більш потужними зовнішніми об'єктами, напруга яких значно перевищує напругу живлення мікропроцесорної системи керування (напругою до 100 В та струмами до 1 А), в якості вихідних формувачів використовуються **потужні транзистори** (наприклад, КТ 814, КТ815, КТ 816, КТ 817, КТ 818, КТ 819 ...) (рис. 13 в, г), вхідні ланцюги яких гальванічно розв'язані із мікропроцесорною системою керування, а також **оптоелектронні тиристорні пари**, що дозволяють комутувати малопотужні (рис. 13, д) та більшої потужності ланцюги змінного струму (рис. 13, е).

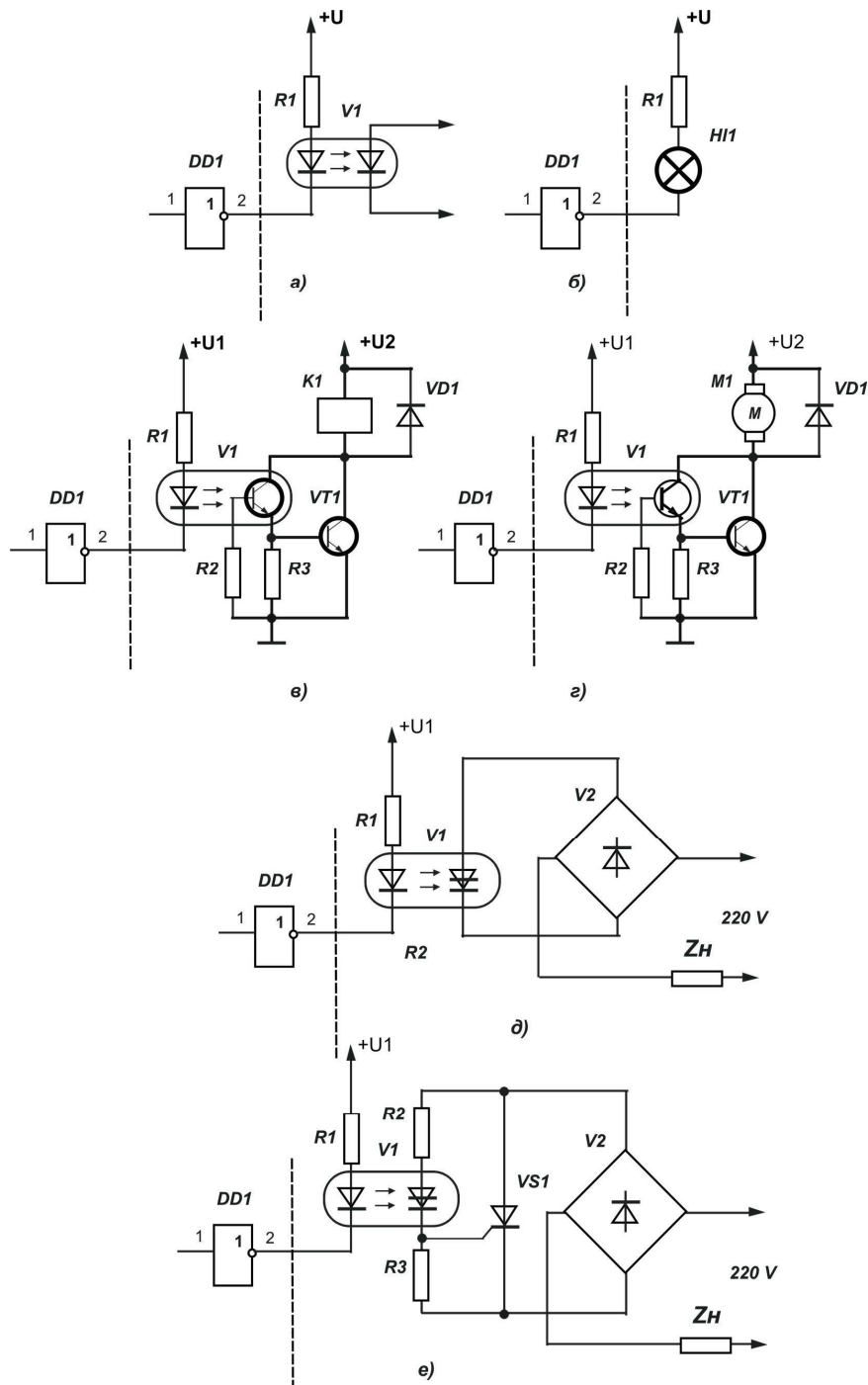


Рис. 13 Пристрої виводу дискретних сигналів

Виконавчими елементами в таких схемах можуть бути обмотки пневмо- та гідроклапанів, пускачів, електродвигуни невеликої потужності і т.п.

На рис. 13, г показана схема керування двигуном постійного струму. Таке вмикання двигуна дозволяє керувати станом «увімкнено-вимкнено» двигуна, але не дозволяє змінювати напрям його обертання.

Використання мостової схеми на біполярних (польових) транзисторах дозволяє при однополярному живленні шляхом комутації пар транзисторів на протилежних сторонах мосту (VT1,VT4 та VT3,VT2), змінювати напрям струму через двигун на протилежний. Це, в свою чергу, викликає зміну напрямку обертання валу двигуна.

Транзисторна мостова схема керування двигуном показана на рис. 13, ж.

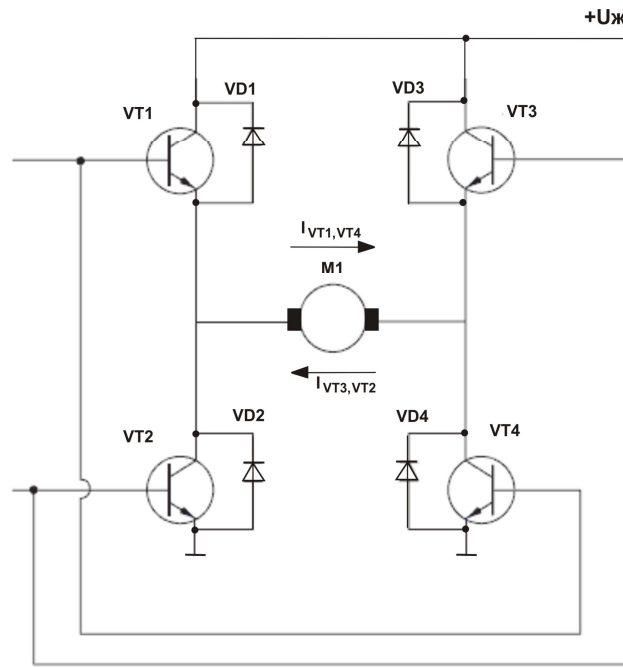


Рис.13, ж

Двигун є індуктивним навантаженням. В моменти включення/ виключення напруга на індуктивності обмоток, яка виникає внаслідок швидкого перемикання транзисторів, може вивести їх з ладу. Для гасіння цієї напруги використовують захисні діоди (VD1...VD4), які включають паралельно до транзисторів.

Управління вихідними формувачами дискретних сигналів здійснюється за допомогою пристроїв виводу даних (регістрів, програмованих паралельних інтерфейсів...), що мають можливість зберігання інформації на своїх вихідних лініях.

Існують два основні способи управління виводом дискретних сигналів:

- окреме управління кожним двійковим розрядом окремою командою від МП-системи;
- групове управління одночасно всіма двійковими розрядами, які можуть бути сформовані МП-системою в залежності від розрядності шини вводу-виводу (8, 16 і більше розрядів).

1.3.3. Ввід дискретних сигналів

Технічні засоби вводу дискретних сигналів призначені для перетворення сигналів від давачів дискретних сигналів в сигнали стандартної амплітуди (наприклад, ТТЛ рівня) для вводу в МП-систему.

Давачами дискретних сигналів можуть бути контактори, кінцеві вимикачі виконавчих механізмів, дискретні вимірювачі рівнів (тиску, температури і т.п.), дискретні вихідні сигнали цифрових приладів.

Під'єднуються давачі дискретних сигналів як безпосередньо до МП-системи, так і через зовнішні електронні формувачі сигналів. Причому, доцільно розв'язувати вхідні ланцюги між давачами та пристроями вводу МП-системи гальванічно.

Найбільшого поширення набули дискретні давачі, які визначаються станом контактів або рівнем постійної напруги.

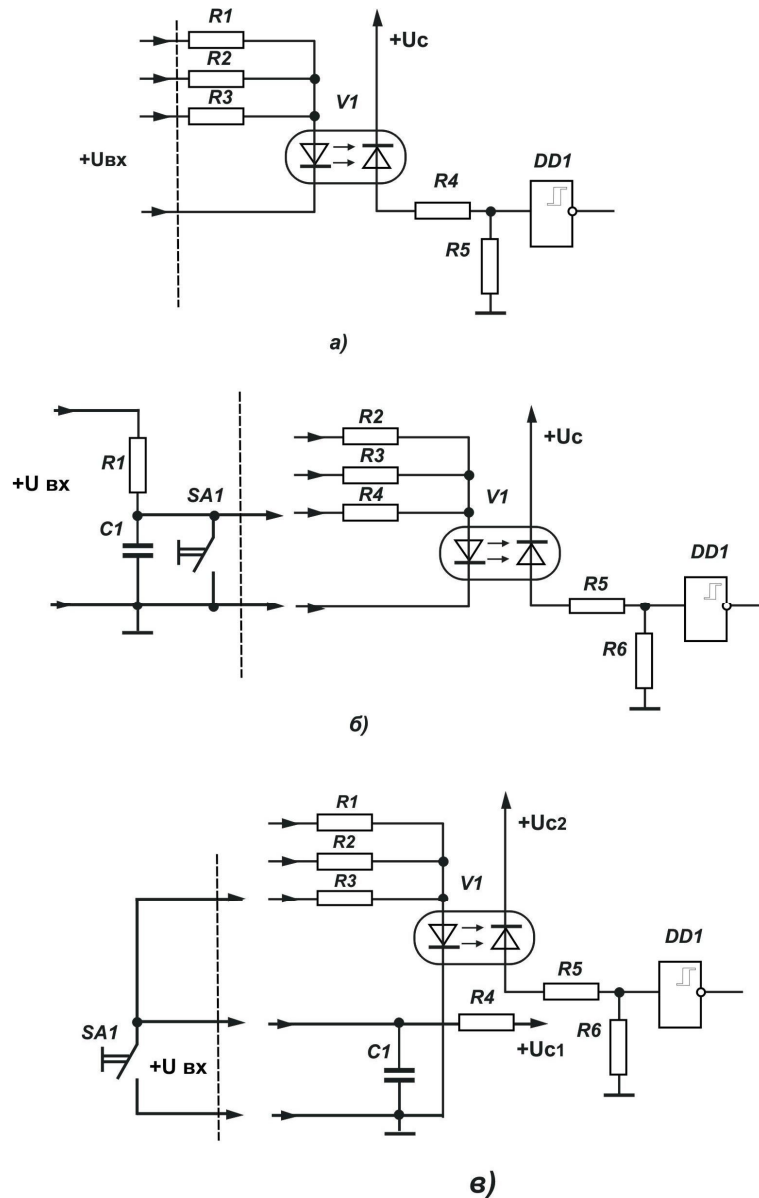


Рис. 14 Схеми вводу дискретних сигналів

Розрізняють три види подання інформації від дискретних давачів:

- двопозиційна, при якій кожен двійковий сигнал, що вводиться, функційно незалежить від інших і визначає стан якоїсь частини об'єкта керування (стан кінцевих вимикачів, наприклад);
- кодова, при якій сукупність двійкових розрядів формує функційно зв'язану групу сигналів (код) (наприклад, вихідний сигнал перетворювача кута повороту в код);
- число-імпульсна, при якому стан об'єкта визначається через кількість імпульсів (наприклад, кількість деталей на конвеєрі відповідає кількості імпульсів від фотоелемента).

Типові схеми під'єднання дискретних давачів показана на рис. 14 а, б, в.

Кожна схема містить обмежуючі резистори різного номіналу, через які, в залежності від його рівня, вхідний дискретний сигнал подається на оптоелектронний елемент, який, крім функцій формувача вхідного сигналу, забезпечує гальванічну розв'язку ланцюгів. З виходу оптопарі сигнал подається на формувач напруги (наприклад, тригер Шмітта), який

перетворює його в сигнал ТТЛ чи іншого рівня, і далі – на пристрій вводу (регістр, програмований паралельний інтерфейс...) для подальшої обробки МП-системою.

1.3.4. Ввід аналогових сигналів

Пристрої вводу аналогових сигналів в загальному випадку включають (рис. 15, а):

- комутатор аналогових сигналів (АК);
- нормуючий підсилювач (НП);
- аналоговий фільтр (АФ);
- підсилювач вибірки-запам'ятовування (ПВЗ);
- аналого-цифровий перетворювач (АЦП).

Вихід АЦП кожного з i -каналів через інтерфейс вводу-виводу (ІВВ) під'єднується до системної шини мікропроцесорного пристрою.

Аналоговий комутатор призначений для під'єднання аналогових давачів до входу підсилювача НП, який, в свою чергу, здійснює операцію пропорціонального підсилення або ослаблення (нормування) аналогових сигналів, що поступають від давачів, і одночасно узгоджує їх з діапазоном вимірювання АЦП. Аналоговий фільтр АФ призначений для подавлення високочастотних завад. Пристрій ПВЗ використовується для розширення частотного діапазону вимірюваних сигналів, а АЦП перетворює аналоговий сигнал в цифровий код, який за командами від інтерфейсу вводу-виводу (ІВВ) поступає в мікропроцесорну систему.

Функцію АК часто можуть виконувати комутатори К176КТ1, К561КТ3, а також аналогові мультиплектори К561КП1 та К564КП2.

Схеми НП та АФ будуються на операційних підсилювачах загального призначення (К140УД6, К140УД7, К140УД8А, К140УД11, К153УД5А, К544УД2А і т.п.).

ПВЗ виконуються у вигляді мікросхем, в яких запам'ятовуючий елемент – конденсатор - є зовнішнім елементом (КР1100СК2, КР1100СК4). ПВЗ працюють в двох режимах:

- вибірки, основним параметром якого є час вибірки T_v ;
- запам'ятовування, основним параметром якого є швидкість розряду запам'ятовуючого конденсатора.

АЦП, в залежності від застосування, діляться на АЦП широкого застосування та спеціалізовані.

До перших відносять мікросхеми, що мають характеристики по точності 10-12 розрядів та швидкодію (час перетворення) - одиниці та десятки мкс.

Спеціалізовані АЦП містять не менше 15-16 розрядів і відрізняються швидкодією в сотні та десятки нс. Найбільш поширеними є АЦП К1113ПВ1А, К1108ПВ1А, К572ПВ1А, К1107ПВ1, К1107ПВ2.

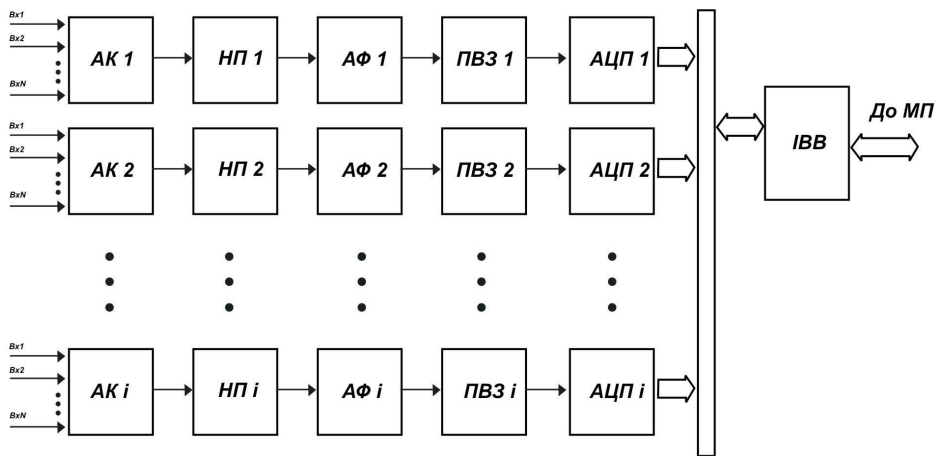
На рис. 15, б показана схема під'єднання АЦП К1113ПВ1А до джерела аналогового сигналу через нормуючий пристрій (К140УД7) та узгодження його з мікропроцесорним пристроєм через програмований паралельний інтерфейс КР580ВВ55.

АЦП К1113ПВ1А забезпечує аналого-цифрове перетворення однополярного або двополярного аналогового сигналу в 10-розрядний двійковий код. Мікросхема містить АЦП послідовного наближення з вбудованим джерелом опорної напруги та генератором синхроімпульсів частотою 300 - 400 КГц. Вихідні каскади є тристабільними.

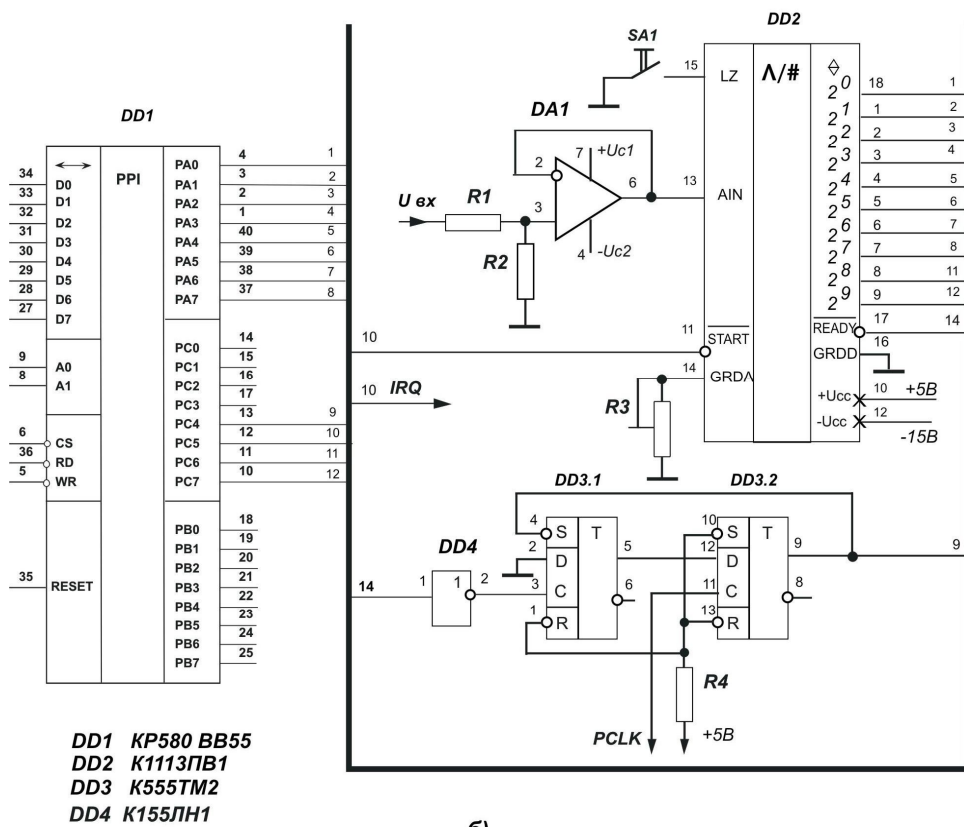
При роботі АЦП в однополярному режимі необхідно вивід LZ мікросхеми заземлити, а в двополярному – залишити вільним.

Вивід аналогової землі GRDA під'єднується через резистор (R3) до загальної шини для регулювання зміщення нуля.

Цифрову землю GRDD заземляють. Для регулювання діапазону вхідного сигналу (від 0 до 10 В) використовують резистивний подільник R1, R2.



а)



б)

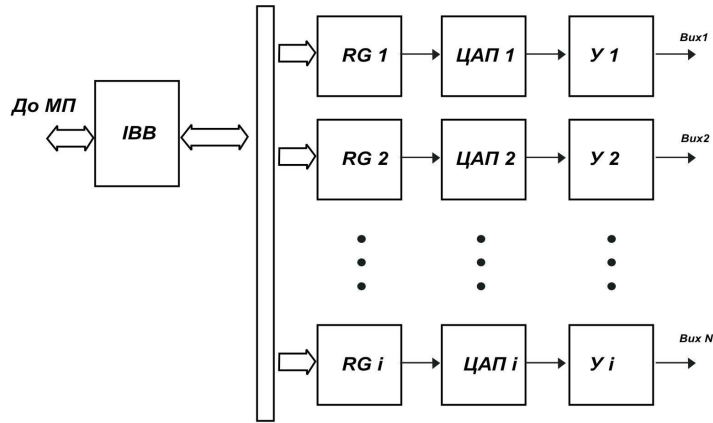
Рис. 15 Пристрої вводу аналогових сигналів та під'єднання АЦП до МП-системи з використанням програмованого паралельного інтерфейсу KP580BB55

За сигналом по входу START, який разом із сигналом "STOP" (виробляється по закінченні перетворення по лінії READY) формується за допомогою тригерів DD3.1 та DD3.2, АЦП починає перетворення аналогового сигналу в код. Одночасно активізується системна лінія IRQ, яка викликає через переривання процедуру обробки аналогового сигналу мікропроцесором. Обмін синхронізується імпульсами PCLK системної шини, що мають ТТЛ рівень. Після закінчення перетворення на лінії READY встановлюється "0", по якому тригер формує сигнал STOP і інформація з виходу АЦП через лінії PA0...PA7 та PC6, PC7 інтерфейсу DD1 поступає в мікропроцесор.

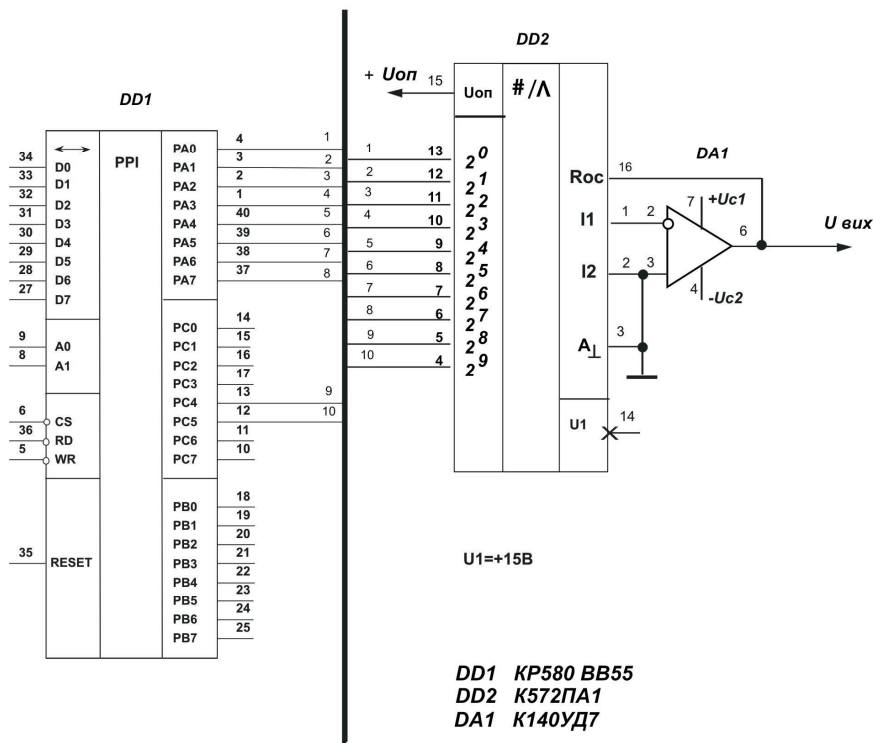
1.3.5. Вивід аналогових сигналів

Пристрої виводу аналогових сигналів в загальному випадку включають (рис. 16, а):

- буферні регістри (RG);
- цифро-аналогові перетворювачі (ЦАП);
- підсилювачі (У).



а)



б)

Рис. 16 Пристрої виводу аналогових сигналів та схема під'єднання ЦАП та узгодження його з МП через програмований паралельний інтерфейс КР580ВВ55

Входи регістрів кожного з i -каналів через інтерфейс вводу-виводу (ІВВ) під'єднується до системної шини мікропроцесорного пристрою.

Буферний регістр призначений для зберігання вхідного коду, що поступає від пристрою виводу. У випадку, коли функції ПІВВ виконує, наприклад, програмований

паралельний інтерфейс (KP580VB55), що має здатність зберігати інформацію на своїх виводах до повторного її виводу, регістри не використовуються.

ЦАП перетворює вхідний цифровий код в аналоговий сигнал у вигляді напруги або струму.

Підсилювач У перетворює струмовий аналоговий сигнал з виходу ЦАП в напругу (якщо ЦАП має струмовий вихід), підсилює аналоговий сигнал за потужністю та нормалізує його.

На рис. 16, б показана схема під'єднання ЦАП та узгодження його з мікропроцесорним пристроєм через програмований паралельний інтерфейс KP580VB55.

Мікросхема K572ПА1 (DD2) є 10-розрядним ЦАП помножуючого типу, побудованого на принципі сумування струмів з живленням від джерела опорної напруги Uоп. Для роботи ЦАП потрібні зовнішній операційний підсилювач (DA1) та стабілізоване джерело опорної напруги.

Найбільш часто також використовуються мікросхеми ЦАП:

- 12-розрядний K572ПА2;
- 12-розрядний паралельний ЦАП K594ПА1;
- швидкодійний 12-розрядний ЦАП з сумуванням струмів K1108ПА1;
- швидкодійний 8-розрядний ЦАП K1118ПА1 та ін.

1.3.6. Керування кроковим двигуном

Кроковий двигун (КД) є традиційним виконавчим пристроєм багатьох електронних приладів і систем. Він являє собою безколекторний двигун постійного струму з фіксованими положеннями валу.

КД призначено для точного позиціонування валу без застосування систем зворотнього зв'язку. Обмотки КД є частиною статора. На роторі розташований постійний магніт або, у випадках зі змінним магнітним опором, зубчастий блок з магнітом'якого матеріалу.

Усі комутації проводяться за зовнішніми схемами керування. **На двигунах з постійними магнітами звичайно є дві незалежні обмотки.**

Крокові двигуни мають широкий діапазон кутових дозволів. Більш грубі двигуни, звичайно с постійними магнітами, обертаються на 90° за крок, у той час як прецизійні двигуни можуть мати дозвіл 1,8° або 0,72° на крок.

Для правильного керування **біполярним кроковим двигуном** необхідна електрична схема, яка повинна виконувати функції старту, зупинки, реверсу й зміни швидкості.

Кроковий двигун транслює послідовність цифрових перемикачів у рух. «Обертове» магнітне поле забезпечується відповідними перемикачними напруг на обмотках. Слідом за цим полем буде обертатися ротор, з'єднаний за допомогою редуктора з вихідним валом двигуна.

Потужність крокових двигунів знаходиться у діапазоні від одиниць ватів до одного кіловату.

Кроковий двигун має не менш двох положень стійкої рівноваги ротора в межах одного оберту. Напруга живлення обмоток керування кроковом двигуном - це послідовність однополярних або двополярних прямокутних імпульсів, що надходять від електронного комутатора (К) або контролера. Результуючий кут відповідає кількості перемикачів комутатора, а частота обертання двигуна – частоті перемикачів електронного комутатора.

На рис. 17 зображено положення ротора крокового двигуна в залежності від комутації обмоток.

Послідовно комутуючи струм в обмотках відповідно до діаграм, наведених на рис. 23, можна змусити обертатися вектор магнітного поля, а за ним і ротор, у прямій або зворотній послідовності. При такому керуванні двигун має 4 стійких стани на одному оберті ротора.

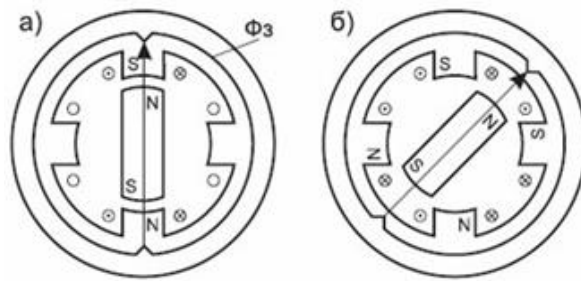


Рис. 17 Положення ротора при кроковому (а) і півкроковому керуванні (б)

Одночасне включення двох обмоток приводить до орієнтування вектора магнітного поля із кроком 45° щодо вертикальної осі. Таке керування називається півкроковим.

Півкроковий режим дозволяє вдвічі підвищити точність позиціонування крокових двигунів. Послідовність комутації обмоток двигуна в півкроковому режимі наведена на рис. 18. При комутації по черзі вмикається одна обмотка, а за нею дві разом і т.п. При такому керуванні двигун має 8 стійких станів.

