



МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНИ

Тернопільський державний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв

КОНСПЕКТ ЛЕКЦІЙ

з курсу

”Проектування мікропроцесорних систем
керування технологічними процесами”

Тернопіль 2009

Посібник "Проектування мікропроцесорних систем керування технологічними процесами" розроблено у відповідності з навчальною програмою курсу "Проектування мікропроцесорних систем керування технологічними процесами".

Укладачі Медвідь В. доц., к.т.н., Козбур В. асистент, Пісьціо В. асистент

Розглянуто й затверджено на засіданні кафедри автоматизації технологічних процесів та виробництв (протокол № 9 від 12 лютого 2008 р.)

Схвалила й рекомендувала до друку вчена рада факультету комп'ютерних технологій Тернопільського державного технічного університету імені Івана Пулюя (протокол № 5 від 22 лютого 2008 р.).

Рецензенти:

Стухляк П.Д., д.т.н., проф., зав. кафедри комп'ютерноінтегрованих технологій
Карпінський М.П., д.т.н., проф., зав. кафедри. "Комп'ютерної інженерії" ТНЕУ

Зміст

ТЕМА 1. Проектування системи керування на базі МП КР580ВМ80А.	9
1.1. Мікропроцесор КР580ВМ80А.	9
1.1.1. Умовне позначення та призначення виводів мікропроцесора	9
1.1.2. Структурна схема мікропроцесора КР580ВМ80А	11
1.1.3. Слово стана МП	13
1.1.4. Робота мікропроцесора в режимі записування інформації	15
1.1.5. Робота мікропроцесора в режимі читання	16
1.1.6. Переривання в МП	17
1.1.7. Прямий доступ до пам'яті	19
1.1.8. Програмування та система команд мікропроцесора	20
1.1.8.1. Мікропроцесорна система на основі МП К580ВМ80А з точки зору програмування	20
1.1.8.2. Стек	21
1.1.8.3. Класифікація команд МП	21
1.2. Мікропроцесорний комплект КР580	23
1.2.1. Тактовий генератор КР580ГФ24	24
1.2.2. Системний контролер КР580ВК28	26
1.2.3. Регістр КР580ИР82 (ИР83)	28
1.2.4. Шинний формувач КР580ВА86 (ВА87)	29
1.2.5. Контролер переривань КР580ВН59А	30
1.2.6. Контролер прямого доступу до пам'яті КР580ВТ57	32
1.2.7. Програмований таймер-лічильник КР580ВИ53	36
1.3. Організація мікропроцесорної системи керування на основі мікропроцесорного комплекту КР580	38
1.3.1. Організація клавіатури мікропроцесорної системи	44
1.3.2. Організація пристрою відображення інформації	47
1.3.3. Побудова технічних засобів для узгодження сигналів мікропроцесорної системи з об'єктами керування	48
1.3.3.1. Вивід дискретних сигналів	48

1.3.3.2. Ввід дискретних сигналів	50
1.3.3.3. Ввід аналогових сигналів	51
1.3.3.4. Виведення аналогових сигналів	54
1.4. Контрольні запитання	56
ТЕМА №2. Проектування системи керування на базі МП	57
2.1. Мікропроцесор K1810BM86	57
2.1.1. Умовне позначення мікропроцесора та загальна характеристика	57
2.1.2. Призначення виводів МП	58
2.1.3. Вибір режиму конфігурації	60
2.1.4. Структурна схема мікросхеми	60
2.1.5. Адресація портів вводу-виводу	63
2.1.6. Адресація пам'яті	64
2.1.7. Фізична організація пам'яті	66
2.1.8. Переривання	67
2.1.9. Часові діаграми роботи МП у мінімальному режимі роботи	70
2.1.9.1. Часова діаграма читання пам'яті або портів вводу-виводу в мінімальному режимі роботи	70
2.1.9.2. Часова діаграма записування у пам'ять або у порти введення-виведення в мінімальному режимі роботи	72
2.1.9.3. Часова діаграма підтвердження переривань у мінімальному режимі роботи	73
2.1.9.4. Часова діаграма прямого доступу до пам'яті у мінімальному режимі роботи	75
2.1.10. Часові діаграми роботи МП у максимальному режимі	75
2.1.10.1. Часова діаграма читання пам'яті або портів вводу-виводу в максимальному режимі роботи	76
2.1.10.2. Часова діаграма записування у пам'ять чи порти вводу-виводу у максимальному режимі роботи	77
2.1.11. Початкова ініціалізація	78
2.1.12. Система команд МП	78

2.1.13. Мультипроцесорні системи	80
2.2. Мікропроцесор K1810BM88	80
2.3. Арифметичний співпроцесор K1810BM87	84
2.3.1. Призначення виводів співпроцесора K1810BM87	84
2.3.2. Робота співпроцесора K1810BM87	85
2.4. Мікропроцесорний комплект K1810	86
2.4.1. Структура мікропроцесорного комплекту	86
2.4.2. Генератор тактових сигналів K1810ГФ24	87
2.4.3. Контролер системної шини K1810ВГ88	89
2.4.3.1. Режими роботи мікросхеми	91
2.5. Організація мікропроцесорної системи керування на основі мікропроцесорного комплекту K1810	94
2.5.1. Мінімальний режим	94
2.5.2. Максимальний режим	96
2.5.3. Арбітр магістралі KP1810ВБ89	97
2.5.4. Призначення виводів мікросхеми	99
2.6. Організація переривань мікропроцесорної системи керування на основі мікропроцесорного комплекту K1810	104
2.6.1. Схема реалізації протоколу переривань	105
2.7. Контрольні запитання	107
ТЕМА №3. Проектування системи керування на базі однокристальних мікроЕОМ KM1816BE48	108
3.1. Призначення виводів та умовне позначення мікросхеми мікроЕОМ KM1816BE48	108
3.2. Структурна схема мікросхеми мікроЕОМ KM1816BE48	111
3.3. Використання зовнішньої пам'яті та розширеного вводу-виводу	114
3.3.1. МК-системи з зовнішньою пам'яттю програм	115
3.3.2. МК-система з зовнішньою пам'яттю даних	116
3.3.3. МК-система з розширеним вводом/виводом	118
3.4. Робота з клавіатурами	120

3.4.1. Різновидності клавіатур.	120
3.4.2. Ввід коду натисненої клавіші	121
3.4.3. Сканування	122
3.4.4. Усунення деренчання контактів	124
3.4.5. Чекання звільнення клавіші	124
3.4.6. Ідентифікація натисненої клавіші	125
3.4.7. Оформлення процедури вводу	125
3.5. Вивід і відображення інформації. Індикатори	126
3.5.1. Вивід символу на індикатори	129
3.5.2. Виведення інформації на лінійний дисплей	130
3.6. Введення інформації з датчиків	133
3.6.1. Опитування двійкового датчика. Чекання події	133
3.6.2. Чекання статичного сигналу	133
3.6.3. Чекання імпульсного сигналу	134
3.6.4. Усунення деренчання контактів	135
3.6.5. Підрахунок кількості імпульсів	137
3.6.5.1. Підрахунок числа імпульсів між двома подіями	137
3.6.5.2. Підрахунок числа імпульсів за заданий проміжок часу	138
3.6.6. Опитування групи двійкових датчиків	139
3.6.7. Опитування групи імпульсних датчиків	141
3.7. Контрольні запитання	142
ТЕМА №4. Проектування системи керування на базі однокристалічних мікроЕОМ серії K1816BE751	144
4.1. Однокристалічні мікроЕОМ сімейства МК51 (MCS51). Основні характеристики	144
4.2. Архітектура ОМЕОМ 80С51	145
4.2.1. Умовне позначення та призначення виводів мікроЕОМ	148
4.3. МК - системи з зовнішньою пам'яттю програм	152
4.4. Розширення ОЗП	153
4.5. Ввід-вивід у МК-системах	154

4.6. Лічильник-таймер у МК51	156
4.6.1. Режими роботи таймера	158
4.7. Послідовний інтерфейс	160
4.7.1. Універсальний асинхронний прийомопередавач	160
4.7.2. Регістр керування/статусу УАПП	161
4.7.3. Робота УАПП у мультимікроконтролерних системах	163
4.7.4. Швидкість послідовного обміну	164
4.7.5. Особливості роботи УАПП у різних режимах	165
4.7.5.1. Режим 0	165
4.7.5.2. Режим 1	166
4.7.5.2. Режими 2, 3	168
4.8. Система переривань	169
4.9. Контрольні запитання	173
ТЕМА №5. Проектування системи керування на базі PIC16X8X	174
5.1. Особливості контролерів PIC16X8X	174
5.2. Призначення виводів та позначення мікросхеми	177
5.3. Архітектура PIC16X8X	179
5.4. Робота мікроЕОМ	183
5.5. Структура та робота операційного блока	183
5.6. Структура ПЗП програм	185
5.7. Структура ОЗП	188
5.7.1. Пряма та непряма адресація даних	190
5.8. EEPROM даних PIC16X8X	191
5.9. Регістри спеціальних функцій	193
5.9.1. Регістр конфігурації (OPTION)	194
5.10. Порти вводу-виводу	195
5.11. Модуль таймера PIC16X8X	200
5.12. Організація переривань PIC16F8X	203
5.13. Початкова ініціалізація та встановлення у початковий стан	206
5.13.1. Джерела скидання	206

5.13.2. Скидання при відключенні живлення	207
5.13.3. Watcdog-таймер	208
5.14. Режим зниженого енергоспоживання	209
5.15. Генератор та синхронізація	210
5.15.1. Типи генераторів	210
5.15.2. Робота з кварцовим резонатором	210
5.15.3. Синхронізація від зовнішніх джерел	211
5.15.4. RC-генератор	212
5.16. Конфігурація та захист	212
5.16.1. Ідентифікаційний код	212
5.16.2. Конфігураційне слово	213
5.17. Система команд PIC-контролерів серії PIC16X8X	214
5.18. Приклади застосування	217
5.18.1. Універсальний таймер	217
5.18.2. Частотомір на PIC-контролері.	220
5.19. Контрольні запитання	223
Список літератури	225

ТЕМА 1. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ МП КР580ВМ80А.

1.1. Мікропроцесор КР580ВМ80А.

Основні характеристики мікропроцесора КР580ВМ80 наведено в таблиці 1.

Таблиця 1. Характеристики мікропроцесора К580ВМ80А.

Параметр	Значення параметра
Тактова частота	2 МГц
Розрядність даних, що опрацьовуються	8 біт
Максимальний об'єм пам'яті, що адресується МП	64 Кбіт

1.1.1. Умовне позначення та призначення виводів мікропроцесора

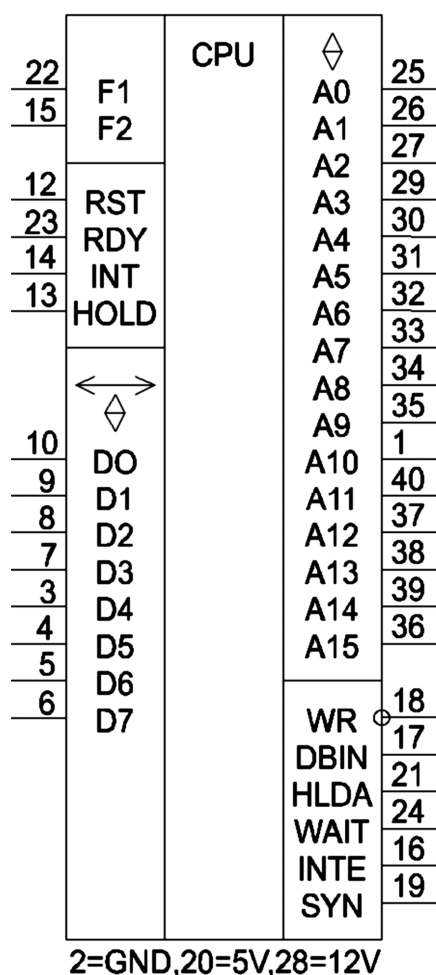


Рис. 1 Умовне позначення мікропроцесора

Призначення виводів мікропроцесора наведено в таблиці 2, умовне позначення наведено на рис. 1.

Таблиця 2 Призначення виводів мікропроцесора

Номер	Позначення	Назва	Назва сигналу	Тип
1	2	3	4	5
19	SYN	Синхронізація	Логічна 1 вказує на початок машинного циклу	Вихід
17	DBIN	Прийом даних	Логічна 1 вказує на зчитування інформації	Вихід
18	\overline{WR}	Запис	Логічний 0 вказує на записування інформації	Вихід
23	READY	Готовність	Логічна 1 вказує на готовність зовнішнього пристрою до обміну інформацією	Вхід
24	WAIT	Очікування	Логічна 1 вказує на те, що мікропроцесор знаходиться в стані очікування готовності до обміну зовнішнього пристрою	Вихід
14	INT	Запит на переривання	Сигнал запиту переривання. Логічна 1 вказує на наявність запиту	Вхід
16	INTE	Дозвіл переривання	Сигнал вказує стан дозволу переривань. Логічна 1 вказує, що переривання дозволені	Вихід
13	HOLD	Запит на захоплення	Сигнал запиту прямого доступу до пам'яті і портів вводу-виводу. Логічна 1 вказує на наявність запиту	Вхід
21	HLDA	Підтвердження захоплення	Логічна 1 вказує на дозвіл мікропроцесора зовнішньому пристрою використовувати прямий доступ до пам'яті і портів вводу-виводу	Вихід
12	RESET	Скид	Логічна 1 встановлює мікропроцесор у початковий стан	Вхід
3-10	D0-D7	Шина даних	Двонаправлена восьмирозрядна шина даних	Вхід
1,25-27, 29-40	A0-A15	Шина адреси	Однонаправлена тристабільна шістнадцятирозрядна адресна шина	Вихід
22	F1	Сигнал синхронізації	Перехід сигналу у високий рівень починає новий машинний такт МП. Рівень сигналу відрізняється від TTL	Вхід
15	F2	Сигнал синхронізації	Рівень сигналу відрізняється від TTL	Вхід

1.1.2. Структурна схема мікропроцесора КР580ВМ80А

Структурна схема мікропроцесора КР580ВМ80А зображена на рис. 2

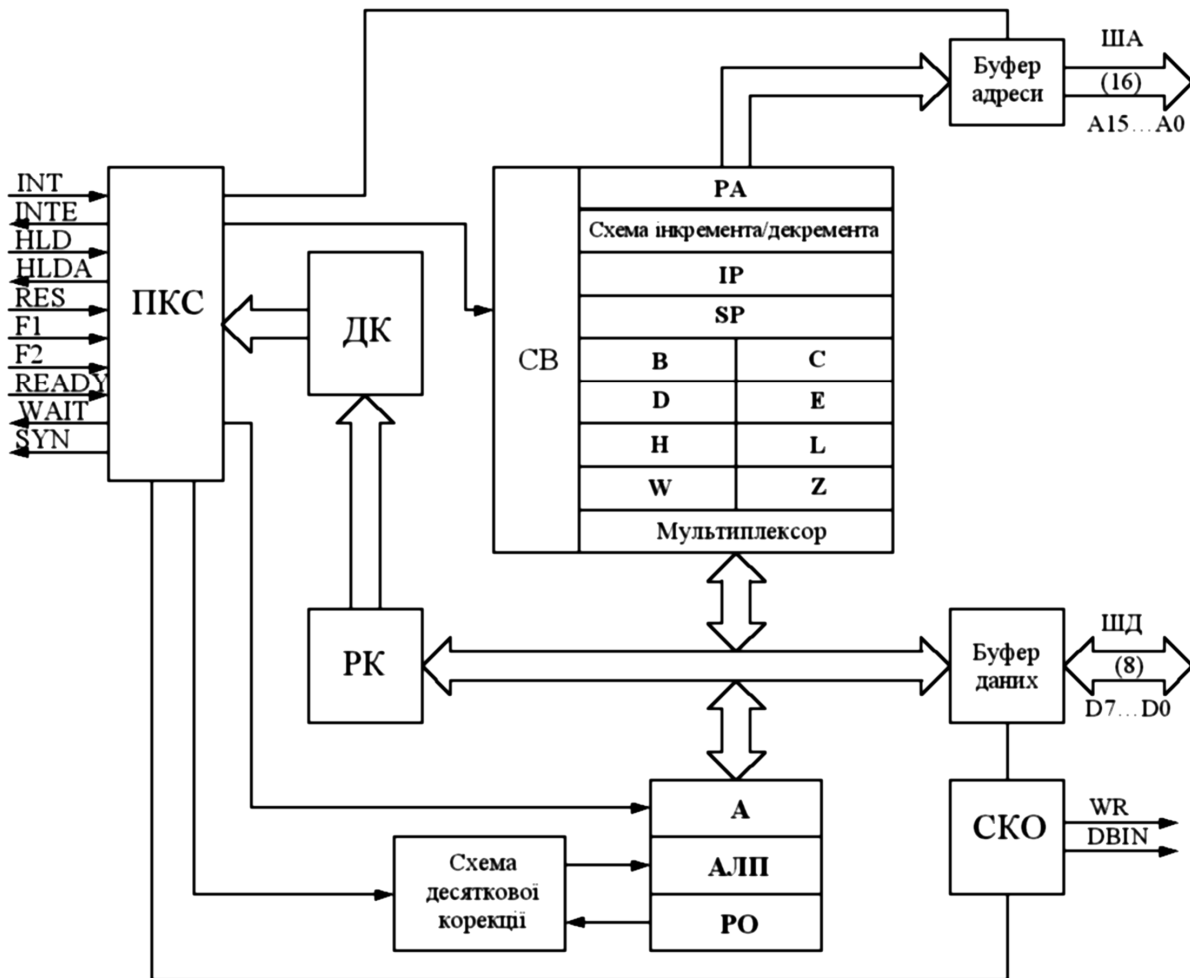


Рис. 2. Структурна схема мікропроцесора КР580ВМ80А

Призначення основних функціональних блоків мікропроцесора наведено в таблиці 3.

Таблиця 3. Призначення виводів МП.

Позначення	Назва	Призначення
1	2	3
АЛП	Арифметико-логічний пристрій	Виконує всі арифметичні (додавання, порівняння, інкремент, декремент), логічні операції (І, АБО, виключне АБО, інверсія) та операції зсуву. Результати більшості операцій заносяться в акумулятор
А	Акумулятор	8-розрядний регістр, куди поміщається більшість результатів арифметичних і логічних команд
РО	Регістр ознак	Зберігає ознаки властивостей результату останньої арифметичної або логічної операції

1	2	3
СДК	Схема десяткової корекції	Блок призначений для перетворення результату роботи в форму двійково-десятькового числа
РК	Регістр команд	8-розрядний регістр, що містить перший байт команди
ДК	Дешифратор команд	Декодує вміст регістра команд, визначає мікропрограму для виконання необхідної команди
ПКС	Пристрій керування та синхронізації	На основі сигналів дешифратора команд та сигналів регістра стана формує сигнали керування для всіх блоків МП, а також виробляє сигнали керування зовнішніми пристроями (ОЗП, ПЗП, ПВВ і т.п.). Крім цього ПКС містить 3 тригери: тригер зупинки ТрЗ, тригер дозволу переривань Тп, тригер дозволу захоплення шин МП
БА	Буфер адреси	Підсилювач потужності сигналів адресної шини
БД	Буфер даних	Буфер шини даних
СКО	Схема керування обміном	Призначена для керування обміном інформацією по шині
БР	Блок регістрів	Блок регістрів, що містить в собі наступні підблоки
ІР	Регістр адреси	16-розрядний регістр - лічильник, що містить 16-розрядну адресу виконуваної команди. Вміст лічильника автоматично збільшується на одиницю (адреса наступної команди) після завантаження кожного байта команди.
SP	Вказівник стеку	16-розрядний регістр, що містить адресу комірки, яка є вершиною стеку
СІД	Схема інкремента - декремента	Схема призначена для збільшення або зменшення на одиницю вмісту 8- або 16-розрядного регістра
B, C, D, E, H, L	Група регістрів загального призначення	Група із 6 восьмирозрядних регістрів (B, C, D, E, H, L) загального призначення, що можуть використовуватись у якості 16-бітних регістрів (BC, DE, HL). Регістри H і L можуть використовуватись також у якості адресного регістра для команд непрямого пересилання даних. Кожен з регістрів доступний для програміста
W, Z	Регістри тимчасового зберігання	Програмно недоступні регістри, що призначені для збереження другого і третього байтів команди.

1	2	3
СВ	Схема вибірки	Призначена для вибірки 16-бітного регістра, або пари 8-бітних регістрів
Мульти-плексор	Мульти-плексор	Призначений для вибірки одного з пари 8 бітних регістрів, з якими відбувається обмін

Розглянемо детальніше деякі блоки МП.

Регістр ознак містить ознаки:

- знаку S, встановлюється в "1", коли результат операції від'ємний;
- нуля Z, якщо результат операції дорівнює 0 (встановлюється в "1");
- парності P, встановлюється в "1", якщо результат містить парне число одиниць;
- перенесення CY, котрий дорівнює "1" за наявності переповнення розрядної сітки при додаванні, та за від'ємності результату при відніманні;
- додаткового перенесення AC ("1") за наявності перенесення з третього розряду в четвертий при додаванні, або займу з четвертого в третій розряд при відніманні.

Формат регістра ознак зображено на рис. 3.

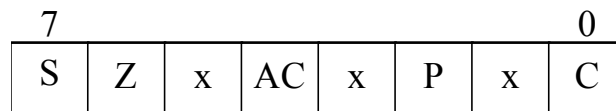


Рис. 3. Формат регістра ознак

1.1.3. Слово стана МП

На першому такті T1 кожного машинного циклу МП видає на шину даних так зване слово стана - код, що ідентифікує вид виконуваного циклу та інформує мікропроцесорну систему про режим роботи МП. Слово стана зберігається на шині даних тільки протягом одного такту, тому для його зберігання на весь час циклу обміну використовують додатково спеціальний регістр слова стана. Інформація в регістр слова стана записується в кінці першого або на початку другого машинного циклу. Індикатором наявності слова стана є сигнал SYN.

У таблиці 4 наведено слова стана для всіх десяти циклів МП KP580BM80.

Таблиця 4. Слово стана для циклів роботи МП КР580ВМ80

Цикл	Назва циклу	Значення розряду слова стана							
		D7	D6	D5	D4	D3	D2	D1	D0
M1	Вибірка команди	1	0	1	0	0	0	1	0
M2	Зчитування з пам'яті	1	0	0	0	0	0	1	0
M3	Записування в пам'ять	0	0	0	0	0	0	0	0
M4	Зчитування з стека	1	0	0	0	0	1	1	0
M5	Записування в стек	0	0	0	0	0	1	0	0
M6	Ввід з зовнішнього пристрою	0	1	0	0	0	0	1	0
M7	Вивід в зовнішній пристрій	0	0	0	1	0	0	0	0
M8	Переривання	0	0	1	0	0	0	1	1
M9	Зупинка	1	0	0	0	0	0	0	0
M10	Переривання в зупинці	0	0	1	0	1	0	1	1

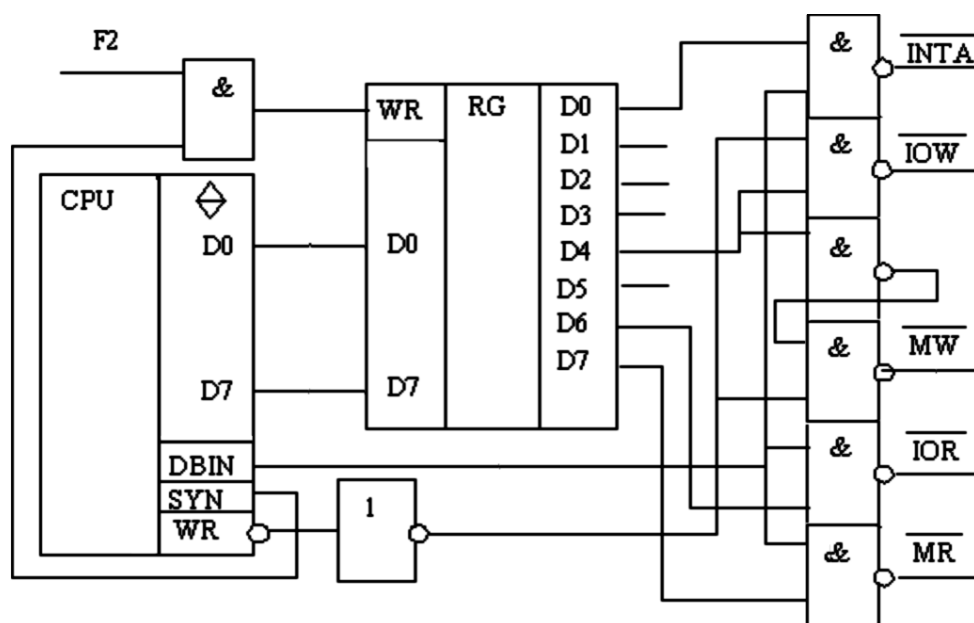


Рис. 4. Типова схема дешифрування слова стана

Типова схема дешифрування слова стана зображена на рис. 4. Позначення на рисунку:

\overline{INTA} - сигнал підтвердження переривань.

\overline{MW} - записування у пам'ять. При $\overline{MW} = 0$ дозволяється запис даних в ОЗП.

\overline{MR} - зчитування з пам'яті. При $\overline{MR} = 0$ дозволяється видача даних з ОЗП чи ПЗП.

\overline{IOR} - зчитування з портів вводу за командою IN. При $\overline{IOW} = 0$ здійснюється зчитування даних з порту.

\overline{IOW} - записування в порти виводу OUT. При $\overline{IOW} = 0$ дозволяється записування даних в порт.

1.1.4. Робота мікропроцесора в режимі записування інформації

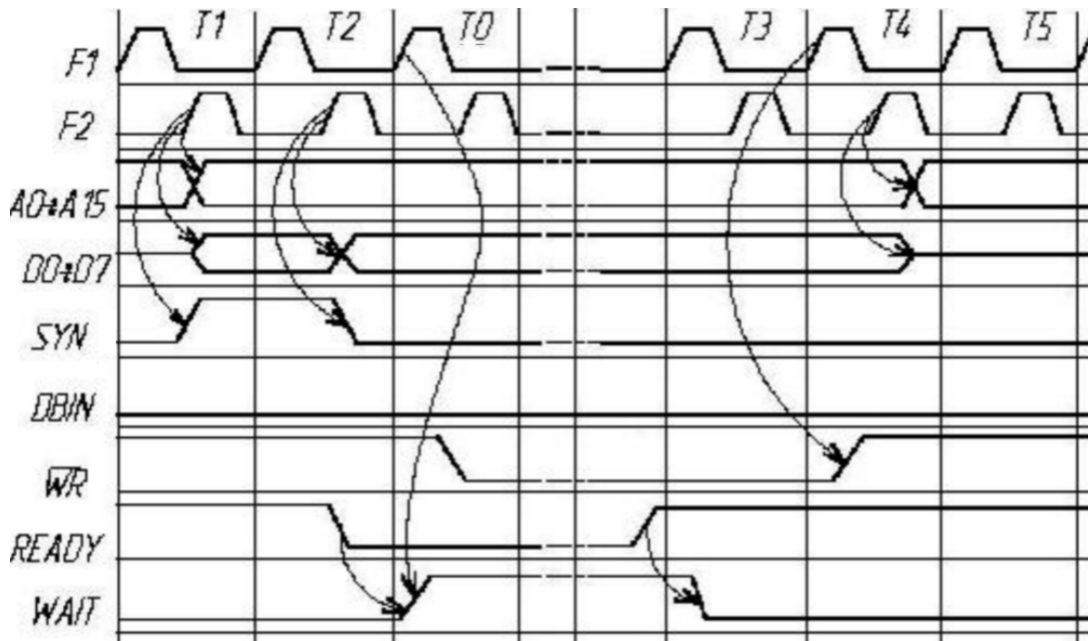


Рис. 5. Часова діаграма запису

Розглянемо часову діаграму роботи мікросхеми в режимі записування зображену на рис. 5. Процес запису інформації починається із такту T1. По фронту сигналу F2 в такті T1 на шині адреси з'являється адреса, за якою буде відбуватись запис інформації, а на шину даних виставляється код слова стана, що визначає тип обміну даними і вказує тип пристрою, куди буде відбуватись записування, одночасно лінія SYN встановлюється в стан логічної 1. По фронту сигналу F2 в такті T2 на шину даних видається код, що записується, а лінія SYN встановлюється в стан логічного 0. По спаду сигналу F2 відбувається аналіз стана лінії READY - лінії, по якій адресований пристрій вказує свою готовність до обміну. Якщо на лінії READY присутній стан логічного нуля (що означає неготовність адресованого пристрою), то наступним тактом буде такт очікування T0, в іншому випадку наступним тактом буде такт T3. По фронту сигналу F1 в такті T0 на лінії WAIT видається лог. 1, що вказує на очікування

мікропроцесором готовності адресованого пристрою, одночасно на лінію \overline{WR} видається лог. 0, котрий вказує на записування інформації. По спаду сигналу F2 процесор знову тестує стан лінії READY, якщо лінія знаходиться в стані лог. 1 (що означає готовність адресованого пристрою до обміну), процесор переходить до виконання такту T3.