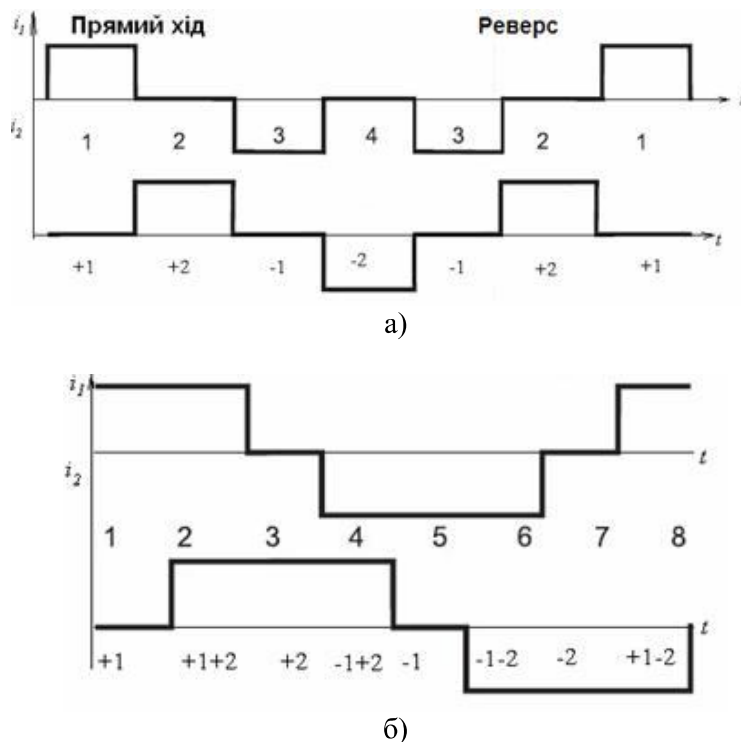


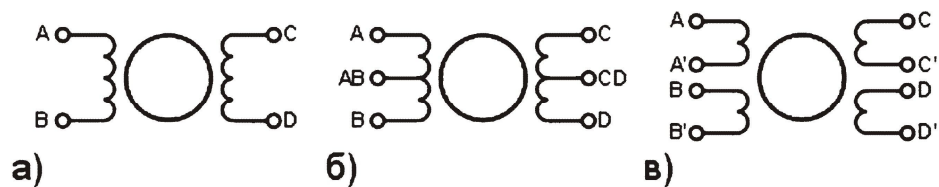
Рис. 18 Крокове (а) та півкрокове (б) керування кроковим двигуном

1.3.6.1. Біполярні і уніполярні крокові двигуни

В залежності від того, якою є форма обмоток крокового двигуна, двигуни діляться на **уніполярні** і **біполярні**.

У **біполярного** двигуна по одній обмотці в кожній фазі, тобто всього дві обмотки і відповідно чотири виводи (рис. а). Для забезпечення обертання валу на ці обмотки подається напруга із змінною полярністю. Тому, для біполярного двигуна необхідний півмостовий або мостовий драйвер, забезпечений двополярним живленням.

Уніполярний двигун також, як і біполярний, для кожної фази має по одній обмотці, але кожна обмотка містить відвід від середини. У зв'язку з цим, шляхом перемикання половинок обмоток крокового двигуна з'являється можливість міняти напрям магнітного поля.



В даному випадку значно спрощується структура драйвера двигуна. Він повинен володіти всього лише чотирма силовими ключами. Відповідно, в уніполярному двигуні застосовується інший метод зміни напрямку магнітного поля. Відводи обмоток часто об'єднуються всередині двигуна, тому даний тип двигуна може мати п'ять або шість проводів (рис. б).

Іноді уніполярні двигуни забезпечуються чотирма обмотками, кожна з яких містить власні виводи - тобто їх всього вісім (рис. в). При певному поєднанні цих обмоток подібний кроковий двигун можливо використовувати або як біполярний, або уніполярний.

Уніполярний двигун, що має дві обмотки з відводами по-середині, можна використовувати і як біполярний. У цьому випадку проводи, що йдуть від середини обмоток, не використовуються.

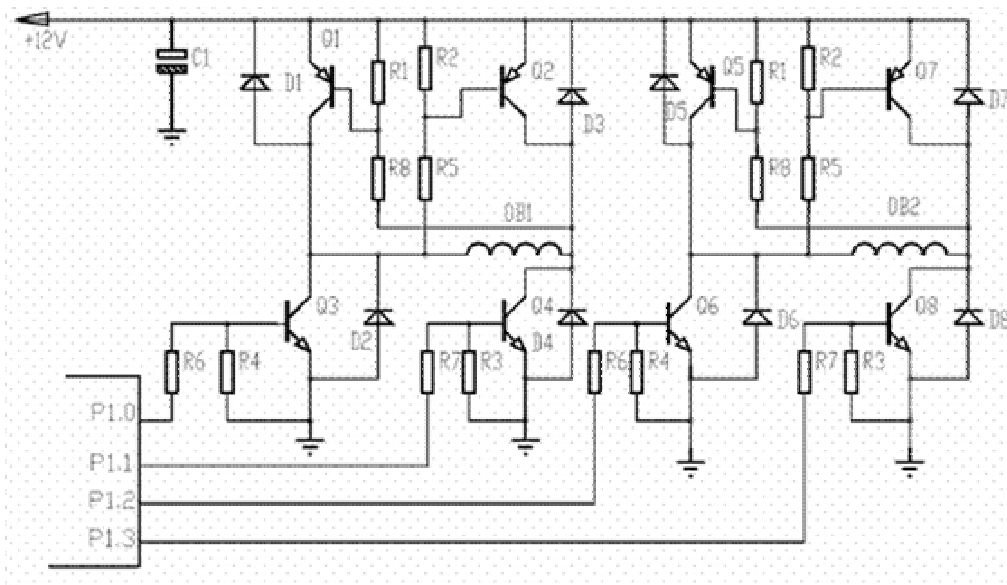


Рис. 19 Під'єднання біполярного крокового двигуна з двома обмотками

Комутатор повинен забезпечувати зміну полярності струму в обмотках. Він являє собою два транзисторні мости, в діагональ яких включені статорні обмотки (рис. 19).

Принципова електрична схема керування уніполярним кроковим двигуном приведена на рис. 20.

Керування двигуном може здійснюватися як під управлінням програми контролера, так і за допомогою кнопок пульта керування S1...S5.

Силова частина схеми керування (рис. 20) складається з чотирьох однакових каскадів на біполярних транзисторах, увімкнених за схемою Дарлінгтона, по одному на кожен фазу крокового двигуна (наприклад, VT1, VT2 для першої обмотки). Транзистори працюють в ключовому режимі для підключення і відключення чотирьох обмоток КД.

У схемі транзистори можуть керуватися безпосередньо логічним сигналом з виводів контролера або інтерфейсних мікросхем, наприклад, від зовнішньої мікро-ЕОМ через програмований паралельний інтерфейс KP580BB55.

Для захисту від перенапруги при комутації фаз двигуна у схемі використані діоди VD2, VD5, VD8, VD11.

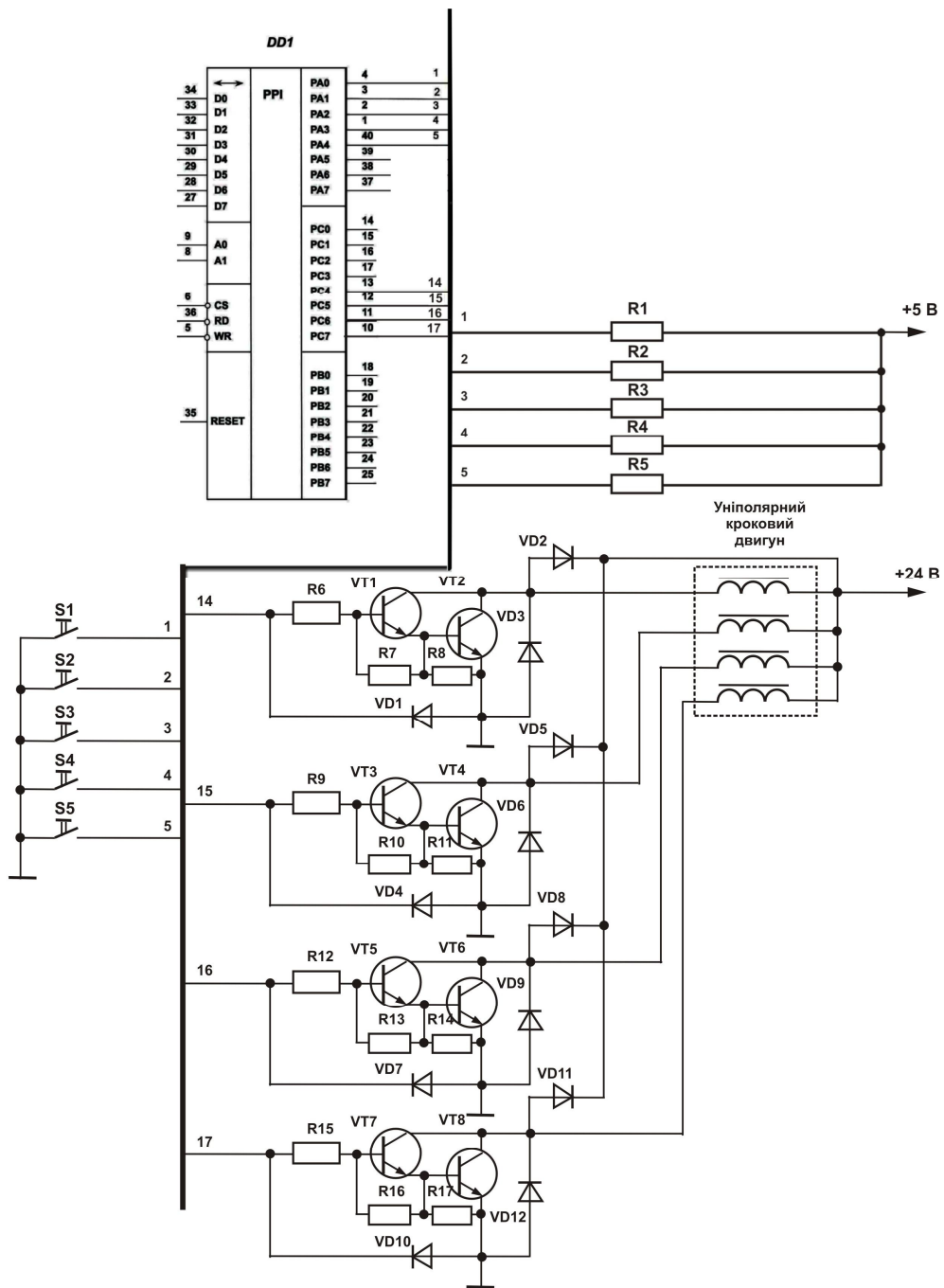


Рис. 20 Принципова електрична схема керування кроковим двигуном через паралельний програмований інтерфейс KP580BV55

1.4. Програмування МП KP580BM80A (i8080)

Команди асемблера для МП KP580BM80A приведені в Додатку 1 методичних вказівок.

1.4.1. Програмування роботи ППІ KP580BV55

Програма для перезапису даних з порту вводу в порт виводу:

```

MVI A, 81H ;запис в акумулятор A керуючого слова
OUT FBH   ;запис вмісту акумулятора в керуючий регістр
VVOD IN FAH ;зчитування даних з півпорту вводу C(L) в
           ;акумулятор
OUT F9H   ;запис вмісту акумулятора в порт виводу B
JMP VVOD  ;повернення на ввід

```

Приклад програмування КР580ВВ55

Розглянемо приклад програмування інтерфейсу вводу-виводу і розробку програми взаємодії процесора з зовнішніми пристроями для фрагменту МПС, поданого на рис. 21.

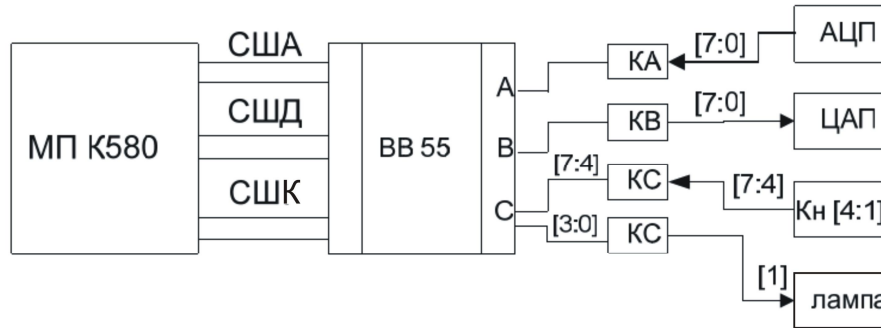


Рис. 21 Фрагмент реалізації МПС

Нижче (рис. 22) представлена послідовність розробки програми і тексти етапів цього процесу:

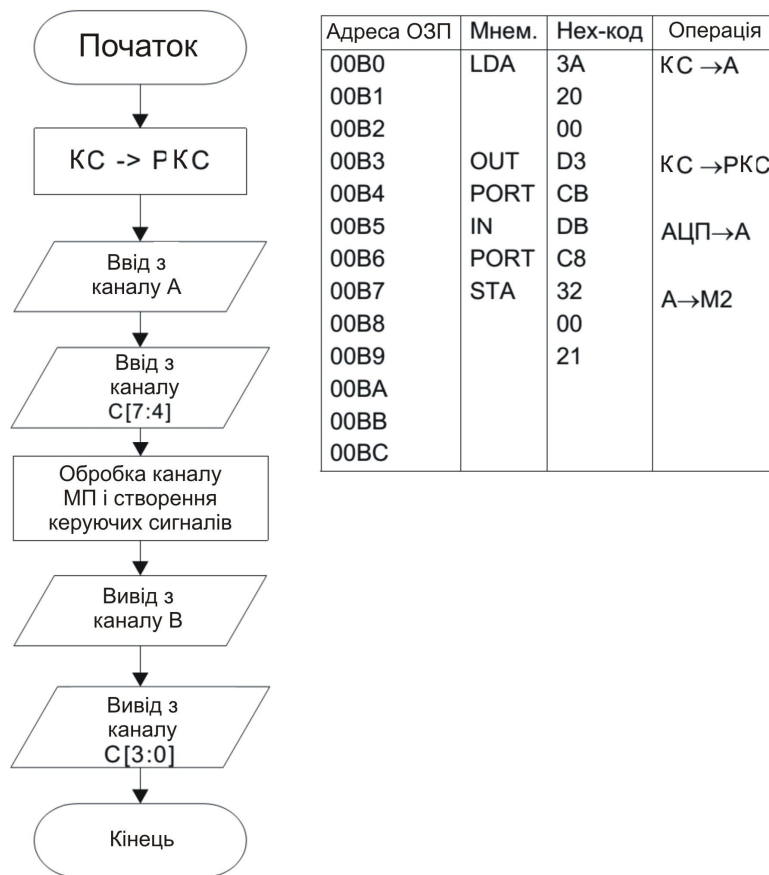


Рис. 22 Розробка програми: а - схема алгоритму; б - фрагмент тексту програми

Керуюче слово:

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

Розподіл пам'яті:

РКС — реєстр керуючого слова
 КС (керуюче слово) → М1- 0020
 Дані каналу АМ2 – 0021 (АЦП)
 Дані каналу ВМ3 – 0022 (ЦАП)
 Дані каналу С[7:4] – 0023 (Кнопки)
 Дані каналу С[3:0] – 0024 (сиг. ламп)
 Адреса контролера [7:2]= 1100 10 11= СВ

1.4.2. Приклад програми для вводу даних з клавіатурної матриці 5x5 (рис. 11)

П'ять молодших ліній порту А використовуються для сканування, а п'ять ліній порту В – для опитування.

1.4.2.1. Процедура сканування служить для виявлення натиснутої клавіші і наступної її ідентифікації. Процедура зводиться до почергового обнулення кожної з ліній сканування й опитування ліній повернення. У порт А видається *байт сканування* (БС), що містить 0 тільки в одному біті. Якщо на перетині лінії сканування і лінії повернення знаходиться натиснена клавіша, то у відповідному біті *байту повернення* (БВ), прийнятого в порт В, буде знаходитися 0.

Послідовність байтів сканування з кодом “біжучий нуль” формується шляхом зсуву попереднього коду на один розряд вліво або вправо. Напрямок зсуву визначає послідовність опитування клавіш. Якщо при повному циклі сканування не було виявлено натиснутої клавіші, то процедура сканування повторюється спочатку.

Після виходу з процедури SCAN у SCANCODE буде знаходитися код натиснутої клавіші. Крім того, процедура SCAN здійснює захист від одночасного натискання кількох клавіш.

Порядок аналізу клавіш такий, що при одночасному натисканні клавіша з більшим кодом ігнорується:

```
                                ;регістр В - лічильник коду сканування
                                ; (SCANCODE)
SCAN:
    MVI B,00H                    ;скидання SCANCODE
    MVI E,05H                    ;завантаження лічильника сканування
    MVI C,11111110B             ;завантаження вихідного байту
                                ;сканування в регістр С
LOOP:
    MOV A,C                      ;завантаження байту сканування в
                                ;акумулятор
    OUT [адр]                    ;вивід поточного байту сканування
                                ;в порт А
    RLC                          ;зсув байту сканування на розряд вліво
    MOV C,A                      ;збереження поточного байту сканування
    MVI D,05H                    ;завантаження лічильника опитувань
                                ;(регістр D)
    IN [адр]                      ;звід байту повернення з порту В
    STA [адр]                     ;збереження байту повернення в пам'яті
ROTATE:
    LDA [адр]                     ;завантаження байту повернення в А
    RRC                          ;зсув байту повернення вправо
    JNC DBNC                      ;вихід із процедури при виявленні
                                ;першої натиснутої клавіші
    INC B                          ;інкремент SCANCODE
    DCR D                          ;декремент лічильника опитувань
    MOV A, D                      ;завантаження вмісту регістра D в А
    JNZ ROTATE                    ;цикл, якщо не нуль
    DCR E                          ;декремент лічильника сканування
    MOV A, E                      ;завантаження вмісту регістра E в А
    JNZ LOOP                      ;цикл, якщо не 0
    JMP SCAN                      ;почати сканування спочатку
```

Після виходу з процедури SCAN керування передається процедурі усунення джеренчання контактів DBNC.

1.4.2.2. Усунення деренчання контактів при введенні символу з клавіатури здійснюється, як правило, програмною реалізацією очікування тривалістю 5 – 20 мс залежно від механічних характеристик клавіатури:

```
DBNC : CALL DELAY ;затримка
```

Якщо можливе виникнення деренчання контактів і при звільненні клавіші, то процедуру DBNC необхідно вставити і після процедури очікування звільнення клавіші.

1.4.2.3. Для того, щоб при повторному звертанні МК до клавіатури не був введений код тієї ж клавіші, вводиться **процедура чекання звільнення**. Після виконання сканування активною залишилася та лінія сканування, в якій виявлена натиснута клавіша. Тому процедура чекання звільнення натиснутої клавіші зводиться до зчитування й аналізу байту повернення:

```
WAITOP: IN [адр] ;ввід байту повернення з порту В
        CMA      ;інверсія байту повернення
        JNZ WAITOP ;якщо клавіша натиснута, то чекати
```

Процедура WAITOP у тому вигляді, в якому вона наведена вище, може бути використану тільки в системах, захищених від "залипання" контактів.

1.4.2.4. Кожній клавіші клавіатури повинен бути поставлений у відповідність код (її вага), що є функцією номерів лінії сканування і лінії повернення, на перетинанні яких натиснена клавіша. **Процедура ідентифікації натисненої клавіші KEYW** може бути поєднана з процедурою сканування (як у розглянутому вище прикладі). Тоді після виходу з процедури SCAN у регістрі SCANCODE буде розміщений код натиснутої клавіші.

Процедура введення коду клавіші KEYBRD оформляється у вигляді лінійної послідовності розглянутих вище процедур:

```
KEYBRD:
SCAN:   ...           ;сканування клавіатури
        ...
        ...
DBNC   :CALL DELAY   ;усунення деренчання при натисканні
WAITOP: ...         ;чекання звільнення клавіші
        ...
        ...
DBNC   :CALL DELAY   ;усунення деренчання контакту
        ;при звільненні клавіші може бути
        ;відсутня
KEYW   : ..         ;ідентифікація натиснутої клавіші
        ;(може бути відсутня)
```

1.4.2.5. Процедура опитування стану клавіатури. Вище зазначалося, що в МП-системах реалізують безупинне керування, процедурі KEYBRD повинна передувати процедура опитування стану клавіатури ASK. Приклад програмної реалізації процедури ASK, оформленої у вигляді підпрограми, наведений нижче.

Вихідний параметр передається в основну програму через ознаку перенесення, що встановлюється, якщо хоча б одна клавіша натиснута:

```
ASK : MVI A,00H      ;скидання акумулятора
      ANA A          ;скидання ознаки перенесення С
      OUT [адр]      ;вивід байту "усі нулі" в порт А для
```

```

;одночасного опитування всіх клавіш
IN [адр] ;ввід байту повернення з порту В
CMA ;інверсія байту повернення
JZ EXIT ;вихід, якщо немає натиснутої клавіші
STC ;встановлення ознаки перенесення в "1"
EXIT: RET ;повернення

```

Підпрограма виконує одночасне опитування всіх клавіш. У випадку, якщо хоча б одна клавіша натиснута (байт повернення – не всі одиниці), встановлюється ознака перенесення, у протилежному випадку вона скидається.

1.4.3. Опитування двійкового давача. Чекування події

В пристроях і системах логічного керування об'єкти та події в об'єкті керування фіксуються з використанням різноманітних датчиків цифрового й аналогового типів.

Найбільше поширення мають двійкові датчики типу так/ні, наприклад, кінцеві вимикачі, що підключаються до порту мікросхеми КР580ВВ55 так, як показано на рис. 23 (D1 – буферний пристрій на основі елемента логіки).

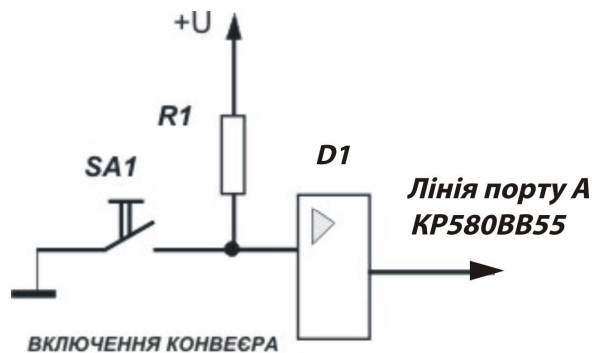


Рис. 23 Підключення цифрового давача до КР580ВВ55

Типова процедура чекування події (WAIT) складається з наступних дій:

- вводу сигналу від давача;
- аналізу значення сигналу і передачі керування в залежності від стану давача.

Конкретна програмна реалізація процедури залежить не тільки від типу давача, але і від того, яким чином він підключений до МП.

Наприклад, при підключенні давача до лінії А2 порту А КР580ВВ55 програма реалізації процедури чекування замикання контакту буде мати вигляд:

```

WAITC: IN [адр] ;ввід сигналу від давача з порту А
ANI 00000100b ;виділення сигналу лінії А2 порту А
JNZ WAITC ;якщо контакт давача розімкнутий, то
;повторювати ввід,
EXIT: ... ;інакше вихід із процедури

```

Програмна реалізація процедури для випадку підключення чотирьох імпульсних давачів до входів 0-3 порту А1 буде мати вигляд:

```

KBRD: IN [адр] ;ввід коду
CMA ;інверсія коду
ANI 00001111b ;виділення сигналів від чотирьох
;молодших ліній порту А
JZ KBRD ;якщо жоден контакт не замкнутий,
;то чекати

```

```

MOV M, A ;передача прийнятого коду в пам'ять
EXIT: ... ;інакше вихід із процедури

```

Аналіз стану контактів здійснюється накладенням маски на прийнятий від давачів код.

1.4.4. Вивід символу на семисегментний індикатор (ССІ)

При використанні зовнішніх (відносно МП) схем перетворювачів кодів (дешифраторів) процедура індикації одного символу зводиться до видачі двійкового коду символу у відповідний порт виводу мікросхеми КР580ВВ55 (рис. 12): на чотири молодші розряди порту С (CL) виводиться двійковий код числа (**код символу**), а на чотири старші розряди порту С (Ch) – номер розряду (**байт індикації**).

При відсутності декодуючого пристрою перетворення двійкового коду в код семи-сегментного індикатора здійснюється програмно з використанням таблиць перекодування.

Розглянемо приклад організації динамічного дисплею.

Наприклад, два порти мікросхеми КР580ВВ55 використовують для індикації (В - для вибору розряду, А — інформаційні, формують байт індикації).

Використовується динамічна індикація. Перевага даного способу індикації полягає в тому, що значно скорочується кількість провідників, необхідних для підключення індикаторів. По-друге, скорочується кількість задіяних портів вводу-виводу, які можна використати для інших потреб.

Адреси регістрів ППІ для даного мікропроцесорного пристрою наступні:

```

6000H – адреса порту А,
6001H - адреса порту В,
6002H – адреса порту С,
6003H – адреса керуючого регістра РКС.

```

```

;програма динамічної індикації символу
;на семисегментному індикаторі;
LXI 60003H ;завантаження в регістри H,L адреси РКС ППІ
MVI A,10000000b ;налаштовуються всі порти на вивід
; (1-Mode Selection,00-режим для групи А,
;0-вивід для А,0-вивід для С4-7,0-режим для
;групи В, 0-вивід для В,0-вивід для С0-3)

```

Приклад процедури виводу на індикатори 16-бітного числа, адреса якого зберігається в регістровій парі D-E, на вісім семисегментних індикаторів:

```

;таблиця семисегментних кодів даних для виводу
;на індикатори
ORG 0500H
DB 0FCH,60H,0DAH,0F2H,66H,0B6H,0BEN,0E0H,0FEN,0F6H
DB 0EEN,3EH,9CH,7AH,9EH,8EH
MVI B,02H
MVI C,01111111b
LOOP:
LXI H, 6001H ;адреса порту В ППА
MOV A, C ;вибір індикатора
MOV M, A
RRC ;зсув вмісту А вправо
MOV C, A ;зберігання вмісту а в регістрі С

```



```

LDAX D
RRC
RRC
RRC
RRC
ANI 0FH
MOV H, 05H
MOV L, A
MOV A, M ;значення з таблиці
MVI H, 6000H
MOV M, A
MVI H, 6001H ;вибір індикатора
MOV A, C
MOV M, A
RRC
MOV C, A
LDAX D
ANI 0FH
MOV H, 05H
MOV L, A
MOV A, M ;значення з таблиці
MVI H, 6000H
MOV M, A
INX D
DCR B
XRA A
CMP B
JNZ LOOP:
RET

```

1.5. Розрахунок електричних і часових параметрів МП - пристроїв

1.5.1. Розрахунок фільтрів по живленню

Для зменшення **високочастотних перешкод** в колі живлення безпосередньо до виводів живлення підключаються високочастотні керамічні конденсатори з розрахунку 0,01- 0,001 мкФ на 2 ІМС.

У колі живлення, крім високочастотних перешкод, існують низькочастотні перешкоди.

Для зменшення **низькочастотної складової** перешкоди, що має частоту 50 Гц, в колі живлення використовуються електролітичні конденсатори, розрахунок параметрів яких приведений нижче.

1.5.2. Розрахунок споживаної потужності та деяких елементів схеми

Для розрахунку споживаної потужності мікропроцесорного пристрою необхідно підсумувати потужності, споживані кожною мікросхемою.

Споживані потужності деяких мікросхем серії КР580 та інших наведені в табл. 2.

Таблиця 2

Назва елемента	Кількість	Споживана потужність, мВт
КР580ВМ80	1	1500

КР580ВК28	1	850
КР580ГФ24	1	560
КР580ВА86	10	800
КР537РУ16	1	50
К753РФ4	1	400
К155ИД7	1	200
КР580ВВ51	1	350
КР580ВВ55	1	350
КР580ВИ53	1	1000
АЛС324А	8	50

Опорну частоту синхронізації визначає кварцовий резонатор, до якого підключені два конденсатори С1 і С2 ємністю 30 пФ. У розробленому пристрої використано кварцовий резонатор, що забезпечує частоту тактових імпульсів 12 МГц.

Якщо **сигнал скидання** для спроектованого пристрою при включенні живлення реалізований апаратно ланкою RC, розрахуємо значення елементів R та C.

Для скидання процесора цей сигнал повинен утримуватися протягом двох машинних циклів (24 періоду резонатора). При частоті процесора 12 МГц інтервал часу утримання сигналу скидання становить $(24/12) \cdot 10^{-6} = 2$ мкс.

Забезпечити дану затримку можна при виборі резистора $R = 200$ Ом і конденсатора $C = 10$ нФ. У цьому випадку тривалість затримки $t = 3 \cdot \tau = 3 \cdot R \cdot C = 6$ мкс.

1.5.3. Розрахунок блоку живлення

Для того, щоб визначити параметри стабілізатора і випрямного мосту, потрібно обчислити сумарний струм споживання мікросхем, що містяться в розробленому мікропроцесорному пристрої (наприклад, для таких, що приведені в табл. 3), і, відповідно, із запасом вибрати стандартизовані елементи стабілізатора із довідників.

Таблиця 3

Позначення	Назва елемента	Струм споживання
DD1	К555ЛН1	8 мА
DD2	К555ИЕ5	20 мА
DD3	К555ЛН1	8 мА
DD4	К155ИЕ5	20 мА
DD5	К155ИЕ5	20 мА
DD6	К555ЛН1	8 мА
DD7	К555ЛН1	8 мА
DD8	К555КП7	28 мА
DD9	К555ЛА1	11 мА
DD10	К555ЛА1	11 мА
DD11	К555ЛА4	23 мА
DD12	К555ЛА4	23 мА
DD13	К555ЛА3	16 мА
DD14	К555ЛЛ1	9,8 мА
DD15	К155ИР13	116 мА
DD16	К555АП6	76 мА
DD17	К555ЛН1	8 мА
Загальний струм споживання:		405,8 мА

Як видно з таблиці, стабілізатор повинен мати вихідний струм I_{cm} більший за 405,8 мА, тобто за загальний струм споживання мікросхем.

Вибираємо стабілізатор за вихідною напругою і струмом.

Вибираємо стабілізатор КР142ЕН8А з наступними параметрами:

$$U_{BX} = 7,5 \dots 15 \text{ В};$$

$$U_{ВИХ} = 4,9 \dots 5,1 \text{ В};$$

$I_{ВИХ.max}$ при $t_{КОР.} = 20 \dots +100 \text{ }^\circ\text{C}$ не більше 3 А;

$I_{втрат}$ не більше 10 мА;

Розсіювана потужність корпусу мікросхеми стабілізатора не більша за 10 Вт.

1.5.3.1. Розрахунок випрямляча

Розрахувати випрямляч означає правильно вибрати випрямні діоди і конденсатор фільтру, а також визначити необхідну змінну напругу вторинної обмотки мережевого трансформатора.