У такті T3 по фронту сигналу F1 процесор переводить лінію \overline{WR} у стан лог. 0, якщо такт T0 був пропущений, у противному випадку лінія \overline{WR} лишається в стані лог. 0, а лінія WAIT повертається в стан лог. 0. У наступному за тактом T3 такті (T1 або T4) по фронту сигналу F1 сигнал \overline{WR} переводиться в стан лог. 1, а по фронту сигналу F2 з шини адрес та шини даних знімаються адреса і дані. На цьому обмін по шині в циклі записування завершено.

Якщо процесор при виконанні циклу потребує додаткові такти (T4, T5), вони ідуть після такту T3, проте в цих тактах новий обмін не здійснюється.

1.1.5. Робота мікропроцесора в режимі читання

Часова діаграма роботи мікропроцесора в режимі читання зображена на рис. 6. Процес читання інформації починається із такту T1. По фронту сигналу F2 в такті T1 на шині адреси з'являється адреса, за якою буде відбуватись читання інформації, а на шину даних виставляється код слова стана що визначає тип передавання даних і вказує тип пристрою, звідки буде відбуватись зчитування, одночасно лінія SYN встановлюється в стан логічної 1. По фронту сигналу F2 в такті T2 шина даних переводиться в z - стан, а лінія SYN встановлюється в стан логічного 0, одночасно сигнал DBIN встановлюється в стан лог. 1, що вказує на приймання даних шиною МП. По спаду сигналу F2 відбувається аналіз стана лінії READY - лінії, по якій адресований пристрій вказує свою готовність до обміну, якщо на лінії READY присутній стан логічного нуля (це означає неготовність адресованого пристрою), то наступним тактом буде такт очікування T0, у зворотньому випадку наступним

тактом буде такт T3.

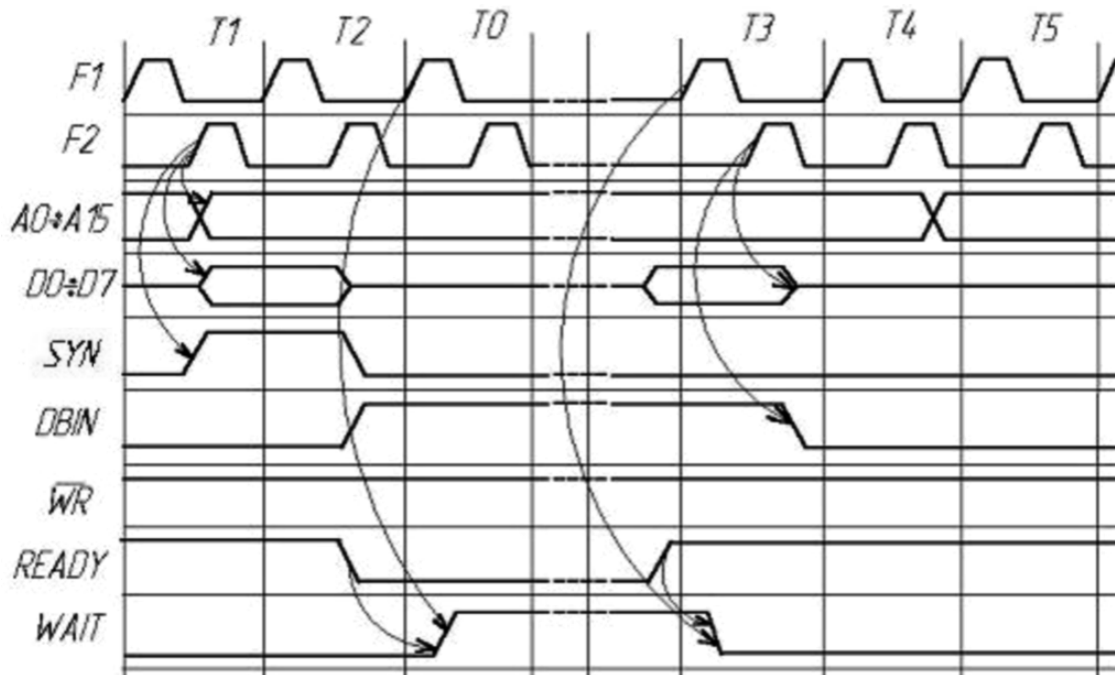


Рис. 6. Часова діаграма читання

По фронту сигналу F1 в такті T0 на лінії WAIT видається лог. 1, що вказує на очікування готовності адресованого пристрою. По спаду сигналу F2 процесор знову тестує стан лінії READY, якщо лінія знаходиться в стані лог. 1 (що означає готовність адресованого пристрою до обміну), то процесор переходить до виконання такту T3. Якщо такт T3 іде за тактом T0, то лінія WAIT по фронту F1 повертається в стан лог. 0. По фронту сигналу F2 дані фіксуються у внутрішньому регістрі, а на лінію DBIN видається лог. 0. У наступному за тактом T3 такті (T1 або T4) по фронту сигналу F2 з шини адрес знімається адреса. На цьому обмін по шині в циклі читання завершено.

1.1.6. Переривання в МП

У мікропроцесорі K580BM80A сигнал, по якому зовнішній пристрій запитує переривання, носить назву INT. Нехай в деякий момент часу на вхід INT надійшла логічна одиниця. В момент закінчення виконання команди мікропроцесор, в останньому машинному циклі в останньому такті по спаду сигналу F2 аналізується стан лінії INT і стан внутрішнього триггеру TrI. Якщо переривання дозволені TrI (і відповідно вивід INTE в стані лог. 1), то наступний машинний цикл буде циклом підтвердження переривань. У

загальному цикл підтвердження переривань близький до циклу зчитування. Проте є деякі відмінності. В першому машинному циклі лінія по фронту F2 тригер TrI та лінія INTE переходить у стан лог. 0, на шини даних видається слово стана, що вказує на цикл підтвердження переривання, а на адресну шину код адреси наступної виконуваної команди, проте ця адреса в циклі підтвердження переривання участі не бере й ігнорується пристроєм, що запросив переривання. Зовнішня відносно МП схема фіксує слово стана та на його основі формує сигнал підтвердження переривань \overline{INTA} , що передається зовнішньому пристрою, котрий запросив переривання, або контролеру переривань. Інші сигнали МП формує аналогічно циклу зчитування (коду команди). По сигналу \overline{INTA} зовнішній пристрій видає на шини даних МП код, що сприймається МП як команда, яку необхідно виконати, і МП в тактах T3, T4, T5 починає її виконувати. Якщо в процесі виконання команди необхідно зчитати другий (або другий та третій) байти команди, цикл зчитування повторюється: знову видається слово стана, що вказує на підтвердження переривання, знову воно фіксується і дешифрується зовнішньою схемою, знову видається сигнал \overline{INTA} і т.д.

Код команди, що передається в МП, може бути будь-який, проте найчастіше такими кодами є коди команд рестарту RST0 - RST7 або код команди виклику підпрограм CALL. За ними викликаються підпрограми опрацювання переривань. При цьому в стеку зберігається адреса команди основної команди, яка має бути виконана після закінчення підпрограми обслуговування.

Підпрограми опрацювання переривань не мають змінювати вміст регістрів МП, що використовуються в основній програмі, тому за необхідності використати регістр у підпрограмі старе значення має бути збережене, наприклад, в стеку, а при закінченні роботи підпрограми воно має бути відновлене. Підпрограми опрацювання переривань мають закінчуватись кодом:

EI ;дозвіл наступних переривань

RET ; повернення до основної програми.

Команда EI дозволяє переривання, причому тригер дозволу переривань TrI перемикається в 1 лише після виконання наступної команди. Ця команда необхідна, тому що тригер TrI не встановлюється автоматично при поверненні із підпрограми опрацювання переривань. Команда RET вибирає зі стеку адресу повернення і повертає керування основній програмі.

Код процедури опрацювання переривання

```
INT_HDL:  
PUSH PSW ; зберегти значення регістрів A та F  
IN 20h ; зчитати дані із пристрою  
STA BUFF_D ; зберегти їх в буфері  
MVI A, 01 ; встановити прапорець буфера в 1  
STA BUFF_F ;  
POP PSW ; повернути збережені  
EI ; дозволити переривання  
RET ; повернення із підпрограми.
```

1.1.7. Прямий доступ до пам'яті

При використанні швидкодіючих зовнішніх пристроїв виникає необхідність збереження даних в оперативній пам'яті або зчитування їх із пам'яті зі швидкістю, котра перевищує швидкість роботи МП. Тоді використовують прямий доступ до пам'яті - ПДП (DMA), при якому процесор відмикається від шин, а керування бере на себе зовнішній пристрій, що запросив ПДП, або контролер прямого доступу до пам'яті.

Зовнішній пристрій може запросити прямий доступ до пам'яті за дорогою сигналу HOLD. Нехай в деякий момент часу сигнал HOLD стає активним (набуває стан лог.1). МП завершує обмін по шині і в такті T3 (T4, T5 або TS) по спаду сигналу F2 аналізує стан лінії HOLD. Якщо на лінії HOLD лог. 1 у наступному такті по фронту сигналу F1 на лінію HLDA видається лог. 1, а по фронту сигналу F2 МП переводить шини даних, адрес та керування в z-стан. Далі процесор виконує внутрішні операції, пов'язані з опрацюванням команди,

і по спаду сигналу F2 в кожному машинному такті аналізує стан лінії HOLD. Зовнішній пристрій може утримувати лінію HOLD в активному стані, протягом часу, який йому потрібен для завершення обміну. Якщо лінія HOLD переходить у стан лог. 0, у наступному машинному такті по фронту F1 знімається сигнал HLDA, а по фронту F2 шини даних, адрес та керування переводяться в робочий режим.

1.1.8. Програмування та система команд мікропроцесора

1.1.8.1. Мікропроцесорна система на основі МП K580BM80A з точки зору програмування

Структура мікропроцесорної системи на основі K580BM80A з точки зору програмування зображена на рис. 7. Мікропроцесор K580BM80A має такі програмно доступні блоки:

A - акумулятор;

F(PO) - реєстр ознак;

B, C, D, E, H, L - реєстри загального призначення;

SP - вказівник стеку;

IP - лічильник команд.

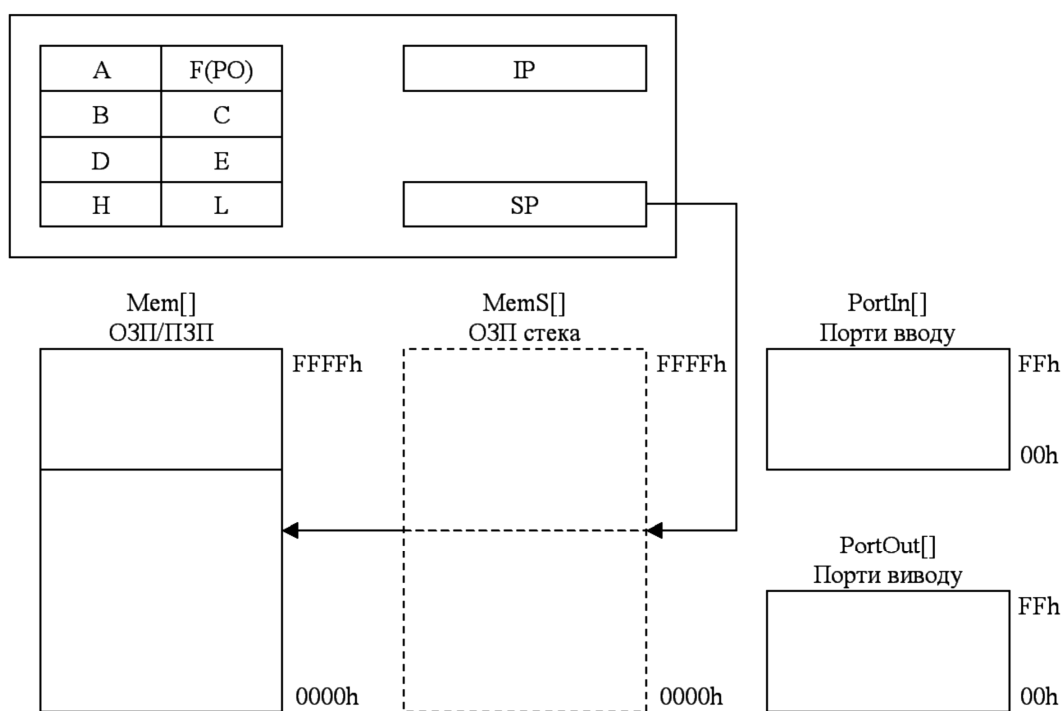


Рис. 7. Структура мікропроцесорної системи на основі K580BM80A з точки

зору програмування

Мікропроцесор у своїй роботі звертається до пам'яті та портів вводу-виводу. Множина всіх адрес об'єктів певного типу, до яких може звернутись мікропроцесор, носить назву адресного простору.

Процесор K580BM80A має три адресних простори:

1. Mem[] - адресний простір ЗП команд/даних, звертання до якого відбувається при виконанні всіх команд, крім команд роботи зі стеком та вводу-виводу. Діапазон адрес 0000-FFFFh.

2. MemS[] - адресний простір ОЗП стеку, звертання до якого відбувається при виконанні операцій додавання в стек та витягування зі стеку. Діапазон адрес 0000-FFFFh. Найчастіше цей адресний простір накладається на адресний простір, діапазон накладається на адресний простір ЗП команд/даних.

3. PortIn[]/PortOut[] - адресний простір портів вводу-виводу. Він використовується для роботи з портами вводу-виводу за допомогою команд IN/OUT. Діапазон адрес 00 - FFh. При використанні команд IN/OUT адреса дублюється на виходах A0 -A7 та A8 -A15.

1.1.8.2. Стек

Стек -це область пам'яті, до якої можна звертатися тільки через комірку, що носить назву вершини стеку. Для стеку визначені такі операції:

POP - вибрати елемент зі стеку;

PUSH - додати елемент у стек.

У МП K580BM80A окрема область пам'яті виділена для розміщення даних стеку, а в мікропроцесорі є регістр, який вказує на вершину стеку котрою є комірка пам'яті з найбільшою адресою не зайнята даними. При додаванні даних у стек дані додаються у його вершину, а вказівник стеку зменшується на довжину даних. Інші дані не переміщуються. В процесі витягання даних дії відбуваються в зворотному порядку.

1.1.8.3. Класифікація команд МП

Команди МП зазвичай складаються із 2 частин: коду операції, що

присутній завжди, і операндів, над якими виконується дія.

За довжиною коду команди поділяють на:

- однобайтні (код команди складається із 1 байта);
- двобайтні (код команди складається із 2 байт);
- трибайтні (код команди складається із 3 байт).

За типом адресації операндів розрізняють команди з:

- з абсолютною адресацією, в яких адреса операнду міститься в другому та третьому байтах команди;

- з реєстровою прямою адресацією, в яких операнд міститься в реєстрі МП, що прямо вказується в команді;

- з реєстровою непрямою адресацією, в яких адреса операнду міститься в реєстрах МП, що вказується в команді;

- з безпосередньою адресацією, в котрих операнд міститься безпосередньо в команді.

За типом команди поділяються на:

1) команди пересилання, котрі виконують пересилання даних, вони з деякими обмеженнями вони еквівалентні командам присвоювання мов високого рівня;

2) команди арифметичних операцій, до яких відносять команди: додавання, віднімання, збільшення, зменшення на 1;

3) команди логічних операцій: побітного логічного І, побітного АБО, побітного виключаючого АБО та побітної інверсії;

4) команди порівняння;

5) команди роботи з бітами, а саме : зсуву вправо та вліво та роботи з ознакою переносу;

6) команди переходу, які поділяються на команди: безумовного прямого та непрямого переходу, та команди умовних переходів;

7) команди виклику підпрограм, котрі поділяють на команди безумовного та умовного виклику підпрограм;

8) команди повернення з підпрограм, що теж поділяють на команди

безумовного та умовного повернення;

9) команди роботи зі стеком, до котрих відносять команди: поміщення та витягування зі стеку, обміну з вершиною та встановлення вершини стеку;

10) команди вводу-виводу;

11) команди керування роботою МП;

12) команда "немає операції".

1.2. Мікропроцесорний комплект КР580

Мікросхеми, що входять до комплекту К580, можна умовно поділити на 4 групи:

1. Універсальні мікропроцесорні ВІС:

К580ВМ80А - базовий мікропроцесор;

К580ВМ1 - базовий мікропроцесор із розширеною системою команд та архітектурою.

2. Загального призначення (системоутворюючі), до них відносять:

К580ГФ24 - тактовий генератор;

К580ВК28 - системний контролер;

К580ВК38 - системний контролер;

К580ІР82 - 8-розрядний регістр із z-станом на виході без інверсії;

К580ІР82/83 - 8-розрядний регістр із z-станом на виході з інверсією сигналів;

К580ВА86 - 8-розрядний двонаправлений шинний формувач без інверсії;

К580ВА87 - 8-розрядний двонаправлений шинний формувач з інверсією сигналів;

К580ВГ18 - контролер шини І-41.

3. Універсальні ВІС. До них відносять:

К580ВВ51 - універсальний синхронно-асинхронний прийомопередавач;

К580ВІ53 - триканальний 16-розрядний лічильник - таймер;

К580ВВ55 - програмований паралельний інтерфейс;

К580ВТ57 - контролер прямого доступу до пам'яті;

К580ВН59 - контролер переривань.

Згадані вище ВІС є універсальними, тобто можуть виконувати відразу

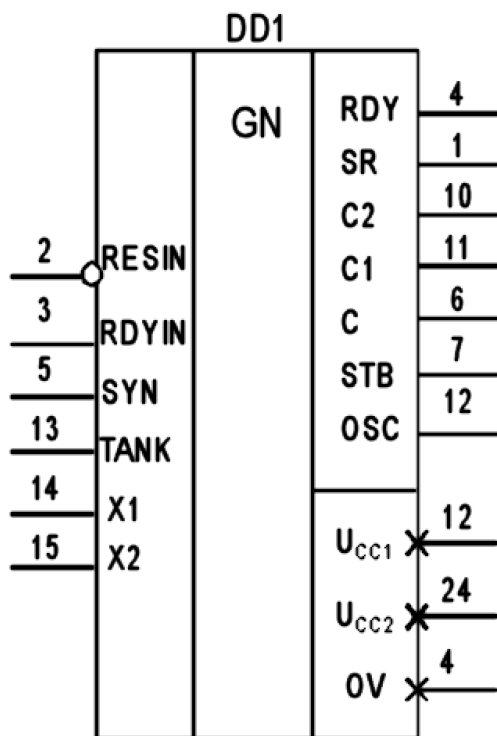
кілька функцій, обслуговувати різне периферійне обладнання, а режим роботи таких мікросхем програмується.

4. Контролери пристроїв та інтерфейсів:

- K580BГ75 - контролер дисплея на основі електронно-променевої трубки;
- K580BB79 - контролер клавіатури та матричного дисплея;
- K580BK91 - контролер інтерфейсу КОП (IEEE-488);
- K580BГ93 - шинний формувач для інтерфейсу КОП (IEEE-488).

Мікросхеми цієї групи призначені для виконання лише однієї операції, проте володіють вищою швидкодією. Розглянемо мікросхеми, які входять до комплекту.

1.2.1. Тактовий генератор КР580ГФ24



**Рис. 8. Тактовий генератор
КР580ГФ24**

Генератор формує:

- ◇ дві послідовності імпульсів (виводи C1, C2), зсунуті в часі, амплітудою 12 В та частотою 0,5...3,0 МГц;
- ◇ тактові сигнали опорної частоти амплітудою TTL;
- ◇ 5 В (вивід OSC);
- ◇ стробуючий сигнал стану STB з періодом $T_{оп}/9$, де $T_{оп}$ – період тактових сигналів опорної частоти (частоти сигналів на лінії OSC);
- ◇ тактові імпульси (вивід C), які синхронізовані з фазою сигналів C2 та амплітудою TTL.

Призначення виводів мікросхеми наведено в таблиці 5.

Таблиця 5. Призначення виводів мікросхеми.

Позначення	Тип	Призначення
1	2	3

SR	Вихід	Скидає в початковий стан МП-систему. Використовується для скидання МП та інших пристроїв, що входять до МП-системи
RESIN	Вхід	Вхід скидання в нуль генератора
RDYIN	Вхід	Сигнал готовності

Продовження таблиці 5

1	2	3
RDY	Вихід	Вихідний сигнал готовності
SYN	Вхід	Сигнал синхронізації
C	Вихід	Тактовий сигнал, синхронізований з фазою сигналу C2
STB	Вихід	Стробуючий сигнал стана. Формується за наявності на вході SYN високого рівня для занесення слова стана в системний контролер КР580ВК28
C1,C2	Виходи	Тактові вихідні сигнали
OSC	Вихід	Тактовий сигнал опорної частоти. Використовується для одночасної синхронізації кількох генераторів
TANK	Вхід	Вивід для під'єднання коливального контура. Використовується для під'єднання коливального контуру, який працює на вищих гармоніках кварцового резонатора з метою стабілізації тактових сигналів опорної частоти.
X1, X2	Входи	Виводи для під'єднання кварцового резонатора
Ucc1	Вхід	Напруга живлення +5 В
Ucc2	Вхід	Напруга живлення +12 В
GND	Вхід	Спільний вивід

1.2.2. Системний контролер KP580BK28

Системний контролер KP580BK28 використовується разом із МП KP580BM80A для формування керуючих сигналів обміну в МП-системі RD, WR, RDIO, WRIO, INTA та як буферний регістр даних, через який відбувається обмін даними між МП та іншими пристроями.

Контролер формує керуючі сигнали залежно від слова стана, яке на початку кожного циклу надходить від МП по каналу даних D0...D7.

Призначення виводів мікросхеми наведено в таблиці 6.

Таблиця 6. Призначення виводів мікросхеми

Позначення	Номер виводу	Тип	Призначення
1	2	3	4
STB	1	Вхід	Стробуючий сигнал стана
HLDA	2	Вхід	Сигнал підтвердження захоплення шини
WR	3	Вхід	Видача інформації. Лінія WR мікропроцесора з'єднується з лінією WR контролера

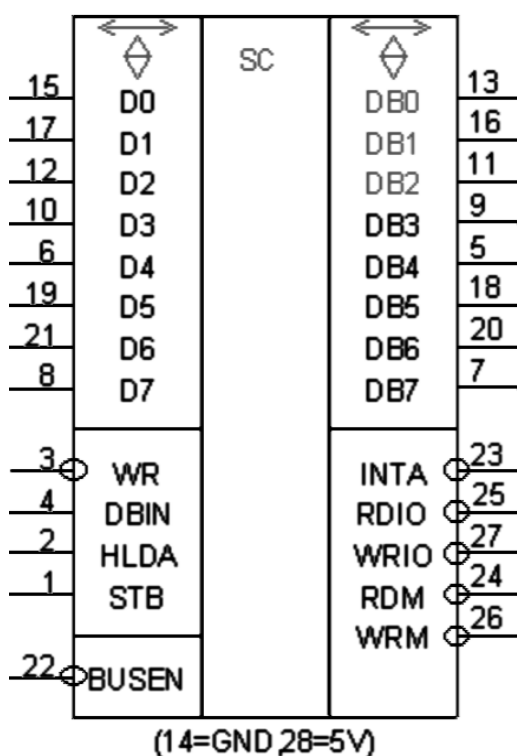


Рис. 9. Системний контролер
KP580BK28

DBIN	4	Вхід	Прийом інформації. Лінія DBIN мікропроцесора з'єднується з лінією DBIN контролера
D0...D7	15, 17, 12, 10, 6, 19,21, 8	Входи- виходи	Двонаправлений канал даних мікропроцесора. Виводи мають обмежену навантажувальну здатність, і призначені для передавання слова стана та даних від мікропроцесора, а також видачі даних мікропроцесору.
DB0...DB7	13, 16, 11, 9, 5, 18, 20, 7	Входи- виходи	Двонаправлений канал даних системи
BUSEN	22	Вхід	Вхід керування передаванням даних і формуванням керуючих сигналів. При BUSEN=0 контролер передає або приймає дані та формує один із керуючих сигналів; при BUSEN=1 лінії контролера переводяться в Z-стан. Зауважимо, що BUSEN - асинхронний сигнал
INTA	23	Вихід	Підтвердження запиту переривання. При роботі з МП КР580ВМ80 у циклі переривання мікропроцесора контролер формує сигнал INTA для приймання байтів команди CALL від контролера переривань КР580ВН59. У простих мікропроцесорних схемах, коли контролер переривань відсутній, якщо вихід INTA під'єднати до напруги 12 В через резистор номіналом 1кОм, під час дії сигналу буферна схема даних контролера формує єдиний вектор переривань за номером 7 і передає його в МП (код команди RST7)
RDM	24	Вихід	Зчитування з пам'яті. Логічний нуль на виході вказує, що мікропроцесор здійснює читання з пам'яті
RDIO	25	Вихід	Зчитування з ПБВ. Логічний нуль на виході вказує, що мікропроцесор здійснює читання з пристрою вводу-виводу
WRM	26	Вихід	Записування в пам'ять. Логічний нуль на виході вказує, що мікропроцесор здійснює записування у пам'ять

Закінчення таблиці 6

1	2	3	4
WRIO	27	Вихід	Записування в ПБВ. Логічний нуль на виході вказує, що мікропроцесор здійснює записування у пристрій вводу-виводу.
Ucc	28	Вхід	Напруга +5 В. Вхід на умовному позначенні не показаний.

GND	14	Вхід	Напруга 0 В. Вхід на умовному позначенні не показаний.
-----	----	------	--

1.2.3. Регістр КР580ІР82 (ІР83)

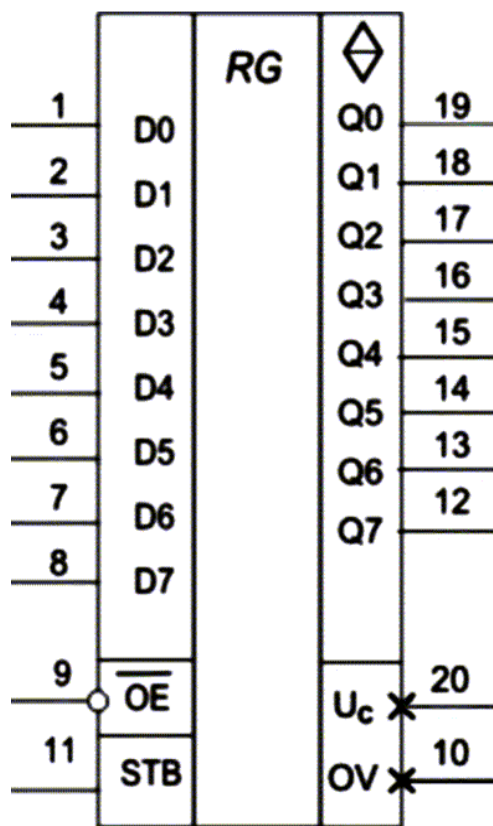


Рис. 10. Регістр КР580ІР82 (ІР83)

Восьмирозрядний регістр, який використовується для організації адресної шини МП-системи, збільшуючи одночасно навантажувальну здатність адресних ліній системи. Має три стани вихідних ліній.

Призначення виводів мікросхеми наведено в таблиці 7, умовне позначення мікросхеми зображено на рис. 10.

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

$OE=1$ лінії $Q0...Q7$ регістра переводяться в Z-стан.

Таблиця 7. Призначення виводів мікросхеми

Позначення	Номер виводу	Тип	Призначення
D0...D7	1-8	Вхід	Входи даних
Q0...Q7	19-12	Вихід	Виходи даних
OE	9	Вхід	Вхід дозволу передавання даних
STB	11	Вхід	Вхід стробуючого сигналу
Uс, 0V	20, 10	-	Входи живлення

Мікросхема КР580ІР83 має в порівнянні з КР580ІР82 інверсні виходи $Q0...Q7$ (нумерація виводів обох мікросхем однакова).