Вхідні дані:

$$U_{вих} = 5 \text{ В};$$

$$q_{вих} = 0,1;$$

$$I_{живл} = 406 \text{ мА}$$

Початковими даними для розрахунку випрямляча (рис. 24) служать:

- напруга на навантаженні ($U_{вих}$),
- споживаний нею максимальний струм ($I_{живл}$).

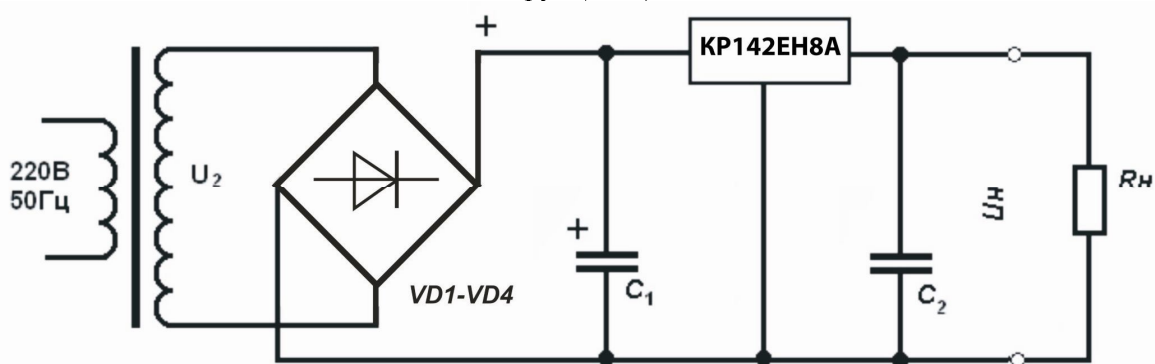


Рис. 24

Розрахунок слід проводити в такому порядку:

1. Визначаємо змінну напругу, яка повинна бути на вторинній обмотці мережевого трансформатора:

$$U_2 = B U_{вих}$$

В даному випадку струм навантаження становить $I_{живл} = 406 \text{ мА}$, який споживає електрична схема (табл. 2).

$$U_2 = 1,9 \cdot 5 = 9,5 \text{ В},$$

де $U_{вих}$ - постійна напруга на навантаженні, В;

B - коефіцієнт, залежний від струму навантаження, який знаходимо за табл. 4.

Таблиця 4

Коефіцієнт	Струм навантаження, А					
	0,1	0,2	0,4	0,6	0,8	1
В	0,8	1	1,9	1,4	1,8	1,7
С	2,4	2,2	2,0	1,9	1,8	1,8

2. За струмом навантаження визначаємо максимальний струм, який протікає через кожен діод випрямного мосту:

$$I_D = 0,5 \cdot C \cdot I_{жс},$$

$$I_D = 0,5 \cdot 2,0 \cdot 0,406 = 0,406 \text{ А}$$

де I_D - струм через діод, А;

$I_{жс}$ - максимальний струм навантаження, А;

C - коефіцієнт, залежний від струму навантаження (визначають по табл. 3)

3. Підраховуємо зворотну напругу, яка буде прикладена до кожного діода випрямляча:

$$U_{звр} = 1,5 \cdot U_{жс},$$

$$U_{звр} = 1,5 \cdot 5 = 7,5 \text{ В}$$

де $U_{звр}$ - зворотна напруга, В;

$U_{н}$ - напруга на навантаженні, В.

4. Вибираємо діоди, у яких значення випрямленого струму і допустимої зворотної напруги рівні або перевищують розрахункові.

Із довідника вибираємо діоди КД212А:

- імпульсна зворотна напруга 200 В,

- максимальний випрямлений струм 1 А.

5. Визначаємо ємність конденсатора фільтру:

Коефіцієнт фільтрації:

$$S_{к.ф.} = q / q_{вих.}$$

$$S_{к.ф.} = 0,67 / 0,1 = 6,7$$

q – коефіцієнт пульсації після двопівперіодного діодного містка ($q = 0,67$);

6. Визначаємо ємність конденсатора: $C = \frac{S_{к.ф.}}{2\pi \cdot m \cdot f \cdot R_n}$

$$C = \frac{6,7}{2 \cdot 3,14 \cdot 2 \cdot 50 \cdot 12,3} = 867,38 \text{ мкФ};$$

m – номер основної гармоніки ($m = 2$)

f – 50 Гц

R_n – опір навантаження;

$$R_n = \frac{U_{вих}}{I_{жс}} = 12,3 \text{ Ом}$$

Таким чином, ємність конденсатора $C_2 > 1 \text{ мкФ}$.

2. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ НА ОСНОВІ ОМЕОМ K1816BE51 (80C51)

2.1. Однокристалні мікро-ЕОМ сімейства МК51 (МС51). Основні характеристики

Однокристалні мікро-ЕОМ (ОМЕОМ) сімейства МК51 відносяться до пристроїв з байтовою організацією, тобто основною одиницею обробки інформації в цих ОМЕОМ є байт, що містить 8 двійкових розрядів. Розроблені по n-МОН технології, ОМЕОМ зберегли свою архітектуру при переході на технологію КМОН, що дозволило, зберігши повну наступність апаратних і програмних засобів, більш ніж на порядок знизити енергоспоживання виробу. В даний час серійно випускаються тільки ОМЕОМ, виконані за КМОН технологією.

Базовим кристалом сімейства є ОМЕОМ K1816BE51 (80C51), що має наступні характеристики:

- розмір резидентної пам'яті програм, Кбайт	4
- тип резидентної пам'яті програм	РПЗП
- розмір резидентної пам'яті даних, байт	128
- мінімальна частота проходження тактових сигналів, МГц	1,2
- максимальна частота проходження тактових сигналів, МГц	12
- напруга живлення, В	+5+10%
- струм споживання, мА	8
- розмір зовнішньої адресованої пам'яті програм, Кбайт	64
- розмір зовнішньої адресованої пам'яті даних, Кбайт	64

До складу ОМЕОМ 80C51 входять наступні додаткові пристрої:

- чотири восьмирозрядні паралельні порти вводу\виводу;
- два 16-ти розрядні таймери - лічильники;
- послідовний порт;
- тактовий генератор;
- блок регістрів спеціальних функцій;
- ЗП криптограм;
- система захисту ЗП від несанкціонованого доступу.

Система команд ОМЕОМ 80C51 містить 111 базових команд.

Дворівнева система переривань підтримує переривання від 5 джерел.

Якщо перед зазначеним позначенням є літери, то вони означають конструктивне виконання:

D - керамічний корпус DIP 40 виводів;

P - пластиковий корпус DIP 40 виводів;

N - корпус PLCC, 44 виводи.

Керамічний корпус DIP має вбудоване вікно, закрите кварцевим склом, для стирання ультрафіолетовим випромінюванням записаної в ЗП програми.

Випускаються також мікросхеми з підвищеною швидкодією - із граничним значенням тактової частоти 16, 20 і 24 МГц; граничне значення тактової частоти в цьому випадку вказується після основного позначення.

2.2 Архітектура ОМЕОМ 80C51

Пам'ять програм і пам'ять даних в ОМЕОМ сімейства МК51 не тільки фізично і логічно розділені, але мають різні тип і систему адресації.

В той же час, для звертання до інформації, що міститься в пам'яті даних і пам'яті програм, використовується та сама восьмирозрядна шина, що в значній мірі сповільнює процес обміну даними і тим самим знижує загальну продуктивність .

Структурна організація ОМЕОМ 80C51.

Позначення ОМЕОМ на електричній схемі показано на рис. 25, а, блок-схема – на рис. 25, б, а позначення та призначення виводів – в табл. 5.

До складу ОМЕОМ входять наступні функціональні вузли:

ЦПП	-	центральний процесорний пристрій (англ. CPU - Central Processor Unit);
ПЗП ПП	-	постійний запам'ятовуючий пристрій пам'яті програми (англ. ROM - Read Only Memory);
ОЗП ПД	-	оперативний запам'ятовуючий пристрій пам'яті даних (англ. RAM-Random Access Memory);
ГЕН	-	генератор, (англ. OSCillator);
ППП	-	програмовані паралельні порти (англ. PROGR.I/O - PROGRammable Input/Output ports);
Посл. П	-	послідовний порт (англ. SPORT - Serial PORT);
Т/Л	-	таймери/лічильники (англ. Timers/Counters);
РШ	-	розширник шини для роботи з зовнішніми ЗП ємністю до 64 Кбайт (англ. 64 KBYTE BUS EXPansion).

Усі вузли зв'язані між собою загальною восьмирозрядною шиною, по якій здійснюється обмін інформацією між ЦПП та іншими пристроями.

ЦПП є сукупністю операційного і керуючого пристроїв, що виконують програму, записану в ПЗП ПП, ємність якого 4Кбайт. ЦПП забезпечує виконання наступних груп операцій:

- арифметичні операції (додавання, додавання з врахуванням перенесення, віднімання з врахуванням позичання, беззнакове множення і ділення, інкремент і декремент, десяткова корекція).
- логічні операції (І, АБО, виключне АБО, інверсія);
- операції зсуву;
- операції пересилання;
- бітові операції;
- операції передачі керування.

Проміжні результати обчислень зберігаються в ОЗП ПД ємністю 128 байт.

Швидкість роботи ЦПП задається генератором, що виробляє необхідні для роботи часові послідовності. Тактова частота задається або кварцовим резонатором, що включається між виводами X1 і X2, або зовнішнім генератором, що підключається до входу X1. З метою забезпечення послідовного доступу до ресурсів процесора при використанні однієї шини генератор формує машинний цикл процесора з дванадцяти тактів резонатора (задаючого генератора).

Машинний цикл містить 6 станів керуючого автомату S1...S6, кожен стан розбитий на дві фази P1, P2, що відповідає різним діям процесора.

Ввід в процесор інформації, що підлягає обробці, може бути здійснений або в паралельній байтовій (ввід восьми розрядів однією командою) або в послідовній (по одному біту) формах, також як і вивід результатів обробки з процесора.

Паралельний обмін інформації можливий через один з чотирьох підтримуваних ОМЕОМ ППП.

Послідовний обмін інформацією може бути організований через будь-який з розрядів ППП, однак для полегшення процесу послідовного обміну і економії обчислювальних ресурсів, необхідних для його реалізації, ОМЕОМ містить вбудований програмований послідовний порт, що дозволяє практично без витрат обчислювальних ресурсів організувати послідовний обмін по декількох видах протоколів.

Крім розглянутих вузлів, до складу ОМЕОМ включені два шістнадцятирозрядні таймери/лічильники, що можуть функціонувати або в режимі таймера, або в режимі лічильника зовнішніх подій.

Режим таймера використовується, головним чином, тоді, коли необхідно організувати циклічні процеси з жорстко фіксованим і незалежним від часу виконання програми періодом циклу, наприклад, при обробці умовних сигналів, коли необхідно забезпечити необхідний (по теоремі Котельникова) інтервал дискретизації.

Режим лічильника зовнішніх подій використовується, наприклад, при підрахунку кількості імпульсів, вимірі частоти і т.п.

Розширювач шини (РШ) використовується для роботи з зовнішнім ЗП - пам'яті програм чи пам'яті даних. Як правило, зовнішнє ЗП використовується тоді, коли для розміщення програми чи даних при вирішенні якоїсь задачі внутрішніх ресурсів ОМЕОМ виявляється недостатньо; режим роботи з зовнішнім ЗП не є типовим для ОМЕОМ.

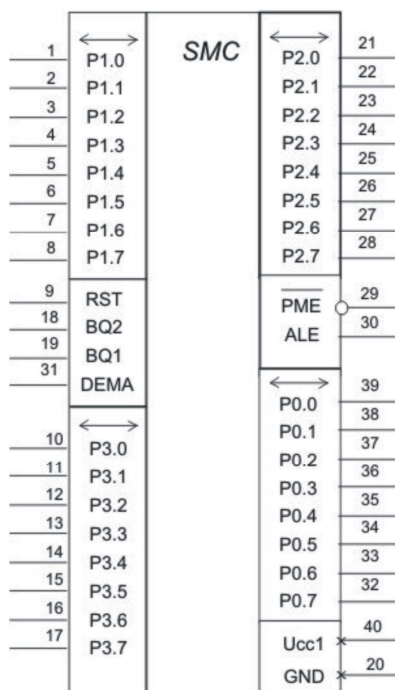
Режим звертання до зовнішнього ЗП, тим часом, використовується не тільки за прямим призначенням. Так, останнім часом відбувається оснащення спеціалізованих мікро-ЕОМ зовнішньою шиною, сумісною за інтерфейсом з ОМЕОМ 80С51 в режимі обміну з зовнішньою пам'яттю, що дозволяє легко програмувати режими спеціалізованих ОМЕОМ і керувати їх роботою за спеціальною програмою безпосередньо з 80С51.

Сам ЦПП розділений на керуючий пристрій КП (англ. CU - Control Unit) і чотири блоки, що є операційним пристроєм: АЛП (англ. ALU), регістри тимчасового збереження операндів TR1 і TR2, і регістр ознак PSW (англ. Program Status Word).

Крім того, з резидентної оперативної пам'яті RAM виділені три регістри спеціальних функцій: акумулятор (А), регістр В і регістр покажчика стеку SP, а блок РШ представлений покажчиком даних DPTR (англ. Data PoinTe) з асоційованими шинами обміну і лічильником команд PC (англ. Program Counter), призначення якого полягає у формуванні адреси команди.

Є два важливих аспекти застосування ОМЕОМ 80С51:

- при використанні зовнішнього ЗП його адресація здійснюється через виводи портів P0 (молодший байт) і P2 (старший байт), а обмін інформацією (ввід кодів команд, вводу\виводу даних) - через виводи порту P0, що, якщо не застосовувати додаткові апаратні засоби, приводить до втрати цих портів для використання;
- таймери/лічильники, послідовний порт і система переривань не мають своїх виводів з корпусу ОМЕОМ, а використовують виводи порту P3 (це називається альтернативними функціями виводів порту P3), і, таким чином, використання таймерів/лічильників у режимі лічильників зовнішніх подій, ліній послідовного порту і зовнішніх переривань знижує розрядність порту P3.



a)

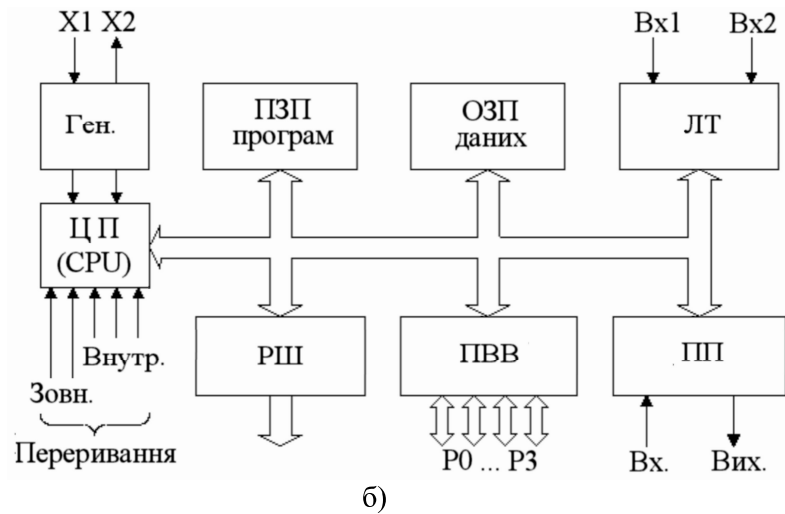


Рис. 25 ОМЕОМ 80С51: а) позначення на електричній схемі, б) блок-схема

Мінімальна схема включення ОМЕОМ 80С51 представлена на рис. 26.

До виводів X1 і X2 ОМЕОМ підключена резонансна схема, що включає кварцовий резонатор.

До входу скидання RST - схема автоматичного рестарту ОМЕОМ при ввімкненні живлення.

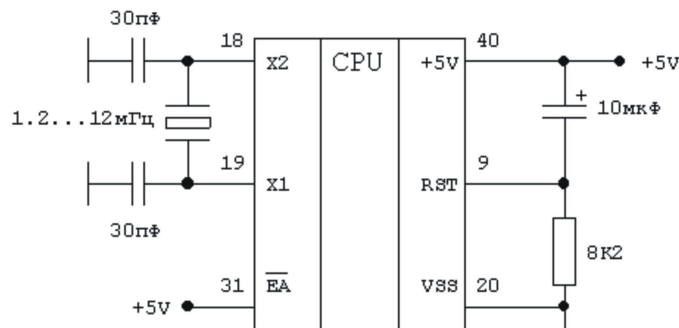


Рис. 26 Схема під'єднання резонатора та автоматичного рестарту до МК51

Таблиця 5

Вивід	Символ	О – вихід; І – вхід	Призначення
1	P1.0	I/O	Виводи паралельного порту P1
2	P1.1	I/O	
3	P1.2	I/O	
4	P1.3	I/O	
5	P1.4	I/O	
6	P1.5	I/O	
7	P1.6	I/O	
8	P1.7	I/O	
9	RST	I	Вхід рестарту ОМЕОМ

10	P3.0/RxD	I/O	Розряд 0 порту P3 або вхід прийому SPORT
11	P3.1/TxD	I/O	Розряд 1 порту P3 або вихід передачі SPORT
12	P3.2/INT0	I/O	Розряд 2 порту P3 або вхід зовнішнього переривання INT0
13	P3.3/INT1	I/O	Розряд 3 порту P3 або вхід зовнішнього переривання INT1
14	P3.4/T0	I/O	Розряд 4 порту P3 або вхід лічильника T0
15	P3.5/T1	I/O	Розряд 5 порту P3 або вхід лічильника T1
16	P3.6/WR	I/O	Розряд 6 порту P3 або вихід WR запису у зовнішній ЗП
17	P3.7/RD	I/O	Розряд 7 порту P3 або вихід RD читання із зовнішнього ЗП
18	BQ2	O	Вивід для підключення резонансної схеми
19	BQ1	I	Вивід для підключення резонансної схеми або вхід підключення зовнішнього генератора
20		I	Вивід підключення від'ємного полюсу джерела живлення (земля)
21	P2.0	I/O	Виводи паралельного порту P2
22	P2.1	I/O	
23	P2.2	I/O	
24	P2.3	I/O	
25	P2.4	I/O	
26	P2.5	I/O	
27	P2.6	I/O	
28	P2.7	I/O	
29	PME	O	Звернення до зовнішньої пам'яті програм
30	ALE	O	Строб адреси зовнішнього ЗП
31	DEMA	I	Вибір роботи ОМЕОМ з внутрішньою (DEMA=1) або зовнішньою (DEMA=0) пам'яттю програм
32	P0.7	I/O	Виводи порту P0
33	P0.6	I/O	
34	P0.5	I/O	
35	P0.4	I/O	
36	P0.3	I/O	
37	P0.2	I/O	
38	P0.1	I/O	
39	P0.0	I/O	
40		I	Вивід підключення "+" джерела живлення

2.3. Використання зовнішньої пам'яті і розширеного вводу/виводу

Тоді, коли функціонально-логічних можливостей МК51 є недостатньо, можна відносно простими засобами розширити МК-систему до наступних розмірів:

- пам'ять програм – до 64 Кбайт;
- пам'ять даних - до 64 Кбайт;
- лінії вводу/виводу - практично необмежено.

Крім того, шляхом підключення спеціалізованих ВІС, що входять у МП-комплект КР580, у МК-системі можуть бути реалізовані різні допоміжні функції: зв'язок з дисплеєм і клавіатурою, багаторівнева система переривань, складна система таймерування, зв'язок з телеграфно-телефонними лініями передачі інформації і т.д. за схемами, які використовуються для роботи з ОМЕОМ 80С51.

2.3.1. МК - системи з зовнішньою пам'яттю програм

Шина BUS (P0) за своїми властивостями подібна до двонаправленої шини даних мікропроцесора КР580, і всі розширення МК виконуються для цієї шини. При звертанні до резидентної пам'яті програм МК не генерує зовнішніх керуючих сигналів (за винятком ALE, що завжди ідентифікує кожен машинний цикл). Починаючи з адреси 4096, МК автоматично формує керуючі сигнали, що забезпечують вибірку команд із зовнішньої пам'яті об'ємом до 64 Кбайт.

Послідовність процесу вибірки команди з зовнішньої пам'яті така:

- вміст лічильника команд виводиться через порт P0 (BUS) і порт P2 (P2.0...P2.7);
- по зрізу сигналу ALE на зовнішньому регістрі фіксується адреса;
- сигналом PМЕ дозволяється робота зовнішньої пам'яті;
- по спаду сигналу PМЕ шина BUS переходить у режим вводу.

На рис. 27 показано структуру МК-системи з зовнішньою пам'яттю програм.

Три додаткові мікросхеми пам'яті (DD4, DD5, DD6) ємністю по 8 Кбайт підключаються до шини BUS своїми інформаційними виходами.

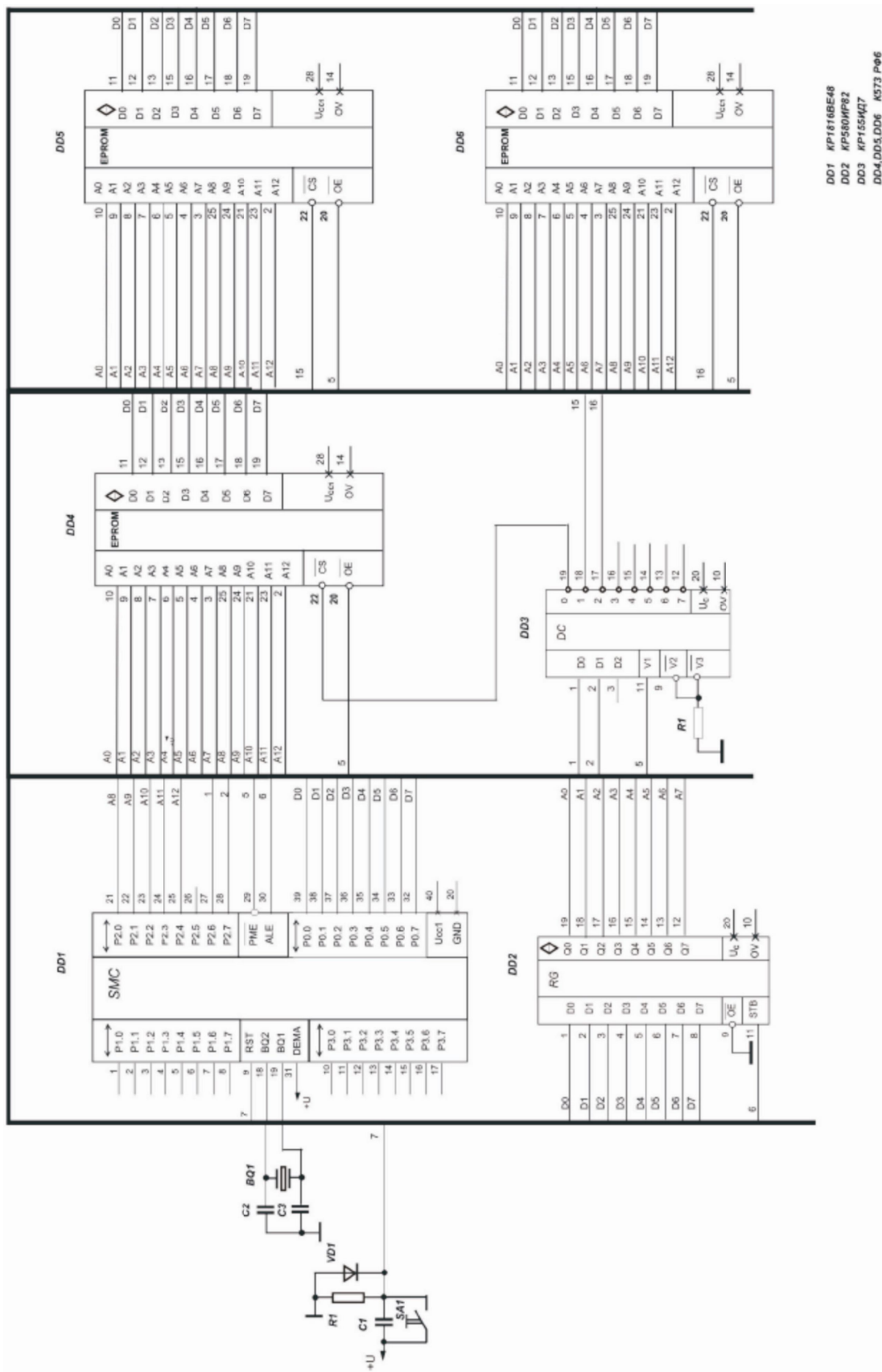
Молодший байт адреси за сигналом ALE фіксується на зовнішньому буферному регістрі DD2.

Старша тетрада адреси, яка виведена через порт P2, не має потреби в буферизації, тому що вона зберігається протягом усього циклу вибірки.

Два старші біти адреси (P2.6, P2.7) заводяться на зовнішній дешифратор DD3 (стробується сигналом PМЕ) для вибору необхідної мікросхеми пам'яті програм.

На рис. 28 показано схему МК-системи, до складу якої включена додаткова мікросхема ОЗП (DD3), що реалізує пам'ять ємністю 2 Кбайт.

Сигналом ALE непряма адреса, виведена по шині BUS, фіксується в буферному регістрі DD2. Сигнали W і R визначають режим роботи ОЗП. Схема на рис. 28 забезпечує адресацію 2 Кбайт з можливих комірок зовнішнього ОЗП на додаток до 256 комірок резидентної пам'яті даних (РПД) МК51.



DD1 КР1816ВЕ48
 DD2 КР550МР82
 DD3 КР1554КД7
 DD4, DD5, DD6 К573 Р96

Рис. 27 МК51 з зовнішньою пам'яттю програм (ПЗП)

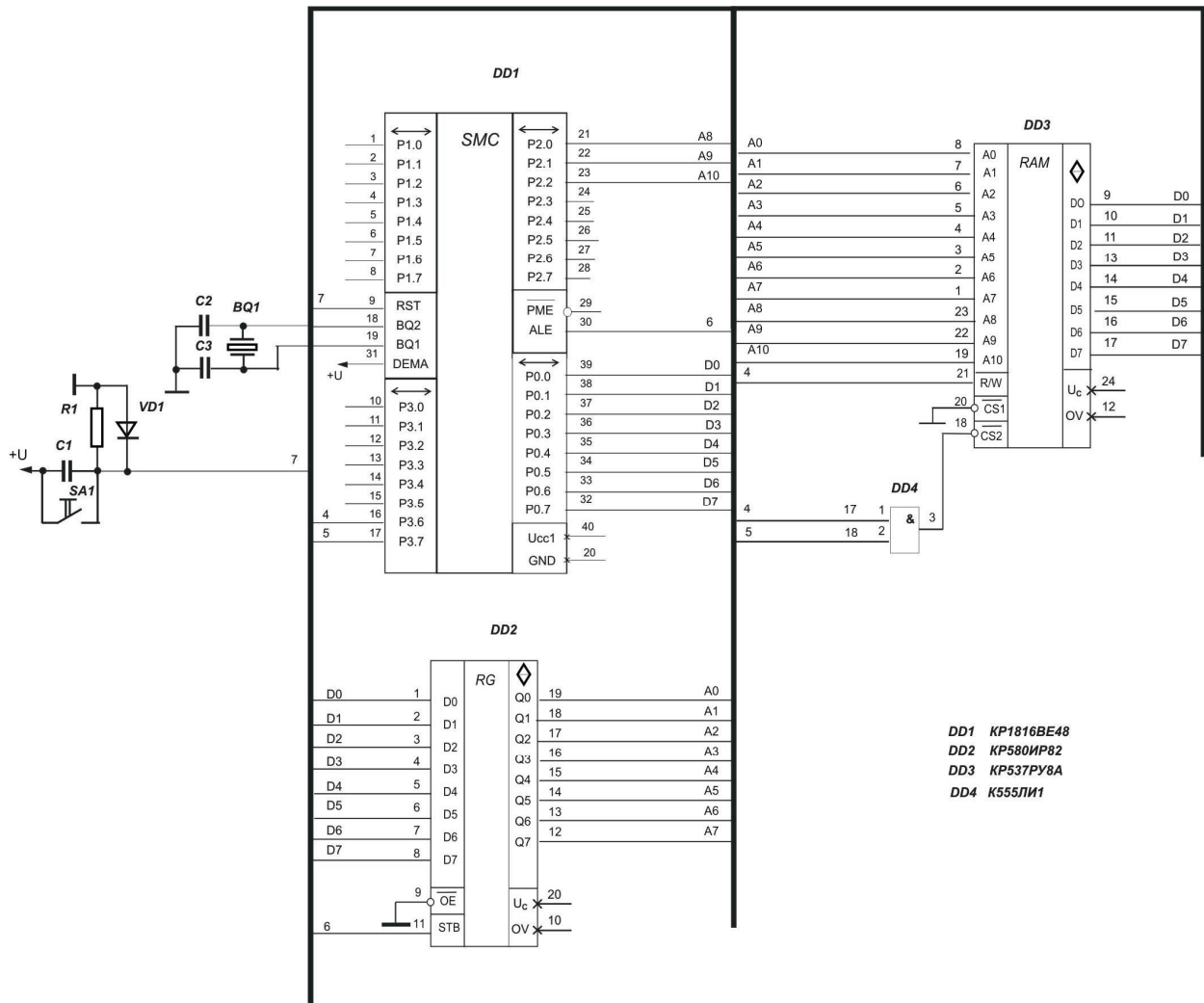


Рис. 28 МК51 з зовнішньою пам'яттю даних (ОЗП)

2.3.2. МК-система з розширенням вводом/виводом

Для з'єднання МК51 з об'єктом, що має велику кількість входів/виходів, можна розширити резидентну систему вводу/виводу за рахунок зовнішніх портів.

Таке розширення може бути виконане двома способами:

- з використанням стандартного розширювача вводу/виводу (РВВ) KP580BP43,
- інтерфейсних ВІС (KP580BB55, KP580BB51).