1.2.4. Шинний формувач КР580ВА86 (ВА87)

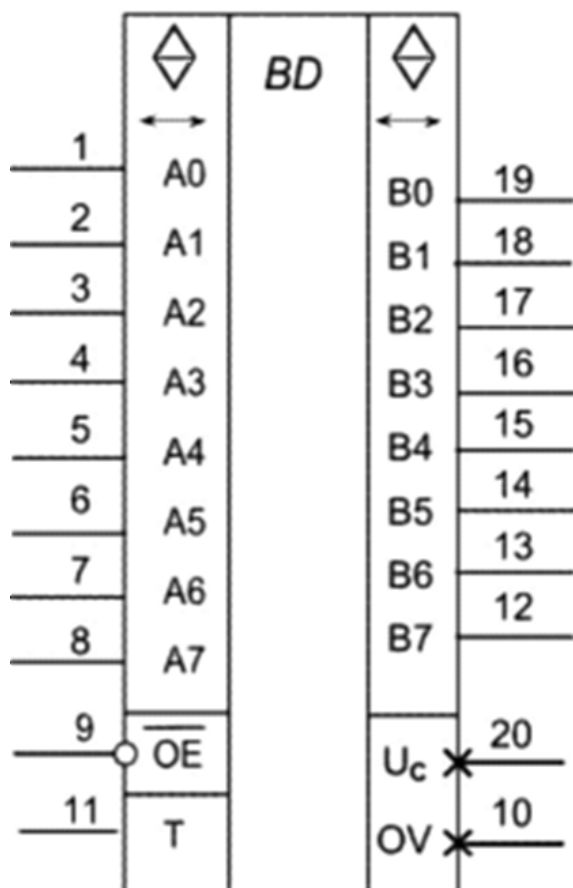


Рис. 11. Шинний формувач (приймодіадавач) КР580ВА86 (ВА87)

Двонаправлений восьмирозрядний шинний формувач, який використовується для обміну даними між МП та системною шиною, збільшуючи одночасно навантажувальну здатність ліній даних системи. Має три стани вихідних ліній.

Призначення виводів мікросхеми наведено в таблиці 8, умовне позначення зображено на рис. 11.

Сигнал Т змінює напрям передачі даних. Якщо Т рівний лог. 0 забезпечується передача даних з ліній В0...В7 до ліній А0...А7, у протилежному

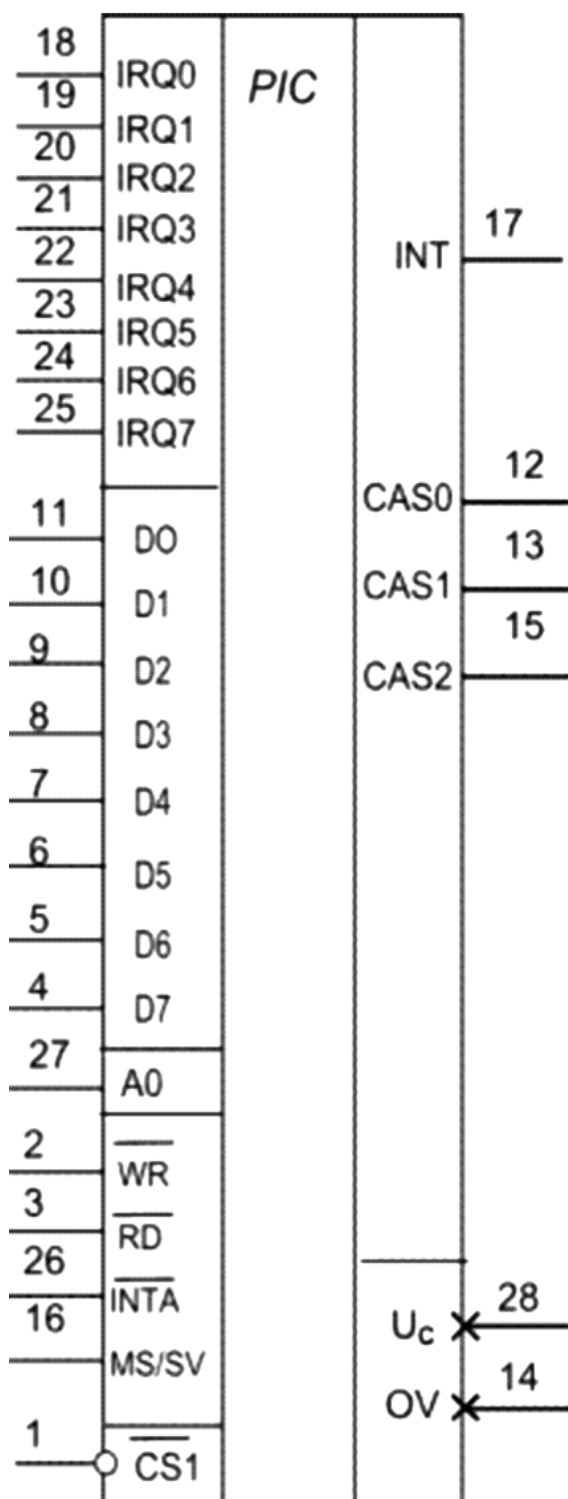
випадку - з ліній А0...А7 до ліній В0...В7. І в першому, і в другому випадку сигнал ОЕ повинен бути в стані лог. 0, при ОЕ=1 лінії шин А та В переводяться в z-стан.

Таблиця 8. Призначення виводів мікросхеми.

Позначення	Номер виводу	Тип	Призначення
А0...А7	1-8	Вхід	Входи-виходи даних А
В0...В7	19-12	Вихід	Входи-виходи даних В
ОЕ	9	Вхід	Вхід дозволу передавання даних
Т	11	Вхід	Вхід напрямку передавання даних
U _с , GND	20, 10	-	Входи живлення

Мікросхема КР580ВА87 має в порівнянні із КР580ВА86 інверсні виходи В0...В7 (нумерація виводів обох мікросхем однакова).

1.2.5. Контролер переривань KP580BH59A



**Рис. 12. Контролер переривань
KP580BH59A**

Контролер використовується для забезпечення процедури переривання в мікропроцесорних системах. Він виконує такі функції:

- ◇ обслуговує до восьми запитів на переривання мікропроцесора;
- ◇ визначає пріоритет запитів;
- ◇ має можливість нарощування кількості ліній запитів переривання за рахунок каскадного з'єднання контролерів переривання загальною кількістю до 64.

Призначення виводів мікросхеми наведено в таблиці 9, умовне позначення зображено на рис. 12.

Після подачі на один із входів IRQ0...IRQ7 лог. 1 мікросхема формує сигнал INT, а також код команди CALL (вектор переривання), що відповідає активному входу IRQ:

- ◇ якщо контролер у системі один, то формуються всі три байти команди CALL;
- ◇ якщо контролерів кілька, то перший байт команди (КОП – код операції) формується

керуючим контролером, а другий та третій байти тією веденою мікросхемою, номер якої виданий на шину CAS2 - CAS0.

◇ Для зчитування команди МП або системний контролер формує три імпульси INTA. За другим імпульсом у МП посилається молодший байт адреси переривання (другий байт команди), за третім – старший байт адреси (третій байт команди). При каскадному з'єднанні контролерів за першим імпульсом INTA формується КОП команди і одночасно на лініях CAS0...CAS2 – номер веденої мікросхеми.

Таблиця 9. Призначення виводів мікросхеми.

Позначення	Номер виводу	Тип виводу	Призначення
CS	1	Вхід	Вибір мікросхеми. Логічний нуль на вході вказує, що з нею відбувається обмін. У процесі запиту й підтвердження переривання сигнал ігнорується
WR	2	Вхід	Запис у мікросхему. Логічний нуль на вході вказує, що відбувається записування даних в мікросхему. Сигнал сприймається лише при нульовому рівні на лінії CS
RD	3	Вхід	Зчитування з мікросхеми. Логічний нуль на вході вказує, що відбувається читання регістрів мікросхеми. Сигнал сприймається лише при нульовому рівні на лінії CS
D0...D7	11-4	Вхід/ вихід	Двонаправлений канал даних. Дані, які зчитуються з мікросхеми та записуються в неї, проходять по цих лініях
CAS0.. CAS2	12, 13, 15	Вхід/ вихід	Лінії каскадування. По лініях передається код номера веденої мікросхеми в процесі підтвердження переривання у режимі каскадного ввімкнення мікросхем
MS/SV	16	Вхід	Вибір ведучої (лог. 1) або веденої мікросхеми. Вивід задає режим роботи мікросхеми при використанні каскадного ввімкнення мікросхем. За наявності єдиного контролера він має працювати в режимі ведучого
INT	17	Вихід	Запит переривання. Логічна одиниця, що видається на вихід, вказує на наявність запиту переривання, яке має обслужитися мікропроцесором
IRQ0...IRQ7	18-25	Вхід	Запит переривання. Лінії призначені для під'єднання зовнішніх пристроїв. Логічна одиниця на лінії вказує на наявність запиту переривань від зовнішнього пристрою
A0	27	Вхід	Адресний вхід. Стан лінії визначає внутрішні регістри мікросхеми, до яких відбувається звертання

Таблиця 9. Призначення виводів мікросхеми.

Позначення	Номер виводу	Тип виводу	Призначення
U _c	28	-	Напруга живлення
0V	14	-	Спільний вивід

Режим роботи контролера встановлюється програмним шляхом: здійснюється подача двох або трьох команд, в першій команді задаються значення розрядів A5...A7 молодшого байта адреси та задається режим роботи мікросхеми, в другій команді задаються значення старшого байта адреси.

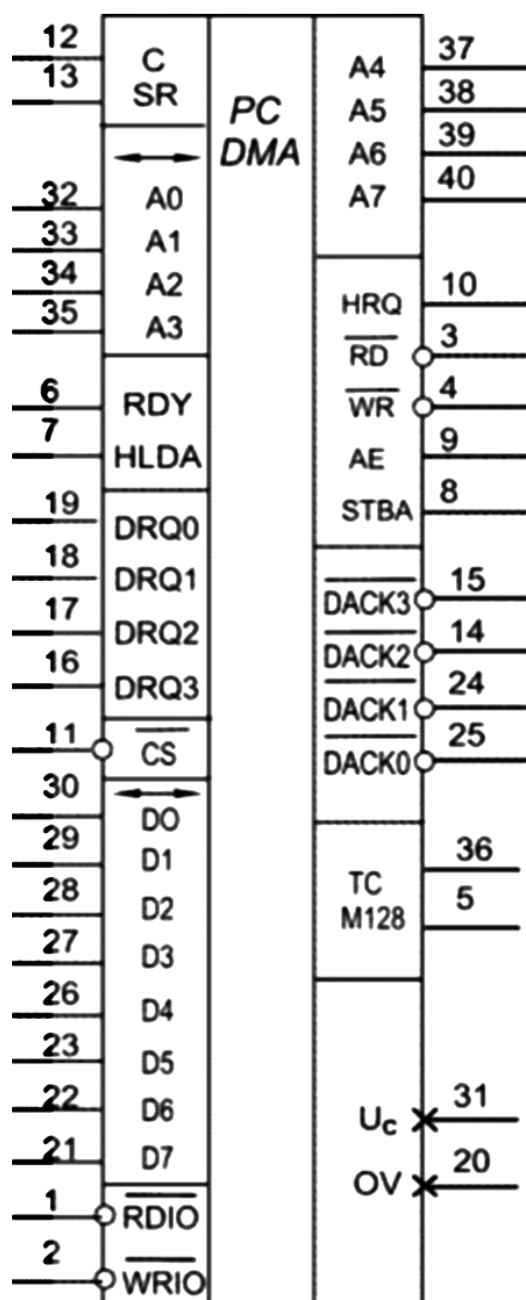


Рис. 13. Контролер прямого доступу до пам'яті (ПДП) KP580BT57

Найвищий пріоритет за замовчуванням має вхід IRQ0, найнижчий – IRQ7. Пріоритет входів IRQ може змінюватись програмно.

1.2.6. Контролер прямого доступу до пам'яті KP580BT57

Для обслуговування швидкодіючих зовнішніх пристроїв та збільшення пропускну здатності каналу зовнішні пристрої – мікропроцесор – пам'ять, використовується контролер прямого доступу до пам'яті (ПДП). Він призначений для:

- ◇ двонаправленого обміну даними між пам'яттю системи та периферійними пристроями;

- ◇ адресації пам'яті через формування масиву адрес комірок пам'яті, включаючи початкову адресу (першу адресу початку обміну) та число циклів звертань до пам'яті;
- ◇ формування керуючих сигналів обміну.

Кожен із чотирьох незалежних каналів контролера забезпечує адресацію зовнішньої пам'яті масивами до 16 Кбайт шляхом інкрементування вибраної адреси із можливістю визначення будь-якої з 64 Кбайт початкових адрес.

Умовне позначення мікросхеми зображено на рис. 13. Призначення виводів мікросхеми наведено в таблиці 10.

Таблиця 10. Призначення виводів мікросхеми

Позначення	Номер виводу	Тип виводу	Призначення
1	2	3	4
CS	11	Вхід	Вибір мікросхеми. Логічний нуль на вході вказує, що з нею відбувається обмін. У процесі запиту й підтвердження прямого доступу до пам'яті сигнал ігнорується
WR	4	Вихід	Запис інформації в пам'ять. Логічний нуль на лінії вказує, що відбувається запис у пам'ять. Мікросхема керує лінією лише в режимі прямого доступу до пам'яті, в інших режимах роботи вивід переведений у високоімпедансний стан і не використовується
RD	3	Вихід	Зчитування інформації з пам'яті. Логічний нуль на лінії вказує, що відбувається читання пам'яті. Мікросхема керує лінією лише в режимі прямого доступу до пам'яті, в інших режимах роботи вивід переведений у високоімпедансний стан і не використовується
D0...D7	30 - 26, 23 - 21	Вхід/ вихід	Шина даних. У режимі прямого доступу до пам'яті лінії використовуються для видачі старшого байта адреси у випадку його зміни, або переведені в високоімпедансний стан. У режимі програмування та зчитування інформації лінії служать входом або виходом даних, що записуються або зчитуються з мікросхеми

Таблиця 10. Призначення виводів мікросхеми

DRQ0 - DRQ3	19 - 16	Вхід	Сигнали запиту прямого доступу до пам'яті від зовнішніх пристроїв. Логічна 1 на лініях вказує на наявність такого запиту. Тип прямого доступу до пам'яті, що має відбутись, визначається при програмуванні мікросхеми. Сигнал може бути заблокований за допомогою програмування мікросхеми
DACK0 - DACK3	25, 24, 14, 15	Вихід	Сигнал підтвердження прямого доступу до пам'яті. Логічний нуль на відповідному виході вказує, що відбувається прямий доступ до пам'яті за відповідним запитом

Продовження таблиці 10

1	2	3	4
M128	5	Вихід	Модуль 128. Є активним, коли поточний цикл обміну є 128, або кратним 128 від кінця масиву даних
HLDA	7	Вихід	Підтвердження запиту прямого доступу до пам'яті. Логічна 1 на лінії вказує, що мікропроцесор звільнив шину і контролер прямого доступу до пам'яті може здійснювати цикли ПДП. З'єднується з виходом HLDA мікропроцесора
A0...A7	32 - 35, 37 - 40	Вхід/ вихід	Канал адреси. В процесі ПДП лінії A0...A7 служать виходами, через які виводиться адреса комірки пам'яті, з якою відбувається обмін. В інших режимах роботи лінії A0...A3 служать входами, що вказують регістр мікросхеми, з яким відбувається обмін, а лінії A4...A7 не використовуються
TC	36	Вихід	Кінець циклу. Сигнал вказує, що цикли прямого доступу по каналу завершені, а поточний цикл ПДП має бути останнім для цього масиву даних і вибраний канал буде автоматично заборонений
SR	13	Вхід	Сигнал початкового встановлення. Логічна 1 на лінії призводить до скиду мікросхеми
HRQ	10	Вихід	Запит прямого доступу. Логічна 1 на лінії вказує на запит прямого доступу до пам'яті від контролера до мікропроцесора. З'єднується з входом HOLD мікропроцесора

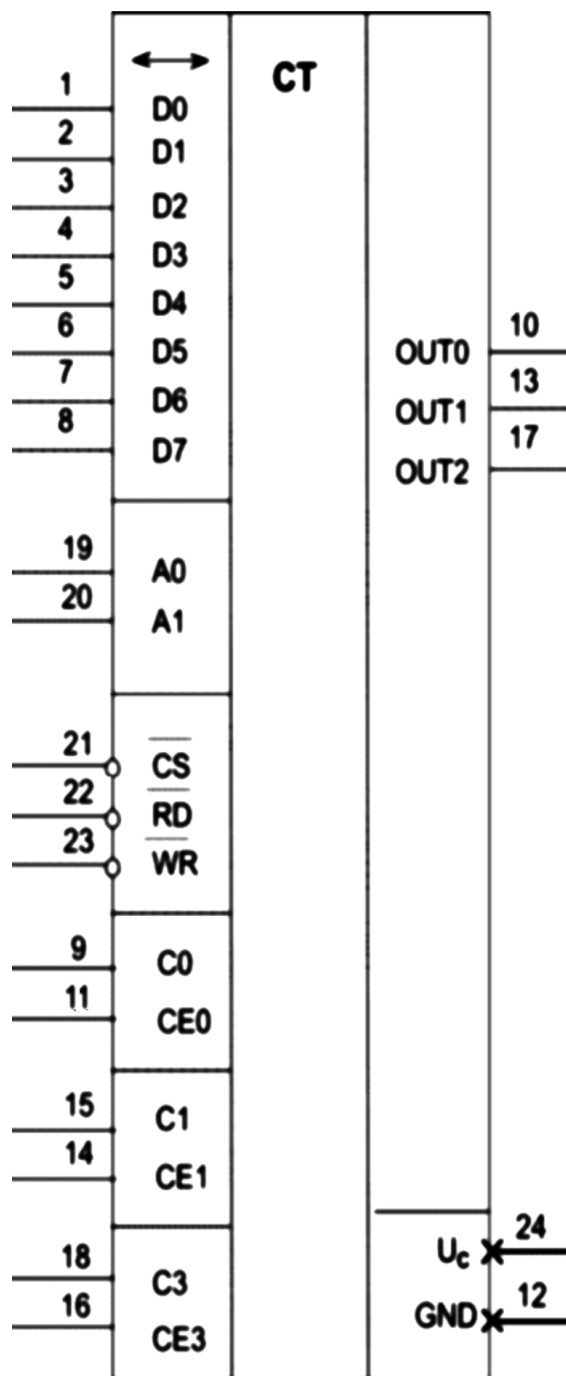
STBA	8	Вихід	Стробуючий сигнал адреси. Сигнал використовується лише в циклі ПДП для вказання, що на шині даних присутній старший байт адреси. Перехід сигналу з логічної одиниці в логічний нуль вказує, що адреса дійсна і може бути записана в зовнішній реєстр. В інших режимах роботи сигнал не використовується
C	12	Вхід	Сигнал синхронізації. Перехід сигналу з логічної одиниці в логічний нуль (задній фронт) вказує на початок нового циклу синхронізації. На лінію подаються тактові імпульси C2 від тактового генератора КР580ГФ24
AE	9	Вихід	Дозвіл адреси. Логічна 1 вказує, що мікросхема працює в режимі ПДП, а на шині адреси адреса сформована самою мікросхемою. Сигнал використовується для блокування інших пристроїв, які не беруть участі в циклі ПДП

Закінчення таблиці 10

1	2	3	4
RDY	6	Вхід	Готовність. Логічний нуль вказує на необхідність продовжити цикл обміну в процесі ПДП. В інших режимах роботи сигнал не використовується
RDIO	1	Вхід/ вихід	Зчитування пристрою вводу-виводу (ПВВ). У процесі ПДП цей сигнал служить виходом і на нього видається лог. 0 при здійсненні циклу записування в пам'ять (зчитування з ПВВ). При цьому сигнал зчитування випереджає сигнал записування в пам'ять. В інших режимах роботи сигнал служить входом і використовується при читанні реєстрів мікросхеми (активний рівень сигналу - нульовий)
WRIO	2	Вхід/ вихід	Запис у пристрій вводу-виводу (ПВВ). У процесі ПДП цей сигнал служить виходом і на нього видається лог. 0 при здійсненні циклу зчитування з пам'яті (запису в ПВВ). При цьому сигнал зчитування пам'яті випереджає сигнал запису. В інших режимах роботи сигнал служить входом і використовується при програмуванні реєстрів мікросхеми (активний рівень сигналу - нульовий)
Ucc,	31	-	Лінія живлення (5В)
GND	20	-	Загальний

Робота контролера в режимі ПДП відбувається в такій послідовності:

1. Активізується один із входів DRQ (вхід DRQ0 має найвищий пріоритет).



**Рис. 14. Програмований таймер-лічильник
KP580BI53**

2. Контролер формує сигнал HRQ, що надходить у мікропроцесор.

3. Після отримання сигналу HLDA контролер:

- ◇ формує сигнал DACK з найвищим пріоритетом;

- ◇ видає вісім молодших розрядів адреси пам'яті на лінії A0...A7, а вісім старших – на лінії D0...D7;

- ◇ генерує сигнали керування RD, WR, RDIO чи WRIO.

4. За один цикл обміну в режимі ПДП контролер здійснює обмін одного байта даних, починаючи з початкової адреси в першому циклі та інкрементуючи її в кожному з наступних циклів.

1.2.7. Програмований таймер-лічильник KP580BI53

Триканальний таймер-лічильник KP580BI53 призначений для організації роботи мікропроцесорної системи в режимі реального часу. Він містить три

незалежні канали, кожен з яких має 16-розрядний лічильник з максимальним значенням відліку:

- ◇ у двійковому коді - 2^{16} ;
- ◇ у десятковому коді – 10^4 ;

Частота синхронізації кожного лічильника 0...2,5 МГц, число режимів роботи кожного каналу 6. Призначення виводів наведено в таблиці 11, умовне позначення мікросхеми зображено на рис. 14.

Кожен із трьох каналів програмується записом у регістр керуючого слова РКС, після чого вже за іншою адресою програмується лічильник цього каналу записом одного чи двох байтів числа відліку.

Внаслідок того, що відсутній вхід встановлення, початкова ініціалізація лічильника здійснюється, лише програмно, при записуванні керуючого слова в регістр режиму обраного каналу.

Лічильник має шість режимів роботи (0...5), з яких найчастіше використовуються режими 2 (подільника частоти) та 3 (генератора меандра). Режим роботи встановлюється розрядами D1...D3, а номер каналу – розрядами D6, D7 керуючого слова.

Таблиця 11. Призначення виводів мікросхеми

Позначення	Номер виводу	Тип виводу	Призначення
D0...D7	1-8	Входи/виходи	Канал даних. По лініях передаються дані про стан мікросхеми та лічильників, а також здійснюється її програмування
C0,C1,C2	9, 15, 18	Входи	Тактові сигнали синхронізації каналів 0...2. Сигнал надходить на тактовий вхід відповідного лічильника
OUT0-OUT2	10, 13, 17	Виходи	Виходи каналів 0...2. Лінія вказує стан відповідного каналу лічильника методом, вказаним при виборі режиму роботи
CE0...CE2	11, 14, 15	Входи	Входи дозволу каналів 0...2. Входи дозволяють відлік відповідного каналу лічильника. Метод вказання дозволу роботи каналу визначається запрограмованим режимом
A0, A1	19, 20	Входи	Адресні лінії. Лінії вказують номер лічильника або допоміжних регістрів (якщо A1A0 = 11), що програмуються або зчитуються в циклі обміну

Таблиця 11. Призначення виводів мікросхеми

Позначення	Номер виводу	Тип виводу	Призначення
CS	21	Вхід	Сигнал вибору мікросхеми. Логічний 0 вказує, що з мікросхемою відбувається обмін по шині даних. Стан лінії не впливає на роботу лічильників
RD	22	Вхід	Зчитування. Логічний 0 при лог. 0 на лінії CS вказує, що відбувається читання мікросхеми
WR	23	Вхід	Запис. Логічний 0 при лог. 0 на лінії CS вказує, що відбувається запис у мікросхему

1.3. Організація мікропроцесорної системи керування на основі мікропроцесорного комплекту КР580

Система керування на основі МП КР580ВМ80А зображена на рис. 15 містить:

- ◇ DD2 МП КР580ВМ80А, робота якого синхронізується імпульсами тактового генератора DD1 КР580ГФ24;
- ◇ системного контролера DD 3 КР580ВК28, який формує системну шину даних ШД (DAT0...DAT7) та шину керування ШК (IACK, IORC, IOWC, MRDC, MWTC та HACK);
- ◇ регістрів DD4, DD5 КР580ИР82, на яких побудована адресна шина ША (ADR0...ADR15).

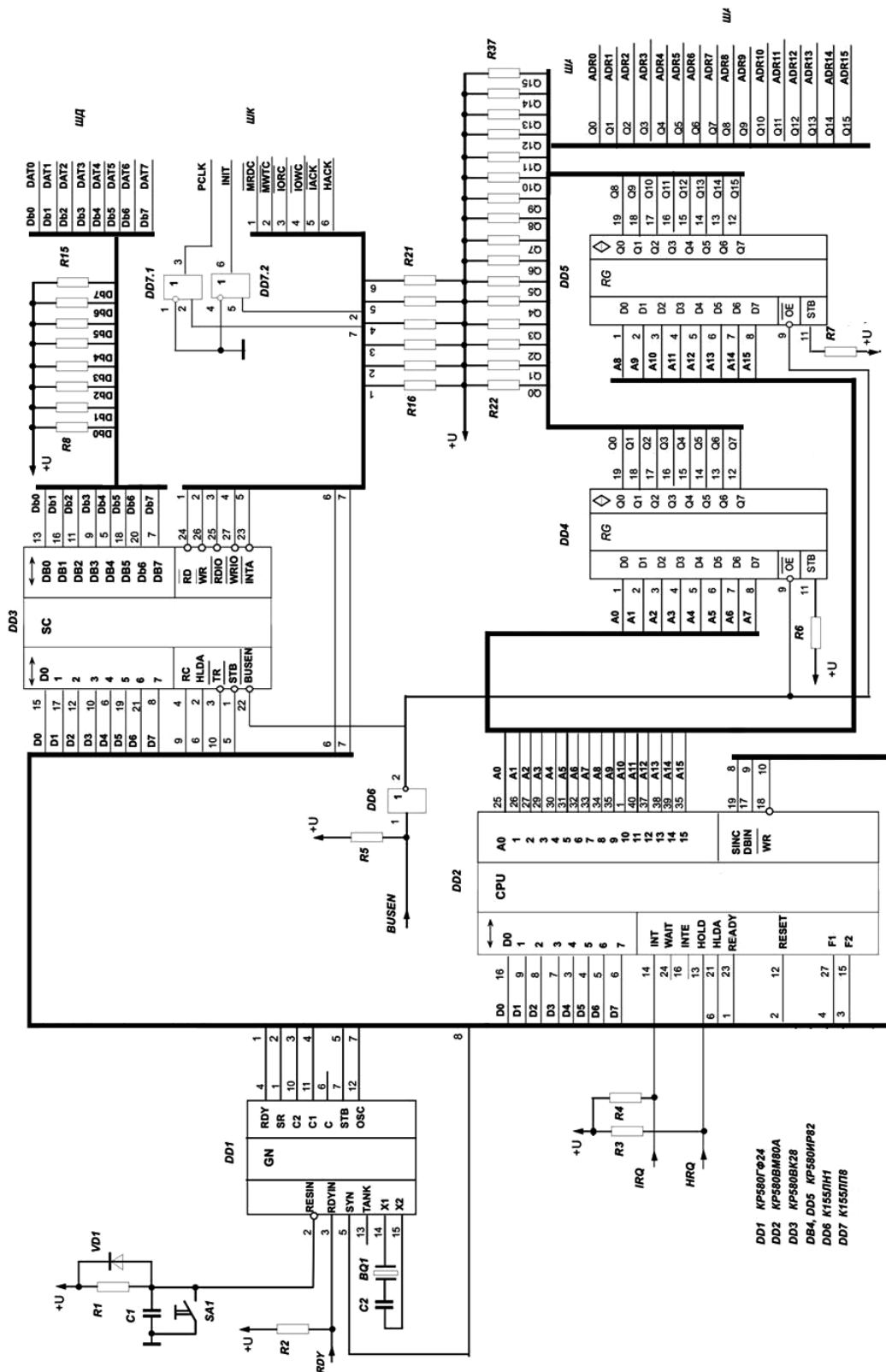


Рис. 15. Процесорний блок

До системної шини, в свою чергу, під'єднують інші пристрої мікропроцесорної системи – постійну та оперативну пам'ять, пристрої вводу-виводу (послідовні та паралельні), контролери переривання та ПДП.

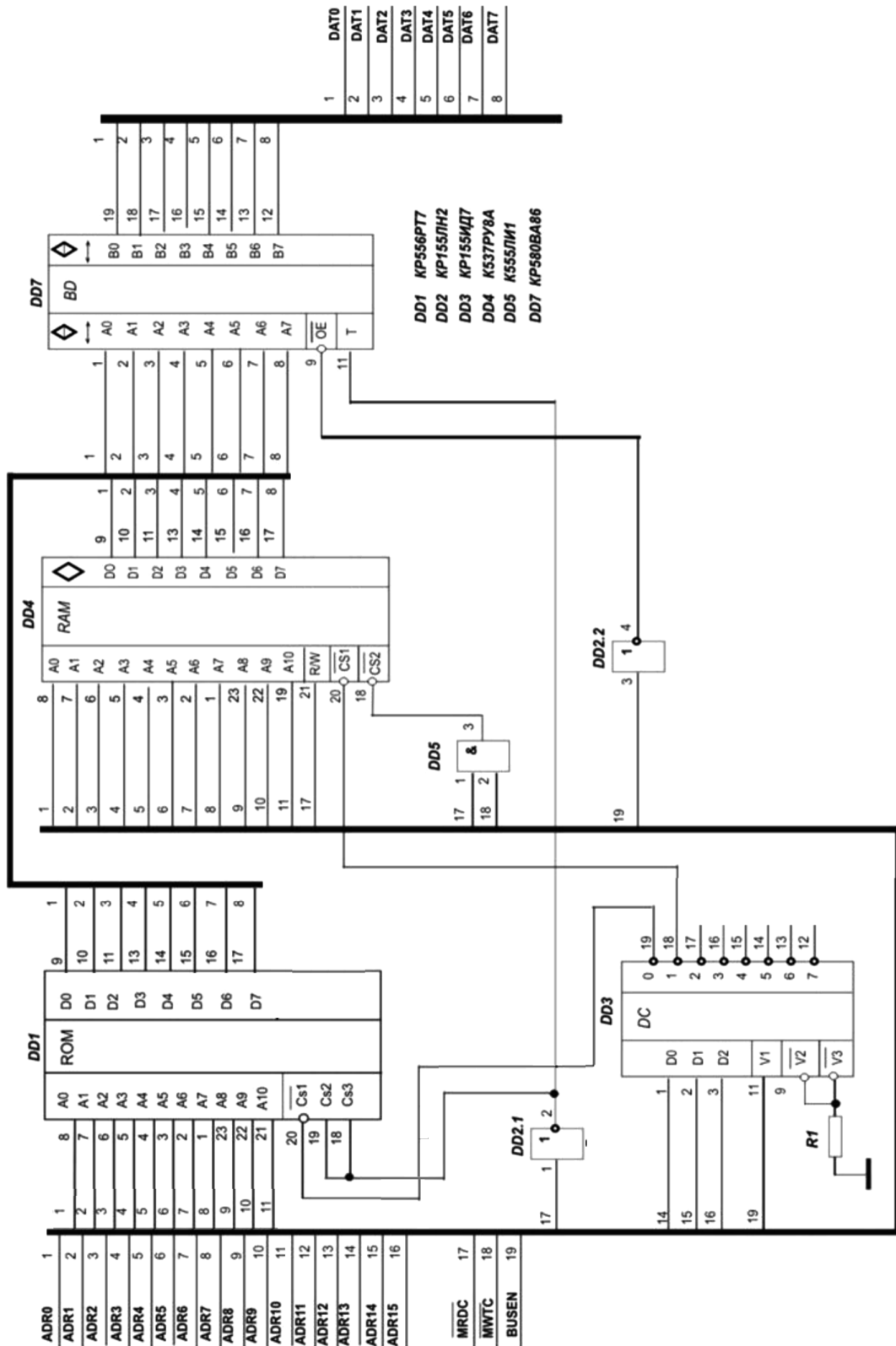


Рис. 16. Блок пам'яті

Для побудови складних мікропроцесорних систем необхідні ще й засоби арбітражу та керування доступом до шини, для чого використовують арбітр шини КР580ВГ18.

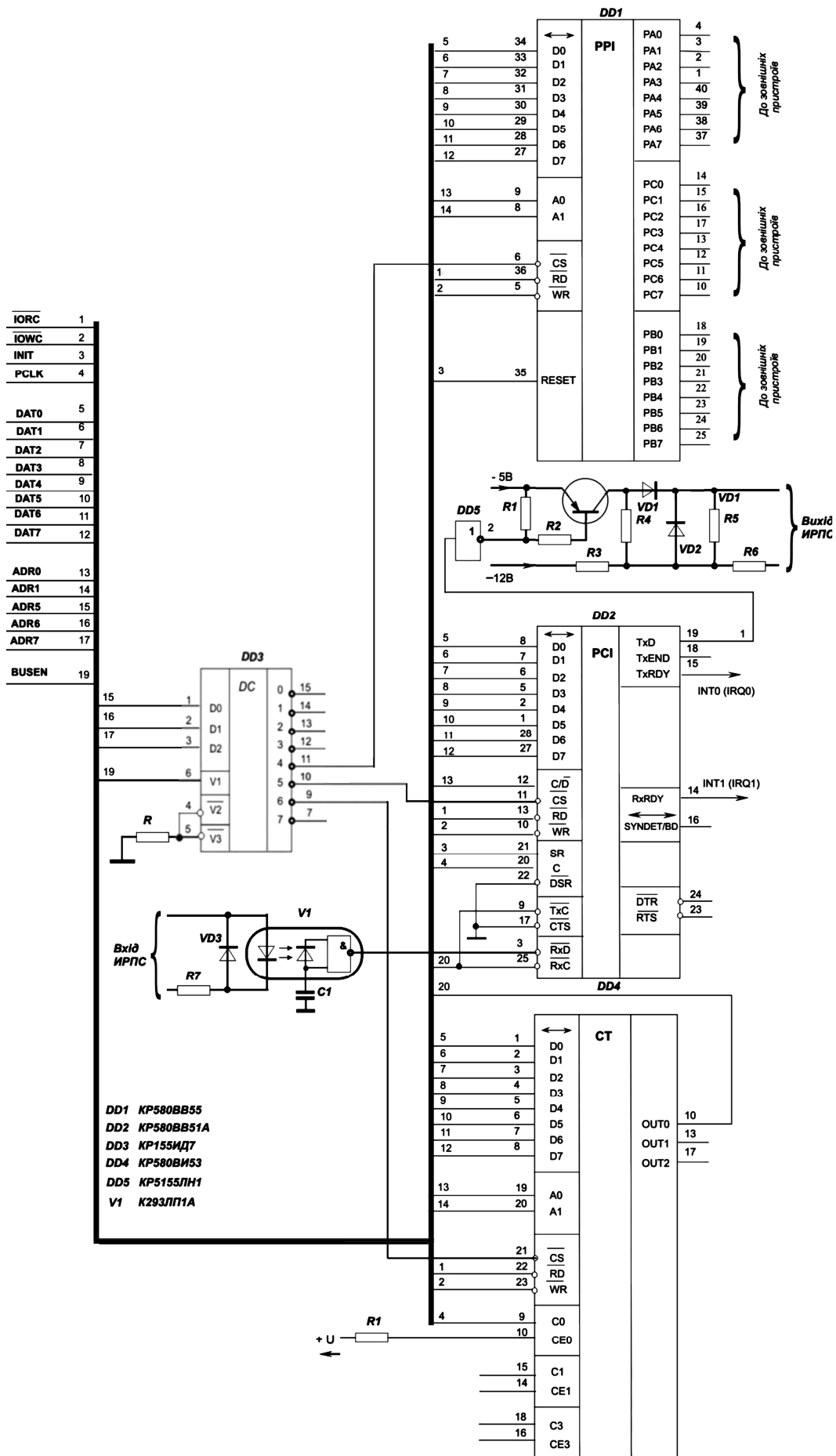


Рис. 17. Блок вводу-виводу

Для поділу адресного простору пам'яті (ЗП) та пристроїв вводу-виводу (ПВВ) мікропроцесора можна використовувати системний дешифратор адреси.

У схемах, що містять контролер ПДП, використовують окремі дешифратори для мікросхем пам'яті та 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 зображений на рис. 16. Вибір оперативної або постійної пам'яті забезпечує дешифратор адреси DD3. Обмін даними між пам'яттю та системною шиною даних забезпечує шинний формувач DD7.

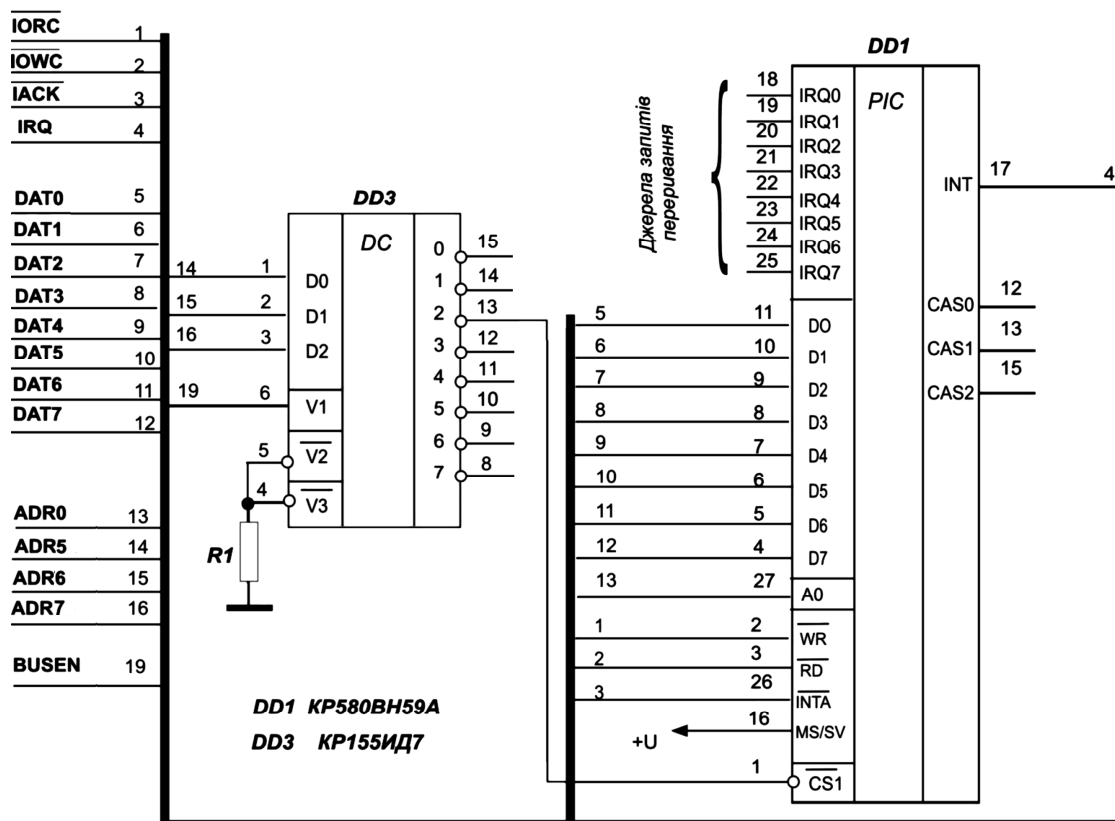


Рис. 18. Контролер переривань

Пристрої вводу-виводу – паралельний та послідовний програмований інтерфейси (КР580В55 та КР580ВВ51) під'єднуються до системної шини аналогічно пам'яті. Їх активізацію по лініях CS здійснює адресний дешифратор (рис. 17). На схемі також показано під'єднання програмованого таймера-лічильника КР580ВІ53 (DD4), а також елементів послідовної лінії ІРПС (приймача та передавача).

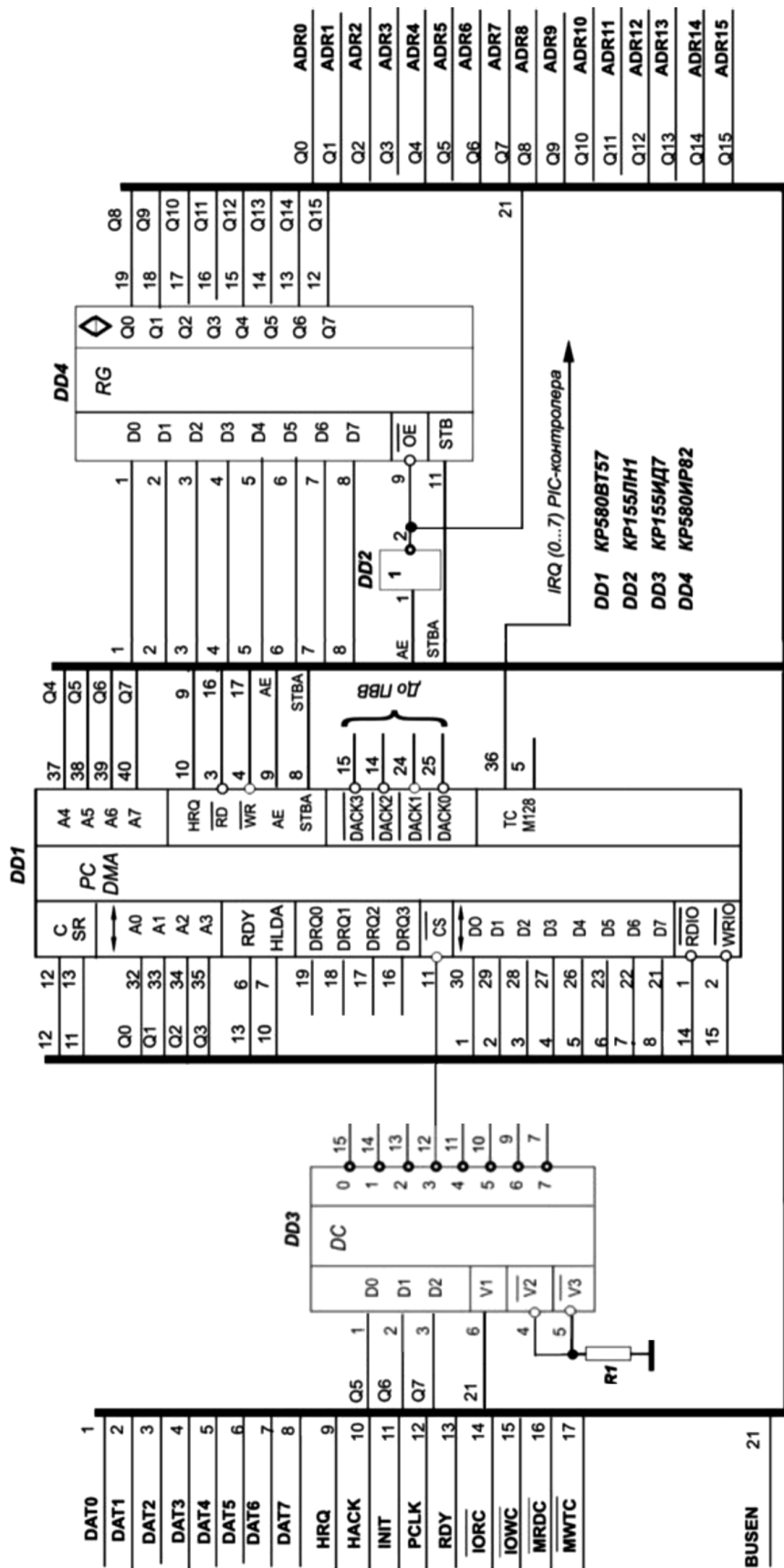


Рис. 19. Контролер прямого доступу до пам'яті

Схеми під'єднань до системної шини контролерів переривання КР580ВН59 та ПДП КР580ВТ57 зображені відповідно на рис. 18 та 19.

1.3.1. Організація клавіатури мікропроцесорної системи

На рис. 20 та 21 зображені можливі варіанти реалізації клавіатурного пульта вводу інформації на основі матриці ключів 5x5.

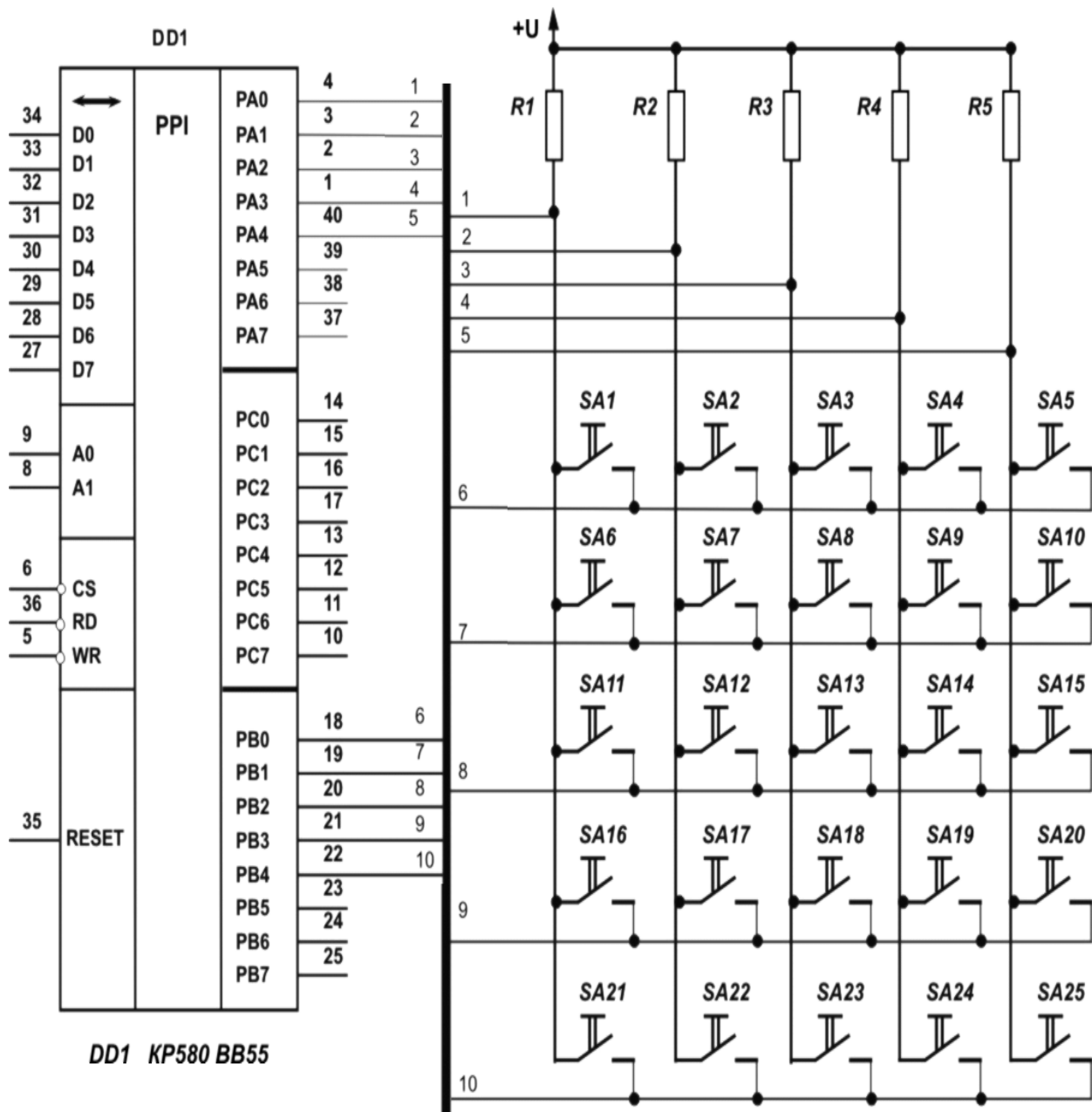


Рис. 20. Клавіатура на основі K580BB55

На рис. 20 ключі під'єднані безпосередньо до портів А та В паралельного програмованого інтерфейса KP580BB55 (DD1), а на рис. 21 – через двонаправлені прийомопередавачі DD4 та DD5 та елементи логіки, які формують сигнали опитування клавіатури.

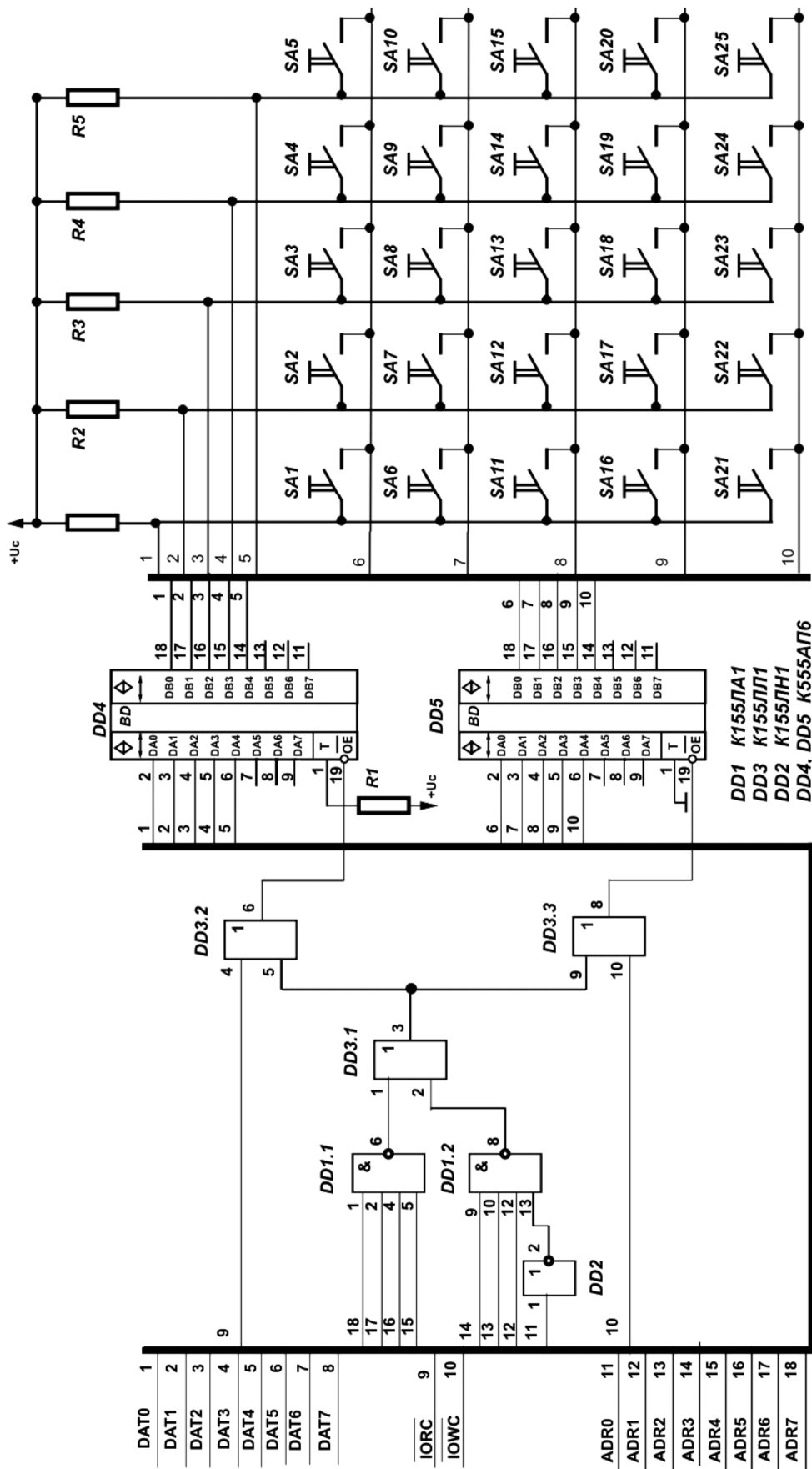


Рис. 21. Клавіатура

Ключі клавішних матриць обох рисунків розташовані так:

- ◇ один вивід кожного нормально розімкненого ключа з'єднаний з однією із вертикальних ліній матриці, 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”;
- ◇ коли на одну із горизонтальних ліній подається “0”, стан вертикальних ліній перевіряється мікропроцесором. Якщо при цьому відбувається замикання ключа з тих, що під'єднані до цієї горизонтальної лінії, то відповідна йому вертикальна лінія також отримує рівень “0”;
- ◇ за допомогою відповідного програмного забезпечення мікропроцесорна система встановлює по чергово рівень “0” на горизонтальних лініях матриці, перевіряючи при цьому стан її вертикальних ліній. Номер горизонтальної та номер вертикальної лінії матриці, які мають рівень “0”, і будуть визначати координати (номер) замкненого ключа.

У схемі на рис. 20 порт А, запрограмований на ввід, здійснює опитування вертикальних ліній матриці, а порт В, запрограмований на вивід, встановлює по чергово на горизонтальних лініях матриці рівень “0”. Для адресації портів використовуються лінії ADR0 та ADR1 адресної шини мікропроцесорної системи.

У схемі на рис. 21 опитування ліній матриці здійснюється через прийомопередавач DD4 (працює на ввід даних), а встановлення “0” на горизонтальних лініях матриці – через прийомопередавач DD5 (працює на вивід даних). Інші мікросхеми забезпечують адресацію клавіатури (фіксована адреса FE), а також формують керуючі сигнали зчитування та записування.

Часто в якості контролера клавіатури з організацією клавіатурної матриці 8x8 використовується запрограмований інтерфейс KP580BB79, який складається з двох функціонально незалежних частин – клавіатурної та дисплейної.

1.3.2. Організація пристрою відображення інформації

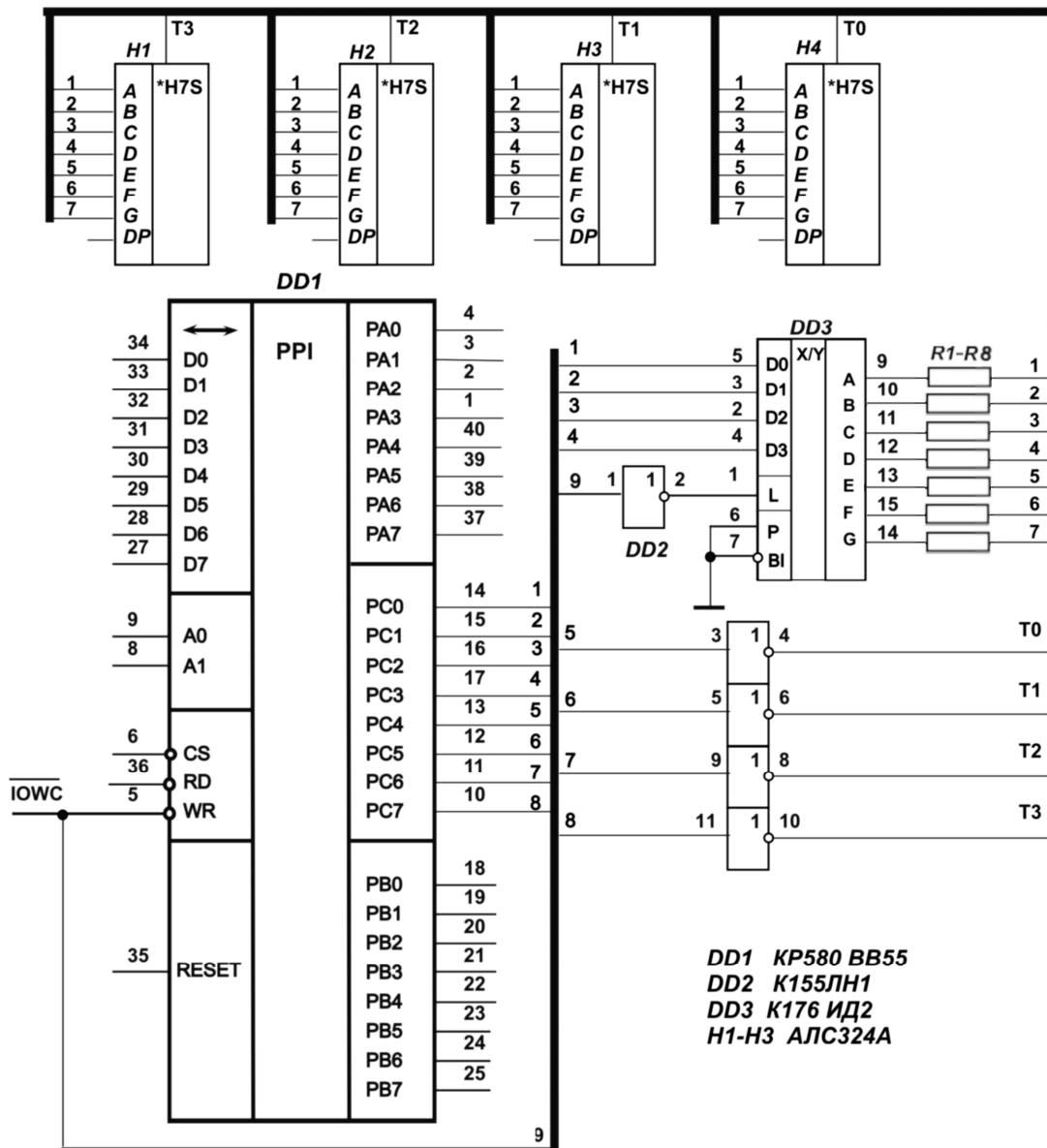


Рис. 22. Дисплей

На рис. 22 зображено один із можливих варіантів організації цифрового табло для відображення чотирьох десяткових цифр за допомогою світлодіодних семисегментних індикаторів АЛС324 А. 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 КР580ВВ55. Для перетворення двійково-десяткового коду в семисегментний код використовується перетворювач, що містить регістр пам'яті, К176ИД2.