Розширювач підключається до МК51 так, як показано рис. 29. Кожен з чотирьох портів РВВ може використовуватися для вводу чи виводу інформації незалежно від інших і забезпечує високу навантажувальну здатність.

Для виводу байту даних в порти P4 і P5 розширювача можна скористатися наступною послідовністю команд:

MOVD P4, A ; вивід A(0...3) в порт 4
 SWAP A ; обмін тетрад акумулятора
 MOVD P5, A ; вивід другої тетради в порт 5.

Розширення РВВ за допомогою програмованих інтерфейсів серії KP580 здійснюється аналогічно до схем для мікропроцесора KP580BM80A.

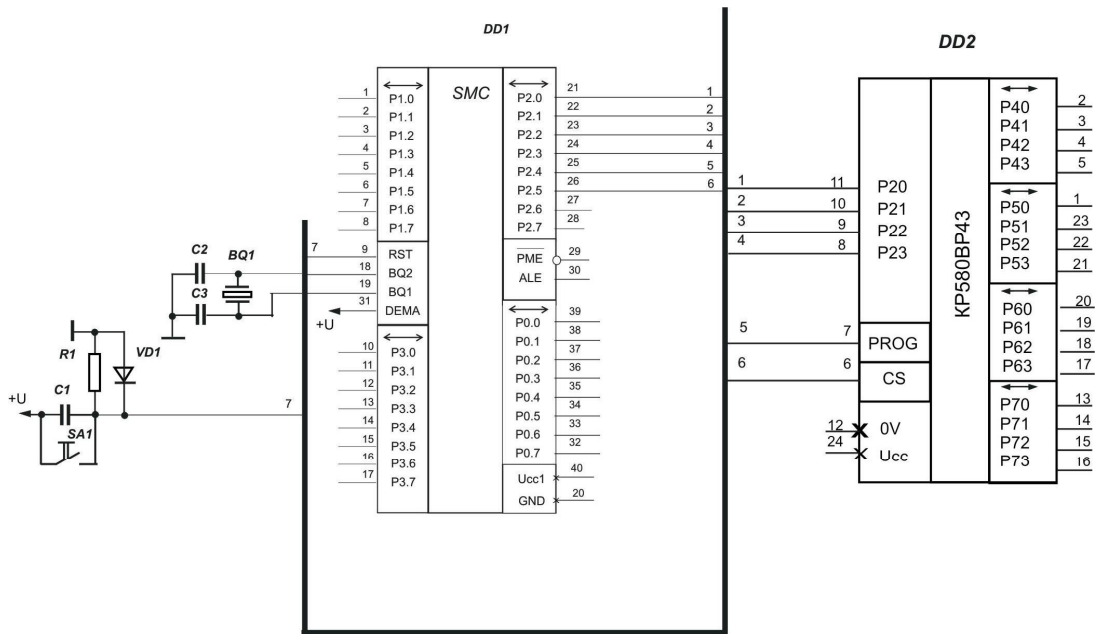


Рис. 29 Розширення ліній вводу-виводу МК51 за рахунок розширювача KP580BP43

Нижче показано два варіанти розширення вводу/виводу з використанням програмованого паралельного інтерфейсу KP580BV55.

У першому варіанті (рис. 30) порти адаптера адресуються як комірки зовнішньої пам'яті даних (ЗПД), доступ до яких здійснюється за командами MOVX.

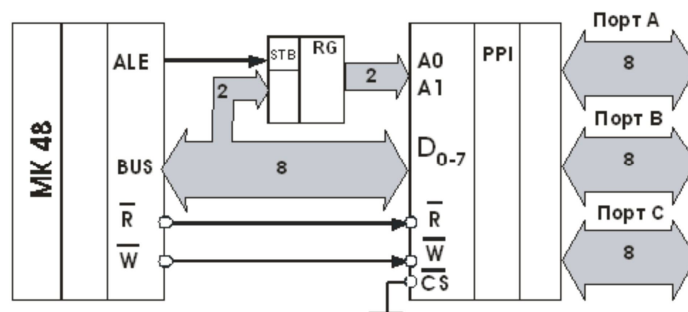
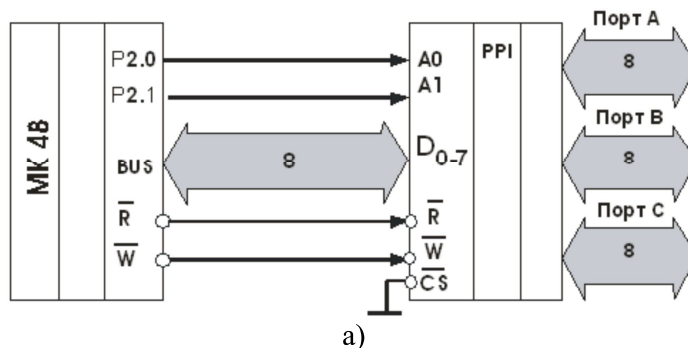


Рис. 30 Адресація PPI через порт BUS з використанням адресного регістру

Для другого варіанту підключення ППІ (рис. 31), вибір порту здійснюється через установку/скидання двох розрядів порту P2, стан яких зберігається протягом всього циклу, а тому в зовнішньому адресному регістрі немає потреби.



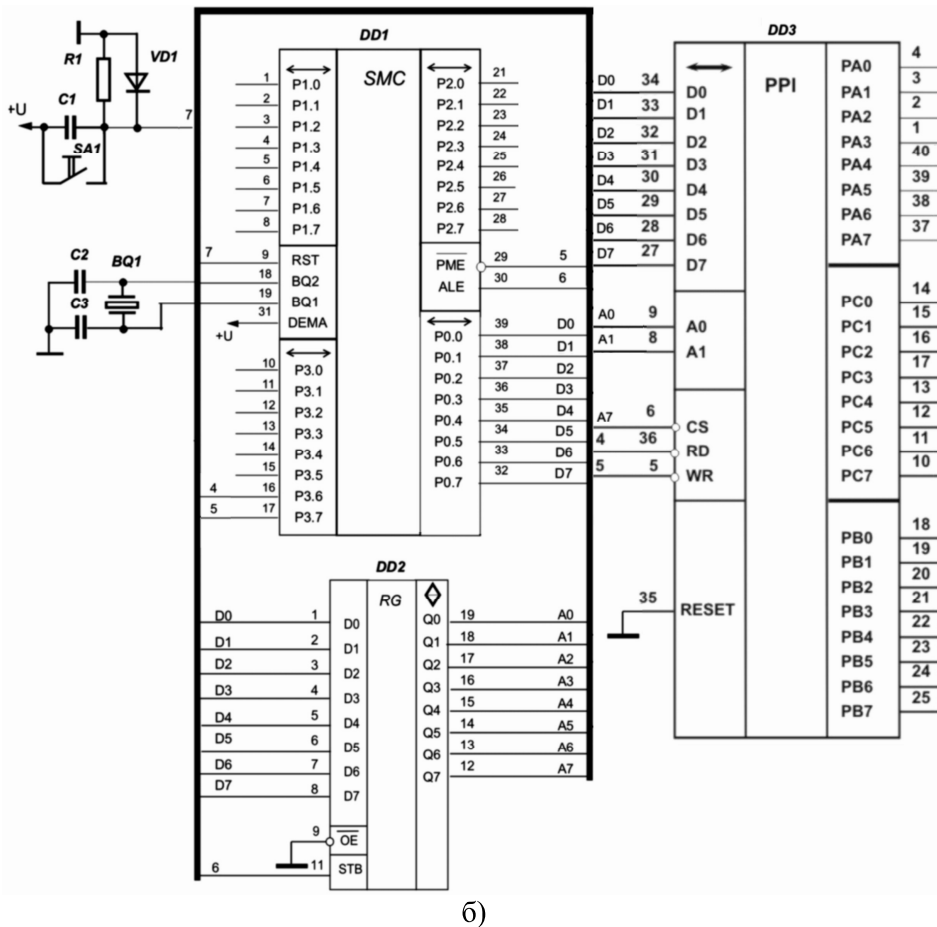


Рис. 31 Адресація PPI через порт P2 без використання адресного регістру:
 а) функціональна схема, б) принципова електрична схема

У розглянутих схемах розширення використовується тільки по одній зовнішній ВІС, тому їхній вхід CS приєднується до землі. При необхідності, до МК може бути підключено декілька PVB і PPI.

2.3.3. Під'єднання клавіатури та дисплею

Під'єднання клавіатури та дисплею, ввід-вивід цифрових та аналогових сигналів в мікропроцесорних системах на основі МК51 виконується за тими ж правилами, що і для схем на основі мікропроцесора КР580ВМ80А (рис. 11... рис. 16).

Процедуру вводу інформації з матричної клавіатури, яка не кодована попередньо, зручно розглянути на прикладі клавіатури 4 * 5, що включає 16 цифрових клавіш (0 - F) і 4 керуючих.

Структура матриці клавіатури аналогічна структурі матриці двійкових давачів, а схема підключення клавіатури до МК представлена на рис. 32.

Лінії порту P1 використовуються для сканування, а лінії порту P2 - для опитування матриці клавіш.

Кожна клавіша в такій матриці має свій номер, що відповідає місцю її розташування. На цифрові клавіші можна нанести позначення, що відповідають їх кодам (від 0 по F).

Коди керуючих клавіш більші за число 15. Діоди забезпечують захист від замикання між собою скануючих ліній у випадку одночасного натискання більш ніж однієї клавіші.

Процедура вводу коду натиснутої клавіші складається з послідовності таких процедур (деякі з них уже були розглянуті раніше):

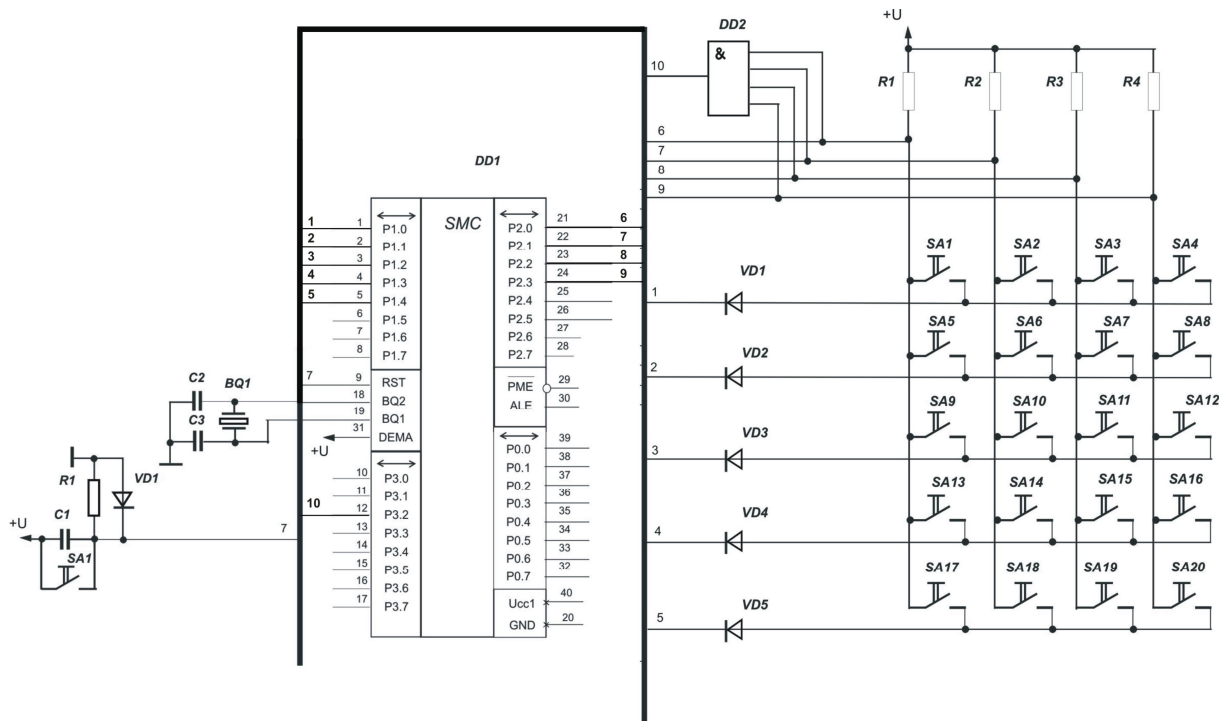


Рис. 32 Під'єднання до МК51 клавіатури

- сканування матриці клавіш;
- усунення деренчання контактів;
- очікування звільнення клавіші;
- ідентифікації коду натиснутої клавіші.

Для деякого типу клавіатур може бути відсутня процедура усунення деренчання контактів (клавіатури на основі герконів). Процедуру сканування іноді буває зручно поєднати з процедурою ідентифікації.

Для програмного вводу інформації з клавіатур, що є некодованими, характерний один недолік, а саме - спрацьовування по відпусканні клавіші, а не по її натисканню. Однак при короткочасних натисканнях клавіш цей ефект не має особливого значення.

2.3.4. Під'єднання ССІ до МК51

Семисегментні індикатори (ССІ) використовуються для відображення цифрової і буквенної інформації. Сім відображаючих елементів дозволяють висвічувати десяткові і шістнадцаткові цифри, деякі літери кирилиці і латинського алфавітів, а також деякі спеціальні знаки. Структура ССІ і способи його підключення до МК показані на рис. 33.

Для засвічування одного сегменту більшості типів ССІ необхідно забезпечити протікання через сегмент струму 10-15 ма при напрузі 2,0-2,5 В. Низька навантажувальна здатність МК не допускає прямого з'єднання їх із ССІ.

Як проміжні підсилювачі струму, можуть використовуватися логічні елементи серії К155 або інтегральні схеми перетворювачів кодів для керування ССІ.

Перетворення двійкових кодів у коди для ССІ може здійснюватися або програмно, або апаратно з використанням перетворювачів К514 ИД1, К514 ИД2, 176 ИД2, 564 ИД5 (за прикладом рис. 33).

При використанні зовнішніх (відносно МК) схем перетворювачів кодів процедура індикації одного символу зводиться до видачі двійкового коду символу у відповідний порт виводу МК.

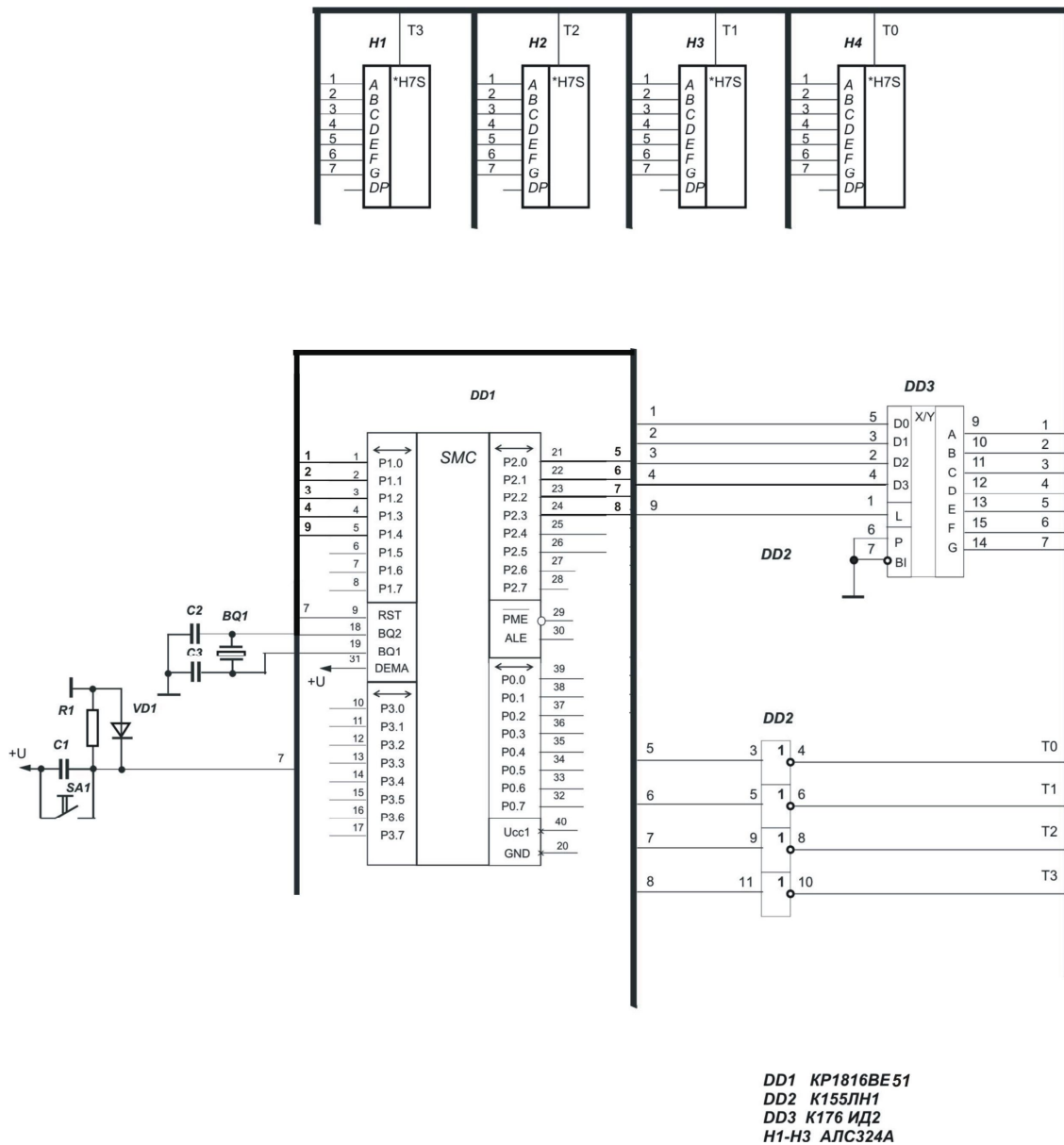


Рис. 33 Під'єднання до МК51 семисегментних індикаторів

2.4. Приклад побудови практичних схем на основі МК51

2.4.1. Вимірювання температури термопарою

Наступна схема дозволяє виміряти температуру об'єкту за допомогою термопари К-типу (хромель-алюмелеві — ТХА), причому діапазон вимірювання лежить в межах від 0 до 1024 °С.

Вимірювання температур за допомогою термопар (thermocouples), так само як, наприклад, за допомогою резистивних датчиків температури (RDT), найбільш часто використовуються в промисловості та наукових дослідженнях.

Роль вимірювального перетворювача сигналу термопари виконує мікросхема MAX6675, цифровий сигнал з якою по інтерфейсу SPI (синхронний послідовний інтерфейс) передається в мікроконтролер, як показано на рис. 34.

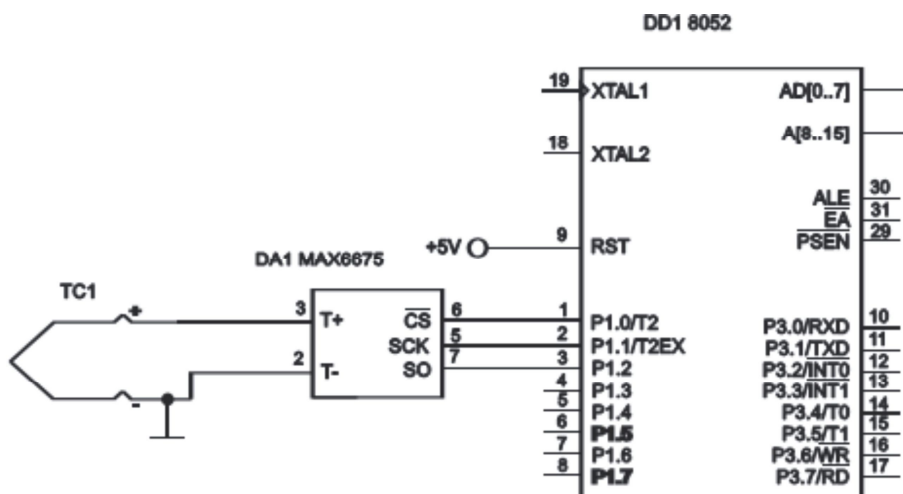


Рис. 34 Вимірювання температури з використанням інтерфейсу SPI

Мікросхема MAX6675 здійснює повну первинну обробку сигналу термопарі К-типу, виконуючи розрахунок компенсування ЕРС для холодного спаю і перетворення отриманого сигналу в цифрову форму. Далі 12-розрядний цифровий сигнал передається по SPI-сумісному інтерфейсу в мікроконтролер (8051, 8052 та ін.).

Даний перетворювач дозволяє виміряти температуру до $1024\text{ }^{\circ}\text{C}$ з похибкою $0,25\text{ }^{\circ}\text{C}$.

Мікросхема підключається до термопарі наступним чином: вивід «Т +» з'єднується з хромелевим виводом термопарі, а вивід «Т-» з алюмелевими виводами термопарі.

Вивід SCK мікросхеми використовується для прийому синхронізуючих імпульсів, які ініціюють прийом даних по лінії SO по наростаючому фронту.

Сигнал CS, встановлений в низький рівень, дозволяє прийом даних по інтерфейсу.

Сигнали SCK і CS є вхідними для мікросхеми, а SO - вихідним.

2.4.2. Вимірювання компонентів електричних схем через вимірювання частоти

Вимірювання частоти дуже широко використовується в промисловості і в лабораторних дослідженнях при:

- аналізі сигналів віддалених давачів, переданих як послідовність імпульсів;
- для вимірювання величин аналогових сигналів, що надходять від віддалених об'єктів через перетворювачі «напруга - частота» (V to F conversion).

На вимірі частоти базується методика вимірювання значень активних компонентів електричних ланцюгів.

В основному, для цих цілей використовуються генераторні схеми, в яких часозадаючі елементами є ємності і опір (RC-генератори для низьких частот) або індуктивності і ємності (LC-генератори для високих частот).

Наприклад, якщо є схема генератора, керованого напругою (VCO, Voltage Controlled Oscillator), показана на рис. 35, то з її допомогою можна легко виконати як вимірювання величини невідомого опору R_t при відомому значенні ємності C_t , так і навпаки, за відомим значенням R_t можна знайти невідому ємність C_t при фіксованих значеннях вхідної напруги V_{in} .

У загальному випадку для подібних вимірювань найчастіше використовуються генератори, керовані напругою, подібні тому, схема якого зображена на рис. 35. Для такої і подібних їй схем, зафіксувавши два з трьох параметрів (V_{in} , R_t , C_t), по виміряному значенню частоти F_{out} схеми легко можна визначити або вхідну напругу (для перетворювача V to F), або опір чи ємність.

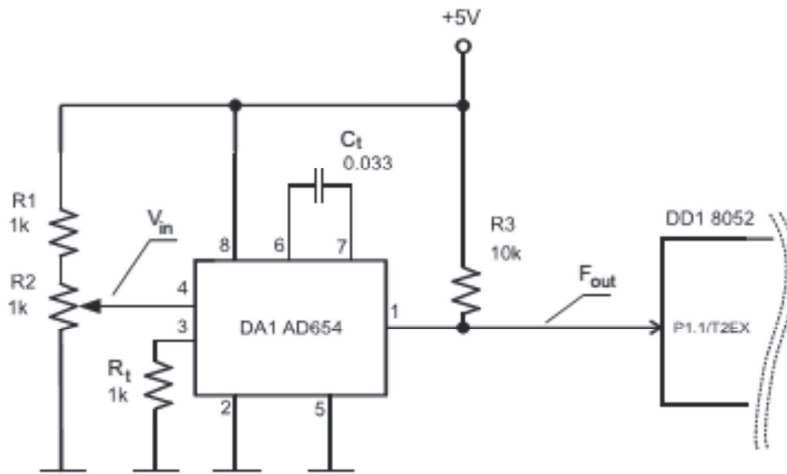


Рис. 35 Вимірювання значень компонентів електричних схем

Важливе зауваження: сама електрична схема подібного перетворювача повинна забезпечувати високу точність перетворення за рахунок застосування спеціалізованих чіпів, як, наприклад, AD654 фірми Analog Devices або AD537 тієї ж фірми.

2.4.3. Керування швидкістю обертання двигуна постійного струму

Мікроконтролери 8051/8052 дозволяють досить легко реалізувати ШІМ, причому це рішення засноване на застосуванні таймера. Багато сучасних 8051-сумісних пристроїв мають вбудовані функціональні вузли для реалізації ШІМ.

Розглянемо на практичному прикладі побудову ШІМ, використовуючи таймер 2 мікроконтролера.

Схема, що ілюструє застосування ШІМ для керування електродвигуном зображена на рис. 36.

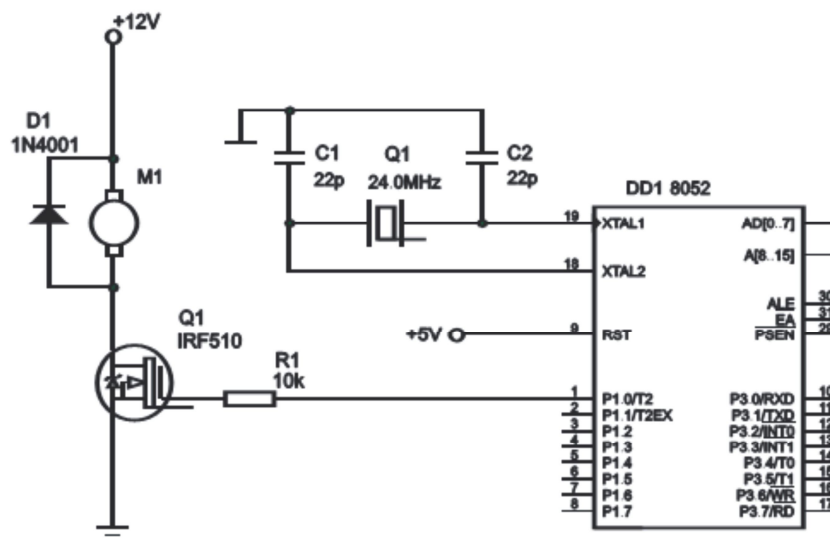


Рис. 36 Реалізація ШІМ в системі з 8051

У цій схемі електродвигун M1 управляється через потужний польовий транзистор (Power MOSFET) типу IRF510, на затвор якого подаються імпульси управління з виводу P1.0 мікроконтролера. Апаратна частина розроблена для тактової частоти 24,0 МГц, тому для інших значень частоти слід змінити параметри налаштування таймера 2, який використовується для генерації послідовності імпульсів.

Замість транзистора Q1 типу IRF510 можна використовувати будь-який інший N-канальний з підходящими характеристиками, а в якості електродвигуна M1 може бути будь-який малопотужний двигун постійного струму.

2.4.4. Ввід даних від цифрових давачів

Наступна схема дозволяє обробляти 8 сигналів від цифрових джерел по одній лінії, використовуючи мультиплексування. Апаратна частина схемотехнічно може бути реалізована так, як показано на рис. 37.

Тут для розширення кількості оброблюваних вхідних сигналів застосована мікросхема 74HC4051, що представляє собою 8-входовий мультиплексор сигналів. Залежно від двійкового коду на входах A, B, C, на виході X з'являється один із вхідних сигналів. Вхід INH мікросхеми при низькому рівні напруги дозволяє роботу мультиплексора, тому він заземлений.

Тригер Шмітта використовується для формування сигналу, що надходить на вхід P1.7 мікроконтролера.

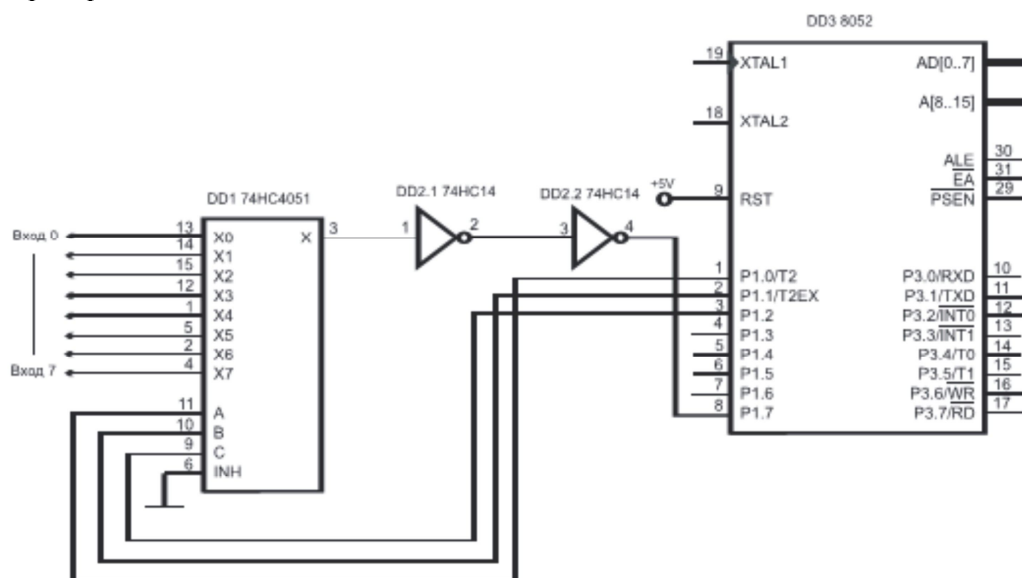


Рис. 37 Мультиплексування вводу дискретних даних

2.4.5. Перетворення паралельного коду в послідовний

На рис. 38 представлена апаратна реалізація обробки вхідних сигналів через перетворення паралельного коду в послідовний.

Для зчитування вхідних даних у цій схемі використовується регістр зсуву на мікросхемі 74HC166, що дозволяє перетворити паралельний код на входах 0-7 в послідовний на виводі SO. Послідовні дані на цьому виводі зчитуються мікроконтролером і перетворюються програмою в паралельний формат.

Використовуваний метод частково уповільнює реакцію на вхідні сигнали в порівнянні з паралельним зчитуванням даних в порт мікроконтролера по 8 лініях.