В якості контролера дисплея використовується також програмований

інтерфейс KP580BB79, 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 виводитись двома способами: зліва направо без зсуву та справа наліво із зсувом. Контролер дозволяє відображати інформацію на люмінесцентних індикаторах, світлодіодних індикаторах та ін. Для виводу інформації на алфавітно-цифрові дисплеї використовується контролер електронно-променевих трубок KP580BG75.

1.3.3. Побудова технічних засобів для узгодження сигналів мікропроцесорної системи з об'єктами керування

1.3.3.1. Вивід дискретних сигналів

Технічні засоби виводу дискретних сигналів від мікропроцесорної системи призначені для передавання 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.

Найбільш розповсюджені схеми керування виконавчими механізмами за допомогою вихідних формувачів зображено на рис. 23.

Найпростішим формувачем є мікросхеми з підвищеною навантажувальною здатністю, що зазвичай мають виходи з відкритим колектором або емітером (рис. 23 а, б DD1 K155ЛН1), навантаженням котрих можуть слугувати оптоелектронні пристрої, світлодіоди або лампочки розжарювання, малопотужні герконові або поляризовані реле, тощо.

Для керування потужнішими зовнішніми об'єктами, напруга яких значно перевищує напругу живлення мікропроцесорної системи керування, в якості вихідних формувачів використовують потужні транзистори (наприклад, КТ814, КТ815, КТ816, КТ817, КТ818, КТ819) (рис. 23 в, г), вхідні ланцюги яких не мають гальванічного зв'язку із мікропроцесорною системою керування, а також оптоелектронні тиристорні пари, що дозволяють комутувати

малопотужні (рис. 23 е) та силові кола змінного струму (рис. 23 д).

Виконавчими елементами в таких схемах можуть бути обмотки пневмота гідроклапанів, пускачів, електродвигуни невеликої потужності.

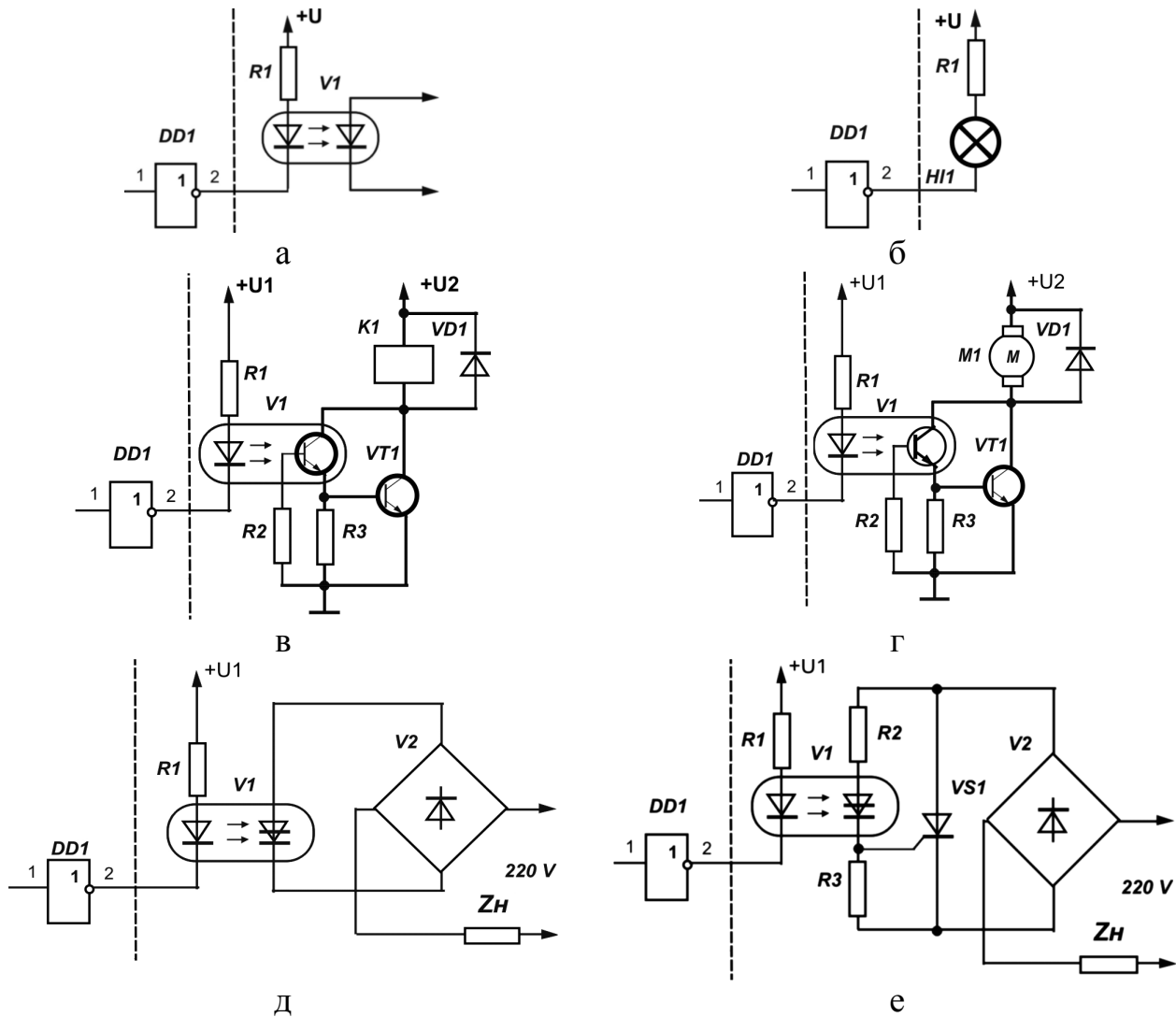


Рис. 23. Вихідні цифрові пристрої

Керування 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 основних способи керування виводом дискретних сигналів:

- ◇ окреме керування кожним двійковим розрядом окремою командою від МП-системи;
- ◇ групове керування одночасно всіма двійковими розрядами, які можуть бути сформовані МП-системою залежно від розрядності

шини вводу-виводу (8, 16 і більше розрядів).

1.3.3.2. Ввід дискретних сигналів

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

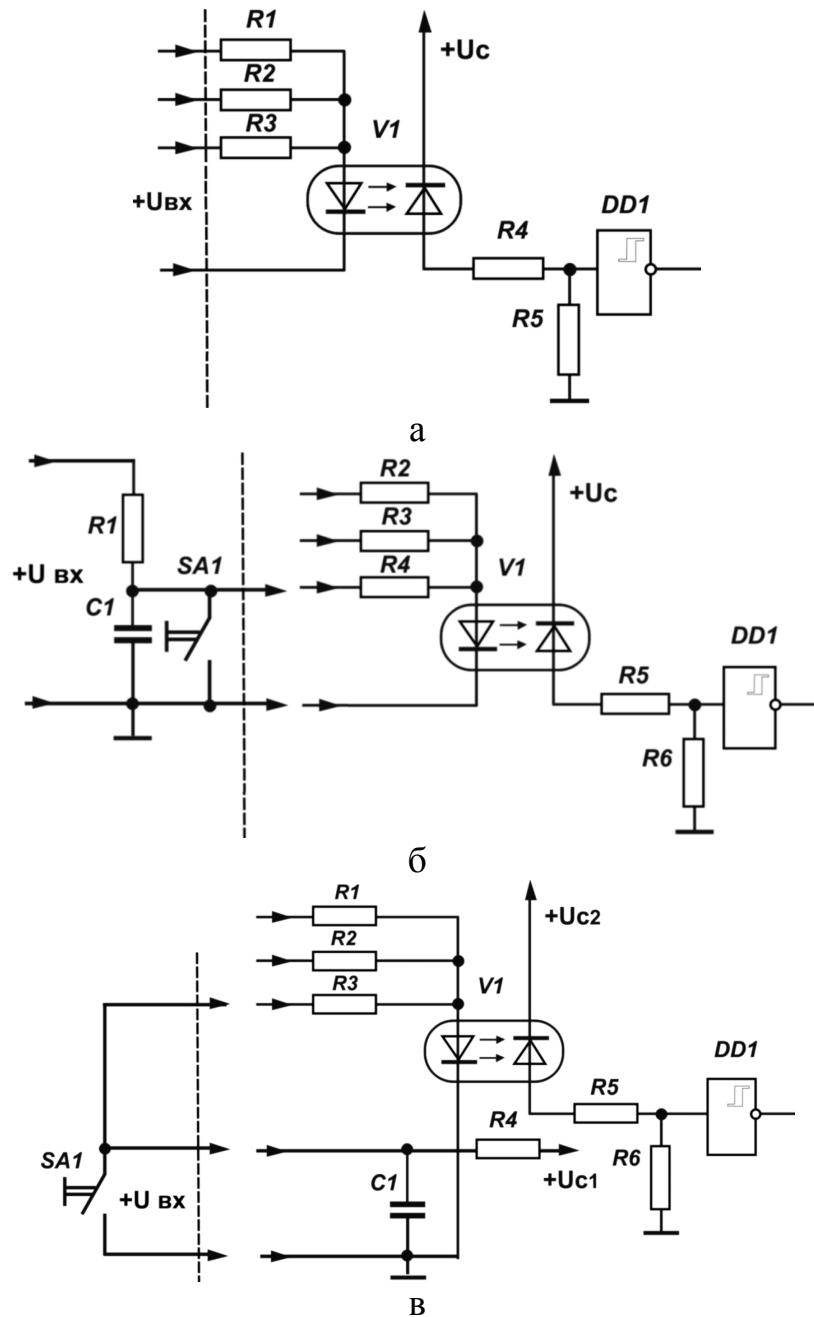


Рис. 24. Ввід дискретних сигналів

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

Під'єднують давачі дискретних сигналів як безпосередньо до МП-системи, так і через зовнішні електронні формувачі сигналів. Причому доцільно вводити гальванічне розділення вхідних кіл давачів та пристроїв вводу МП-системи.

Розрізняють три види подання інформації від дискретних давачів:

- ◇ двопозиційний, при якому кожен двійковий сигнал, що вводиться, функційно не залежить від інших і визначає стан частини об'єкта керування (стан кінцевих вимикачів, наприклад);
- ◇ кодовий, коли сукупність двійкових розрядів формує функційно пов'язану групу сигналів (код). Наприклад, код, котрий вказує кут повороту;
- ◇ імпульсний, за якому стан об'єкта визначається через кількість імпульсів. Наприклад, кількість деталей на конвеєрі відповідає кількості імпульсів від фотоелемента.

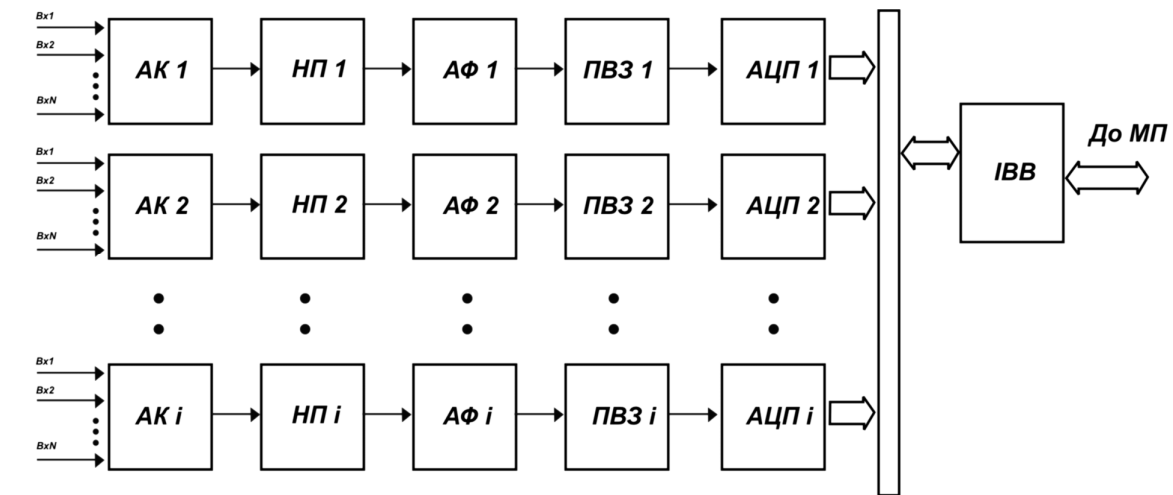
Типові схеми під'єднання дискретних давачів показана на рис. 24 а, б, в. Кожна схема містить обмежуючі резистори різного номіналу, через які, залежно від його рівня, вхідний дискретний сигнал подається на оптоелектронний елемент, який крім функцій формувача вхідного сигналу, забезпечує гальванічну розв'язку ланцюгів. З виходу оптопари 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 подальшого опрацювання МП-системою.

1.3.3.3. Ввід аналогових сигналів

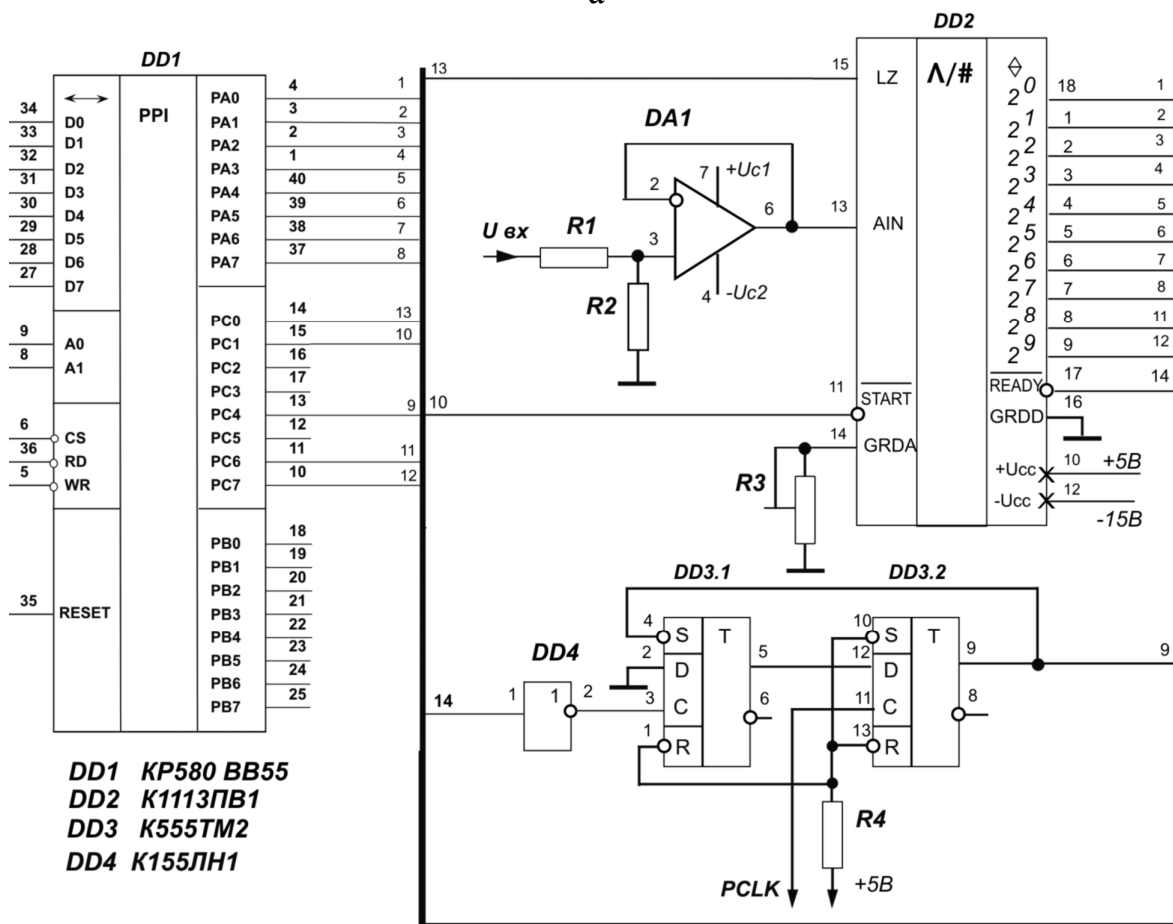
Пристрої вводу аналогових сигналів у загальному містять (рис. 25 а):

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

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



а



б

Рис. 25. Ввід аналогових сигналів

Аналоговий комутатор призначений для під'єднання аналогових датчиків до входу підсилювача НП, який в свою чергу здійснює операцію пропорційного підсилення або послаблення (нормування) аналогових сигналів, що надходять від датчиків, і одночасно узгоджує їх із діапазоном вимірювання АЦП.

Аналоговий фільтр АФ призначений для видалення високочастотних завад. Пристрій ПВЗ використовується для розширення частотного діапазону вимірюваних сигналів, а АЦП перетворює аналоговий сигнал у цифровий код, який за командами від інтерфейсу вводу-виводу ІВВ надходить у мікропроцесорну систему.

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

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

◇ ПВЗ виконують 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_b ;

◇ запам'ятовування, основним параметром якого є швидкість розряду запам'ятовуючого конденсатора.

АЦП, в залежності від застосування, поділять на АЦП широкого застосування та спеціалізовані. До перших відносять мікросхеми, що мають 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. Найпоширенішими є АЦП К1113ПВ1А, К1108ПВ1А, К572ПВ1А, К1107ПВ1, К1107ПВ2.

На рис. 25 б зображена 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'єднують через резистор (R3) до загальної шини для регулювання зміщення нуля. Цифрову землю GRDD заземляють. Для регулювання діапазону вхідного сигналу (від 0 до 10В) використовують подільник R1, R2.

Сигнали START і STOP формуються за допомогою тригерів DD3.1 та DD3.2. За сигналом по входу START, 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 TTL 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 надходить в мікропроцесор.

1.3.3.4. Виведення аналогових сигналів

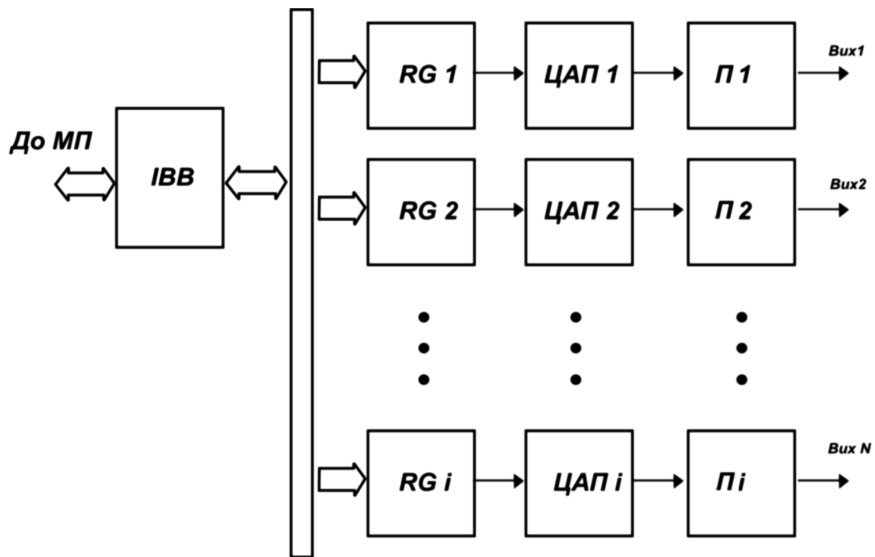
Пристрої виводу аналогових сигналів у загальному випадку містять (рис. 26 а):

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

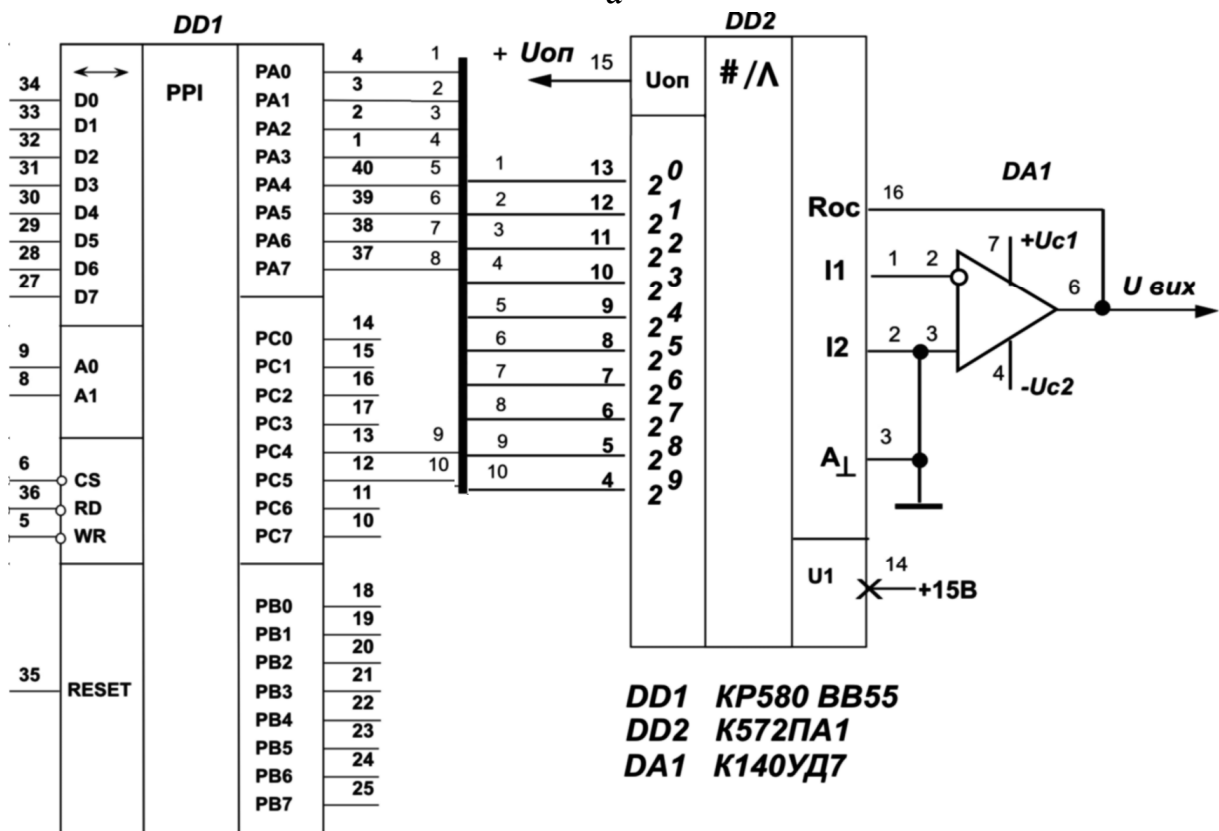
Входи регістрів кожного з -каналів через інтерфейс вводу-виводу (IBV) під'єднується до системної шини мікропроцесорного пристрою.

Буферний регістр призначений для зберігання вхідного коду, що надходить від пристрою виводу. У випадку, коли функції ПВВ виконує, наприклад, програмований паралельний інтерфейс (KP580BB55), 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.



а



б

Рис. 26. Аналоговий вивід сигналів

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

На рис. 26 б зображена схема під'єднання ЦАП та узгодження його з мікропроцесорним пристроєм через програмований паралельний інтерфейс KP580BB55.

Мікросхема К572ПА1 (DD2) є 10-розрядним ЦАП помножуючого типу, побудованого на принципі сумування струмів із живленням від джерела опорної напруги $U_{оп}$. Для роботи ЦАП потрібні зовнішній операційний підсилювач (DA1) та стабілізоване джерело опорної напруги.

Найчастіше також використовують мікросхеми ЦАП:

- ◇ 12-розрядний К572ПА2;
- ◇ 12-розрядний паралельний ЦАП К594ПА1;
- ◇ швидкодіючий 12-розрядний ЦАП зі сумуванням струмів К1108ПА1;
- ◇ швидкодіючий 8-розрядний ЦАП К1118ПА1 та ін.

1.4. Контрольні запитання

1. Структура мікропроцесорної системи на базі МП КР580ВМ80А.
2. Формування адресної шини, шини даних та шини керування з використанням регістрів, шинних формувачів, мікросхем цифрової логіки.
3. Побудова оперативної та постійної пам'яті в системі на базі МП КР580ВМ80А.
4. Організація шини вводу-виводу.
5. Організація роботи МП-системи в циклі переривання та режимі прямого доступу до пам'яті.
6. Використання контролерів ПКП та ПДП серії КР580.
7. Під'єднання цифрових та аналогових датчиків до МП-системи.
8. Під'єднання силових каскадів на основі транзисторних, тиристорних ключів та оптоелектронних пар.

ТЕМА №2. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ МП

2.1. Мікропроцесор К1810ВМ86

2.1.1. Умовне позначення мікропроцесора та загальна характеристика

17	NMI	CPU	A/D0	16	
18	INT		A/D1	15	
19	CLK		A/D2	14	
21			RST	A/D3	13
22			RDY	A/D4	12
23	$\overline{\text{TEST}}$		A/D5	11	
33			A/D6	10	
30			A/D7	9	
31	MN/ $\overline{\text{MX}}$		A/D8	8	
30			A/D9	7	
31			A/D10	6	
31			A/D11	5	
31			HLDA (RQ/GT1)	A/D12	4
31			HOLD (RQ/GT0)	A/D13	3
31			HOLD (RQ/GT0)	A/D14	2
31		HOLD (RQ/GT0)	A/D15	39	
31		HOLD (RQ/GT0)	A16/S3	38	
31		HOLD (RQ/GT0)	A17/S4	37	
31	HOLD (RQ/GT0)	A18/S5	36		
31	HOLD (RQ/GT0)	A19/S6	35		
31	HOLD (RQ/GT0)	A18/S5	36		
31	HOLD (RQ/GT0)	A19/S6	35		
31	HOLD (RQ/GT0)	$\overline{\text{BHE/S7}}$	34		
31	HOLD (RQ/GT0)	$\overline{\text{WR(LOCK)}}$	29		
31	HOLD (RQ/GT0)	$\overline{\text{RD}}$	32		
31	HOLD (RQ/GT0)	M/ $\overline{\text{I0}}$ (S2)	28		
31	HOLD (RQ/GT0)	DT/R(S1)	27		
31	HOLD (RQ/GT0)	DEN(S0)	26		
31	HOLD (RQ/GT0)	ALE(QS0)	25		
31	HOLD (RQ/GT0)	$\overline{\text{INTA(QS1)}}$	24		

Рис. 27. Умовне позначення мікропроцесора К1810ВМ86

Переваги процесора порівняно з 8-бітними:

- ◇ розширена система команд;
- ◇ більший об'єм адресованої пам'яті;
- ◇ можливість використання співпроцесорів;
- ◇ розширена система переривань;
- ◇ використання черги команд та конвеєризації їх виконання;
- ◇ збільшена швидкодія при виконанні програм.

Умовне позначення мікропроцесора зображено на рис. 27.

2.1.2. Призначення виводів МП

Призначення виводів МП наведено в таблиці.

Таблиця 12. Призначення виводів МП

Вивід	Позначення виводу	Призначення виводу
1	2	3
Виводи МП, призначення яких не змінюється від режиму роботи.		
1, 20	GND	Загальний
16-2, 39	A/D0 – A/D15	Мультиплексована шина (канал) адреси / даних
17	NMI	Немаскований запит переривання. Логічна 1 на вході призводить до переходу МП до опрацювання переривання INT 2
18	INT	Маскований запит переривання. Логічна 1 на вході призводить до переходу МП до процедури підтвердження переривання у випадку наявності дозволу маскованого переривання
19	CLK	Тактовий синхросигнал. Початок машинного такту відбувається при переході сигналу в стан лог. 0. Тривалість лог. 1 становить 1/3 періоду сигналу
21	RST	Початкове встановлення (скидання). Логічна 1 на вході, що триває більше 4 періодів тактового сигналу, призводить до початкового скидання мікропроцесора
22	RDY	Готовність. Логічна 1 на вході свідчить про готовність зовнішнього пристрою до обміну
23	$\overline{\text{TEST}}$	Перевірка. Логічний 0 на вході вказує, що пристрій який перевіряється програмним шляхом, готовий до виконання наступної операції. Перевірка здійснюється за спеціальною командою WAIT
32	$\overline{\text{R}}$	Зчитування. Логічний 0 на виході вказує, що в цьому циклі обміну відбувається зчитування

1	2	3
33	MN/\overline{MX}	Мінімальний або максимальний режим. Логічний 0 вказує на роботу МП у максимальному режимі роботи, лог. 1 - у мінімальному
34	$\overline{BHE}/ST7$	Дозвіл передавання старшої половини каналу даних AD8 – AD15 та сигнал стану
35-38	A19/S6, A18/S5, A17/S4, A16/S3	Старші сигнали шини адреси мультиплексовані з сигналами стану
40	+5 В	Напруга живлення
Виводи МП у мінімальному режимі роботи		
24	\overline{INTA}	Сигнал підтвердження переривання. Активний рівень – лог. 0
25	ALE	Строб адреси. Задній фронт (перехід зі стану лог. 1 у стан лог. 0) сигналу на виході вказує, що на шині даних/адреси знаходиться адреса
26	\overline{DEN}	Дозвіл для зовнішніх буферів на передачу адреси. Логічний нуль на виході вказує, що буферні регістри, котрі запам'ятали адресу, мають видати її на шину адреси
27	DT/\overline{R}	Ввід/вивід даних. Логічна 1 на виході вказує, що відбувається приймання даних, логічний 0 - передавання. Сигнал призначений для керування вихідними буферами шини даних
28	M/\overline{IO}	Пам'ять/зовнішній пристрій. Логічна 1 на виході вказує, що відбувається обмін з пам'яттю, логічний ноль - обмін з пристроями вводу-виводу, або з підтвердження переривання
29	\overline{WR}	Запис. Логічний нуль вказує, що відбувається записування .
30	HLDA	Підтвердження захоплення каналу. Логічна 1 вказує, що зовнішній пристрій, який запросив прямий доступ до пам'яті, може захопити шини МП
31	HOLD	Запит захоплення каналу. Логічна 1 вказує, що зовнішній пристрій запросив прямий доступ до пам'яті
Виводи МП у максимальному режимі.		
24,25	QS1, QS0	Сигнал стану черги команд
26 – 28	S0 – S2	Сигнал стану циклу каналу
29	\overline{LOCK}	Канал зайнятий. Логічний 0 вказує на те, що відбувається обмін з пам'яттю чи портами вводу-виводу, котрий не можна переривати
30, 31	RQ/GT1, RQ/GT0	Дві однобітних двонаправлених шини запиту та дозволу доступу до каналу

Біти S3-S7 інформують про:

- ◇ S3 - S4 – сегментний регістр, за допомогою якого здійснюється обмін з пам'яттю;
- ◇ S5 – стан тригера дозволу переривань;
- ◇ S6, S7 – ідентифікують процесор.

Біти S2 S1 S0 вказують тип обміну по шині згідно з таблицею.

Таблиця 13. Стан біт S2 S1 S0 залежно від типу обміну по шині

S2	S1	S0	Тип обміну
0	0	0	Підтвердження переривання
0	0	1	Вибірка команди
0	1	0	Записування в порт
0	1	1	Записування в пам'ять
1	0	0	Читання порту
1	0	1	Читання пам'яті
1	1	0	Зупинка
1	1	1	Пасивний стан

2.1.3. Вибір режиму конфігурації

Мінімальний режим МП (вивід MN/MX під'єднати до шини живлення) призначений для використання МП у простих однопроцесорних системах. Крім сигналів керування обміном з пам'яттю та зовнішніми пристроями, МП забезпечує доступ до системного каналу за запитом від контролера прямого доступу до пам'яті (ПДП) за сигналами HLD та HLDA.

Максимальний режим МП (вивід MN/ $\overline{\text{MX}}$ під'єднаний до загальної шини) призначений для використання МП у складних однопроцесорних і мультипроцесорних системах. У цьому режимі системний контролер КР1810ВГ88 декодує вихідні сигнали стана МП ST0...ST2 і видає сигнали керування обміном. Інші п'ять з восьми виводів МП використовуються для роботи МП у мультипроцесорних системах.

2.1.4. Структурна схема мікросхеми

Структурна схема складається з наступних пристроїв:

- ◇ пристрій спряження з каналом;
- ◇ операційний пристрій.

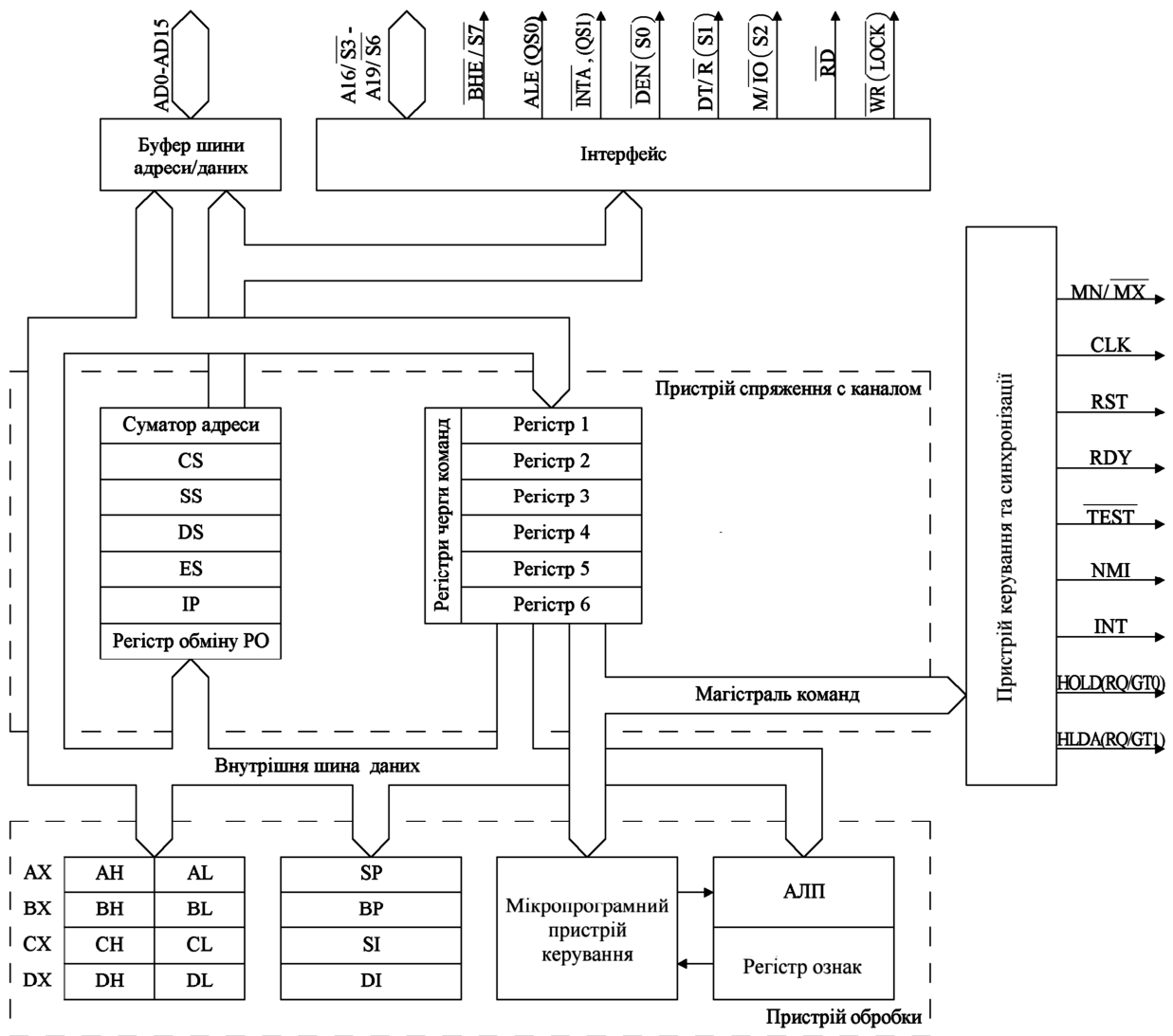


Рис. 28. Структурна схема мікросхеми K1810BM86

Присрій спряження з каналом складається із:

1. Буфера адреси даних, що призначений для підсилення та буферизації сигналів шини даних та адреси).
2. Пристрою керування та синхронізації, що формує всі сигнали, які керують роботою МП.
3. Інтерфейсу – пристрою, що формує сигнали керування шиною МП.
4. Регістрів черги команд, що зберігають у собі команди які будуть виконуватись у подальшому.
5. Блока формувача адреси, який складається із:
 - 5.1. Суматора адреси, що формує 20-бітну адресу із сегментної частини та зміщення.

5.2. Групи сегментних реєстрів, призначених для збереження сегментної частини адреси. До них відносять реєстри: сегменту коду (CS), сегменту стеку (SS), сегменту даних (DS), додаткового сегменту (ES).

5.3. Реєстра обміну, призначеного для збереження зміщення під час обміну даними.

5.4. Лічильник команд, що зберігає зміщення команди IP.

Операційний пристрій, що виконує всі операції МП, складається із:

1. Блока реєстрів загального призначення, до яких відносять 16-розрядні реєстри, які можуть також використовуватись як 8-розрядні, а саме:

1.1. Акумулятора AX, що складається із 2 восьмирозрядних реєстрів: AL та AH.

1.2. Базового реєстра BX, що складається із 2 восьмирозрядних реєстрів: BH та BL.

1.3. Лічильника циклів CX, що складається із 2 вісьмирозрядних реєстрів: CH та CL.

1.4. Реєстра даних DX, що складається із 2 вісьмирозрядних реєстрів: DH та DL.

Фактично всі реєстри загального призначення функціонують аналогічно акумулятору МП КР580ВМ80, проте кожен із них має свою додаткову специфічну функцію, що й визначає його назву.

2. Блока індексних реєстрів, що мають розрядність 16 біт. До них відносять:

2.1. Індекс джерела SI.

2.2. Індекс отримувача DI.

2.3. Вказівник бази BP.

2.4. Вказівник стеку SP.

Кожен із індексних реєстрів також може виконувати функції акумулятора МП КР580ВМ80, але тільки при операціях з 16-розрядними числами, кожен із цих реєстрів має свою додаткову специфічну функцію.

3. Шеснадцятирозрядного АЛП, де виконуються всі арифметичні та

логічні команди.

4. Мікропрограмний пристрій керування, що керує роботою інших блоків МП.

5. Регістр ознак. Регістр ознак вказує стан процесора та ознаки результату виконання останньої арифметичної або логічної операції. Формат регістра ознак зображено на рис. 29. Призначення біт регістра ознак наведено в таблиці 14.

D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	OF	DF	IF	TF	SF	ZF	0	AF	O	PF	I	CF

Рис. 29. Структура регістра ознак

Таблиця 14. Призначення біт регістра ознак

Ознака	Значення
AF	Якщо біт дорівнює 1, то відбулось перенесення з третього розряду в четвертий, або позичання з четвертого розряду в третій. Ознака призначена для роботи з двійково-десятковими числами
CF	Якщо біт дорівнює 1, то відбулось перенесення або позичання із старшого розряду результату
OF	Якщо біт дорівнює 1, то відбулось переповнення, тобто втрачена значуща цифра, якщо розрядність результату перевищує розрядність регістра
SF	Якщо біт дорівнює 1, то результат від'ємний
PF	Якщо біт дорівнює 1, то результат містить парне число одиниць
ZF	Якщо біт дорівнює 1, то результат операцій дорівнює нулю
DF	Якщо біт дорівнює 1, при виконанні операцій з рядками вміст індексного регістра автоматично зменшується, тобто рядки опрацьовуються від старших адрес до молодших (справа наліво). Якщо біт дорівнює 0, – відбувається збільшення відповідного індексного регістра
IF	Якщо біт дорівнює 1, то МП реагує на зовнішні масковані запити переривання
TF	Якщо біт дорівнює 1, то МП переходить у покроковий режим виконання програми: після виконання кожної команди МП генерує внутрішні переривання типу 1.

2.1.5. Адресація портів вводу-виводу

Як і в МП K580BM80 порти вводу виводу та пам'ять знаходяться в окремих адресних просторах. Для звертання до портів вводу-виводу використовуються команди IN, OUT, які можуть працювати з прямою адресацією в діапазоні адрес 0-FFh, (наприклад, команда IN AL, 16h,) та із непрямою в більш широкому діапазоні адрес 0-FFFFh (наприклад, команда IN AX, DX).

2.1.6. Адресація пам'яті

Кожна комірка пам'яті, адресована МП K1810BM86, має два типи адрес: фізичну, що однозначно визначає її положення в 1Мбайті пам'яті, та логічну. Логічна адреса, містить 32 біти та складається із 2 частин – сегмента та зміщення, які є 16 бітними числами. Зазвичай, при програмуванні МП користуються логічною адресою, яка записується у формі SEG:OFFSET, де SEG - сегментна частина адреси, OFFSET - зміщення. Наприклад: B800:0024, 0040:0000, FF00:0000 тощо. Між фізичною та логічною адресами є проста відповідність:

$$PhA = 16 \cdot SEG + OFFSET,$$

де PhA – фізична адреса.

Слід зауважити, що у мікропроцесорі K1810BM86 у фізичній адресі використовуються тільки 20 молодших бітів результату додавання. При переповненні розрядної сітки старший 21 біт результату відкидається. Наприклад, при логічній адресі FFFF:0010 фізична адреса буде 00000, а не 100000. Зрозуміло, що одній і тій же фізичній адресі відповідає 16К різних логічних адрес. Це полегшує написання та використання програм, які можуть розміщатись у різних областях пам'яті.

Для того, аби налаштувати програму на використання в новій області пам'яті, необхідно лише змінити значення сегментної частини адреси. Зазвичай таке настроювання проводиться операційною системою. В літературі сегментом також часто називають сукупність усіх комірок пам'яті, які мають однакову сегментну частину адреси. Довжина такого блока в МП K1810BM86 становить 64 Кбайт.

В будь-який момент часу програма може звертатись до одного із 4 сегментів, що відповідають одному з 4 сегментних реєстрів. Кожен тип даних, які використовує МП, має свій сегментний реєстр, що визначає адресу цих даних. Можливі сегментні реєстри, які використовуються при адресації різних типів даних, наведені в таблиці 15.

Таблиця 15. Використання сегментів при адресації об'єктів у пам'яті

Дані, що адресуються	Сегмент			
	CS	DS	SS	ES
Коди команд	+	-	-	-
Дані, індексовані регістрам BP	S	S	+	S
Елементарні дані, та індексовані дані, крім даних, де зміщення формується за допомогою регістра BP	S	+	S	S
Стек	-	-	+	-
Рядок джерела	S	+	S	S
Рядок приймача	-	-	-	+
Таблиця	S	+	S	S
У таблиці прийняті такі позначення: + - сегмент по замовчуванню; S - сегмент, що може використовуватись за наявності префікса перекриття сегменту; - - сегмент неможливо використати.				

Формування зміщення залежить від режиму адресації типу команда та типу передавання по шині (вибірка команди, вибірка даних і.т.п.), і визначається такими правилами:

1. При вибірці команди зміщення береться із регістра IP.
2. При роботі зі стеком зміщення береться із регістра SP.
3. При роботі з даними формування зміщення здійснюється залежно від

типу команди, що виконується.

- 3.1. При прямій адресації зміщення визначається в самій команді

MOV AX, [1040H] ; AX:=Mem[DS:1040].

Де MEM[SEG:OFFSET] - комірка пам'яті з логічною адресою SEG:OFFSET.

- 3.2. При непрямій базовій адресації зміщення визначається вказівником бази

MOV AX, [BX] ; AX:=MEM[DS:BX].

- 3.3. При непрямій індексній адресації зміщення визначається індексним регістром

MOV AX, [DI] ; AX:=MEM[DS:BX].

- 3.4. При непрямій базово-індексній адресації зміщення визначається сумою індексного і базового регістрів

MOV AX, [BX] [SI] ; AX:=MEM[DS:BX+SI].

У випадку, коли сума індексного і базового реєстрів не вміщується в 16 розрядів, старші розряди відсікаються. Це правило розповсюджується і на інші види адресації.

3.5. При непрямій адресації з базовим реєстром і зміщенням, зміщення даних визначається сумою базового реєстра та зміщення, заданого в команді

MOV AL, [BP] [1020h] ; AL:=MEM[SS:BP+1020h] .

3.6. При непрямій адресації з індексним реєстром і зміщенням, зміщення даних визначається сумою індексного реєстра та зміщення, заданого в команді

MOV AX, [DI] [1000h] ; AD:=MEM[DS:DI+1000h]

3.7. При непрямій адресації з індексним і базовим реєстрами та зміщенням, зміщення даних визначається сумою індексного та базового реєстрів до якої додається зміщення, задане в команді

MOV AX, [SI][DI] [2000h] ; AD:=MEM[DS:SI+DI+2000h]

Слід зауважити, що зміщення може бути як 16-, так і 8-бітним. У випадку 8 бітного зміщення при формуванні адреси зміщення розширюється до 16 біт за допомогою копіювання в старший байт сьомого біта зміщення, заданого в команді.

3.8. При табличній адресації адреса формується шляхом сумування вмісту реєстра AL з реєстром BX. Така адресація використовується при виконанні команди XLAT AL – команди перекодування по таблиці. Дії, що виконує команда, можна записати у вигляді: AL:=MEM[DS:BX+AL].

3.9. При адресації рядка з якого беруться дані (рядка-джерела) в якості зміщення обов'язково використовується реєстр SI.

3.10. При адресації рядка в яку поміщаються дані (рядка-приймача) в якості зміщення обов'язково використовується реєстр DI.

2.1.7. Фізична організація пам'яті

Пам'ять організована, як 2 банка пам'яті по 512к – старший банк D15-D8 та молодший (D7-D0). Для адресації комірок пам'яті використовують розряди A19-A1. Байт з парною адресою пересилається по лінії D7-D0, непарний – по

лініях D15-D8.

Для вибірки використовуються сигнали: A0 (вибирає молодший банк при A0=0) та \overline{BNE} , як наведено в таблиці 16.

Таблиця 16. Вибірка банків пам'яті

Ошиб	A0	Банк, що вибирається	Лінії шини даних, по яких передаються дані
1	2	3	4
0	0	Обидва	D15-D0
1	0	Молодший	D7-D0
0	1	Старший	D15-D8
1	1	Заборонений стан	–

2.1.8. Переривання

Мікропроцесор дозволяє опрацьовувати 256 типів переривань із номерами 0 - 255, які поділяють на:

1. Зовнішні апаратні:

1.1.) масковані – блокуються програмно (по входу INT);

1.2.) немасковані – по входу NMI (програмно не блокуються).

2. Внутрішні апаратні.

3. Програмні.

Запити на зовнішні переривання МП сприймає та опрацьовує після виконання біжучої команди. Зовнішні переривання надходять на МП по двох виводах INT та NMI і діляться на масковані та немасковані.

Запити на масковані переривання видаються за сигналом INT, реакція на запит залежить від стана біта IF регістра ознак. Заборона переривання INT здійснюється установкою в „0” внутрішнього тригера дозволу переривань командою CLI, дозвіл – командою STI, що встановлює тригер в „1”. Якщо переривання дозволені, після його розпізнання мікропроцесор формує цикл підтвердження переривання для зовнішнього пристрою, що запросив переривання. Зовнішній пристрій у відповідь на цикл підтвердження переривання формує та видає в МП номер переривання, яке відбулось. Цей номер носить назву вектора переривання.

При немаскованому перериванні запит надходить по входу NMI.

Опрацювання переривання не залежить від стану біта IF регістра ознак. Немасковані переривання використовуються для повідомлення процесора в „загрозуючих” режимах (можливість від’єднання живлення, помилка пам’яті і т.д.). Для переривань NMI немає підтвердження, а МП реагує на перехід в стан лог. 1 на цьому вході, активний сигнал NMI має залишатися не менше як у двох машинних тактах.

Внутрішні апаратні переривання ділять на переривання через помилку ділення та покрокового виконання програми. Переривання через помилку ділення проходять під час виконання команд ділення DIV та IDIV, якщо результат ділення перевищує можливу допустиму величину. Покрокове переривання виникає при TF=1 після виконання кожної команди, або пари команд, перша з яких – префікс заміни сегмента.

Програмні переривання ділять на:

1. Переривання по відлагоджувальним точкам.
2. Переривання по переповненню.
3. Переривання за вектором.

Переривання за відлагоджувальними точками генерується за однобайтною командою INT3.

Переривання за переповненням генерується по однобайтній команді INTO. Якщо біт OF регістра ознак рівний одиниці, то по команді INTO відбувається перехід на відповідну процедуру, інакше виконується наступна команда.

Переривання за вектором виконується по двобайтній команді INTx, де $x = 0..FF$, усі можливі переривання можуть бути викликані за допомогою команди INTx із відповідним номером вектора. Номери вектора переривання для всіх можливих переривань наведені в таблиці 17.

Запити на переривання опрацьовуються після виконання біжучої команди. Якщо одночасно надходять переривання різного типу, то опрацьовується переривання з вищим пріоритетом.

Таблиця 17. Переривання в KP1810BM86

Тип переривання	Номер вектора	Пріоритет
Переривання через помилку ділення	0	1(вищий)
Покрокове переривання	1	4 (найнижчий)
Немасковане переривання (по вх. NMI)	2	2
Переривання за відлагоджувальній точці INT3	3	1
Переривання за переповненням INT0	4	1
Переривання за вектором INTx	x	1
Масковані переривання (по входу INT)	x	3

Процедура опрацювання переривання визначається номером вектора переривання і береться з таблиці векторів переривань, що розміщена в початкових адресах адресного простору МП. Кожен із елементів таблиці має довжину 32 біта і складається зі зміщення (молодші 16 біт) та сегмента (старші 16 біт) адреси початку підпрограми переривань. При надходженні будь-якого переривання МП визначає номер вектора переривання, зберігає в стеку вміст лічильника команд, сегмента коду та регістра ознак, потім вибирає з таблиці згідно з номером зміщення та сегмент адреси початку підпрограми опрацювання переривань, скидає біт дозволу переривань IF в регістрі ознак і переходить до викання команди, записаної за згаданою адресою. Якщо в підпрограмі опрацювання переривання використовуються регістри МП, їх попередній стан має бути збережений у стеку. У випадку, коли в процесі опрацювання переривання необхідно дозволити масковані переривання, то можна скористуватись командою STI. В кінці підпрограми опрацювання переривання стан усіх регістрів МП, що були збережені у стеку, необхідно відновити. Підпрограма опрацювання переривань повинна закінчуватися командою IRET, яка повертає зі стеку збережений вміст лічильника команд, вміст регістра ознак та вміст сегментного регістра. Так як перед виконанням підпрограми обробки переривань біт дозволу переривань був встановлений в 1, то після повернення із підпрограми переривання і далі будуть дозволені.

2.1.9. Часові діаграми роботи МП у мінімальному режимі роботи

Як і в МП КР580ВМ80 робота мікропроцесора К1810ВМ86 здійснюється по тактах, проте в МП К1810ВМ86 виконання команд у випадку, коли команда не виконує передавання по шині даних, не прив'язане до шинних циклів МП, достатньо лише, щоб у черзі команд була хоча б одна невиконана команда.

Кожен обмін по шині називають машинним циклом, що складається із машинних тактів. Ознакою початку машинного такту є перехід сигналу CLK зі стана лог. 1 в стан лог. 0. Ознакою початку машинного циклу в мінімальному режимі роботи є перехід сигналу ALE зі стана лог. 0 у стан лог. 1.

2.1.9.1. Часова діаграма читання пам'яті або портів вводу-виводу в мінімальному режимі роботи

Часова діаграма читання пам'яті чи портів вводу-виводу в мінімальному режимі роботи зображена на рис. 30. У кінці попереднього машинного такту, на лінію $\overline{DT/R}$ видається сигнал, що вказує на наявність приймання даних у наступному циклі (лог. 0). На початку машинного циклу по спаду CLK, в першому машинному такті мікропроцесор видає на лінію $\overline{M/IO}$ ознаку обміну з пам'яттю чи портом вводу-виводу, а на лінію ALE – логічну 1. З певною затримкою відносно до сигналу ALE, на шині AD0-AD15, лініях A16/S4 - A19/S6 та лінії $\overline{BHE}/S7$ видається адреса, по якій відбувається обмін. Потім (з деякою затримкою відносно до фронту сигналу CLK) на лінії ALE встановлюється лог. 0. Перехід сигналу ALE в стан логічного нуля вказує, що адресу видану на шину, необхідно зафіксувати. По спаду сигналу CLK лінії AD0-AD15 переводяться на прийом даних, а на лінії A16/S4 - A19/S6 та лінію $\overline{BHE}/S7$ видається слово стана.

Після цього на лінії \overline{R} видається лог. 0, що вказує на прийом даних, а по фронту сигналу CLK на лінії \overline{DEN} встановлюється лог. 0, котрий дозволяє

роботу вихідних буферів. По спаду сигналу CLK на початку 3 машинного такту аналізується стан лінії RDY, якщо лінія знаходиться в стані лог. 0, то після такту T2 буде такт очікування T0, в пролежному випадку - T3.

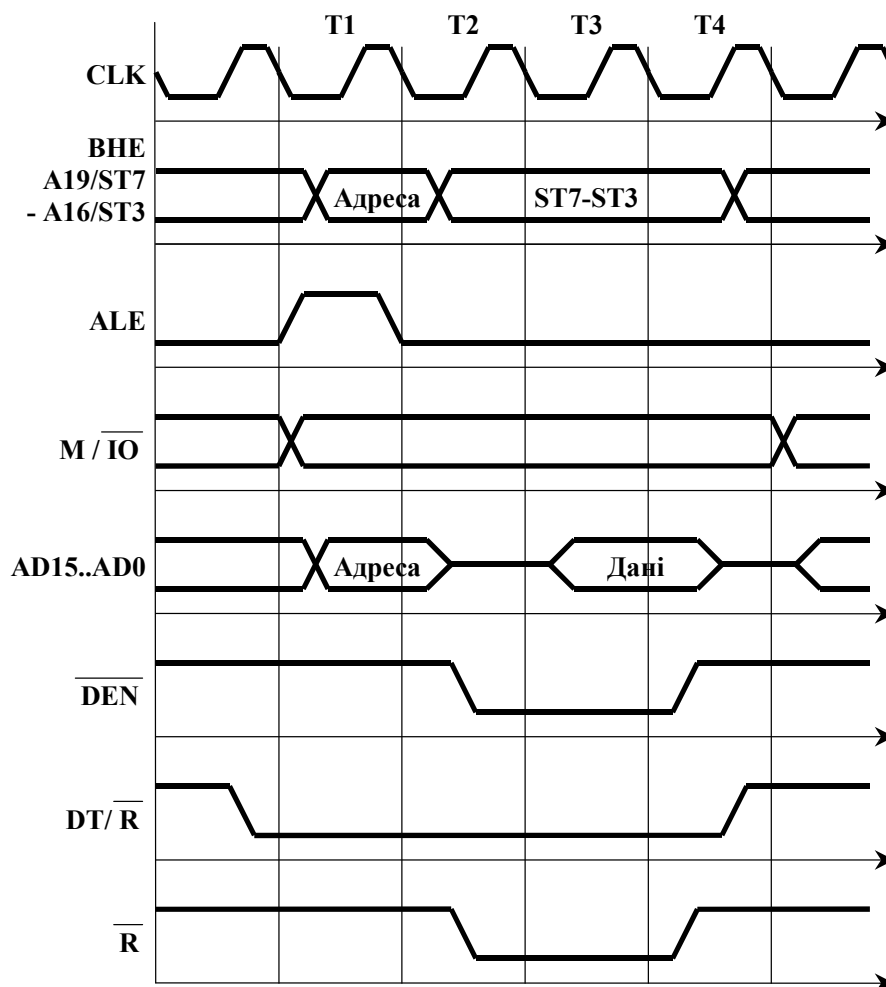


Рис. 30. Часова діаграма роботи МП при читанні в мінімальному режимі роботи

У такті T0 всі лінії зберігають свій стан і по спаду сигналу CLK знову аналізується стан лінії RDY. Якщо лінія в стані лог. 1, то наступний такт – T3, в пролежному випадку повторюється знову такт T0. У такті T4 по спаду сигналу CLK дані видані на лінії зовнішнім пристроєм приймаються МП. Протягом такту T4 мікропроцесор знімає сигнали керування в такому порядку: на початку такту знімається сигнал \overline{DEN} та \overline{R} , по фронту сигналу CLK лінія DT/ \overline{R} переходить у стан, що відповідає наступному обміну. На цьому обмін у режимі читання завершено.