Схема працює наступним чином. Дані з входів D0 - D7 завантажуються в паралельному форматі в регістр DD1 за низьким рівнем сигналу на виводі PE мікросхеми.

Потім по фронту синхроімпульсу, що надходить на вхід CLK, дані побітово зсуваються на вихід SO, де зчитуються мікроконтролером.

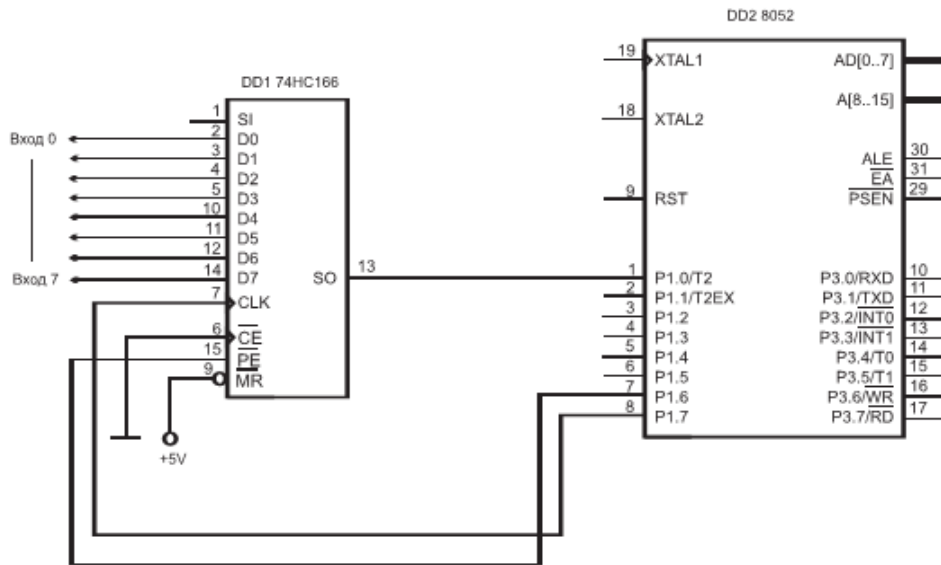


Рис. 38 Перетворення паралельних вхідних даних в послідовний код

Замість мікросхеми 74HC166 можна застосувати аналогічні чіпи, при цьому необхідно враховувати швидкодію конкретної мікросхеми і використовувати, де потрібно, програмні затримки.

2.4.6. Вимірювання аналогового сигналу з використанням МК51

Розробимо апаратно-програмний проект, в якому необхідно виміряти величину зовнішнього аналогового сигналу, що знаходиться в діапазоні 0-5 В, і вивести його значення через послідовний порт на термінальний пристрій або дисплей персонального комп'ютера.

Для цього слід під'єднати ПК через послідовний порт до мікроконтролера, потім запустити на ПК будь-яку програму, що працює з послідовним портом, і виконати її.

Принципова схема пристрою зображена на рис. 39.

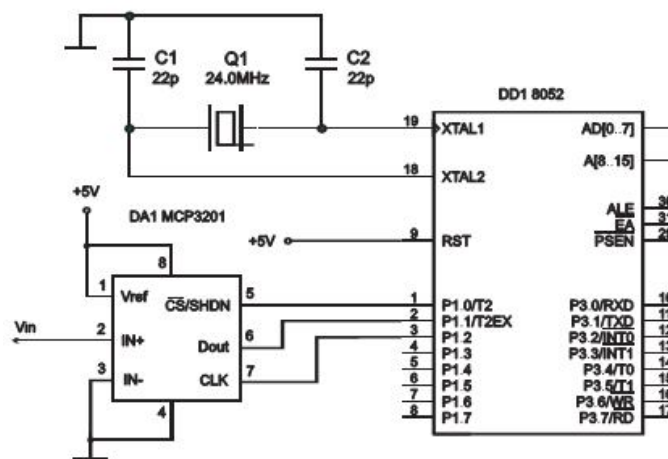


Рис. 39 Схема обробки аналогового сигналу в МК51

- АЦП позначений як DA1 і приєднаний до мікроконтролера наступним чином:
- вхід дозволу перетворення CS / SHDN з'єднується з виводом P1.0 мікроконтролера;
 - вхід синхронізації CLK з'єднується з виводом P1.2 мікроконтролера;
 - вихід даних Dout з'єднується з виводом P1.1 мікроконтролера.

Програмну частину повинна зчитувати вхідний сигнал V_{in} на вході АЦП і відобразити його значення на екрані приблизно кожні 10 сек.

2.4.7. Вимірювання аналогових сигналів від декількох джерел

У практичних розробках відносно рідко доводиться вимірювати аналогові сигнали, що надходять від одного джерела. Розглянемо проект, який дозволяє одночасно вимірювати аналогові сигнали з 8 джерел. Апаратна частина цього проекту містить мікросхему LTC1292 і мультиплексор аналогових сигналів 4051. Схема апаратної частини пристрою представлена на рис. 40.

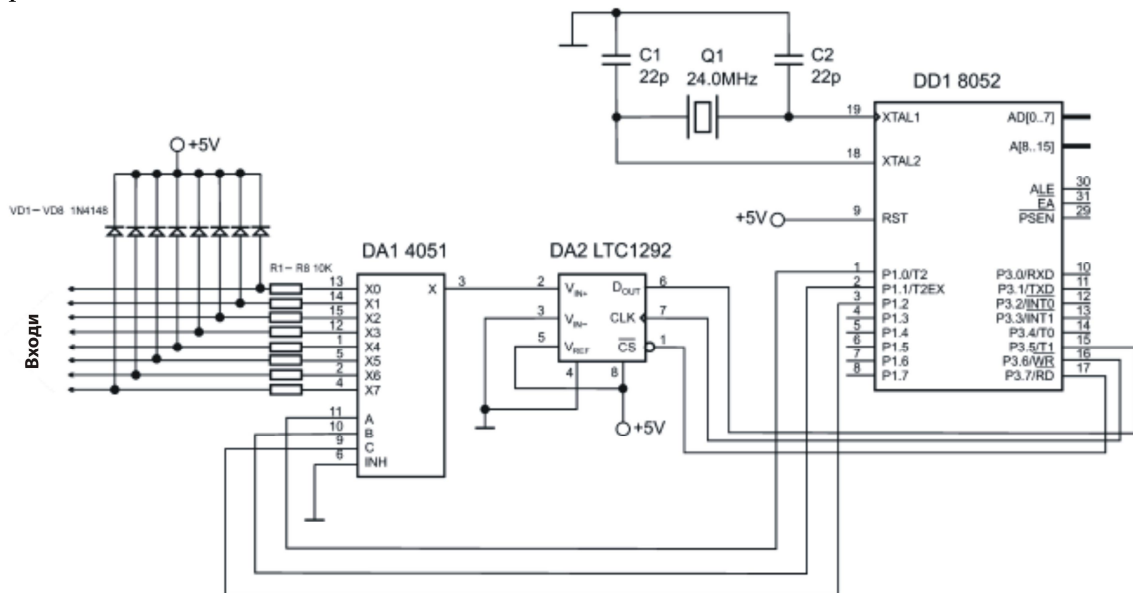


Рис. 40 Схема 8-канального АЦП на LTC1292

У цій схемі аналогові вхідні сигнали подаються на 8 входів мультиплексора на мікросхемі DA1. Мікроконтролер через певні інтервали часу, що задаються таймером 2, сканує входи і після перетворення посилає дані на термінал. Для підвищення точності перетворення можна зменшити номінали резисторів R1 - R8. На практиці, в багатьох випадках можна обійтися без обмежувачих ланцюгів на резисторах і діодах, якщо заздалегідь відомо, що вхідні напруги не перевищують 5 В.

2.4.8. Формування вихідного аналогового сигналу пристроєм на основі МК51

Розглянемо проект пристрою, в якому вихідний аналоговий сигнал генерується за допомогою мікросхеми LTC1456, що є однокристальним 12 розрядним ЦАП фірми Linear Technology.

Цей перетворювач керується 12— розрядним двійковим кодом, що надходить через модифікований варіант SPI інтерфейсу.

Виводи мікросхеми мають наступні призначення:

- CLK - строб синхронізації при прийомі даних;
- Din - вивід, на який надходять дані. Дані на цьому виводі запам'ятовуються в регістрі мікросхеми по наростаючому фронту сигналу CLK;
- CS / LD - сигнал дозволу / установки інтерфейсу.

При низькому рівні сигналу на вході дозволяється прийом даних по лінії Din, при високому рівні прийом даних по лінії Din забороняється, а прийняті дані завантажуються з регістра зсуву в регістр перетворення, змінюючи значення вихідної напруги перетворювача.

Принципова схема цифро-аналогового перетворювача з керуванням від мікроконтролера показана на рис. 41.

Програмне забезпечення для цього випадку дозволяє ввести з клавіатури терміналу або ПК необхідне значення вихідної напруги. Власне, перетворення цифрового коду в аналогову напругу виконує програма, написана на асемблері.

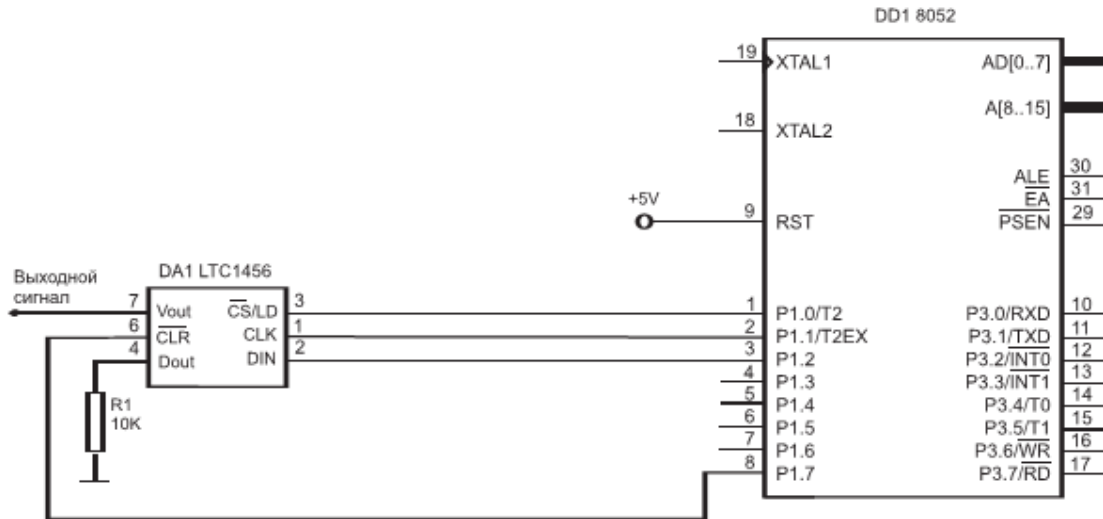


Рис. 41 Генерація вихідного аналогового сигналу з використанням LTC1456

2.5. Програмування контролера МК51

Команди асемблера МК51 приведені в Додатку 2.

2.5.1. Ввід інформації з клавіатури

Процедура вводу коду клавіші KEYBRD:

```

;версія для МК51
KEYBRD:
SCAN : ... ;сканування клавіатури
...
...
DBNC : CALL DELAY ;усунення деренчання при натисканні
WAITOP: MOV A,#0FFH
WAIT : CJNE A,P2,WAIT ;якщо клавіша натиснута, то чекати
...
...
DBNC : CALL DELAY ;усунення деренчання контакту
;при звільненні клавіші (може бути
;відсутнє)
KEYW : ... ;ідентифікація натиснутої клавіші
(може
;бути відсутнє)

```

Процедура опитування стану клавіатури. Вище відзначалося, що в МК-системах, реалізують безупинне керування, процедурі KEYBRD повинна передувати процедура опитування стану клавіатури ASK

Приклад програмної реалізації процедури ASK, оформленої у виді підпрограми, приведений нижче.

Вихідний параметр передається в основну програму через флажок перенесення, який встановлюється, якщо хоча б одна клавіша натиснута:

```
ASK : CLR A ;скидання акумулятора
```

```

CLR C      ;скидання флажка перенесення
OUTL P1,A ;вивід байту "усі нулі" для одночасного
           ;опитування всіх клавiш
IN A,P2    ;ввід байту повернення
CPL A      ;інверсія байту повернення
JZ EXIT    ;вихід, якщо немає натиснутої клавiші
CPL C      ;встановлення флажка перенесення
EXIT: RET  ;повернення

```

Підпрограма виконує одночасне опитування всіх клавiш. У випадку, якщо хоча б одна клавiша натиснута (байт повернення - не всі одиниці), встановлюється флажок перенесення, інакше – скидається

2.5.2. Опитування двійкового давача. Чекання події

```

;версія для МК51
WAITO:
    JNB P1.3,
    WAITO      ;чекання розмикання контакту давача

WAITC:  JB P1.3, WAITC ;чекання замикання контакту давача

```

2.5.3. Опитування імпульсного давача

```

;версія для МК51
WAITC:
    JB P1.3,
WAITO:  WAITC
           ;чекання P1.3=0
    JNB P1.3,
WAITO:  WAITO      ;чекання P1.3=1

```

2.5.4. Усунення “деренчання” контакту

```

;версія для МК51
DBNC:   MOV R3, #20

DBNC1:  JB P3.4,      ;ініціалізація лічильника
        DBNC
           ;якщо контакт розімкнений, то почати
        DJNZ         ;відлік опитувань спочатку
        R3,DBNC1    ;повторяти, поки R3 не стане рівним 0

```

2.5.5. Підрахунок кількості імпульсів

```

;версія для МК51
MOV TMOD,#01000000B ;налаштування лічильника 1
MOV TH1,#0          ;скидання лічильника деталей
WAITO:  JB P3.4,WAITO ;чекання включення конвейера
        SETB TCON.6   ;пуск лічильника 1
WAIT1:  JNB P3.4,WAIT1 ;чекання виключення конвейера
        CLR TCON.6    ;зупинка лічильника 1
        MOV A,TH1     ;(акумулятор) <---число деталей
EXIT:   ...          ;вихід з процедури

```

Якщо давач імпульсів під’даний до входу T1:

```

;версія для МК51
TIME EQU NOT(10000)+1 ;визначення константи TIME для відліку
;інтервалу 10 мс
MOV TMOD, #01010001B ;налаштування таймера, 16 біт. 1 -
;лічильник, 0 - таймер
CLR A ;скидання акумулятора
MOV TH1, A ;скидання таймера/лічильника 1
MOV TL1, A
MOV TH0, #HIGH(TIME) ;завантаження в таймер/лічильник 0
MOV TLO, #LOW(TIME) ;константи TIME
ORL TCON, #50H ;пуск т/л 1 і т/С0
WAIT: JBC TCON.5, EXIT ;перевірка переповнення т/С0
SJMPL WAIT ;цикл, якщо TF=0
EXIT: MOV B, TH1 ;(B) (A) <---кількість імпульсів за 10
;мс
MOV A, TL1
... ;вихід з процедури

```

2.5.6. Програмна реалізація процедури чекання заданого коду (WTCODE) для випадку підключення групи з восьми взаємозалежних статичних давачів до входів порту P1 МК:

```

;версія для МК51
WTCODE:MOV A,#10 ;завантаження в акумулятор еталонного коду
WAIT: CJNE A,P1,WAIT ;якщо кодова комбінація на входах порту P1 не
;співпала з еталонним значенням, то чекати
EXIT: ... ;вихід з процедури

```

2.5.7. Організація часової затримки тривалістю 50 мс в МК51

Вважаємо, що біт IE.7 встановлено.

```

;організація переходу до мітки NEXT при
;переповненні Т/С0
ORG 0BH ;адреса вектора переривання від Т/С0
CLR TCON.4 ;зупинка Т/С0
RETI ;вихід з підпрограми
;обробка переривання
ORG 100H ;початкова адреса програми
MOV TMOD, #01H ;налаштування Т/С0
MOV TLO, #LOW(NOT(5000-1)) ;завантаження таймера
MOV TH0, #HIGH(NOT(5000-1))
SETB TCON.4 ;старт Т/С0
SETB IE.1 ;дозвіл переривання від Т/С0
SETB PCON.0 ;перехід МК51 в режим холостого ходу
NEXT: ...

```

Для вимірювання тривалості сигналу може використовуватися таймер. Особливо ефективно його використання в МК51, що має вхід дозволу відліку (альтернативна функція на вході INT). Вимірюваний сигнал можна, наприклад, подавати на вхід INT0, вимірювання тривалості при цьому буде виконуватися в Т/С0.

Програма вимірювання тривалості "додатнього" імпульса буде мати вигляд:

```

;версія для МК51
MOV TMOD, #00001001B ;налаштування Т/С0
MOV TH0, #0 ;скидання таймера

```



```

MOV TL0, #0
SETB TCON.4           ;старт Т/С0
WAITO: JNB P3.2, WAITO ;чекання "1"
WAITC: JB P3.2, WAITC  ;чекання "0"
CLR TCON.4           ;стоп Т/С0
EXIT:   ...           ;вихід з процедури

```

2.5.8. Передача і прийом даних через послідовний порт

Вхідні параметри: А – дані, Р – ознака парності.

Передача методом опитування готовності:

```

S P OUT:
mov C,P
mov TB9,C
jnb TI,$           ;M1: jnb TI,M1
clr TI
mov SBUF,A
ret

```

Передача методом переривання:

```

INT_RSO:           ;вектор переривання
clr TI
mov SBUF,A
reti

```

Використання методу опитування готовності (з очікуванням):

```

SPIN:
jnb RI,SPIN       ;jnb RI,$
clr RI
mov A,SBUF
mov C,RB9         ;прийнята ознака
xrl C,P           ;обчислена ознака
ret

```

Використання методу опитування готовності (без очікування):

```

SPIN:
jnb RI,SP RET
clr RI
mov A,SBUF
mov C,RB9         ;прийнята ознака
xrl C,P           ;обчислена ознака
SP RET:
ret

```

Використання методу переривання:

```

INT_RSI:           ;вектор переривання
clr RI
mov SBUF,A
mov C,RB9         ;прийнята ознака
xrl C,P           ;обчислена ознака
reti

```

2.5.11. Передача символьного рядка через послідовний порт:

```
CR equ 0Dh
LF equ 0Ah
...
call STRING
db CR,LF           ;адреса в DPTR
db "Повідомлення", "$"
nop
....
STRING:
pop DPH           ;завантаження адреси повідомлення
pop DPL
STR1:
clr A
movc A, @ A + DPTR ;завантаження символу
STR2:
jnb TI, $         ;очікування готовності
clr TI
mov SBUF, A       ;видача символу
inc DPTR
clr A
movc A, @ A + DPTR ;завантаження символу
cjne A, # "$", STR2 ;цикл читання
mov A, # 1
jmp @ A + DPTR    ;повернення до основної програми
```

2.5.10. Реалізація клавіатури (ще один варіант)

Алгоритм роботи:

1. Вивід скан-коду.
2. Прийом коду натискання.
3. Якщо клавіша натиснута, повторний прийом.
4. Якщо клавіша натиснута, прийом коду відпускання.
5. Повторний прийом коду відпускання.
6. Перекодування коду клавіші.

```
PORT_STR equ P0
PORT_STLB equ P1
```

Сканування клавіатури:

```
SCAN:
mov R4, # 0       ;скидання скан-коду
mov R7, # 4       ;лічильник сканування
mov R6, 11111110b ;початковий байт сканування (рядок)
LOOP:
mov PORT_STR, R0  ;вивід байту сканування
rl A              ;зсув біту 0 вліво
mov R6, A         ;збереження поточного байту
mov A, PORT_STLB ;введення байту повернення (стовпець)
                  ;виявлення натиснутої клавіші і переведення
                  ;позиційного коду стовпця в числовий код
mov R5, # 3       ;лічильник опитувань
```

```
ROTATE:
rrc A           ;зсув байту повернення
jnc END SCAN   ;вихід при виявленні 0
inc R4         ;інкремент скан-коду
djnz R5, ROTATE
djnz R7, LOOP  ;лічильник циклів сканування
jmp SCAN
END SCAN:
```

II. ПРОЕКТУВАННЯ ЕЛЕКТРИЧНИХ СХЕМ АДРЕСНИХ ДЕШИФРАТОРІВ НА ОСНОВІ ЕЛЕМЕНТІВ ЛОГІКИ

Технічне завдання

1. Необхідно розробити схеми адресної дешифрації у залежності від вхідного набору адресних кодів A15...A12 для вибору мікросхем оперативної (ОЗП) та постійної (ПЗП) відповідно до завдання на проектування мікропроцесорної системи керування (розділ I завдання на курсове проектування, табл. 1).

2. Необхідно розробити схеми адресної дешифрації у залежності від вхідного набору адресних кодів A7...A4 для вибору мікросхем пристроїв вводу-виводу (ПВВ) відповідно до завдання на проектування мікропроцесорної системи керування (розділ I завдання на курсове проектування, табл. 1).

Вимоги до змісту пояснювальної записки

Пояснювальна записка повинна містити наступні розділи:

1. Вступ.
2. Короткий опис функціональної схеми мікропроцесорної схеми з вказанням мікросхем пам'яті (ОЗП та ПЗП) та пристроїв вводу-виводу та їх характеристик.
3. Синтезувати керуючі вихідні сигнали схеми адресної дешифрації CS_n в базисі логічних функцій I-НЕ, АБО-НЕ, І, АБО (наприклад, на основі серії K555).
4. **Скласти за даними таблиці розподілу адресного простору вдосконалені диз'юнктивні нормальні функції (ДДФ) для сигналів адресних дешифраторів.**
5. Принципові електричні схеми адресних дешифраторів для кожного з функціональних блоків (пам'яті та пристроїв вводу-виводу) до і після оптимізації.
6. Перелік використаної літератури.

1. Вступ

Електроніка охоплює великий розділ науки та техніки, що пов'язаний з вивченням та використанням різноманітних фізичних явищ, а також розробкою та застосуванням пристроїв, що працюють на основі протікання електричного струму у вакуумі, газі та твердому тілі. Промислова електроніка (застосування електроніки у промисловості, на транспорті, в електроенергетиці) є найважливішою складовою електроніки, що розглядається в широкому розумінні цього слова.

В свою чергу, в промислову електроніку входять:

1) інформаційна електроніка, до якої належать електронні системи та пристрої, які пов'язані з вимірюванням, контролем та керуванням промисловими об'єктами та технологічними процесами;

2) енергетична електроніка (перетворююча техніка), пов'язана з перетворенням виду електричного струму для кіл електроприводу, електротяги, електротермії, електроенергетики та іншим.

Сучасний етап розвитку інформаційної електроніки характеризується широким використанням компонентів мікроелектроніки, включаючи великі інтегральні схеми.

Все це дозволяє створити більш ефективні умови генерування, передачі і розподілу електроенергії, а також використовувати більш продуктивну технологію в різних галузях промисловості.

Сучасні електронні системи дозволяють проводити контроль точності обробки безпосередньо в процесі виготовлення виробу без зупинки виробництва (так звані системи активного контролю), дозволяють повністю або частково повести автоматизацію виробництв, що дозволить зменшити собівартість виробу, підвищити його якість, а також зменшити число зайнятих працівників на важких та шкідливих роботах.

3. Проектування адресних дешифраторів на основі елементів логіки

Розглянемо процес дешифрації адреси оперативної (ОЗП), постійної (ПЗП) пам'яті та пристроїв вводу-виводу (ПВВ) відповідно до розробленої Вами згідно вказаного варіанту (розділ I, табл. 1) схеми електричної принципової мікропроцесорної системи керування.

Як приклад, розглянемо приведену нижче схему на рис. 42, в якій дешифрація адреси реалізується двома дешифраторами: окремо для пам'яті (DD7) та пристроїв вводу-виводу (DD8).

Відповідно до схеми (рис. 42) створюємо таблицю розподілу адресного простору для адресації пам'яті та пристроїв вводу-виводу (табл. 5).

Таблиця 5

		A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ПЗП (DD6)	0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1FFFh	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ОЗП (DD5)	8000h	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9FFFh	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ППП (DD9)	A0A0h	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0
	B0BFh	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1
RG (DD10)	E0E0h	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
	FFFFh	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Побудуємо дешифратори з використанням елементів логіки.

У відповідності до табл. 5 створюємо таблицю істинності (табл. 6) для формування сигналу CS1 вибірки мікросхеми постійної пам'яті (DD6). Активний рівень цього сигналу – логічний «0».

Таблиця 6

A15	A14	A13	A12	CS1
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Використовуючи таблицю істинності (табл. 6), складаємо Удосконалену Диз'юнктивно Нормальну Функцію (УДНФ) для функції CS1 із тих входних сигналів, при яких він має низький рівень, тобто "0" (це активний рівень).

$$CS = (\overline{A15} \cdot \overline{A14} \cdot \overline{A13} \cdot \overline{A12}) \cup (\overline{A15} \cdot \overline{A14} \cdot \overline{A13} \cdot A12) \quad (1)$$

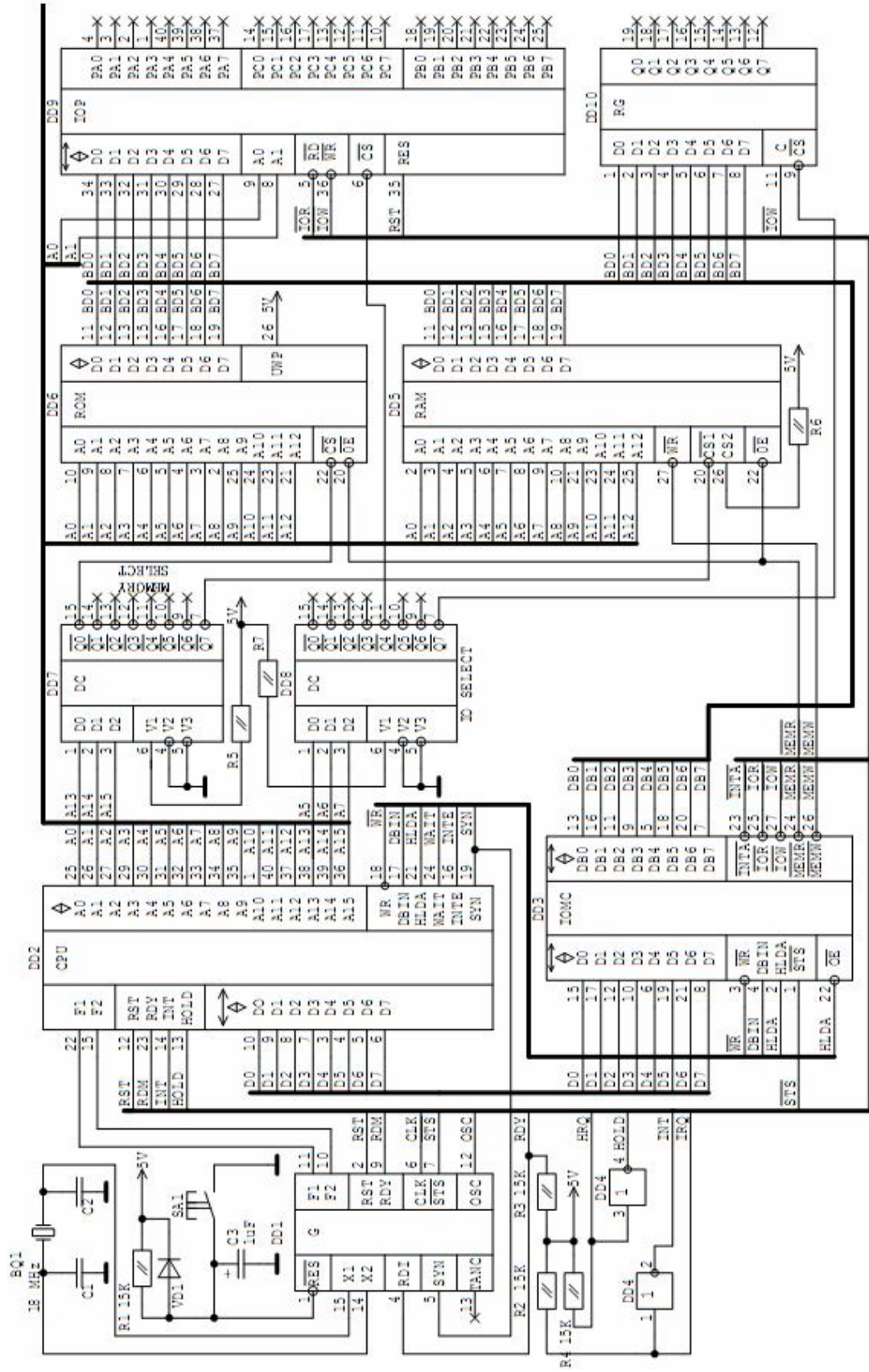


Рис. 42 Приклад виконання МП-пристрою на основі мікропроцесорного комплекту КР580

Використовуючи вираз (1), побудуємо схему дешифрації для сигналу CS1 (схема 1).

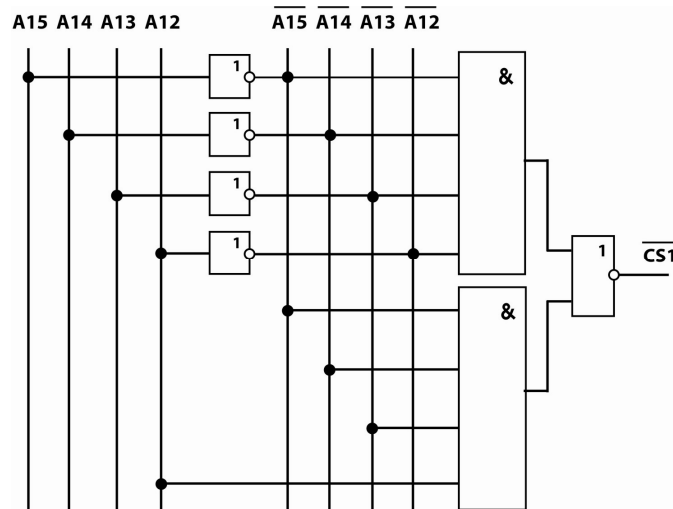


Схема 1 Побудова функції CS1 для ПЗП (DD6)

Для отримання інверсного значення CS використовуємо останній елемент схеми 1 (двовходовий АБО-НЕ) з інверсією.

Спростимо вираз (1) наступним чином:

$$CS = (\overline{A15} \cdot \overline{A14} \cdot \overline{A13})(A12 \cup A12) = \overline{A15} \cdot \overline{A14} \cdot \overline{A13} = \overline{(A15 \cup A14 \cup A13)} \quad (2)$$

$$\overline{CS} = A15 \cup A14 \cup A13 \quad (3)$$

Побудуємо схему дешифрації за спрощеним виразом (3) (схема 2).

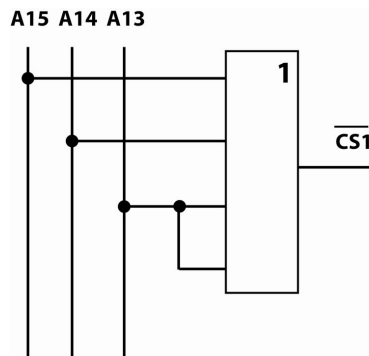


Схема 2 Спрощена схема функції CS1 для ПЗП (DD6)

Проведемо мінімізацію виразу для CS1 з використанням **карти Карно**.

Для того, щоб зменшити кількість логічних елементів, які необхідні для побудови, функцію CS1 потрібно мінімізувати.

Для мінімізації використаємо табличний метод карт Карно. Карта Карно представляє собою певну форму таблиці істинності для відповідної кількості аргументів, для нашого випадку для 4-х аргументів A12 – A15.

Карта Карно містить всі 2^n можливих вхідних комбінацій (n- кількість аргументів) і значень функцій, які відповідають кожній із комбінацій.

Для чотирьох змінних існує 16 вхідних наборів, карта Карно розділена на 16 комірок.

Рядки визначають можливими комбінаціями A15A14, а стовбчики A13A12 (табл. 1).

Слід зауважити, що сусідні комбінації двох змінних в рядках і стовпчиках відрізняються лише однією змінною, наприклад 00-01, або 01-11. Не слід розміщувати комбінації по сусідству, які відрізняються двома змінними (наприклад 00-11 або 10-01). Це правило важливе як для стовбців, так і для рядків.

Карта для функції CS1 подана в табл. 7.

Таблиця 7

		A13 A12			
		00	01	10	11
A15 A14	00	0	0	1	1
	01	1	1	1	1
	10	1	1	1	1
	11	1	1	1	1

Проводимо склеювання мінтермів. Склеювати (групувати) можна мінтерми, які стоять по сусідству в рядку чи стовпчику.

Для спрощення склеюємо мінтерми, що містять «0». В нашому випадку це (1) і (2) мінтерми. Випишуємо ці мінтерми, виносимо за дужки спільний член і спрощуємо отриманий вираз:

$$CS = (\overline{A15} \cdot \overline{A14} \cdot \overline{A13} \cdot \overline{A12}) \cup (\overline{A15} \cdot \overline{A14} \cdot A13 \cdot \overline{A12})$$

Функція CS1 після спрощення (мінімізації) набуває вигляду:

$$\overline{CS} = A15 \cup A14 \cup A13$$

З чотирох мінтермів її мінімізували до трьох.

Основне призначення карт Карно - спрощення логічних виразів шляхом знаходження спільних членів.

Склеювати також можна мінтерми, які стоять у крайніх клітинках рядків і стовпчиків.

У відповідності до табл.5 створюємо таблицю істинності (табл. 8) для формування сигналу CS2 вибірки мікросхеми оперативної пам'яті (DD5). Активний рівень цього сигналу також логічний «0».

Таблиця 8

A15	A14	A13	A12	CS2
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Використовуючи таблицю істинності (табл. 8), складаємо УДНФ для функції CS2 із тих вхідних сигналів, при яких він має низький рівень, тобто «0» (активний рівень).

$$CS2 = (A15 \cdot \overline{A14} \cdot \overline{A13} \cdot \overline{A12}) \cup (A15 \cdot \overline{A14} \cdot A13 \cdot A12) \quad (4)$$

Використовуючи вираз (4), побудуємо схему дешифрації для сигналу CS2 (схема 3).

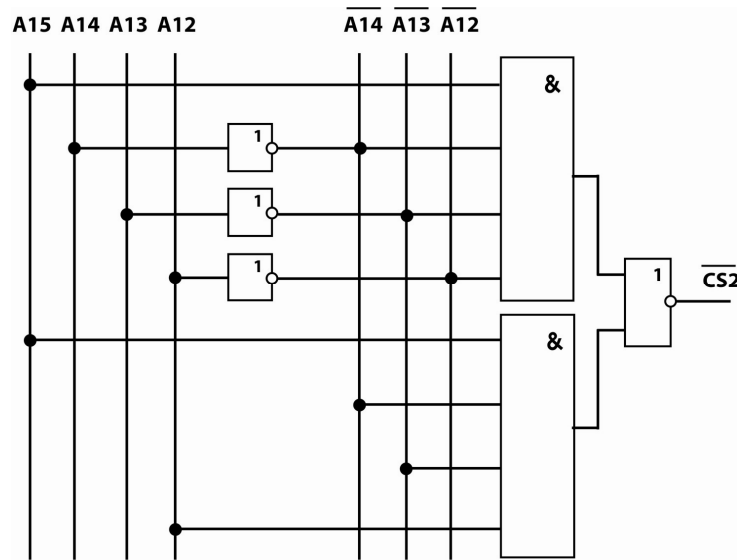


Схема 3 Побудова функції CS2 для ПЗП (DD5)

Спростимо вираз (4):

$$CS2 = (A15 \cdot \overline{A14} \cdot \overline{A13})(\overline{A12} \cup A12) = A15 \cdot \overline{A14} \cdot \overline{A13} \quad (5)$$

$$\overline{CS2} = \overline{A15} \cup A14 \cup A13 \quad (6)$$

Побудуємо схему дешифрації за спрощеним виразом (6) (схема 4).

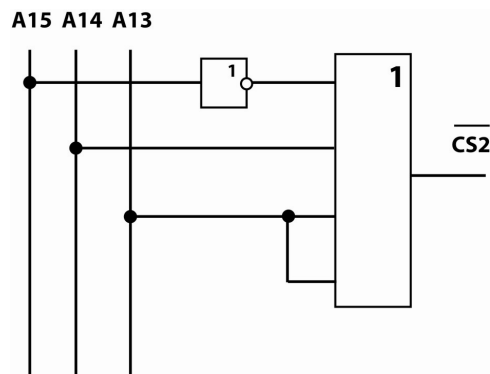


Схема 4 Спрощена схема функції CS2 для ПЗП (DD5)

У відповідності до табл. 5 створюємо таблицю істинності (табл. 9) для формування сигналу CS3 вибірки мікросхеми KP580BB55 (DD9). Активний рівень цього сигналу - логічний «0».

За таблицею істинності (табл. 9) складаємо УДНФ для функції CS3 із тих входних сигналів, при яких він має рівень «0» (також активний рівень).

$$CS3 = (A7 \cdot \overline{A6} \cdot A5 \cdot \overline{A4}) \cup (A7 \cdot \overline{A6} \cdot A5 \cdot A4) \quad (7)$$

Таблиця 9

A7	A6	A5	A4	CS3
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

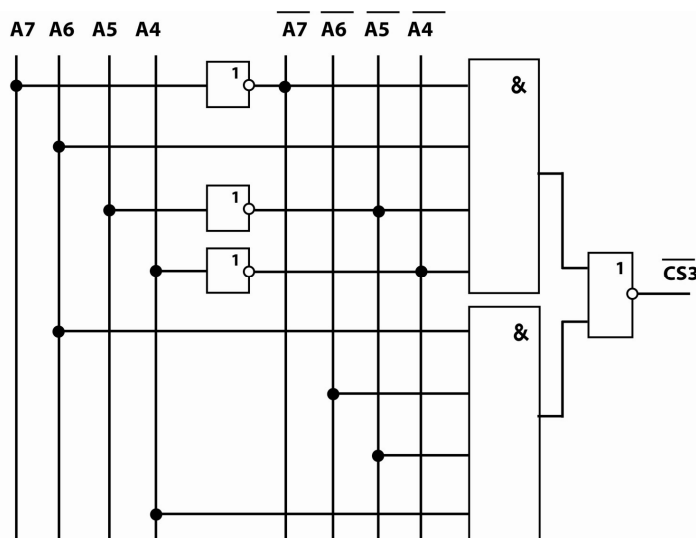


Схема 5 Побудова функції CS3 для ППІ (DD9)

Спростимо вираз (7):

$$CS3 = (A7 \cdot \overline{A6} \cdot A5)(\overline{A4} \cup A4) = A7 \cdot \overline{A6} \cdot A5 \quad (8)$$

$$\overline{CS3} = \overline{A7} \cup \overline{A6} \cup \overline{A5} \quad (9)$$

Побудуємо схему дешифрації за спрощеним виразом (9) (схема 6).

Відповідно до табл. 5 створюємо таблицю істинності (табл. 10) для формування сигналу CS4 вибірки мікросхеми регістра (DD10). Активний рівень цього сигналу логічний «0».

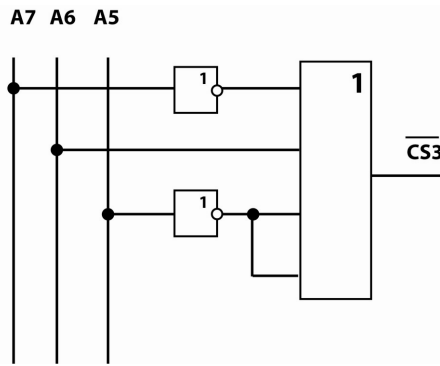


Схема 6 Спрощена схема функції CS3 для ППІ (DD9)

Таблиця 10

A7	A6	A5	A4	CS4
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Використовуючи таблицю істинності (табл. 10) складаємо УДНФ для функції CS4 із вхідних сигналів, при яких він має рівень "0" (активний рівень).

$$CS4 = (A7 \cdot A6 \cdot A5 \cdot \overline{A4}) \cup (A7 \cdot A6 \cdot A5 \cdot A4) \quad (10)$$

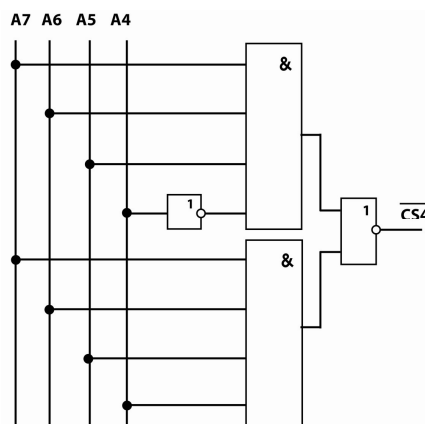


Схема 7 Побудова функції CS4 для регістра (DD10)

Спростимо вираз (10):

$$CS4 = (A7 \cdot A6 \cdot A5)(\overline{A4} \cup A4) = A7 \cdot A6 \cdot A5 \quad (11)$$

$$\overline{CS3} = \overline{A7} \cup \overline{A6} \cup \overline{A5} \quad (12)$$

Побудуємо схему дешифрації за спрощеним виразом (12) (схема 8).

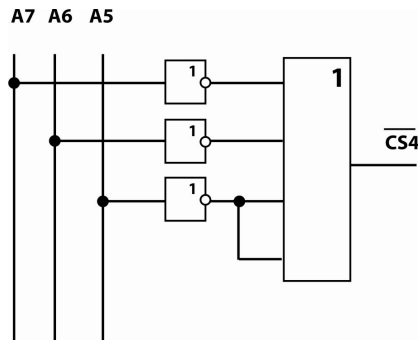


Схема 8 Спрощена схема функції CS4 для регістра (DD10)

Всі приведені схеми дешифрації на логічних елементах в практичних схемах мікропроцесорних пристроїв замінюються однією або декількома мікросхемами дешифраторів. Прикладом є електрична схема на рис. 42.

ЛІТЕРАТУРА

1. Магда Ю.С. Микроконтроллеры серии 8051: практический подход. — М.: ДМК Пресс, 2008. — 228 с.
2. Угрюмов Е.П. Цифровая схемотехника. Спб.: BHV, 2001. 528 с.
3. Щелкунов Н.Н., Дианов А.П. Микропроцессорные средства и системы. -М.: Радио и связь, 1989, -288с.
4. Боборыкин А.В., Липовецкий Г.П. и др. Однокристалльные микроЭВМ. Справочник. М.: БИНОМ, 1994. 400с.
5. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах. М.: Энергоатомиздат, 1990.
6. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник / С.Т. Хвощ, Н.Н. Варлинский, Е.А. Попов; под общ. ред. С.Г. Хвоща. Л.: Машиностроение, 1987. 640 с.
7. Тули М. Справочное пособие по цифровой электронике: Пер с англ. -М.: Энергоатомиздат, 1990, 176с.
8. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения: Справочник. 2-е изд., перераб. и доп. -М.: Радио и связь, 1990, -512с.
9. Применение интегральных микросхем памяти: Справочник /А.А. Дерюгин, В.В. Цыркин, В.Е. Красовский и др.; Под ред. А.Ю. Гордонова. А.А. Дерюгина. -М.: Радио и связь, 1994. - 232с.
10. Лебедев О.Н. Применение микросхем памяти в электронных устройствах: Справ. пособие. -М.: Радио и связь, 1994, -216с.
11. В.Я. Нерода, В.Э. Торбинский, Е.Л. Шлыков. Однокристалльные микро-ЭВМ MCS-51. Архитектура. Серия Однокристалльные микроЭВМ, М.: Диджитал Компонентс, 1995.
12. ЖКИ индикаторы: <http://www.melt.com.ru>
13. Информационный сервер по микроэлектронике www.gaw.ru.

14. Термрмопреобразователи. <http://www.ao-tera.com.ua/commonfiles/t-sensors.pdf>
15. Буняк А.М. “Електроніка та мікросхемотехніка”, Тернопіль, 2001.
16. «Цифровые интегральные схемы». Справочник под. ред. Мальцева П.П., М.: 1994.
- Забродин Ю.С. “Промышленная электроника”. М.: 1982.
17. Терещук Р.М. и др. “Полупроводниковые приемно–усилительные устройства”: Справочник радиолюбителя. Киев. Наукова думка, 1987.-800 с.
19. Шило В.Л. ”Популярные цифровые микросхемы”. Справочник, Челябинск “Металлургия”, 1988.
20. Галкин В.И. и др. “Полупроводниковые приборы”. Справочник. Минск, “Беларусь”, 1987.
21. Мікросхемні стабілізатори серії КР142. Журнал “Радіо” № 8, 9 1990 р.
22. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник. В 2 т. / В.-Б. Абрайтис, Н. Н. Аверьянов, А. И. Белоус и др.; Под ред. В. А. Шахнова. –М.; Радио и связь, 1988.-Т.1. – 368 с.: ил.
23. Гук М. Аппаратные средства IBM PC. Энциклопедия – СПб: Питер Ком, 1999.- 816 с.: ил.
24. Аналоговые и цифровые интегральные микросхемы. Справочное пособие / С. В. Якубовский, Н. А. Барканов, Л. И. Ниссельсон и др.; под ред. С. В. Якубовского. – 2-е изд., перераб. и доп.- М.: Радио и связь, 1985.-432с., ил.- (Проектирование РЭА на интегральных микросхемах).
25. Самофалов К. Г., Викторов О. В. Микропроцессоры. – Б-ка инженера – 2-е изд., перераб. и доп. - К: Техника, 1989.-312 с.
26. Нестеренко И. И. Цветовая и кодовая маркировка радиоэлектронных компонентов, отечественных и зарубежных. М.: «Солон», 1997.-128 с.

Команди МП КР580ВМ80А (і8080)**1. Класифікація команд**

Всі команди МП КР580ИК80 розділяються на п'ять груп.

1. Команди пересилання даних. Вони здійснюють пересилання даних між регістрами або між пам'яттю і регістрами.

2. Арифметичні команди. Призначені для виконання додавання, віднімання, збільшення або зменшення вмісту регістрів або комірок пам'яті.

3. Логічні команди. Виконують логічні операції "І", "АБО", "виключаюче АБО", порівняння, зсув і доповнення даних в регістрах та комірках пам'яті.

4. Команди переходів (умовних та безумовних), виклику підпрограм та повернення з підпрограм.

5. Команди вводу/виводу, керування та роботи із стеком, керування флагами, команди дозволу та заборони переривань.

Умовне позначення всіх команд МП та їх кодів подано в таблиці 1.

Часто оброблювані дані зберігаються в пам'яті. При використанні багатобайтових даних такі дані, як команди, зберігаються в наступних одна за іншою комірках пам'яті.

Молодший байт записується першим, за ним слідує більші старші байти. МП КР580ВМ80А має чотири різних способи адресації даних, що зберігаються в пам'яті чи регістрах.

Пряма - другий і третій байти команди містять точну адресу комірки пам'яті, у якій зберігаються потрібні дані. Молодші біти адреси містяться в другому байті, старші - у третьому.

Регістрова - команда вказує чи регістр пару регістрів, де містяться дані.

Непряма регістрова - команда вказує пару регістрів, у яких поміщає адреса комірки пам'яті, де розташовані потрібні дані. Старші біти адреси зберігаються в першому регістрі пари, молодші - у другому.

Безпосередня - команда сама поміщає потрібні дані. Дані можуть бути довжиною 8 чи 16 біт (спочатку йде молодший байт, потім старший).

Якщо немає команд переходу чи переривань, процесор виконує команди, що зберігаються в комірках пам'яті, що слідує одна за іншою.

Команда переходу може визначити адресу наступної команди, яку потрібно виконати, одним із двох способів.

Прямий - команда переходу поміщає адресу наступної команди, яку потрібно виконати. За винятком команди RST другий байт поміщає молодші біти адреси, третій байт - старші біти.

Непрямий регістровий - команда переходу вказує пари регістрів, у якій знаходиться адреса наступної команди, що виконується. Старші біти адреси зберігаються в першому регістрі пари, молодші - у другому.

Код команди	x0	x1	x2	x3	x4	x5	x6	x7
0x	NOP	LXI B, 2 байти	STAX B	INX B	INR B	DCR B	MVI B, байт	RLC
1x	-	LXI D, 2 байти	STAX D	INX D	INR D	DCR D	MVI D, байт	RAL
2x	-	LXI H, 2 байти	SHLD 2 байти	INX H	INR H	DCR H	MVI H, байт	DAA
3x	-	LXI SP, 2 байти	STA 2 байти	INX SP	INR M	DCR M	MVI M, байт	STC
4x	MOV B, B	MOV B, C	MOV B, D	MOV B, E	MOV B, H	MOV B, L	MOV B, M	MOV B, A
5x	MOV D, B	MOV D, C	MOV D, D	MOV D, E	MOV D, H	MOV D, L	MOV D, M	MOV D, A
6x	MOV H, B	MOV H, C	MOV H, D	MOV H, E	MOV H, H	MOV H, L	MOV H, M	MOV H, A

7x	MOV M, B	MOV M, C	MOV M, D	MOV M, E	MOV M, H	MOV M, L	HLT	MOV M, A
8x	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A
9x	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A
Ax	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A
Bx	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A
Cx	RNZ адреса	POP B	JNZ адреса	JMP адреса	CNZ адреса	PUSH B	ADI байт	RST 0
Dx	RNC адреса	POP D	JNC адреса	OUT порт	CNC адреса	PUSH D	SUI байт	RST 2
Ex	RPO адреса	POP H	JPO адреса	XTHL	CPO адреса	PUSH H	ANI байт	RST 4
Fx	RP адреса	POP PSW	JP адреса	DI	CP адреса	PUSH PSW	ORI байт	RST 6
0x	-	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C, байт	RRC
1x	-	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E, байт	RAR
2x	-	DAD H	LHLD 2 байти	DCX H	INR L	DCR L	MVI L, байт	CMA
3x	-	DAD SP	LDA 2 байти	DCX SP	INR A	DCR A	MVI A, байт	CMC
4x	MOV C, B	MOV C, C	MOV C, D	MOV C, E	MOV C, H	MOV C, L	MOV C, M	MOV C, A
5x	MOV E, B	MOV E, C	MOV E, D	MOV E, E	MOV E, H	MOV E, L	MOV E, M	MOV E, A
6x	MOV L, B	MOV L, C	MOV L, D	MOV L, E	MOV L, H	MOV L, L	MOV L, M	MOV L, A
7x	MOV A, B	MOV A, C	MOV A, D	MOV A, E	MOV A, H	MOV A, L	MOV A, M	MOV A, A
8x	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
9x	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
Ax	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
Bx	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A
Cx	RZ адреса	RET	JZ адреса	-	CZ адреса	CALL адреса	ACI байт	RST 1
Dx	RC адреса	-	JC адреса	IN порт	CC адреса	-	SBI байт	RST 3
Ex	RPE адреса	PCHL	JPE адреса	XCHG	CPE адреса	-	XRI байт	RST 5
Fx	RM адреса	SPHL	JМадреса	EI	CM адреса	-	CPI байт	RST 7

2. Команди МП

2.1. Команди пересилання даних

Команди цієї групи здійснюють пересилання даних в/із реєстр (а) і комірки пам'яті і не змінюють стану флагів. Еквівалентні командам присвоювання мов високого рівня.

Пересилання даних з реєстра в реєстр

MOV r1, r2 ; (r1) ← (r2)

Вміст реєстра r2 пересилається в реєстр r1

Формат команди

0	1	D	D	D	S	S	S
---	---	---	---	---	---	---	---

MOV	B, A	47	MOV	C, A	4F	MOV	D, A	57	MOV	E, A	5F
MOV	B, B	40	MOV	C, B	48	MOV	D, B	50	MOV	E, B	58
MOV	B, C	41	MOV	C, C	49	MOV	D, C	51	MOV	E, C	59
MOV	B, D	42	MOV	C, D	4A	MOV	D, D	52	MOV	E, D	5A
MOV	B, E	43	MOV	C, E	4B	MOV	D, E	53	MOV	E, E	5B
MOV	B, H	44	MOV	C, H	4C	MOV	D, H	54	MOV	E, H	5C
MOV	B, L	45	MOV	C, L	4D	MOV	D, L	55	MOV	E, L	5D
MOV	H, A	67	MOV	L, A	5F	MOV	A, A	7F			
MOV	H, B	60	MOV	L, B	58	MOV	A, B	78			
MOV	H, C	61	MOV	L, C	59	MOV	A, C	79			
MOV	H, D	62	MOV	L, D	5A	MOV	A, D	7A			
MOV	H, E	63	MOV	L, E	5B	MOV	A, E	7B			
MOV	H, H	64	MOV	L, H	5C	MOV	A, H	7C			
MOV	H, L	65	MOV	L, L	5D	MOV	A, L	7D			

Циклів - 1, тактів - 4. Тип адресації - регістрова, регістр ознак не змінюється

Пересилання даних з пам'яті

MOV r, M ;

Вміст комірки пам'яті, адреса якої зберігається в регістрах H і L, пересилається в регістр r.

0	1	D	D	D	1	1	0
---	---	---	---	---	---	---	---

MOV	A, M	7E
MOV	B, M	46
MOV	C, M	4E
MOV	D, M	56
MOV	E, M	5E
MOV	H, M	66
MOV	L, M	6E

Циклів - 2, тактів 7, адресація: непряма регістрова. Регістр ознак не змінюється.

Пересилання даних в пам'ять

MOV M, r () ((H) (L)) ← (r)

Вміст регістра r пересилається в комірку пам'яті, адреса якої знаходиться в регістрах H і L.

0	1	1	1	0	S	S	S
---	---	---	---	---	---	---	---

MOV	M, A	77
MOV	M, B	70
MOV	M, C	71
MOV	M, D	72
MOV	M, E	73
MOV	M, H	74
MOV	M, L	75

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Безпосереднє пересилання

MVI r, байт () ; (r) ← (байт 2)

Вміст другого байту команди пересилається в регістр r

0	0	D	D	D	1	1	0
---	---	---	---	---	---	---	---

Дані

MVI	A, байт	3E
MVI	B, байт	06
MVI	C, байт	0E
MVI	D, байт	16
MVI	H, байт	26
MVI	L, байт	2E

Циклів - 2, тактів - 7. Тип адресації - безпосередня. Регістр ознак не змінюється.

Безпосереднє пересилання в пам'ять

$MVI\ M, \text{байт } ((H) \ (L)) \leftarrow (\text{байт } 2)$

Вміст другого байту команди пересилається в комірку пам'яті, адреса якої вказана у регістрах H і L.

0	0	1	1	0	1	1	0
Дані							

MVI M, байт 36 байт

Циклів - 3, тактів - 10. Тип адресації - безпосередня/непряма регістрова. Регістр ознак не змінюється.

Безпосереднє завантаження пари регістрів

$LXI\ rp, 2 \text{ байти } (rh) \leftarrow (\text{байт } 3)$
 $(rl) \leftarrow (\text{байт } 2)$

Третій байт команди пересилається в старший регістр (rh) пари регістрів, другий - у молодший регістр (rl) пари регістрів.

0	0	R	P	0	0	0	1
Молодший байт даних							
Старший байт даних							

LXI B, 2 байта 01 (безпосереднє завантаження пари регістрів B і C),

LXI D, 2 байта 11 (безпосереднє завантаження пари регістрів D і E),

LXI H, 2 байта 21 (безпосереднє завантаження пари регістрів H і L),

LXI SP, 2 байта 31 (безпосереднє завантаження вказівника стеку),

Циклів - 3, тактів - 10, тип адресації - безпосередня. Регістр ознак не змінюється.

Пряме завантаження акумулятора

$LDA\ [adr.] \ (A) \leftarrow ((\text{байт } 3) \ (\text{байт } 2))$

Вміст комірки пам'яті, адреса якої вказана у другому і третьому байтах команди, завантажуються в акумулятор.

0	0	1	1	1	0	1	1
Молодший байт адреси							
Старший байт адреси							

$LDA\ [adr.] \ 3A$

Циклів - 4, тактів - 13, тип адресації - пряма. Регістр ознак не змінюється.

Прямий запис вмісту акумулятора в пам'ять

$STA \quad [adr.] \quad ((байт\ 3) \quad (байт\ 2)) \leftarrow (A)$

Вміст акумулятора пересилається в комірку пам'яті, адреса якої вказана у другому і третьому байтах команди.

0	0	1	1	0	0	1	0
Молодший байт адреси							
Старший байт адреси							

$STA \quad [adr.] \quad 32 \quad [adr.]$

Циклів - 4, тактів - 13, тип адресації - пряма. Регістр ознак не змінюється.

Пряме завантаження H та L

$LHLD \quad [adr.]$

$(L) \leftarrow ((байт\ 3) \quad (байт\ 2))$

$(H) \leftarrow ((байт\ 3) \quad (байт\ 2)+1)$

Вміст комірки пам'яті, адреса якої вказана у другому і третьому байтах команди, пересилається в регістр L. Вміст комірки пам'яті з наступною адресою $((adr.) + 1)$ пересилається в регістр H.

0	0	1	0	1	0	1	0
Молодший байт адреси							
Старший байт адреси							

$LHLD \quad [adr.] \quad 2A$

Циклів - 5, тактів - 16, тип адресації - пряма. Регістр ознак не змінюється.

Прямий запис вмісту регістрів H, L у пам'ять

$SHLD \quad [adr.]$

$((байт\ 3) \quad (байт\ 2)) \leftarrow (L)$

$((байт\ 3) \quad (байт\ 2)+1) \leftarrow (H)$

Вміст регістра L пересилається в комірку пам'яті, адреса якої вказана у другому і третьому байтах команди. Вміст регістра H пересилається в наступну комірку пам'яті.

0	0	1	0	0	0	1	0
Молодший байт адреси							
Старший байт адреси							

$SHLD \quad [adr.] \quad 22$

Циклів - 5, тактів - 16, тип адресації - пряма. Регістр ознак не змінюється.

Непряме завантаження акумулятора

$LDAX \quad rp \quad (A) \leftarrow ((rp))$

Вміст комірки пам'яті, адреса якої вказана у парі регістрів rp, пересилається в регістр A. Можуть бути вказані тільки пари регістрів $rp = B$ (регістри B і C) та $rp = D$ (регістри D і E).

0	0	R	P	1	0	1	0
---	---	---	---	---	---	---	---

$LDAX \quad B \quad 0A$

$LDAX \quad D \quad 1A$

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Непрямий запис вмісту акумулятора в пам'ять

$STAX \quad rp \quad ((rp)) \leftarrow (A)$

Вміст акумулятора пересилається в комірку пам'яті, адреса якої вказана у парі регістрів гр. Можуть бути вказані тільки перші регістри з пари регістрів гр. Наприклад, В (регістри В и С) та D (регістри D і E).

0	0	R	P	0	0	1	0
---	---	---	---	---	---	---	---

$STAX \quad B \quad 02$

$STAX \quad D \quad 12$

Циклів - 2, тактів - 7, тип адресації - непряма регістрація. Регістр ознак не змінюється.

Обмін даними між регістрами H, L і D, E

$XCHG \quad (H) \leftrightarrow (D) \quad (L) \leftrightarrow (E)$

Вміст регістрів H и L обмінюється з вмістом регістрів D і E.

1	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

$XCHG \quad EB$

Циклів - 1, тактів - 4, тип адресації - регістрова. Регістр ознак не змінюється.

2.2. Арифметичні команди

Команди цієї групи виконують арифметичні операції над даними в регістрах і комірках пам'яті. Якщо немає особливих вказівок, то команди змінюють стан флагів нуля, парності, перенесення і допоміжного перенесення у відповідності зі стандартними правилами.

Всі операції віднімання виконуються через арифметичне доповнення до 2, причому, флаг перенесення встановлюється в логічну 1, якщо є позичання, і скидається, якщо позичання не було.