2.1.9.2. Часова діаграма записування у пам'ять або у порти введення-виведення в мінімальному режимі роботи

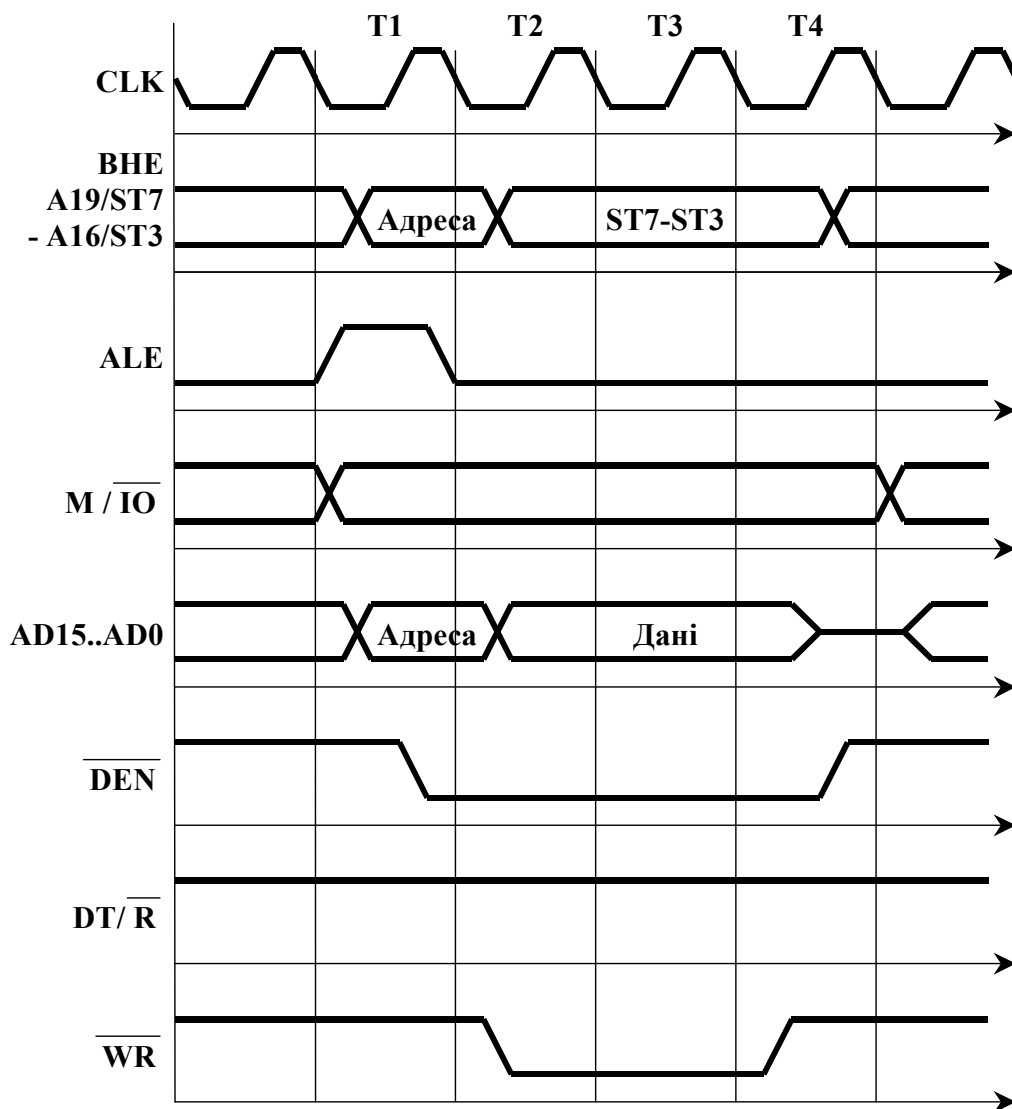


Рис. 31. Часова діаграма роботи МП при записуванні в мінімальному режимі

Часова діаграма записування пам'яті чи портів вводу-виводу в мінімальному режимі роботи зображена на рис. 31.

Подібно до часової діаграми читання, у кінці машинного такту, що передуює початку машинного циклу, який розглядається, на лінію $\overline{DT/R}$ видається сигнал, що вказує на наявність передавання даних у наступному циклі (лог. 1). На початку машинного циклу по спаду CLK, у першому машинному такті мікропроцесор видає на лінію $\overline{M/IO}$ ознаку обміну з пам'яттю чи портом вводу-виводу, а на лінію ALE – логічну 1. З певною

затримкою відносно сигналу ALE, на шині AD0-AD15, лініях A16/S4 - A19/S6 та лінії $\overline{\text{BHE}}/\text{S7}$ видається адреса, за якою відбувається обмін. Потім, з деякою затримкою відносно фронту сигналу CLK на лінії ALE встановлюється лог. 0, а на лінію $\overline{\text{DEN}}$ видається лог. 0, що дозволяє роботу вихідних буферів. Перехід сигналу ALE в стан лог. 0 вказує, що адресу, видану на шину, необхідно зафіксувати. По спаду сигналу CLK на лінії AD0-AD15 видаються дані, що записуються, а на лінії A16/S4 - A19/S6 та лінію $\overline{\text{BHE}}/\text{S7}$ – слово стана. Після цього на лінії $\overline{\text{WR}}$ видається лог. 0, вказуючи на запис даних. По спаду сигналу CLK на початку 3 машинного такту аналізується стан лінії RDY. Якщо лінія знаходиться в стані лог. 0, то після такту T2 буде такт очікування T0, в протилежному випадку - T3. У такті T0 всі лінії зберігають свій стан і по спаду сигналу CLK знову аналізується стан лінії RDY. Такти T0 повторюються поки лінія не перейде в стан лог. 1. У такті T3 сигнали лишаються незмінними. Наступним за T3 іде такт T4. Протягом такту T4 мікропроцесор знімає сигнали керування в такому порядку: на початку такту знімається сигнал $\overline{\text{WR}}$; по фронту сигналу CLK лінія DT/ $\overline{\text{R}}$ переходить у стан, що відповідає наступному обміну, з шини даних знімаються дані, що записувались, а лінія $\overline{\text{DEN}}$ переводиться в пасивний стан. На цьому обмін у режимі записування завершено.

2.1.9.3. Часова діаграма підтвердження переривань у мінімальному режимі роботи

Часова діаграма підтвердження переривань у мінімальному режимі роботи зображена на рис. 32. Підтвердження переривання виконується у відповідь на сигнал маскованого запиту переривань у випадку, коли масковані переривання дозволені. В процесі підтвердження переривання виконуються два шинних цикли, що розділені двома холостими тактами. Часова діаграма циклу підтвердження переривання аналогічна часовій діаграмі зчитування з порту

вводу-виводу з наступними відмінностями: протягом обох циклів адреса не видається, але формується строб ALE і сигнали, що вказують на приймання даних, замість сигналу \overline{RD} формується сигнал \overline{INTA} .

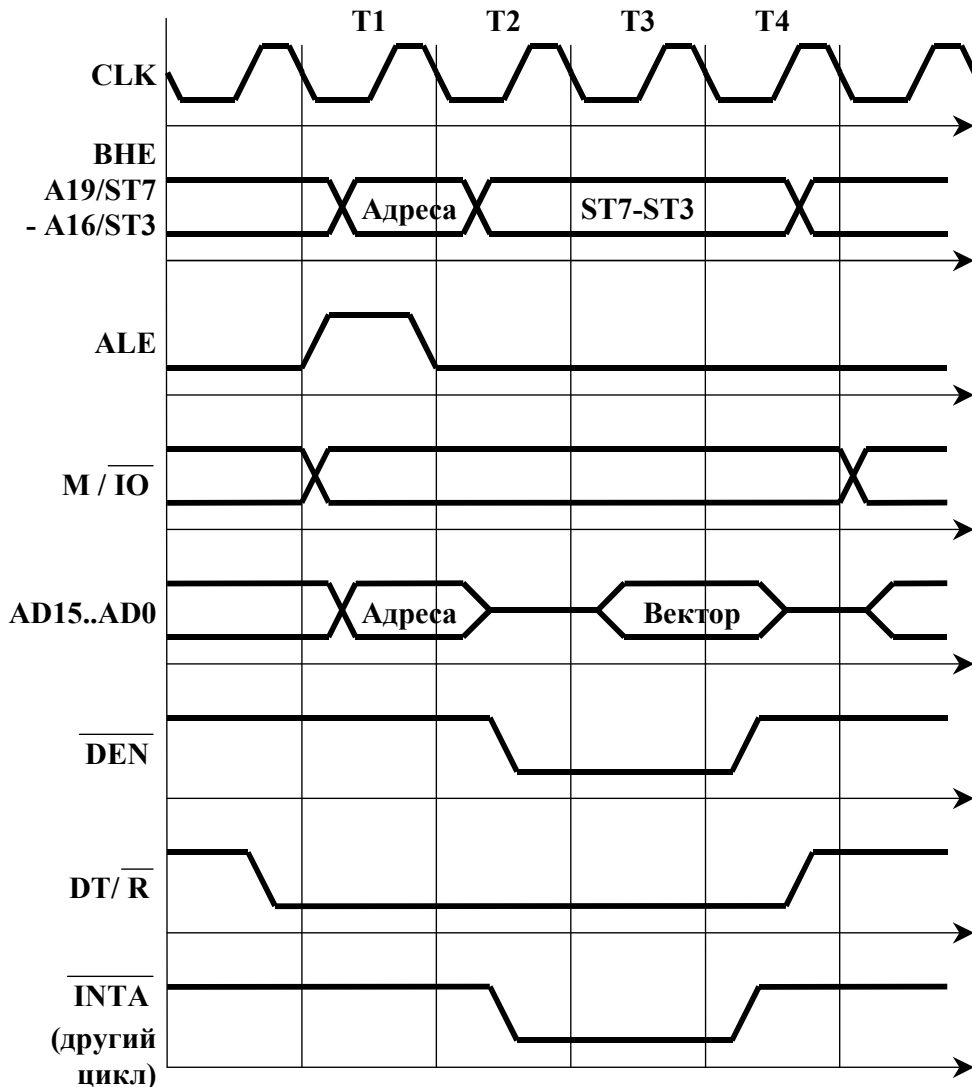


Рис. 32. Часова діаграма роботи МП при підтвердженні переривання в мінімальному режимі роботи

Перший цикл підтвердження переривання вказує контролеру переривань, що йому необхідно підготувати номер переривання призначений, для передавання в МП. У цьому циклі дані, які видає контролер не сприймаються і не аналізуються. В другому циклі байт номера переривання (вектор), що видається на лінії AD0-AD7, сприймається МП. У процесі підтвердження переривання лінія HOLD не аналізується, тобто між циклами підтвердження переривання не може бути викликаний цикл ПДП.

2.1.9.4. Часова діаграма прямого доступу до пам'яті у мінімальному режимі роботи

Часова діаграма прямого доступу до пам'яті в мінімальному режимі роботи зображена на рис. 33.

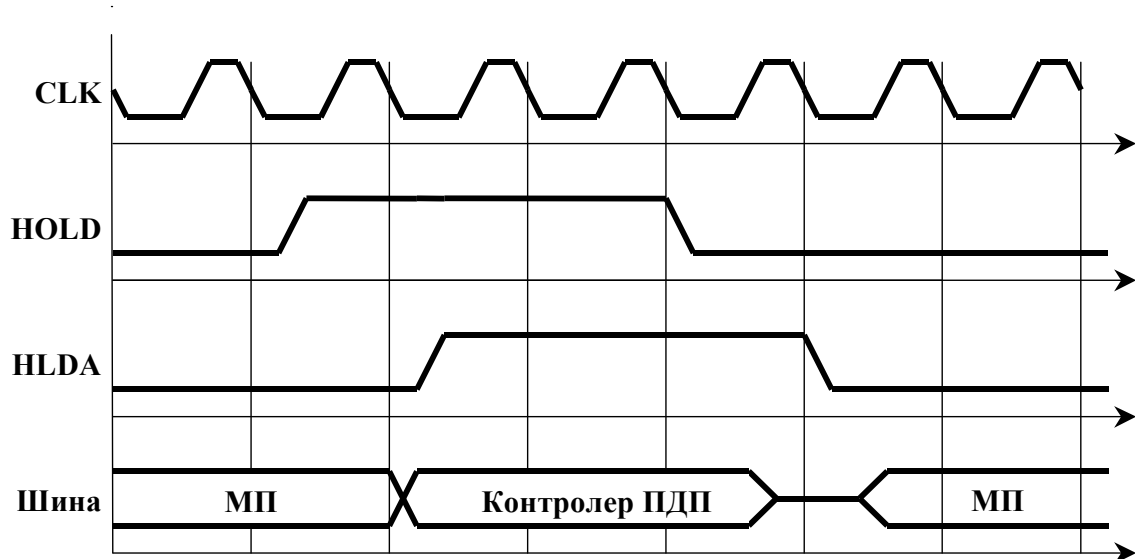


Рис. 33. Часова діаграма роботи МП при прямому доступі до пам'яті в мінімальному режимі роботи

У режим захоплення шин у мінімальному режимі роботи МП входить по сигналу $HOLD = 1$ не менше, ніж через один період синхронізації після завершення обміну. Запит сприймається, якщо він з'являється не більше ніж за 35 нс до фронту сигналу CLK. На вихід HLDA видається сигнал підтвердження захоплення і всі виводи МП з трьома станами переходять в z-стан. Пристрій обробки при цьому продовжує виконувати команди із черги команд, поки не з'явиться команда обміну по шині або черга команд не спустошиться. Вихід із режиму захоплення шин здійснюється через один чи два такти після зняття сигналу HOLD. При виконанні команди LOCK захоплення шин можливе після виконання команди, що іде за командою LOCK.

2.1.10. Часові діаграми роботи МП у максимальному режимі

У загальному в максимальному режимі робота МП теж проходить по циклах і тактах. Але, в максимальному режимі роботи сигнал ALE на виходах МП відсутній, тому ознакою початку машинного циклу є перехід ліній S2 S1 S0

в будь-який стан, крім стана 1 1 1.

2.1.10.1. Часова діаграма читання пам'яті або портів вводу-виводу в максимальному режимі роботи

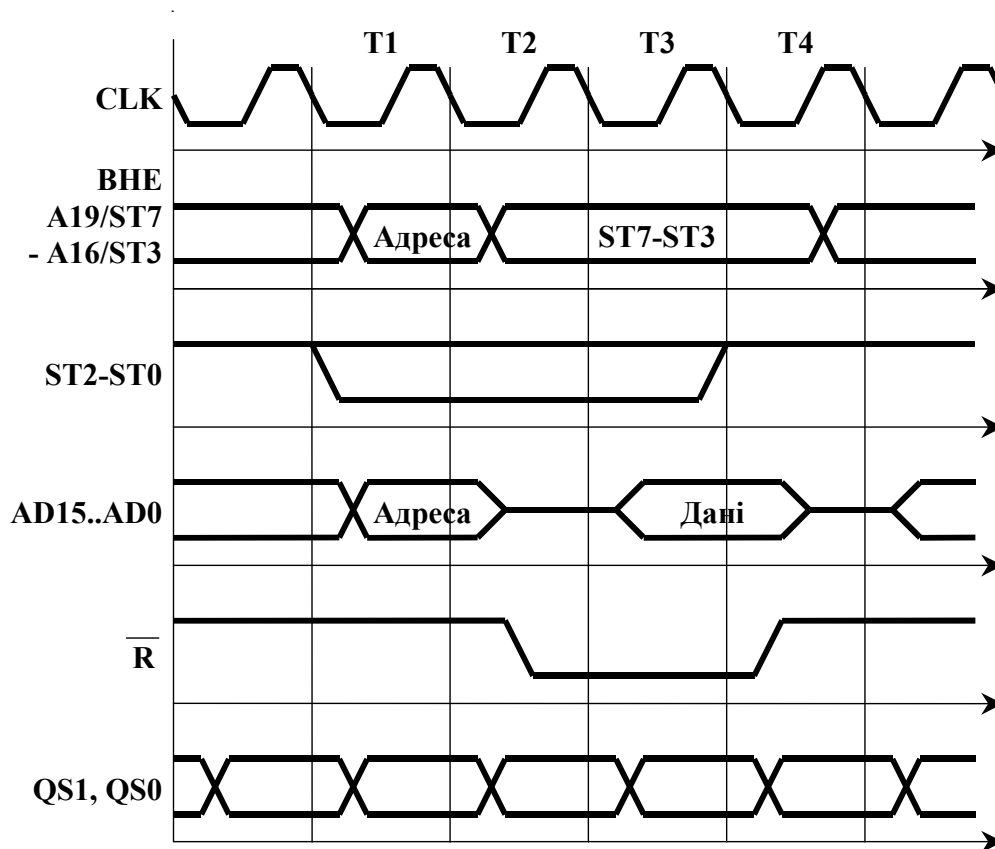


Рис. 34. Часова діаграма роботи МП при читанні в максимальному режимі роботи

Часова діаграма читання пам'яті чи портів вводу-виводу в максимальному режимі роботи зображена на рис. 34.

У кінці машинного такту, що передуює початку машинного циклу, який розглядається, по фронту сигналу CLK на лінії $\overline{S0} - \overline{S2}$ видається код стана, що відповідає обміну, який відбуватиметься. На початку машинного циклу по спаду CLK з певною затримкою відносно нього, на шині AD0-AD15, лініях A16/S4 - A19/S6 та лінії $\overline{BHE} / S7$ видається адреса, по якій відбувається обмін. На початку другого машинного такту лінії AD0-AD15 переводяться на приймання даних, а на лінії A16/S4 - A19/S6 та лінію $\overline{BHE} / S7$ видається слово стана. Після цього на лінії \overline{RD} видається лог. 0, який вказує на приймання

даних. По спаду сигналу CLK на початку третього машинного такту аналізується стан лінії RDY, якщо лінія знаходиться в стані лог. 0, то після такту T2 буде такт очікування TO, в протилежному випадку – T3. У такті TO усі лінії зберігають свій стан і в кінці такту по спаду сигналу CLK знову аналізується стан лінії RDY. Якщо лінія в стані лог. 1, то наступний такт T3, у протилежному випадку знову повторюється такт TO. На початку такту T3 по спаду сигналу CLK лінії $\overline{S0}$ - $\overline{S2}$ переводяться в пасивний стан. А на початку четвертого машинного такту дані, видані на лінії зовнішнім пристроєм, приймаються МП. З деяким запізненням відносно сигналу CLK на початку четвертого машинного такту МП знімає сигнал \overline{RD} – зчитування інформації завершено.

2.1.10.2. Часова діаграма записування у пам'ять чи порти вводу-виводу у максимальному режимі роботи

Часова діаграма запису пам'яті чи порти вводу-виводу в максимальному режимі роботи зображена на рис. 35.

У кінці машинного такту, що передує початку машинного циклу, який розглядається, по фронту сигналу CLK на лінії $\overline{S0}$ - $\overline{S2}$ видається код стана, що відповідає обміну, який відбуватиметься. На початку машинного циклу по спаду CLK, з певною затримкою відносно нього, на шині AD0-AD15, лініях A16/S4 - A19/S6 та лінії \overline{BHE} /S7 – адреса, по якій відбувається обмін. На початку другого машинного такту на лінії AD0-AD15 видаються дані, що записуються, а на лінії A16/S4 - A19/S6 та лінію \overline{BHE} /S7 – слово стану. На початку такту T3 по спаду сигналу CLK лінії $\overline{S0}$ - $\overline{S2}$ переводяться в пасивний стан. А в 4 машинному такті по фронту сигналу CLK дані, видані на лінії AD0-AD15, знімаються – записування інформації завершено. Між тактами T2 та T3 можуть бути добавлені такти очікування (TO), якщо лінія RDY знаходиться в стані лог. 0. Аналіз лінії RDY здійснюється на початку кожного такту такту TO або T3.

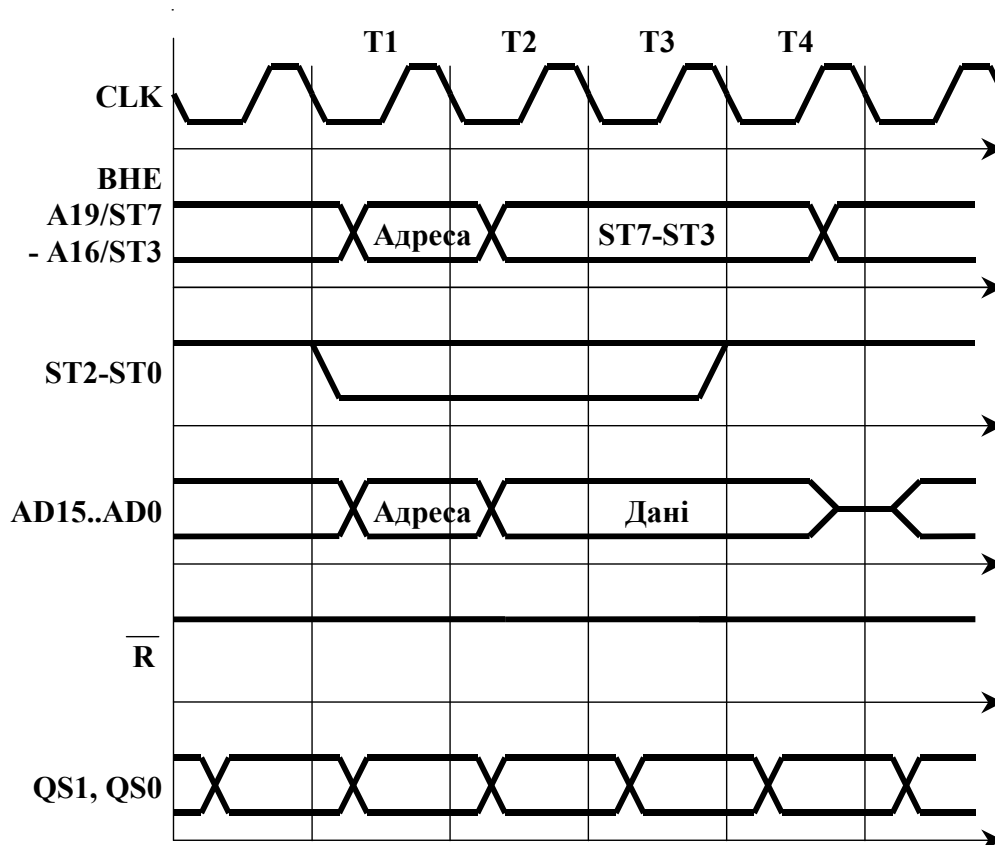


Рис. 35. Часова діаграма роботи МП при записуванні в максимальному режимі

2.1.11. Початкова ініціалізація

Початкове встановлення і запуск мікропроцесора здійснюється за сигналом SR (по високому рівні на час понад чотири періоди сигналу CLK). За сигналом SR робота МП зупиняється і відбувається початкове встановлення регістрів МП.

Регістр ознак	F	F002h
Регістр адреса команди	IP	0000h
Сегментні регістри	CS	FFFFh
	DS	0000h
	SS	0000h
	ES	0000h

2.1.12. Система команд МП

Система команд МП містить 135 команд, поділених на 6 категорій:

1. Команди пересилання даних, призначені для пересилання вмісту джерела на місце приймача. Є 3 групи команд:

1.1. Загального призначення MOV BX,CX; BX=CX.

1.2. Вводу-виводу OUT DX, AL.

1.3. Завантаження логічної адреси LEA BX.

2. Арифметичні команди призначені для виконання основних арифметичних операцій (додавання, віднімання, ділення, множення). Операнди можуть міститися в регістрах пам'яті, результат вміщується на місце одного з операндів.

3. Команди порозрядного опрацювання даних призначені для виконання логічних операцій лінійного та циклічного зсувів на один або n розрядів.

4. Команди опрацювання елементів рядків даних призначені для пересилання, порівняння, записування в пам'ять, завантаження в акумулятор елементів рядків даних. Команди опрацювання рядків дозволяють обробляти строки довжиною 64К (У випадку використання із префіксом REP – командою повторення команди).

5. Команди передавання керування призначені для організації переходів у програмі. Є чотири класи команд:

5.1. Команди безумовного переходу.

5.2. Команди умовного переходу.

5.3. Команди керування циклами.

5.4. Команди переривань.

6. Команди керування МП дозволяють програмно встановити режим роботи МП.

Команди МП забезпечують виконання операцій над одним або двома операндами і результат записується на місце одного з операндів. Залежно від типу команди операнди можуть бути розміщені в: регістрах МП, коді команди, пам'яті, або портах вводу виводу. Безпосередні дані можуть мати розмір байт або слово. Операнди в програмно доступних регістрах можуть мати розмір байта або слова, а для команд множення/ділення – подвійного слова.

Операнди в пам'яті можуть бути розміром в байт, слово, подвійне слово а в регістрах вводу-виводу розміром в слово або байт. Для вказання розміщення

операндів у типових командах пересилання, логічних та арифметичних командах, використовується формат команди.



де code – код команди,

w = 0 – операція 8-розрядна,

w = 1 – 16-розрядна,

mod – режим формування адреси,

reg – вказує регістр обміну 1,

r/m – вказує регістр обміну 2.

2.1.13. Мультипроцесорні системи

Мікропроцесор K1810BM86 може використовуватись у багатопроцесорних системах. У багатопроцесорних системах можливе використання процесорів 2 типів: незалежних процесорів та співпроцесорів. Незалежні процесори виконують свій власний програмний код, співпроцесори – команди, послані основним процесором.

Для забезпечення роботи системного каналу при роботі кількох процесорів K1810BM86 по команді LOCK виробляється сигнал LOCK, що забороняє (низьким рівнем) користуватися системним каналом. Мікропроцесор K1810BM86 може протестувати стан готовності співпроцесора за допомогою команди WAIT, що призупиняє роботу програми до появи сигналу на вході TEST.

До системи команд МП входить команда ESC, що дає можливість отримувати співпроцесору команди та дані з програми, котра виконуються МП. Команда ESC разом із WAIT використовуються для організації паралельних процесів у МП системах.

2.2. Мікропроцесор K1810BM88

Крім мікропроцесора K1810BM86 мікропроцесорний комплект K1810 містить мікропроцесор K1810BM88, який має такі відмінності від базового процесора:

- ◇ шина даних 8-розрядна, AD0-AD7;
- ◇ при виконанні команди, яка вимагає 16-розрядного обміну виконуються 2 цикли обміну, які неможливо розділити;
- ◇ сигнал ВНЕ відсутній, замість нього на лінію видається сигнал стана S0;

17	NM1	CPU	A/D0	16
18	NMD		A/D1	15
			A/D2	14
19	CLK		A/D3	13
21		RST	A/D4	12
22		DRY	A/D5	11
			A/D6	10
23	$\overline{\text{TEST}}$		A/D7	9
			A8	8
33	MN/ $\overline{\text{MX}}$		A9	7
			A10	6
30	HLDA (RQ/GT1)		A11	5
31		HLD (RQ/GT0)		A12
			A13	3
			A14	2
			A15	39
			A16/S3	38
			A17/S4	37
			A18/S5	36
			A19/S6	35
			S0/S7	34
			$\overline{\text{WR}}(\text{LOCK})$	29
			RD	32
			$\overline{\text{M}}/\text{D}(\text{S}2)$	28
			DT/R(S1)	27
			DEN(S0)	26
			ALE(QS0)	25
			INTA(GSD)	24

Рис. 36. Умовне позначення мікропроцесора K1810BM88

- ◇ лінії AD8-AD15 використовуються тільки для передавання адреси, яка утримується на виходах упродовж усього циклу обміну;
- ◇ черга команд складається не з 6, а з 4 регістрів;
- ◇ сигнал M/IO інвертований.

Умовне позначення мікросхеми зображено на рис. 36. Призначення виводів МП близьке до призначення виводів K1810BM86 (таблиця 18).