Додавання вмісту акумулятора до вмісту регістра

$ADD \quad r \quad (A) \leftarrow (A) + (r)$

Вміст регістра r додається до вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	0	0	S	S	S
---	---	---	---	---	---	---	---

ADD	A	87	ADD	E	83
ADD	B	80	ADD	H	84
ADD	C	81	ADD	L	85
ADD	D	82			

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки Z, S, P, CY, AC.

Додавання вмісту акумулятора до вмісту комірки пам'яті

$ADD \quad M \quad (A) \leftarrow (A) + ((H) \quad (L))$

Вміст комірки пам'яті, адреса якої розміщена в регістрах H и L, додається до вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

$ADD \quad M \quad 86$

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, CY, AC.

Безпосереднє додавання

$ADI \text{ байт } (A) \leftarrow (A) + (\text{байт } 2)$

Вміст другого байту команди додається до вмісту акумулятора. Результат завантажується в акумулятор.

1	1	0	0	0	1	1	0
Дані							

ADI байт C6

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC

Додавання вмісту акумулятора до вмісту регістра і біту флага перенесення

$ADC \ r \ (A) \leftarrow (A) + (r) + (CY)$

Вміст регістра r і біт флага перенесення додаються до вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	0	1	S	S	S
---	---	---	---	---	---	---	---

ADC	A	8F
ADC	B	88
ADC	C	89
ADC	D	8A
ADC	E	8B
ADC	H	8C
ADC	L	8D

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, CY, AC

Додавання вмісту акумулятора до вмісту комірки пам'яті і біту перенесення

$ADC \ M \ (A) \leftarrow (A) + ((H) \ (L)) + (CY)$

Вміст комірки пам'яті, адреса якої вказана у регістрах H і L, і вміст біту флага перенесення додаються до вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---

ADC M 8E

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, CY, AC.

Безпосереднє додавання з врахуванням біту перенесення

$ACI \ \text{байт} \ (A) \leftarrow (A) + (\text{байт } 2) + (CY)$

Вміст другого байту команди і біту флага перенесення додається до вмісту акумулятора. Результат завантажується в акумулятор.

1	1	0	0	1	1	1	0
Дані							

ACI байт CE

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC

Віднімання вмісту регістра

$SUB \quad r \quad (A) \leftarrow (A) - (r)$

Вміст регістра r віднімається від вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	1	0	S	S	S
---	---	---	---	---	---	---	---

SUB	A	97
SUB	B	90
SUB	C	91
SUB	D	92
SUB	E	93
SUB	H	94
SUB	L	95

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, CY, AC

Віднімання вмісту комірки пам'яті

$SUB \quad M \quad (A) \leftarrow (A) - ((H) \quad (L))$

Вміст комірки пам'яті, адреса якої розміщена в H та L , віднімається від вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	1	0	1	1	0
---	---	---	---	---	---	---	---

$SUB \quad M \quad 96$

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, CY, AC.

Віднімання константи

$SUI \quad \text{байт} \quad (A) \leftarrow (A) - (\text{байт } 2)$

Вміст другого байту команди віднімається від вмісту акумулятора. Результат завантажується в акумулятор.

1	1	0	1	0	1	1	0
Дані							

$SUI \quad \text{байт} \quad D6$

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC

Віднімання вмісту регістра та біту перенесення

$SBB \quad r \quad (A) \leftarrow (A) - (r) - (CY)$

Вміст регістра r і біт флага перенесення CY віднімаються від вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	1	1	S	S	S
---	---	---	---	---	---	---	---

SBB	A	9F
SBB	B	98
SBB	C	99
SBB	D	9A
SBB	E	9B
SBB	H	9C
SBB	L	9D

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, CY, AC.

Віднімання вмісту комірки пам'яті та біту перенесення

$$SBB \ M \ (A) \leftarrow (A) - ((H) \ (L)) - (CY)$$

Вміст комірки пам'яті, адреса якої розміщена в регістрах H і L, та біт ознаки перенесення CY віднімаються від вмісту акумулятора. Результат завантажується в акумулятор.

1	0	0	1	1	1	1	0
---	---	---	---	---	---	---	---

SBB M 9E

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, CY, AC.

Віднімання константи та біту перенесення

$$SBI \ \text{байт} \ (A) \leftarrow (A) - (\text{байт } 2) - (CY)$$

Вміст другого байту команди і біт ознаки перенесення CY віднімаються від вмісту акумулятора. Результат завантажується в акумулятор.

1	1	0	1	1	1	1	0
Дані							

SBI байт DE

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC.

Збільшення вмісту регістра

$$INR \ r \ (r) \leftarrow (r) + 1$$

Вміст регістра r збільшується на одиницю.

0	0	D	D	D	1	0	0
---	---	---	---	---	---	---	---

INR	A	3C
INR	B	04
INR	C	0C
INR	D	14
INR	E	1C
INR	H	24
INR	L	26

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, AC. Ознака перенесення CY не змінюється.

Збільшення вмісту комірки пам'яті

$$INR \ M \ ((H) \ (L)) \leftarrow ((H) \ (L)) + 1$$

Вміст комірки пам'яті, адреса якої завантажена в регістрах H и L, збільшується на одиницю.

0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

INR M 34

Циклів - 3, тактів - 10, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, AC. Ознака перенесення CY не змінюється.

Зменшення вмісту регістра

$$DCR \ r \ (r) \leftarrow (r) - 1$$

Вміст регістру r зменшується на одиницю.

0	0	D	D	D	1	0	1
---	---	---	---	---	---	---	---

DCR	A	3D	DCR	E	1D
DCR	B	05	DCR	H	25
DCR	C	0D	DCR	L	2D
DCR	D	15			

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, AC. Ознака перенесення CY не змінюється.

Зменшення вмісту комірки пам'яті.

$$DCR \quad M \quad ((H) \quad (L)) \leftarrow ((H) \quad (L)) - 1$$

Вміст комірки пам'яті, адреса якої розміщена в регістрах H і L, зменшується на одиницю.

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

$$DCR \quad M \quad 35$$

Циклів - 3, тактів - 10, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, AC. Ознака перенесення CY не змінюється.

Збільшення вмісту пари регістрів

$$INX \quad rp \quad (rh) \quad (rl) \leftarrow (rh) \quad (rl) + 1$$

Вміст пари регістрів rp збільшується на одиницю. Команда не впливає на регістр ознак.

0	0	R	P	0	0	1	1
---	---	---	---	---	---	---	---

$$INX \quad B \quad 03 \quad (\text{пара регістрів } B \text{ і } C)$$

$$INX \quad D \quad 13 \quad (\text{пара регістрів } D \text{ і } E)$$

$$INX \quad H \quad 23 \quad (\text{пара регістрів } H \text{ і } L)$$

$$INX \quad SP \quad 33 \quad (\text{вказівник СТЕКУ})$$

Циклів - 1, тактів - 6, тип адресації - регістрова. Регістр ознак не змінюється.

Зменшення вмісту пари регістрів

$$DCX \quad rp \quad (rh) \quad (rl) \leftarrow (rh) \quad (rl) - 1$$

Вміст пари регістрів rp зменшується на одиницю. Команда не впливає на флаги.

0	0	R	P	1	0	1	1
---	---	---	---	---	---	---	---

$$DCX \quad B \quad 0B \quad (\text{пара регістрів } B \text{ і } C)$$

$$DCX \quad D \quad 1B \quad (\text{пара регістрів } D \text{ і } E)$$

$$DCX \quad H \quad 2B \quad (\text{пара регістрів } H \text{ і } L)$$

$$DCX \quad SP \quad 3B \quad (\text{вказівник СТЕКУ})$$

Циклів - 1, тактів - 6, тип адресації - регістрова. Регістр ознак не змінюється.

Додавання вмісту пари регістрів із вмістом регістрів H та L

$$DAD \quad rp \quad (H) \quad (L) \leftarrow (H) \quad (L) + (rh) \quad (rL)$$

Вміст пари регістрів rp додається до вмісту регістрів H і L. Результат завантажується в пари регістрів H, L. Використовується тільки флаг перенесення CY. Він встановлюється, якщо є перенесення при додаванні з подвійною точністю, інакше скидається.

0	0	R	P	1	0	0	1
---	---	---	---	---	---	---	---

DAD	B	09	(пара регістрів B і C)
DAD	D	19	(пара регістрів D і E)
DAD	H	29	(пара регістрів H і L)
DAD	SP	39	(вказівник СТЕКУ)

Циклів - 3, тактів - 10, тип адресації - регістрова. Змінюються ознака СУ.

Корекція результату при виконанні додавання двійково-десяткових чисел

DAA

8-бітне число в акумуляторі коректується по правилам двійково-десятьової арифметики за допомогою наступних операцій.

Якщо число, представлене чотирма молодшими бітами акумулятора, більше за 9 або встановлений флаг допоміжного перенесення АС, то до вмісту акумулятора додається 6.

Якщо тепер число, представлене чотирма старшими бітами акумулятора, більше за 9 або встановлений флаг перенесення СУ, то число 6 додається до числа, утвореного чотирма старшими бітами акумулятора.

0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---

DAA 27

Циклів - 1, тактів - 4. Змінюються ознаки: Z, S, P, СУ, АС.

2.3. Логічні команди

Команди цієї групи виконують побітні (логічні) операції над даними, що зберігаються в регістрах, комірках пам'яті і регістрі ознак. Якщо немає особливих вказівок, то всі команди впливають на ознаки: нуля, парності, знака, перенесення і допоміжного перенесення у відповідності зі стандартними правилами.

Логічне І над вмістом регістра та акумулятора

$$ANA \quad r \quad (A) \leftarrow (A) \wedge (r)$$

Над вмістом регістра r і акумулятора виконується операція логічного І. Результат завантажується в акумулятор. Скидається флаг перенесення СУ, встановлюється флаг допоміжного перенесення АС.

1	0	1	0	0	S	S	S
---	---	---	---	---	---	---	---

ANA	A	A7
ANA	B	A0
ANA	C	A1
ANA	D	A2
ANA	E	A3
ANA	H	A4
ANA	L	A5

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, СУ, АС, P.

Логічне І над вмістом комірки пам'яті та акумулятора

$$ANA \ M \quad (A) \leftarrow (A) \wedge ((H) \ (L))$$

Над вмістом комірки пам'яті, адреса якої поміщена в регістри H і L, і вмістом акумулятора виконується операція логічного І. Результат завантажується в акумулятор. Скидається флаг перенесення СУ і допоміжного перенесення АС.

1	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

ANA M A6

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, AC, CY.

Логічне І над вмістом акумулятора та константою

$ANI \text{ байт } (A) \leftarrow (A) \wedge (\text{байт } 2)$

Над вмістом другого байту команди і акумулятора виконується операція логічного І. Результат поміщається в акумулятор. Скидаються флаги перенесення CY і допоміжного перенесення AC.

1	1	1	0	0	1	1	0
Дані							

ANI байт E6

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC.

Виключаюче АБО над вмістом регістра та акумулятора

$XRA \quad r \quad (A) \leftarrow (A) \vee (r)$

Над вмістом регістра r і акумулятора виконується операція "виключаюче АБО". Результат поміщається в акумулятор. Скидаються флаг перенесення CY і допоміжного перенесення AC.

1	0	1	0	1	S	S	S
---	---	---	---	---	---	---	---

XRA	A	AF
XRA	B	A8
XRA	C	A9
XRA	D	AA
XRA	E	AB
XRA	H	AC
XRA	L	AD

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, CY, AC

Виключаюче АБО над вмістом комірки пам'яті та акумулятора

$XRA \text{ M } (A) \leftarrow (A) \vee ((H) \quad (L))$

Над вмістом комірки пам'яті, адреса якого поміщена в регістри H й L, і вмістом акумулятора виконується операція "виключаюче АБО". Результат завантажується в акумулятор. Скидаються флаги перенесення CY і допоміжного перенесення AC.

1	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

XRA M AE

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, CY, AC

Виключаюче АБО над вмістом акумулятора та константою

$(A) \leftarrow (A) \vee (\text{байт } 2)$

Над вмістом акумулятора і другого байта команди виконується операція "виключаюче АБО". Результат завантажується в акумулятор. Скидаються флаги перенесення CY і допоміжного перенесення AC.

1	1	1	0	1	1	1	0
Дані							

XRI байт EE

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC.

Логічне АБО над вмістом акумулятора та регістра

$$ORA \quad r \quad (A) \leftarrow (A) \vee (r)$$

Над вмістом регістра r та акумулятора виконується команда логічного АБО. Результат завантажується в акумулятор. Скидаються флаги перенесення CY і допоміжного перенесення AC.

1	0	1	1	0	S	S	S
---	---	---	---	---	---	---	---

ORA	A	B7
ORA	B	B0
ORA	C	B1
ORA	D	B2
ORA	E	B3
ORA	H	B4
ORA	L	B5

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, CY, AC

Логічне АБО над вмістом акумулятора та комірки пам'яті

$$ORA \ M \quad (A) \leftarrow (A) \vee ((H) \quad (L))$$

Над вмістом комірки пам'яті, адреса якої визначена в регістрах H и L, та акумулятора виконується операція логічного АБО. Результат завантажується в акумулятор. Скидаються флаги перенесення CY і допоміжного перенесення AC.

1	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---

ORA M B6

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, CY, AC.

Логічне АБО над вмістом акумулятора та константою

$$ORI \ \text{байт} \quad (A) \leftarrow (A) \vee (\text{байт} \ 2)$$

Над вмістом другого байта команди та акумулятора виконується операція логічного АБО. Результат завантажується в акумулятор. Скидаються флаги перенесення CY і допоміжного перенесення AC.

1	1	1	1	0	1	1	0
Дані							

ORI байт F6

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC.

Порівняння вмісту регістра з вмістом акумулятора

$$CMP \quad r \quad (A) - (r)$$

Вміст регістра r віднімається від вмісту акумулятора, результат віднімання відкидається. Акумулятор не змінюється. В результаті віднімання флаги встановлюються аналогічно виконанню команди віднімання.

1	0	1	1	1	S	S	S
---	---	---	---	---	---	---	---

CMP	A	BF
CMP	B	B8
CMP	C	B9
CMP	D	BA
CMP	E	BB
CMP	H	BC
CMP	L	BD

Циклів - 1, тактів - 4, тип адресації - регістрова. Змінюються ознаки: Z, S, P, CY, AC

Порівняння вмісту комірки пам'яті з вмістом акумулятора

$CMP\ M(A) - ((H) (L))$

Вміст комірки пам'яті, адреса якої визначена у регістрах H и L, віднімається від вмісту акумулятора. Результат віднімання відкидається. Акумулятор не змінюється. Регістр ознак встановлюється аналогічно виконанню команди віднімання.

1	0	1	1	1	1	1	0
---	---	---	---	---	---	---	---

CMP M BE

Циклів - 2, тактів - 7, тип адресації - непряма регістрова. Змінюються ознаки: Z, P, S, CY, AC

Порівняння вмісту акумулятора з константою

$CPI\ (байт) (A) - (байт\ 2)$

Вміст другого байту команди віднімається від вмісту акумулятора. Результат віднімання відкидається. Регістр ознак встановлюється аналогічно виконанню команди віднімання.

1	1	1	1	1	1	1	0
Дані							

CPI байт FE

Циклів - 2, тактів - 7, тип адресації - безпосередня. Змінюються ознаки: Z, S, P, CY, AC

Циклічний зсув вліво

RLC

Вміст акумулятора зсувається вліво на одну позицію. Вміст самого старшого біту заноситься в молодший біт і біт флага перенесення.

0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

RLC 07

Циклів - 1, тактів - 4. Команда впливає тільки на ознаку переносу CY.

Циклічний зсув вправо

$RRC(A_n) \leftarrow (A_{n+1}), (A_7) \leftarrow (A_0), (CY) \leftarrow (A_0)$

Вміст акумулятора зсувається на одну позицію вправо. Вміст молодшого біта заноситься в самий старший біт і біт флага перенесення.

0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---

RRC OF

Циклів - 1, тактів - 4. Команда впливає тільки на ознаку переносу CY.

Зсув вліво через перенесення

$$\text{RAL } (A_{n+1}) \leftarrow (A_n), (CY) \leftarrow (A_7), (A_0) \leftarrow (CY)$$

Вміст акумулятора зсувається на одну позицію вліво через біт ознаки перенесення CY. Молодший біт встановлюється рівним попередньому стану ознаки перенесення, а біт ознаки перенесення CY - рівним величині попереднього значення старшого біту акумулятора.

0	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---

RAL 17

Циклів - 1, тактів - 4. Команда впливає тільки на ознаку перенесення CY.

Зсув вправо через перенесення

$$\text{RAR } (A_n) \leftarrow (A_{n+1}), (CY) \leftarrow (A_0), (A_7) \leftarrow (CY)$$

Вміст акумулятора зсувається на одну позицію вправо через біт ознаки перенесення. Старший біт акумулятора встановлюється рівним попередньому значенню ознаки перенесення, а ознаки перенесення - рівним попередньому значенню молодшого біту акумулятора.

0	0	0	1	1	1	1	1
---	---	---	---	---	---	---	---

RAR 1F

Циклів - 1, тактів - 4. Команда впливає тільки на ознаку переносу CY.

Доповнення акумулятора

$$\text{CMA } (A) \leftarrow (\bar{A})$$

Вміст акумулятора інвертується (біт, рівний 1, стає рівним 0; біт, рівний 0, стає рівним 1).

0	0	1	0	1	1	1	1
---	---	---	---	---	---	---	---

CMA 2F

Циклів - 1, тактів - 4. Регістр ознак не змінюється.

Доповнення ознаки перенесення

$$\text{CMC } (CY) \leftarrow (\overline{CY})$$

Інвертується біт ознаки перенесення CY. Інші ознаки не використовуються.

0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---

CMC 3F

Циклів - 1, тактів - 4. Змінюються ознаки: CY

Встановлення ознаки перенесення

$$\text{STC } (CY) \leftarrow 1$$

Біт ознаки перенесення встановлюється в 1. Інші ознаки не використовуються.

0	0	1	1	0	1	1	1
---	---	---	---	---	---	---	---

STC 37

Циклів - 1, тактів - 4. Змінюються ознаки: CY

2.4. Команди переходу

Команди цієї групи змінюють нормальне послідовне виконання програми. Виконання команд не впливає на стан регістра ознак.

Є два типи команд переходу: безумовні й умовні.

Команди безумовного переходу здійснюють передачу керування, змінюючи значення програмного лічильника IP.

Команди умовного переходу спочатку перевіряють стан однієї з чотирьох ознак регістра ознак процесора, щоб визначити, чи виконана вказана в команді умова.

Є наступні умови переходу:

Умова	Код
NZ - не нуль ($Z = 0$)	000
Z - нуль ($Z = 1$)	001
NC - немає переносу ($CY = 0$)	010
C - є перенос ($CY = 1$)	011
PO - непарний результат ($P = 0$)	100
PE - парний результат ($P = 1$)	101
P - плюс ($S = 0$)	110
M - мінус ($S = 1$)	111

Безумовний перехід

JMP [адр.] (IP) ← (байт 3) (байт 2)

Керування передається команді, адреса якої вказана у другому і третьому байтах команди переходу.

1	1	0	0	0	0	1	1
Молодший байт адреси							
Старший байт адреси							

JMP [адр.] C3

Циклів - 3, тактів - 10, тип адресації - безпосередня. Регістр ознак не змінюється.

Непрямий безумовний перехід

PCNL (IP_H) ← (H)
 PCNL (IP_L) ← (L)

Вміст регістру H пересилається в 8 старших біт лічильник команд IP, вміст регістру L пересилається в 8 молодших біт.

1	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

PCNL E9

Циклів - 1, тактів - 6, тип адресації - регістрова. Регістр ознак не змінюється

Умовний перехід

J (УМОВА) [адр.] Якщо умова (CCC) вірна, то (PC) ← (байт 3) (байт 2)

Якщо вказана умова справджується, то керування передається команді, адреса якої вказана у другому і третьому байтах команди переходу. Якщо умова не виконується, то послідовний код програми не змінюється.

1	1	C	C	C	0	1	0
Молодший байт адреси							
Старший байт адреси							

JNZ	[адр.]	C2
JZ	[адр.]	CA

JNC	[адр.]	D2
JC	[адр.]	DA
JPO	[адр.]	E2
JPE	[адр.]	EA
JP	[адр.]	F2
JM	[адр.]	FA

Циклів - 3, тактів - 10, тип адресації - безпосередня. Регістр ознак не змінюється.

Безумовний виклик підпрограми

$$\begin{aligned} & ((SP) - 1) \leftarrow (IP_H) \\ \text{CALL } [адр.] & \quad ((SP) - 2) \leftarrow (IP_L) \\ & (SP) \leftarrow (SP) - 2 \\ & (IP) \leftarrow (\text{байт } 3) \quad (\text{байт } 2) \end{aligned}$$

Старші 8 біт адреси наступної команди пересилаються в комірку пам'яті, адреса якої на 1 менша вмісту вказівника стеку SP. Молодші 8 біт адреси наступної команди пересилаються в комірку пам'яті, адреса якої на 2 менша величини вказівника стеку SP. Вміст вказівника стеку зменшується на 2. Керування передається команді, адреса якої вказана в другому і третьому байтах команди.

1	1	0	0	1	1	0	1
Молодший байт адреси							
Старший байт адреси							

CALL [адр.] CD

Циклів - 5, тактів - 18, тип адресації - безпосередня/непряма регістрова. Регістр ознак не змінюється.

Умовний виклик підпрограми

$$\begin{aligned} \text{C (УМОВА)} [адр.] \text{ (УМОВНИЙ ВИКЛИК) якщо (CCC), то } & ((SP) - 1) \leftarrow (IP_H), \\ & ((SP) - 2) \leftarrow (IP_L), (SP) \leftarrow (SP) - 2, (IP) \leftarrow (\text{байт } 3) \quad (\text{байт } 2) \end{aligned}$$

Якщо вказана умова виконується, то виконуються дії, описані в команді CALL, в іншому випадку послідовність виконання команд не змінюється.

1	1	C	C	C	1	0	0
Молодший байт адреси							
Старший байт адреси							

CNZ	(адр.)	C4
CZ	(адр.)	CC
CNC	(адр.)	D4
CC	(адр.)	DC
CPO	(адр.)	E4
CPE	(адр.)	EC
CP	(адр.)	F4
CM	(адр.)	FC

Циклів - 5, тактів - 18, тип адресації - безпосередня/непряма регістрова. Регістр ознак не змінюється.

Безумовне повернення з підпрограми

$$\begin{aligned} & (IP_L) \leftarrow ((SP)) \\ \text{RET} & (IP_H) \leftarrow ((SP) + 1) \\ & (SP) \leftarrow (SP) + 2 \end{aligned}$$

Вміст комірки пам'яті, адреса якого завантажується в вказівник стеку SP, пересилається в 8 молодших біт програмного лічильника PC. Вміст комірки пам'яті, адреса якого на 1 більша вмісту вказівника стеку, пересилається в 8 старших біт програмного лічильника. Вміст вказівника стеку збільшується на 2.

1	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---

RET C9

Циклів - 3, тактів - 10, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Умовне повернення з підпрограми

R умова

якщо (CCC), то $(PCL) \leftarrow ((SP))$, $(PCH) \leftarrow ((SP) + 1)$, $(SP) \leftarrow (SP) + 2$

Якщо вказана умова виконується, то виконуються дії, описані в команді RET, в іншому випадку послідовність виконання команд не порушується.

1	1	C	C	C	0	0	0
---	---	---	---	---	---	---	---

RNZ	C0
RZ	C8
RNC	D0
RC	D8
RPO	E0
RPE	E8
RP	F0
RM	F8

Циклів - 3, тактів - 12, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Рестарт

$$\begin{aligned} & ((SP) - 1) \leftarrow (IP_H) \\ & ((SP) - 2) \leftarrow (IP_L) \\ \text{RSTn} & (SP) \leftarrow ((SP) - 2) \\ & (IP) \leftarrow 8 * (NNN) \end{aligned}$$

Старші 8 біт адреси наступної команди пересилаються в комірку пам'яті, адреса якої на 1 менша вмісту вказівника стеку. Молодші 8 біт адреси наступної команди пересилаються в комірку пам'яті, адреса якої на 2 менша вказівника стеку. Вміст вказівника стеку зменшується на 2. Керування передається команді, адреса якої дорівнює коду NNN, помноженому на 8.

1	1	N	N	N	1	1	1
---	---	---	---	---	---	---	---

RST	0	C7
RST	1	CF
RST	2	D7
RST	3	DF
RST	4	E7
RST	5	EF

RST	6	F7
RST	7	FF

Циклів - 3, тактів - 12, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Програмний лічильник після рестарту

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	N	N	N	0	0	0

2.5. Команди вводу/виводу, керування, роботи із стеком

Команди цієї групи виконують операції вводу/виводу, змінюють вміст стеку і стан флагів.

Завантаження в стек вмісту пари регістрів

$$((SP)-1) \leftarrow (rh)$$

$$\text{PUSH } rp \quad ((SP)-2) \leftarrow (rl)$$

$$(SP) \leftarrow (SP) - 2$$

Вміст старшого регістра пари регістрів rp пересилається в комірку пам'яті, адреса якої на 1 менша вмісту вказівника стеку SP . Вміст молодшого регістра пари регістрів rp пересилається в комірку пам'яті, адреса якої на 2 менша вказівника стеку. Вміст вказівника стеку зменшується на 2. Пара регістрів $rp = SP$ не може бути вказана в команді.

1	1	R	P	0	1	0	1
---	---	---	---	---	---	---	---

PUSH B 35 (заслати пари регістрів B і C в стек)

PUSH D D5 (заслати пари регістрів D і E в стек)

PUSH H E5 (заслати пари регістрів H і L в стек)

Циклів - 3, тактів - 12, тип адресації - непряма регістрова.

Завантаження в стек слова стану програми

$$((SP)-1) \leftarrow (A)$$

$$((SP)-2)_0 \leftarrow (CY), \quad ((SP)-2)_1 \leftarrow X$$

$$\text{PUSH PSW} \quad ((SP)-2)_2 \leftarrow (P), \quad ((SP)-2)_3 \leftarrow X$$

$$((SP)-2)_4 \leftarrow (AC), \quad ((SP)-2)_5 \leftarrow X$$

$$((SP)-2)_6 \leftarrow (Z), \quad ((SP)-2)_7 \leftarrow (S)$$

$$(SP) \leftarrow (SP) - 2$$

x - будь-яке значення

Вміст акумулятора пересилається в комірку пам'яті, адреса якої на 1 менша вмісту вказівника стеку SP . Вміст регістра ознак пересилається в комірку пам'яті, адреса якої на 2 менша вказівника стеку. Вміст вказівника стеку зменшується на 2.

1	1	1	1	0	1	0	1
---	---	---	---	---	---	---	---

PUSH PSW F5

Циклів - 3, тактів - 12, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Зчитування із стеку вміст пари регістрів

$$(rl) \leftarrow ((SP))$$

$$\text{POP } rp \quad (rh) \leftarrow ((SP)+1)$$

$$(SP) \leftarrow (SP) + 2$$

Вміст комірки пам'яті, адреса якої визначається вмістом вказівника стеку SP , пересилається в молодший регістр пари регістрів rp . Вміст комірки пам'яті, адреса якого на 1

більша вмісту вказівника стеку, пересилається в старший регістр пари регістрів гр. Вміст вказівника стеку збільшується на 2. У команді не може бути вказана пара регістрів гр = SP.

1	1	R	P	0	0	1
---	---	---	---	---	---	---

POP B C1 (зчитати з стека пари регістрів B і C)

POP D D1 (зчитати з стека пари регістрів D і E)

POP H E1 (зчитати зі стека пари регістрів H і L)

Циклів - 3, тактів - 10, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Зчитування із стеку слова стану програми

$$((CY)) \leftarrow ((SP))_0$$

$$(P) \leftarrow ((SP))_2$$

$$(AC) \leftarrow ((SP))_4$$

POP PSW $(Z) \leftarrow ((SP))_6$

$$(S) \leftarrow ((SP))_7$$

$$(A) \leftarrow ((SP)) + 1$$

$$(SP) \leftarrow ((SP)) + 2$$

Вміст комірки пам'яті, адреса якої визначається вмістом вказівника стеку SP, використовується для відновлення стану флагів. Вміст комірки пам'яті, адреса якої на 1 більша вмісту вказівника стеку, пересилається в акумулятор. Вміст вказівника стеку збільшується на 2.

1	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---

POP PSW F1

Циклів - 3, тактів - 10, тип адресації - непряма регістрова. Змінюються ознаки: Z, S, P, CY, AC

Обмін вмісту верхівки стеку і вмісту регістрів H та L

$$(L) \leftrightarrow ((SP))$$

XTHL $(H) \leftrightarrow ((SP) + 1)$

Вміст регістра L обмінюється з вмістом комірки пам'яті, адреса якого завантажена в вказівнику стеку SP. Вміст регістра H обмінюється з вмістом комірки пам'яті, адреса якої на 1 більша вмісту вказівника стеку.

1	1	1	0	0	0	1	1
---	---	---	---	---	---	---	---

XTHL E3

Циклів - 5, тактів - 16, тип адресації - непряма регістрова. Регістр ознак не змінюється.

Пересилання вмісту регістрів H та L у вказівник стеку

SPHL $(SP) \leftrightarrow (H) (L)$

Вміст регістрів H, L пересилається у вказівник стеку.

1	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---

SPHL F9

Циклів - 1, тактів - 6, тип адресації - регістрова. Регістр ознак не змінюється.

Ввід даних з порту

IN порт $A \leftarrow \text{Порт[адреса]}$

Дані, що зчитуються з вказаного у команді порту, пересилаються в акумулятор.

1	1	0	1	1	0	1	1
Порт							

IN байт DB байт

Циклів - 3, тактів - 10, тип адресації - пряма. Регістр ознак не змінюється.

Вивід даних у порт

OUT порт Порт[адреса] ← (A)

Вміст акумулятора передеться у вказаний порт.

1	1	0	1	0	0	1	1
Порт							

OUT байт D3 байт

Циклів - 3, тактів - 10, тип адресації - пряма. Регістр ознак не змінюється.

Дозвіл переривань

EI

Переривання дозволяються при виконанні наступної команди. Переривання не розпізнаються під час виконання команди EI.

1	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---

Циклів - 1, тактів - 4. Регістр ознак не змінюється.

Заборона переривань

DI

Переривання забороняються безпосередньо при виконанні наступної за DI команди. Переривання не розпізнаються під час виконання команди.

1	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---

DI FB

Циклів - 1, тактів - 4. Регістр ознак не змінюється.

Зупинка процесора

HLT

У IP заноситься адреса наступної команди. Процесор не діє до приходу переривання у випадках, коли вони дозволені. Коли переривання заборонені, вихід з виконання команди HLT можливий тільки по сигналу RST. Регістри і флаги не змінюють стану.

HLT 76

Циклів - 1, тактів - 5. Регістр ознак не змінюється.

Команда "Немає операції"

NOP

У IP заноситься адреса наступної команди, і МП переходить до її обробки. Стан флагів і регістрів не змінюється. Команда використовується для утворення коротких витримок часу, відлагодження програми тощо.

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

NOP 00

Циклів - 1, тактів - 4. Регістр ознак не змінюється.

Система команд ОМЕОМ МК51

Мікроконтролери сімейства МК51 підтримують систему з 111 базових команд, які можна розбити на групи, розглянуті нижче.

Надалі використовуються наступні позначення:

Rn, n=0...7- реєстр поточного реєстрового банку;

Ri, i=0.1 - реєстри R0 чи R1 поточного реєстрового банку, використовувані при непрямій адресації;

#d - число довжиною 1 байт (8біт);

#d16 - число довжиною 2 байти ;

ad - пряма адреса в ОЗП чи даних в адресному просторі РСФ;

bit - адреса біту (у бітових операціях);

PC - лічильник команд;

(ad) - вміст комірки пам'яті з адресою (ad);

((Ri)) - вміст комірки пам'яті, адреса якого знаходиться в Ri;

<- стрілка вказує розміщення результату операції.

При виконанні арифметичних операцій операнди і результати операції подаються в стандартних для восьмирозрядних обчислювальних систем форматах, причому молодший розряд числа відповідає розряду D0, а старший - D7.

1. ГРУПА КОМАНД ПЕРЕДАЧІ ДАНИХ

1.1. ПЕРЕСИЛАННЯ В МЕЖАХ РЕЗИДЕНТНОГО ЗУПД

Здійснюються командою **MOV X,Y**, де X - адреса призначення, за якою записується число чи операнд, зазначений як Y. Вміст комірки пам'яті, зазначеної як операнд Y, при виконанні операції MOV не змінюється. Припустимі варіанти пересилань в межах резидентного ЗУПД показані у вигляді графа на рис. 1.

Об'єм пам'яті в ЗУПД, що займається командами MOV, швидкість їхнього виконання в машинних циклах, а також виконувані функції приведені в табл. 1.

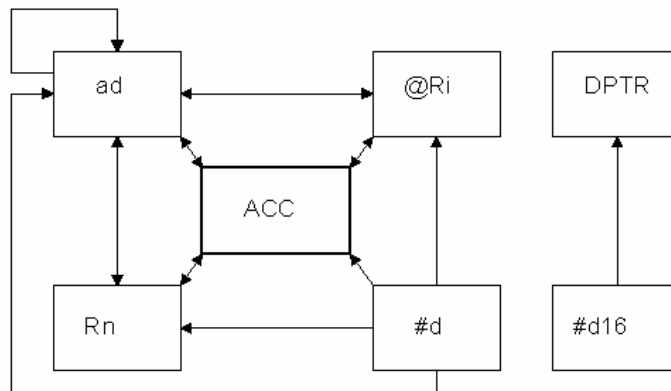


Рис. 1

Таблиця 1

Мнемокод	Б	Ц	Операція
MOV A, Rn	1	1	(A) ← (Rn)
MOV A, ad	2	1	(A) ← (ad)
MOV A, @Ri	1	1	(A) ← ((Ri))
MOV A, #d	2	1	(A) ← #d
MOV Rn, A	1	1	(Rn) ← (A)
MOV Rn, ad	2	2	(Rn) ← (ad)
MOV Rn, #d	2	1	(Rn) ← #d

MOV ad, A	2	1	$(ad) \leftarrow (A)$
MOV ad, Rn	2	2	$(ad) \leftarrow (Rn)$
MOV add, ads	3	2	$(add) \leftarrow (ads)$
MOV ad, @Ri	2	2	$(ad) \leftarrow ((Ri))$
MOV ad, #d	3	2	$(ad) \leftarrow \#d$
MOV @Ri, A	1	1	$((Ri)) \leftarrow (A)$
MOV @Ri, ad	2	2	$((Ri)) \leftarrow (ad)$
MOV @Ri, #d	2	1	$((Ri)) \leftarrow \#d$
MOV DPTR, #d16	3	2	$(DPTR) \leftarrow \#d16$

1.2. ПЕРЕСИЛАННЯ ЗА МЕЖАМИ РЕЗИДЕНТНОГО ЗУПД здійснюються з ЗУПП командою MOVC (MOVE Code memory), а з/у зовнішню пам'ять даних – командою MOVX (MOVE eXternal memory). Система підтримуваних процесором команд вигляду MOVX і MOVC зведена в табл. 2.

Таблиця 2

Мнемокод	Операція
MOV A, @A+DPTR	$(A) \leftarrow ((A)+(DPTR))$
MOVC A, @A+PC	$(PC) \leftarrow (PC)+1$ $(A) \leftarrow ((A)+(PC))$
MOVX A, Ri	$(A) \leftarrow ((Ri))$
MOVX A, @DPTR	$(A) \leftarrow ((DPTR))$
MOVX @Ri, A	$((Ri)) \leftarrow (A)$
MOVX @DPTR, A	$((DPTR) \leftarrow (A)$

1.3. КОМАНДИ ОБМІНУ мають мнемокод XCH A, X

При виконанні цієї команди вміст акумулятора міняється місцем із вмістом комірки пам'яті, зазначеної як операнд X. Допустимі види команд обміну і їх характеристики зведені в табл. 3.

Таблиця 3

Мнемокод	Б	Ц	Операція
XCH A, Rn	1	1	$(A) \leftrightarrow (Rn)$
XCH A, ad	2	1	$(A) \leftrightarrow (ad)$
XCH A, @Ri	1	1	$(A) \leftrightarrow ((Ri))$

Крім зазначених команд, до групи команд передачі даних прийнято відносити операції:

PUSH ad - запис в стек,

POP ad - витяг зі стеку.

Ці команди виконуються за 2 машинних цикли і займають у пам'яті програм по 2 байти.

Як відомо, стеком називається область ОЗП зі спрощеним доступом за адресою, автоматично формованою вказівником стеку SP. Відмінність роботи стека ОМЕОМ сімейства МК51 полягає в тому, що при записі інформації в стек спочатку інкрементується (збільшується на одиницю) вміст SP, а потім за адресою, зазначеною в ньому, записується операнд, зазначений у команді.

Витяг операнду зі стека здійснюється в зворотньому порядку.ГУ, причому адреса вершини стеку, записана на початку програми в SP, може збігатися з адресою комірки, використовуваної як звичайне ОЗУ.

Природно, що розмір стеку повинен враховувати не тільки об'єм інформації, записуваної за командам PUSH, але й об'єм, записуваний адресами повернення при використанні переривань і підпрограм з врахуванням глибини їхньої вкладеності (кожна адреса повернення займає 2 байти).

Варто мати на увазі, що після рестарту процесора в SP записується число 7, і, якщо в програмі використовуються переривання, чи підпрограми команди PUSH, для запобігання втрати інформації, записаної в регістри банку 1, варто змінити значення SP (положення вершини стека).

2. ГРУПА КОМАНД АРИФМЕТИЧНИХ ОПЕРАЦІЙ містить у собі:

- ADD A,X - додавання;
- ADDC A,X - додавання з врахуванням перенесення;
- SUBB A,X - віднімання з врахуванням запозичення;
- INC X - одиничне позитивне збільшення (інкремент);
- DEC X - одиничне негативне збільшення (декремент);
- MUL AB - беззнакове множення;
- DIV AB - беззнакове ділення;
- DA A - десяткову корекцію акумулятора.

Команда ADD реалізує функцію обчислення суми вмісту акумулятора з операндом X (або як цілих чисел без знаку, або як цілих зі знаком у додатковому коді) і запис результату в акумулятор. Вміст операнду X (якщо це комірка пам'яті) при цьому не зміниться. При виконанні команди формується ознака перенесення C.

Команда ADDC (SUBB) на додаток до обчислення суми (різниці) враховує перенесення між байтами і може бути використана для обробки операндів, довжина яких перевищує 8 розрядів.

Розглянемо як приклад додавання трьохбайтових чисел. Нехай перше число розміщене в трьох комірках пам'яті за адресами 30H, 31H, 32H, причому, молодший байт розміщений в комірці 30H, а старший - в комірці з адресою 32H.

Аналогічно, друге число розміщене в комірках 38H, 39H и 3AH. Суму двох чисел необхідно розмістити за місцем другого числа (див. рис. 2).

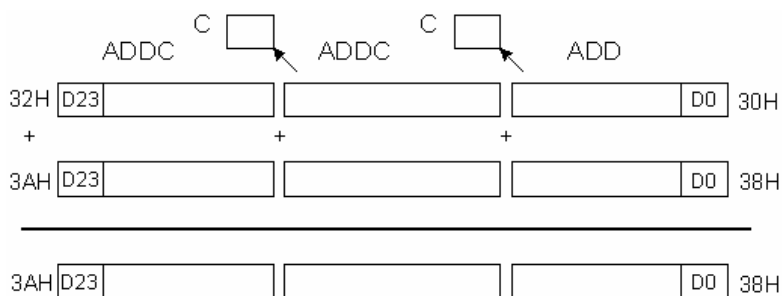


Рис. 2

Як і при будь-якому виді додавання, обчислення суми варто починати з молодших розрядів за допомогою команди ADD. Якщо отриманий в результаті такого додавання результат виявиться меншим, ніж максимально представимо в рамках байту число (255 при беззнаковому додаванні чи 127 при знаковому), то перенесення з байту не виникне й ознака C=0.

Якщо ж результат буде більшим (наприклад, якщо кожний з операндів більший за половину максимального числа), те результат не може бути вірогідно представлений у межах байта; виникає перенесення C=1, яке варто врахувати далі при додаванні старших байтів. У загальному вигляді при додаванні старших байтів повинна бути виконана операція:

$$X + Y + C$$

при цьому якщо C=0 (перенесення молодших байтів не виникло), буде обчислена сума у вигляді

$$X + Y,$$

а якщо перенесення виникло, він буде врахований за всіма правилами арифметичного додавання:

$$X + Y + 1.$$

Функція $X + Y + C$ може бути виконана за допомогою команди ADDC, при цьому фрагмент програми, що реалізує функцію додавання двох трьохбайтових чисел, буде мати вигляд (задача оптимізації програми тут не ставиться):

```
MOV A, 30H
ADD A, 38H
MOV 38H, A
MOV A, 31H
ADDC A, 39H
MOV 39H, A
MOV A, 32H
ADDC A, 3AH
MOV 3AH, A.
```

Команда MUL AB перемножує цілі числа без знаків з акумулятора і регістра B.

Добуток двох восьмирозрядних чисел буде мати довжину 16 розрядів і займати 2 байти. Старший байт добутку міститься в регістрі B, а молодший - в акумуляторі. Якщо результат добутку перевищує 255, то встановлюється флажок переповнення OV в регістрі ознак, в іншому випадку він скидається.

У будь-якому випадку скидається флажок перенесення.

Команда DIV AB здійснює беззнакове ділення восьмибітного цілого числа з акумулятора на восьмибітне число з регістра B. Частка від ділення заноситься в акумулятор, залишок - у регістр B.

Флажки перенесення C і переповнення OV скидаються.

Основні види арифметичних операцій і їх характеристики зведені в табл. 4.

Таблиця 4

Мнемокод	Б	Ц	Операція
ADD A, Rn	1	1	$(A) \leftarrow (A) + (Rn)$
ADD A, ad	2	1	$(A) \leftarrow (A) + (ad)$
ADD A, @Ri	1	1	$(A) \leftarrow (A) + ((Ri))$
ADD A, #d	2	1	$(A) \leftarrow (A) + \#d$
ADDC A, Rn	1	1	$(A) \leftarrow (A) + (Rn) + (C)$
ADDC A, ad	2	1	$(A) \leftarrow (A) + (ad) + (C)$
ADDC A, @Ri	1	1	$(A) \leftarrow (A) + ((Ri)) + (C)$
ADDC A, #d	2	1	$(A) \leftarrow (A) + \#d + (C)$
DA A	1	1	Якщо $(A_{0-3}) > 9 \vee ((C) = 1)$, то $(A_{0-3}) \leftarrow (A_{0-3}) + 6$, потім якщо $(A_{4-7}) > 9 \vee ((C) = 1)$, то $(A_{4-7}) \leftarrow (A_{4-7}) + 6$
SUBB A, Rn	1	1	$(A) \leftarrow (A) - (C) - (Rn)$
SUBB A, ad	2	1	$(A) \leftarrow (A) - (C) - (ad)$
SUBB A, @Ri	1	1	$(A) \leftarrow (A) - (C) - ((Ri))$
SUBB A, #d	2	1	$(A) \leftarrow (A) - \#d$
INC A	1	1	$(A) \leftarrow (A) + 1$
INC Rn	1	1	$(Rn) \leftarrow (Rn) + 1$
INC ad	2	1	$(ad) \leftarrow (ad) + 1$
INC @Ri	1	1	$((Ri)) \leftarrow ((Ri)) + 1$
INC DPTR	1	2	$(DPTR) \leftarrow (DPTR) + 1$
DEC A	1	1	$(A) \leftarrow (A) - 1$
DEC Rn	1	1	$(Rn) \leftarrow (Rn) - 1$
DEC ad	2	1	$(ad) \leftarrow (ad) - 1$
DEC @Ri	1	1	$((Ri)) \leftarrow ((Ri)) - 1$
MUL AB	1	4	$(B)(A) \leftarrow (A) \times (B)$
DIV AB	1	4	$(A).(B) \leftarrow (A) / (B)$

3. ГРУПА КОМАНД ЛОГІЧНИХ ОПЕРАЦІЙ включає команди логічного АБО (ORL), І (ANL) і виключне АБО (XRL), які реалізують відповідні порозрядні операції над вмістом операндів.

Характерним для ОМЕОМ сімейства МК51 є необов'язковість використання акумулятора в логічних командах в якості одного з джерел операндів, та як приймач результату операції.

Таблиця 5 дає узагальнення підтримуваних типів логічних команд і їх характеристики.

Таблиця 5

Мнемокод	Б	Ц	Операція
ANL A, Rn	1	1	$(A) \leftarrow (A) \wedge (Rn)$
ANL A, ad	2	1	$(A) \leftarrow (A) \wedge (ad)$
ANL A, @Ri	1	1	$(A) \leftarrow (A) \wedge ((Ri))$
ANL A, #d	2	1	$(A) \leftarrow (A) \wedge \#d$
ANL ad, A	2	1	$(ad) \leftarrow (ad) \wedge (A)$
ANL ad, #d	3	2	$(ad) \leftarrow (ad) \wedge \#d$
ORL A, Rn	1	1	$(A) \leftarrow (A) \vee (Rn)$
ORL A, ad	2	1	$(A) \leftarrow (A) \vee (ad)$
ORL A, @Ri	1	1	$(A) \leftarrow (A) \vee ((Ri))$
ORL A, #d	2	1	$(A) \leftarrow (A) \vee \#d$
ORL ad, A	2	1	$(ad) \leftarrow (ad) \vee (A)$
ORL ad, #d	3	2	$(ad) \leftarrow (ad) \vee \#d$
XRL A, Rn	1	1	$(A) \leftarrow (A) \nabla (Rn)$
XRL A, ad	2	1	$(A) \leftarrow (A) \nabla (ad)$
XRL A, @Ri	1	1	$(A) \leftarrow (A) \nabla ((Ri))$
XRL A, #d	2	1	$(A) \leftarrow (A) \nabla \#d$
XRL ad, A	2	1	$(ad) \leftarrow (ad) \nabla (A)$
XRL ad, #d	3	2	$(ad) \leftarrow (ad) \nabla \#d$

4. ГРУПА КОМАНД ОПЕРАЦІЙ З БІТАМИ

ОМЕОМ сімейства МК51 підтримують групу операцій: скидання CLR, встановлення SETB, інверсію CPL, кон'юнкцію ANL, диз'юнкцію ORL і пересилання MOV операндів, в якості яких використовуються біти - окремі розряди побітноадресованих регістрів РСФ чи біти з області прямоадресованих бітів ОЗП.

Так, наприклад, за допомогою команди SETB P1.0

можна встановити в стан логічної одиниці сигнал на виводі 1 ОМЕОМ, а командою MOV P2.5, C вивести значення ознаки перенесення на вивід 26.

Для адресації біт використовується пряма восьмирозрядна адреса; непряма адресація біт не підтримується.

Основні характеристики і допустимі комбінації бітових операцій зведені в табл. 6.

Таблиця 6

Мнемокод	Б	Ц	Операція
CLR C			$(C) \leftarrow 0$
CLR bit			$(bit) \leftarrow 0$
SET B C			$(C) \leftarrow 1$
SET B bit			$(bit) \leftarrow 1$
CPL C			$(C) \leftarrow (C)$
CPL bit			$(bit) \leftarrow (bit)$
ANL C, bit			$(C) \leftarrow (C) (bit)$
ANL C, /bit			$(C) \leftarrow (C) (bit)$
ORL C, bit			$(C) \leftarrow (C) \vee (bit)$
ORL C, /bit			$(C) \leftarrow (C) \vee (bit)$
MOV C, bit			$(C) \leftarrow (bit)$
MOV bit, C			$(bit) \leftarrow (C)$

5. ГРУПА КОМАНД ПЕРЕДАЧІ КЕРУВАННЯ включає стандартні для будь-якого процесора команди безумовного переходу, команди умовних переходів за значеннями ознак і біт, команди виклику / повернення з підпрограм, а також ряд оригінальних операцій, характерних тільки для ОМЕОМ сімейства МК51.

Команди передачі керування виконуються за 2 машинних цикли.

Розрізняють два формати команди безумовних переходів і аналогічні їм 2 формати команди виклику підпрограм:

- LJMP <address> - безумовний перехід
- LCALL <address> - виклик підпрограми в адресному просторі 64 Кбайт.
- AJMP <address> - безумовний перехід і
- ACALL <address> - виклик підпрограми в адресному просторі 2 Кбайт.

Команди AJMP і ACALL займають у порівнянні з LJMP і LCALL менше місця в ЗУПП - 2 байти замість трьох. Це обставина в деяких випадках може мати дуже важливе значення, оскільки будь-які умовні переходи можливі тільки на відстань + 127 байт у ЗУПП щодо коду команди умовного переходу.

Умовні переходи можуть виконуватися:

- за значенням біта перенесення C:

JC <address> - переходити на <address>, якщо C=1, а

JNC <address> - якщо C=0.

- за нульовим значенням акумулятора:

JZ <address> - переходити на <address>, якщо вміст акумулятора дорівнює 0 і

JNZ <address> - якщо не дорівнює 0.

- за значенням біта:

JB bit, <address> - переходити на <address>, якщо значення біта bit=1,

JNB bit, <address> - переходити, якщо значення біта bit=0,

JBC bit, <address> - переходити, якщо значення біта bit=1

і скидання значення цього біта в 0. Використання останньої з цих команд зручно при обслуговуванні переривань.

До команд умовних переходів відносять також команду організації циклу

DJNZ X, <address>

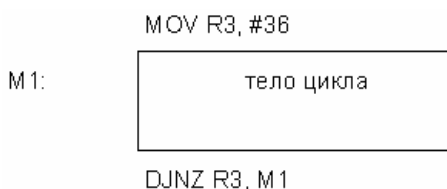
і команди чекання події

CJNE X, #d, <address>

CJNE A, ad, <address>

Команда DJNZ X, <address> віднімає одиницю з вмісту операнда X, у якості якого може бути зазначений регістр чи прямоадресована комірка пам'яті, зберігаючи його нове значення, і порівнює результат з нулем.

Якщо результат не дорівнює 0, то процесор переходить до виконання команди, розташованої за адресою <address>, а якщо дорівнює, то до наступної за DJNZ команди. Команда DJNZ використовується для організації циклу.



Так, наприклад, фрагмент програми дозволяє організувати 36 проходів тіла циклу.

Команда CJNE X, #d, <address> порівнює значення операнда X, у якості якого можуть бути зазначені акумулятор, чи регістр з непрямою адресою комірки пам'яті, і числа #d, і, якщо вони не рівні, процесор переходить до виконання команди, розташованої за адресою <address>. Якщо ж рівні, то до виконання наступної за CJNE команди.

Дія команди CJNE A, ad, <address> аналогічна.

Так, наприклад, команда

STP: CJNE A, P0, STP

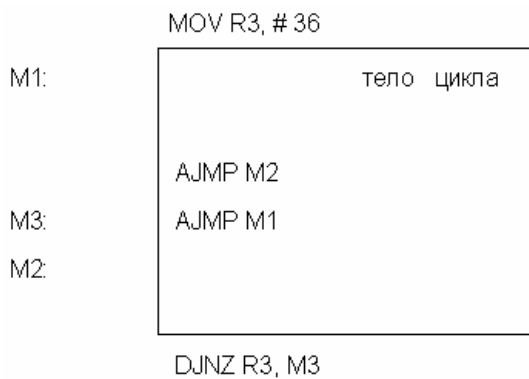
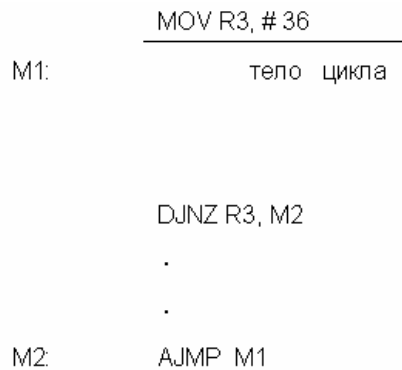
зупиняє виконання програми доти, поки на виводи порту P0 не надійде комбінація логічних сигналів, еквівалентна вмісту акумулятора.

Та обставина, що умовні переходи можливі тільки на порівняно "короткі відстані", часто викликає незручності при програмуванні, однак практично завжди ці незручності можуть бути обійдені.

Наприклад, якщо тіло циклу займає в ЗУПП більш 127 байт, можна застосувати наступну конструкцію.

У випадку, якщо той самий фрагмент алгоритму використовується в декількох місцях програми, прийнято з метою скорочення загального обсягу програми оформляти його як підпрограму.

Звертання до підпрограми можливо з будь-якої місця програми чи іншої підпрограми, при цьому процесор переходить до виконання викликаної підпрограми, запам'ятовуючи в стеку адресу повернення. Адреса повернення займає в стеку два байти.



Повернення з підпрограми здійснюється за командою RET, розміщеної наприкінці підпрограми.

Команда RETI використовується для повернення процесора з підпрограми переривань; її відмінності від команди RET будуть розглянуті далі при описі процесу переривань.

Можливі команди передачі керування і їхня довжина приведені в табл. 7, де:

ad 16 - шестнадцятирозрядна адреса (64 Кбайт);

ad 11 - одинадцятирозрядна адреса (2 Кбайт);

rel - восьмирозрядна адреса (127 байт);

Таблиця 7

Мнемокод	Б	Ц	Операція
LJMP ad 16	3		(PC) ← ad 16
AJMP ad 11	2		(PC) ← (PC) + 2

			$(PC_{0-10}) \leftarrow ad\ 11$
SJMP rel	2		$(PC) \leftarrow (PC) + 2$ $(PC_{0-10}) \leftarrow rel$
JMP @A+DPTR	1		$(PC) \leftarrow (A) + (DPTR)$
JZ rel	2		$(PC) \leftarrow (PC) + 2,$ якщо $(A) = 0$, то $(PC) \leftarrow (PC) + rel$
JNZ rel	2		$(PC) \leftarrow (PC) + 2,$ якщо $(A) \neq 0$, то $(PC) \leftarrow (PC) + rel$
JC rel	2		$(PC) \leftarrow (PC) + 2,$ якщо $(C) = 0$, то $(PC) \leftarrow (PC) + rel$
JNC rel	2		$(PC) \leftarrow (PC) + 2,$ якщо $(C) \neq 0$, то $(PC) \leftarrow (PC) + rel$
JB bit, rel	3		$(PC) \leftarrow (PC) + 3,$ якщо $(bit) = 0$, то $(PC) \leftarrow (PC) + rel$
JNB bit, rel	3		$(PC) \leftarrow (PC) + 3,$ якщо $(bit) \neq 0$, то $(PC) \leftarrow (PC) + rel$
JBC bit, rel	3		$(PC) \leftarrow (PC) + 3,$ якщо $(bit) = 0$, то $(bit) \leftarrow 0$ і $(PC) \leftarrow (PC) + rel$
DJNZ Rn, rel	2		$(PC) \leftarrow (PC) + 2, (Rn) \leftarrow (Rn) - 1,$ якщо $(Rn) = 0$, то $(PC) \leftarrow (PC) + rel$
DJNZ ad, rel	2		$(PC) \leftarrow (PC) + 2, (ad) \leftarrow (ad) - 1,$ якщо $(ad) = 0$, то $(PC) \leftarrow (PC) + rel$
CJNE A, ad, rel	3		$(PC) \leftarrow (PC) + 3,$ якщо $(A) = (ad)$, то $(PC) \leftarrow (PC) + rel,$ якщо $(A) < (ad)$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
CJNE A, #d, rel	3		$(PC) \leftarrow (PC) + 3,$ якщо $(A) = \#d$, то $(PC) \leftarrow (PC) + rel,$ якщо $(A) < \#d$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
CJNE Rn, #d, rel	3		$(PC) \leftarrow (PC) + 3,$ якщо $(Rn) = \#d$, то $(PC) \leftarrow (PC) + rel,$ якщо $(Rn) < \#d$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
CJNE @Ri, #d, rel	3		$(PC) \leftarrow (PC) + 3,$ якщо $((Ri)) = \#d$, то $(PC) \leftarrow (PC) + rel,$ якщо $((Ri)) < \#d$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
LCALL ad 16	3		$(PC) \leftarrow (PC) + 3, (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{0-7}), (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{8-15}), (PC) \leftarrow ad\ 16$
ACALL ad 11	2		$(PC) \leftarrow (PC) + 2, (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{0-7}), (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{8-15}), (PC_{0-10}) \leftarrow ad\ 11$
RET	1		$(PC_{8-15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1,$ $(PC_{0-7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
RETI	1		$(PC_{8-15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1,$ $(PC_{0-7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$

Команди операцій зсуву OMEOM сімейства МК51 підтримують операції циклічного зсуву акумулятора:

- замкнуті - RR (вправо) і RL (вліво),
- через біт C - RRC(вправо) і RLC (вліво).

Рух бітів у розрядах акумулятора при виконанні команд RLA і RLCA ілюструються рис. 3. У ЗУПП команди зсувних операцій займають 1 байт і виконуються за 1 машинний цикл.

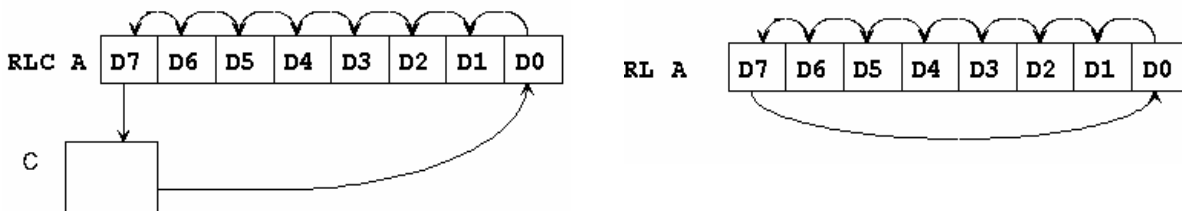


Рис. 3

6. ІНШІ КОМАНДИ

До інших команд відносяться:

- команда очищення акумулятора CLR A, виконання якої перетворює в 0 вміст акумулятора;
- команда інверсії значення акумулятора CPL A;
- “порожня” операція NOP;
- операція обміну тетрадами в акумуляторі SWAP A;
- операція обміну молодшими тетрадами вмісту акумулятора і непрямо адресованої комірки пам'яті XCHD A,@Ri (молодша тетрада - чотири молодших розряди байту)

Ці команди займають у ЗУПП по одному байту і виконуються за один машинний цикл.

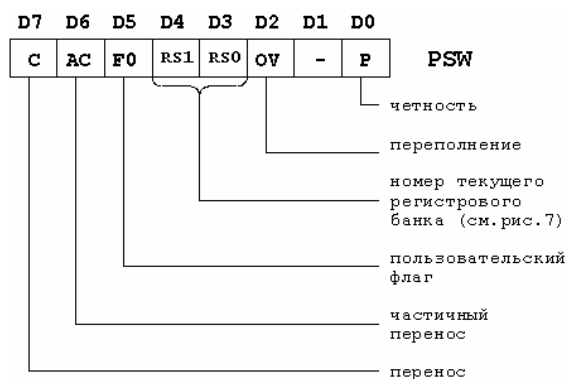


Рис. 4

Формування ознак.

У процесі виконання операцій формуються ознаки:

C - перенесення;

AC — часткове перенесення (перенесення між тетрадами);

Z - нульове значення вмісту акумулятора;

OV - переповнення;

P - парність.