Таблиця 18. Призначення виводів МП

Вивід	Позначення виводу	Призначення виводу
1	2	3
Виводи МП, призначення яких не змінюється від режиму роботи		
9-16	A/D7 - A/D0	Мультиплексована шина (канал) адреси/даних
39,2-8	A15 - A8	Шина адреси
17	NMI	Немаскований запит переривання. Логічна 1 на вході призводить до переходу МП до опрацювання переривання INT 2.
18	INT	Маскований запит переривання. Лог. 1 на вході призводить до переходу МП до процедури підтвердження переривання у випадку наявності дозволу маскованого переривання
19	CLK	Тактовий сигнал. Початок машинного такту відбувається при переході сигналу в стан лог. 0. Тривалість лог. 1 становить 1/3 від періоду сигналу
21	RST	Початкове встановлення (скидання). Логічна 1 на вході, що триває більше 4 періодів тактового сигналу, призводить до початкового скидання мікропроцесора
22	RDY	Готовність. Логічна 1 на вході свідчить про готовність зовнішнього пристрою до обміну
23	$\overline{\text{TEST}}$	Перевірка. Логічний 0 на вході вказує, що пристрій який перевіряється програмним шляхом, готовий до виконання наступної операції. Перевірка здійснюється за спеціальною командою – WAIT
32	$\overline{\text{R}}$	Зчитування. Логічний 0 на виході вказує, що в цьому циклі обміну відбувається зчитування
33	MN/ $\overline{\text{MX}}$	Мінімальний/максимальний режим. Логічний 0 вказує на роботу МП у максимальному режимі роботи, лог. 1 – в мінімальному.
34	S0/S7	Сигнал слова стана 0/сигнал слова стана 7
35-38	A19/S6, A18/S5, A17/S4, A16/S3	Старші сигнали шини адреси мільтиплексовані з сигналами слова стана
40	+5 В	Напруга живлення

1	2	3
Виводи МП та їх призначення у мінімальному режимі роботи.		
24	\overline{INTA}	Сигнал підтвердження переривання. Активний рівень – лог. 0
25	ALE	Строб адреси. Задній фронт (перехід зі стана лог. 1 в стан лог. 0) сигналу на виході вказує, що на шині даних/адреси знаходиться адреса
26	\overline{DEN}	Дозвіл для зовнішніх буферів на передавання адреси. Логічний нуль на виході вказує, що буферні регістри, які запам'ятали адресу, мають видати її на шину адреси
27	DT/ \overline{R}	Ввід/вивід даних. Лог. 1 на виході вказує, що відбувається приймання даних, лог. 0 – передавання. Сигнал призначений для керування вихідними буферами шини даних
28	\overline{M}/IO	Пам'ять/зовнішній пристрій. Логічний 0 на виході вказує, що відбувається обмін з пам'яттю, логічна 1 – з пристроями вводу-виводу або цикл підтвердження переривання
29	\overline{WR}	Записування. Логічний нуль вказує, що відбувається записування
30	HLDA	Підтвердження захоплення каналу. Логічна 1 вказує, що зовнішній пристрій, що запросив прямий доступ до пам'яті може захопити шини
31	HOLD	Запит захоплення каналу. Логічна 1 вказує, що зовнішній пристрій запросив прямий доступ до пам'яті
Виводи МП, та їх призначення у максимальному режимі		
24,25	QS0, QS1	Сигнал стана черги команд
26 – 28	S0 – S2	Сигнали стана мікропроцесора
29	\overline{LOCK}	Канал зайнятий. Логічний 0 вказує, що відбувається обмін з пам'яттю чи портами вводу - виводу, котрий не можна переривати
30, 31	RQ/GT1, RQ/GT0	Дві однобітні двонаправлені шини запиту та дозволу доступу до каналу

Стан біт S2 S1 S0 вказує тип обміну по шині аналогічно процесору K1810BM86 з поправкою, на те що біти S6, S7 ідентифікують інший процесор. Часові діаграми роботи, режими адресації, система команд, типове ввімкнення МП K1810BM88 ідентичні МП K1810BM86 з поправкою на застосування восьмирозрядної шини даних та інверсного призначення лінії \overline{M}/IO .

2.3. Арифметичний співпроцесор K1810BM87

Для прискорення виконання арифметичних операцій використовується співпроцесор – спеціалізований процесор, що виконує обмежену систему команд. Застосування співпроцесора K1810BM87 додає в систему команд МП K1810BM86 68 нових команд для роботи з числами, серед яких:

- арифметичні команди для чисел із плаваючою точкою;
- команди визначення квадратного кореня та піднесення до ступеня;
- команди обчислення тригонометричних функцій;
- команди обчислення логарифмічних функцій;
- команди обчислення експоненціальних функцій.

Співпроцесор прискорює виконання арифметичних команд більше, ніж у сто разів порівняно з його програмною емуляцією.

2.3.1. Призначення виводів співпроцесора K1810BM87

Співпроцесор розрахований на роботу МП у максимальному режимі роботи, тому більшість його виводів за призначенням та умовним позначенням співпадають із відповідними виводами МП K1810BM86. Призначення виводів – у таблиці 19.

Таблиця 19. Призначення виводів співпроцесора

Вивід	Позначення	Призначення
1	2	3
1,20	GND	Загальний
16- 2,39	A/D0 – A/D15	Мультиплексована шина (канал) адреси / даних
19	CLK	Тактовий сигнал. Початок машинного такту відбувається при переході сигналу в стан лог. 0. Тривалість лог. 1 становить 1/3 періоду сигналу
21	RST	Початкове встановлення (скидання). Логічна 1 на вході, що триває більше 4 періодів тактового сигналу, призводить до початкового скидання мікропроцесора
22	RDY	Готовність. Логічна 1 на вході свідчить про готовність зовнішнього пристрою до обміну
23	BUSY	Зайнято. Логічна 1 вказує на зайнятість співпроцесора
32	INT	Запит переривання. Логічною 1 співпроцесор може сигналізувати МП про помилку при виконанні арифметичних операцій

1	2	3
34	$\overline{BNE} / S7$	Дозвіл передавання старшої половини каналу даних AD8 – AD15 та сигнал стана
35-38	A19/S6 – A16/S3	Старші сигнали шини адреси мільтиплексовані з сигналами слова стана
40	+5 V	Напруга живлення
24,25	QS1, QS0	Входи сигналів стана черги команд
26 – 28	S0 – S2	Сигнали стана мікропроцесора
31 33	RQ/GT0 RQ/GT1	Однобітна лінія запиту (вихід) та дозволу (вхід) доступу до каналу

2.3.2. Робота співпроцесора K1810BM87

Співпроцесор K1810BM87 синхронізується від спільного джерела сигналів CLK і має спільні з МП входи початкового встановлення та готовності. В процесі роботи МП співпроцесор відслідковує сигнали стана МП S0-S2, сигнали черги команд QS1, QS0 та користуючись ними, проглядає та дешифрує інструкції МП. Будь-яка команда, адресована співпроцесору (ESC), опрацьовується МП та співпроцесором одночасно. Якщо команда призводить до обміну з пам'яттю, центральний процесор формує адресу операнда та виконує цикл читання першого слова (для МП K1810BM88 - байту), але ігнорує зчитані дані. Дані та адреса перехоплюються співпроцесором, що фіксує їх у внутрішньому регістрі. Якщо команда вимагає зчитування більш одного слова (для K1810BM88 – байта) співпроцесор по лінії RQ/GT0 запитує керування локальною шиною і отримавши його, продовжує завантаження подальших байтів операнду, послідовно збільшуючи адресу. Якщо команда вимагає записування даних, то зчитані в першому машинному циклі дані ігноруються і за тією ж самою адресою співпроцесор, отримавши керування шиною, здійснює записування даних, у випадку запису багатобайтових даних байти записуються за послідовно зростаючими адресами.

Після обчислювальної інструкції співпроцесора перед наступним використанням центральний процесор повинен перевірити закінчення виконання команди, що здійснюється інструкцією WAIT, за якою процесор очікує надходження по лінії TEST низького рівня від співпроцесора з виходу BUSY. Цей

сигнал встановлюється в стан лог. 1, коли співпроцесор зайнятий і не може прийняти наступну команду. Команду WAIT бажано вводити не відразу після команди ESC, а коли будуть потрібні результати обчислень, тоді центральний процесор і співпроцесор будуть деякий час працювати паралельно. Також бажано чергувати команди, які виконує основний процесор та співпроцесор.

2.4. Мікропроцесорний комплект K1810

2.4.1. Структура мікропроцесорного комплекту

У мікропроцесорному комплекті K1810 виділяють мікросхеми таких типів:

1. Мікропроцесори та співпроцесори:

1.1. K1810BM86 – 16-розрядний мікропроцесор.

1.2. K1810BM88 – 16-розрядний мікропроцесор з 8-розрядною шиною даних.

1.3. K1810BM87 – арифметичний співпроцесор.

1.4. K1810BM89 – співпроцесор вводу-виводу.

2. Мікросхеми загального призначення (системоутворюючі):

2.1. Генератор тактових імпульсів K1810ГФ84.

2.2. Системний контролер K1810ВГ88.

2.3. Арбітр системної шини K1810ВБ89.

2.4. Регістр з інверсним входом K1810ИР83.

2.5. Регістр з прямим входом K1810ИР82.

2.6. Буферний формувач без інверсії K1810ВА86.

2.7. Буферний формувач з інверсією K1810ВА86.

3. Універсальні інтерфейсні ВІС:

3.1. K1810ВТ37 – контролер ПДП.

3.2. K1810ВН39 – контролер переривань.

3.3. K1810ВІ54 – лічильник таймер.

4. Мікросхеми контролери пристроїв та інтерфейсів:

4.1. K1810ВГ72 – контролер накопичувача на гнучких дисках.

2.4.2. Генератор тактових сигналів К1810ГФ24

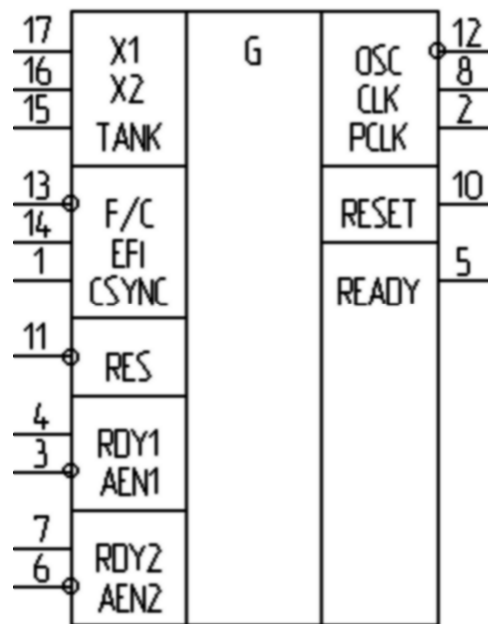


Рис. 37. Умовне позначення мікросхеми К1810ГФ24

Генератор тактових імпульсів (ГТІ) К1810ГФ84 призначений для керування ЦП К1810ВМ86 та периферійними пристроями, а також для синхронізації сигналів READY із тактовими сигналами ЦП і сигналів інтерфейсної шини Multibus. Умовне позначення мікросхеми зображено на рис. 37, призначення виводів наведено в таблиці 20.

Генератор тактових імпульсів містить схеми формування тактових імпульсів (OSC, CLK, PCLK), формування сигналу скидання (RST) та формування сигналу готовності (RDY). Схема формування тактових імпульсів виробляє сигнали: CLK – тактової частоти для ЦП К1810ВМ86, PCLK – тактової частоти для керування периферійними ВІС, OSC – сигнал тактової частоти задаючого генератора, необхідні для керування та синхронізації пристроїв, що входять в МП систему. Частоти сигналів синхронізації пов'язані співвідношенням

$$F_{OSC} = 3 F_{CLK} = 6 F_{PCLK}$$

у режимі внутрішнього генератора та

$$F_{EFI} = 3 F_{CLK} = 6 F_{PCLK}$$

у режимі зовнішнього генератора. В загальному джерело сигналу

вибирається виводом F/C. Якщо цей сигнал в стані лог. 0, то сигнал формується внутрішнім генератором (SGN), якщо в стані лог. 1, то сигнал синхронізації сприймається по входу EFI.

Таблиця 20. Призначення виводів K1810ГФ24

Вивід	Позначення	Тип	Призначення виводу
1	CSYNC	Вхід	Дозвіл запуску синхронізації: 1 - на виходах PCLK та CLK; 0 - на виходах PCLK та CLK –сигнал синхронізації
2	PCLK	Вихід	Меандр із частотою 1/2 CLK з перемиканням по задньому фронту CLK
3	$\overline{\text{AEN1}}$	Вхід	Вибірка готовності шини лог 1 на вході забороняє сигнал готовності RDY1
4	RDY1	Вхід	Сигнал готовності шини 1
5	RDY	Вихід	Готовність
6	RDY2	Вхід	Сигнал готовності шини 2
7	$\overline{\text{AEN2}}$	Вхід	Адреса готовності 2
8	CLK	Вихід	Тактовий сигнал МП
9	GND	–	Загальний
10	RST	Вихід	Скид МП (низьким рівнем)
11	$\overline{\text{RES}}$	Вхід	Вхід сигналу скиду
12	OSC	Вихід	Вихід мультивібратора
13	$\overline{\text{F/C}}$	Вхід	Вибір джерела частоти
14	EFI	Вхід	Зовнішня частота
15	TANK	–	Вивід під'єднання зовнішнього LC резонатора
16	X1	–	Виводи під'єднання зовнішнього кварцового резонатора
17	X2	–	
18	Ucc	–	Напруга живлення

Схема формування скиду RST має на вході тригер Шмітта, а на виході - D-тригер, який формує фронт сигналу RST по спаду CLK. Зазвичай, на вхід RES під'єднується RC-ланка, що забезпечує формування сигналу при ввмиканні джерела живлення. Схема формування тактових імпульсів має вхід синхронізації (CSYNC), за допомогою якого можлива синхронізація кількох ГТІ, що входять у систему. Сигнал READY фіксується в мікросхемі по спаду сигналу CLK та формується за законом

$$\text{READY} = \overline{\text{AEN1}} * \text{RDY1} + \text{RDY2} * \overline{\text{AEN1}} .$$

2.4.3. Контролер системної шини K1810BG88



Рис. 38. Структура контролера системної шини K1810BG88

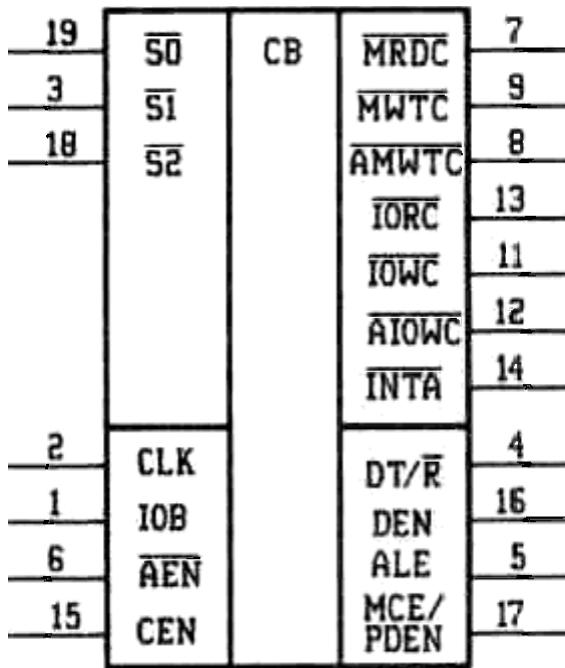


Рис. 39. Умовне графічне зображення K1810BG88

Контролер системної шини K1810BG88 призначений для роботи в складі мікропроцесорної системи на базі МП K1810BM86. Залежно від стану МП контролер керує обміном даних між локальною шиною (ЛШ) процесора та системною шиною (СШ) за наявності доступу до керування шинами МП, а також між локальною шиною та шиною вводу-виводу або резидентною шиною.

Контролер шини (КШ) синхронізується тактовим

генератором МП і здійснює керування шинними формувачами, регістрами, фіксаторами адреси, пристроями вводу-виводу та пам'яттю. Структурна схема контролера шини зображена на рис. 38, а його умовне графічне позначення – на рис. 39.

Призначення виводів мікросхеми наведено в таблиці 21.

Таблиця 21. Призначення виводів мікросхеми

Позначення	Номер	Призначення
1	2	3
S0, S1, S2	19, 3, 18	Входи сигналів стана МП К1810ВМ86/ВМ87/ВМ89. Контролер шини декодує ці сигнали та формує командні сигнали й сигнали керування
CLK	2	Вхідний сигнал генератора тактових імпульсів ГФ84, що синхронізує роботу контролера шини
AEN	6	Дозвіл на видавання командних сигналів контролера. Логічний нуль дозволяє видавання вихідних сигналів керування обміном із пам'яттю та портами вводу-виводу. В режимі роботи з шиною вводу-виводу (IOB = 1) сигнал AEN не впливає на видавання сигналів, що керують вводом-виводом
CEN	15	Дозвіл на видавання командних сигналів та сигналів керування DEN, PDEN. При CEN = 0 вихідні командні сигнали перебуває в пасивному стані (сигнал високого рівня)
IOB	1	Сигнал керування режимом роботи контролера. IOB = 1 задає режим роботи з шиною вводу-виводу, IOB = 0 задає режим роботи з системною шиною
MRDC	7	Сигнал читання з пам'яті. Логічний 0 вказує на читання з пам'яті
MWTC	9	Сигнал записування в пам'ять. Логічний 0 вказує на записування в пам'ять. Сигнал встановлюється після видавання даних, що записуються
AMWC	8	Випереджуючий сигнал записування в пам'ять. Логічний 0 вказує на запис в пам'ять. Сигнал встановлюється до видавання даних, що записуються.
IORC	13	Сигнал читання з портів вводу-виводу. Лог. 0 вказує на читання
IOWC	11	Сигнал записування в порти вводу-виводу. Лог. 0 вказує на записування
AIOWC	12	Сигнал записування в порти вводу-виводу. Логічний 0 встановлюється до видавання даних
MCE/ PDEN	17	Сигнал керування, що виконує дві функції залежно від режиму: У режимі роботи з шиною вводу-виводу (IOB = 1) використовується як сигнал PDEN – сигнал керування станом «ввімкнено» шинних формувачів, увімкнених між локальною шиною та шиною вводу-виводу. У режимі роботи з системною шиною (IOB = 0) використовується як сигнал MCE – сигнал керування зчитуванням номера веденого контролера переривань, що підлягає обслуговуванню

1	2	3
DT/R	4	Сигнал керування роботою шинних формувачів. DT/R= 1 перемикає шинні формувачі на передавання даних. DT/R = 0 переключає шинні формувачі на примання
DEN	16	Сигнал дозволу роботи шинних формувачів. Лог. 0 дозволяє їх роботу
ALE	5	Строб адреси. Задній фронт сигналу вказує на дійсність адреси
5V	20	Напруга живлення
0V	10	Загальний

Основною інформацією для формування командних сигналів і сигналів керування контролером системної шини є код стана МП, що надходить на входи S0, S1, S2. Дешифратор стана МП декодує код у відповідності з таблицею 22.

Таблиця 22. Декодування коду стана

Код стана S2 S1 S0	Стан МП	Командний сигнал контролера ВГ88
1	2	3
000	Підтвердження переривань	\overline{INTA}
001	Ввід з ПБВ	\overline{IORC}
010	Вивід з ПБВ	$\overline{IOWC}, \overline{AIOWC}$
011	Зупинка	-
100	Вибірка команди	\overline{MRDC}
101	Читання з пам'яті	\overline{MRDC}
110	Записування в пам'ять	$\overline{MWTC}, \overline{AMWC}$
111	Пасивний стан	-

2.4.3.1. Режими роботи мікросхеми

Вихідні командні сигнали та сигнали керування виробляються контролером під керуванням вхідних сигналів IOB, SEN та AEN, що визначають режим роботи контролера, активність командних сигналів і можливість доступу до системної шини. Контролер працює в двох режимах: із системною шиною та з шиною вводу-виводу.

Режим роботи з системною шиною встановлюється шляхом подавання на

вхід IOB напруги низького рівня ($IOB = 0$). У цьому режимі контролер формує командні сигнали та сигнали ALE, DEN, DT/R керування фіксаторами адреси й шинними формувачами. На входах формуються постійні значення $AEN = 0$ та $CEN = 1$, що дозволяють видавати командні сигнали та сигнали керування. На виході MCE/PDEN при такому ввімкненні формується сигнал MCE, що використовується в системах із каскадуванням контролерів переривань для визначення моменту передавання номера веденого контролера, який запитав переривання. Сигнал ALE служить для визначення моменту фіксації адреси, встановленої на локальній шині у фіксаторах IP82. Вихідні сигнали DT/R та DEN використовуються для керування роботою шинних формувачів. Сигнал DEN має високий активний рівень, тому його слід інвертувати перед подаванням на вхід OE шинних формувачів. На рис. 8 сигнал DEN використовується сумісно з вихідним сигналом SP/EN контролера переривань, що забороняє роботу шинних формувачів, коли дані передаються з контролера переривань у МП. Режим роботи з СШ застосовується в багатопроцесорних системах, коли кілька МП вимагають доступ до пристроїв вводу-виводу та пам'яті, що під'єднані до системної шини.

В цьому випадку кожен МП обслуговується своїм контролером і доступ до системної шини має той із них, контролер якого отримує сигнал AEN дозволу доступу від арбітра шин. У випадку надання доступу до системної шини I-41 мікропроцесору контролер шини формує командні сигнали через 115 нс після надходження сигналу AEN, а також сигнали ALE, DEN, DT/R керування фіксаторами адреси та шинними формувачами.

Поряд із входом AEN для керування видаванням командних сигналів, а також сигналів керування використовується вхід CEN. Зазвичай він застосовується у випадках, коли МП має доступ до двох шин: системної та резидентної, причому кожній зі них відповідає свій адресний простір. Для формування сигналу CEN у таких випадках використовується дешифратор адреси, що розділяє адресний простір між системною (I-41), та резидентною шинами. Вихід цього дешифратора (прямий та інверсний) служить в якості

сигналу CEN.

Режим роботи з шиною вводу-виводу встановлюється шляхом формування сигналу IOB=1. Цей режим використовується у випадках, коли контролер керує доступом до двох шин: резидентної вводу-виводу та системної. Командні сигнали IORC, IOWC, AIOWC та INTA в цьому режимі завжди дозволені, тобто їх поява не залежить від вхідного сигналу AEN.

Як тільки мікропроцесор починає виконувати команду вводу-виводу, формується відповідний командний сигнал, а також сигнали PDEN та DT/R керування моментом і напрямком передавання даних по резидентній шині вводу-виводу. Системна шина в цьому випадку може працювати тільки з пам'яттю (або з пристроями вводу-виводу, відображеними на пам'ять), а мікропроцесор отримує доступ до системної шини (И -41) тільки по сигналу AEN від арбітра шини.

Командні сигнали $\overline{\text{IORC}}$, $\overline{\text{IOWC}}$, $\overline{\text{AIOWC}}$ та $\overline{\text{INTA}}$ для роботи з СШ не використовуються. У відповідності з виконуваною мікропроцесором командою, що вимагає звертання до пам'яті, контролер формує потрібний командний сигнал $\overline{\text{MRDC}}$ або $\overline{\text{MWTC}}$, $\overline{\text{AMWTC}}$, а також сигнали керування моментами фіксації адреси ALE, передавання даних DEN та ігнорування передавання даних по системній шині DT/R.

Вихідний сигнал MCE разом із сигналом $\overline{\text{INTA}}$ використовується в циклі підтвердження переривання в системах із каскадованими контролерами переривань. Сигнал MCE формується в режимі роботи з системною шиною (IOB≠0). Коли МП відповідає на запит переривання, він виставляє код стана S2S1S0 = 000, по якому системний контролер формує два від'ємних імпульси на виході INTA. У відповідь на перший імпульс по шині адрес ША та шині даних ШД не передається ніякої інформації. Перед початком іншого імпульсу сигнал MCE змушує ведучий контролер переривань видати на локальну шину ЛШ процесора код веденого контролера, що запитав переривання. Цей код по сигналу ALE записується у фіксатор адреси.

По фронту іншого імпульсу \overline{INTA} ведений контролер, який запитає переривання, виставляє вектор переривання на системну шину даних, звідки він зчитується ЦП.

У системах, що використовують один контролер переривань, сигнал MCE не використовується. В цьому випадку контролер переривань по іншому імпульсу \overline{INTA} виставляє вектор на локальну шину (ЛШ) процесора.

Вихідний сигнал ALE формується в кожному машинному циклі та служить для записування біжучої адреси у фіксатори адреси. Цей сигнал використовується також для фіксації коду стана S2 S1 S0 мікропроцесора у фіксаторах для розпізнавання стана зупинки, а також коду веденого контролера переривань у системах із програмованим контролером переривань.

Вхідний сигнал керування CEN діє як визначник можливості використання командних сигналів, сформованих контролером системної шини. При CEN = 0 командні сигнали утримуються в пасивному стані. Така особливість використовується для поділу адресного простору та уникнення адресних конфліктів між зовнішніми пристроями, під'єднаними до системної та резидентної шин.

2.5. Організація мікропроцесорної системи керування на основі мікропроцесорного комплекту K1810

Мікропроцесор K1810BM86 можна використовувати як у простих мікропроцесорних схемах (мінімальний режим роботи), так і в мультипроцесорних схемах (максимальний режим роботи мікропроцесора).

2.5.1. Мінімальний режим

Мінімальний режим, за якого вивід MN/MX МП під'єднується до шини +5В, орієнтований на застосування МП в однопроцесорних системах, що містять невелику кількість пристроїв. У мініальному режимі роботи (рис. 40) МП K1810 BM86 використовує для керування обміном даних по шині власні лінії керування:

- ◇ WR, RD – відповідно сигнали записування та зчитування даних з пам'яті або ПБВ;
- ◇ M/IO – вибір пам'яті або ПБВ;
- ◇ INTA – строб зчитування вектора переривання;
- ◇ DT/R – лінія керування формувачами шини даних;
- ◇ DEN – дозвіл для обміну даними (активізує лінію OE шинних формувачів);
- ◇ ALE – строб записування адреси в адресні регістри адресної шини.

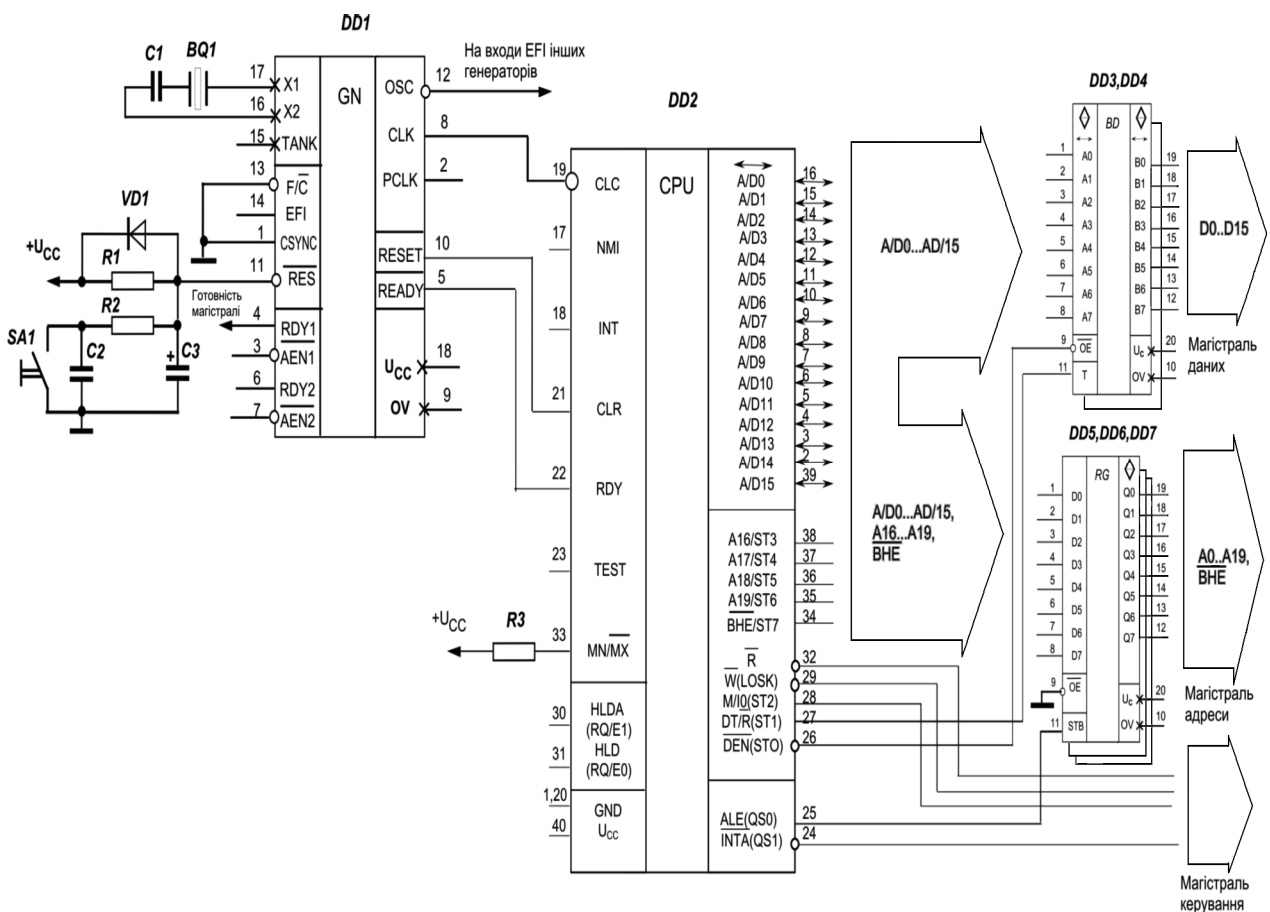


Рис. 40. Структура системної магістралі на базі МП К1810ВМ86 у мінімальному режимі роботи

До складу магістралі в мінімальному режимі роботи МП входять:

- ◇ тактовий генератор КР1810ГФ84 (мікросхема DD1);
- ◇ мікропроцесор КР1810ВМ86 (DD2);
- ◇ шинні формувачі КР580ВА86 (DD3, DD4);
- ◇ адресні регістри КР580ІР82 (DD5-DD7).

2.5.2. Максимальний режим

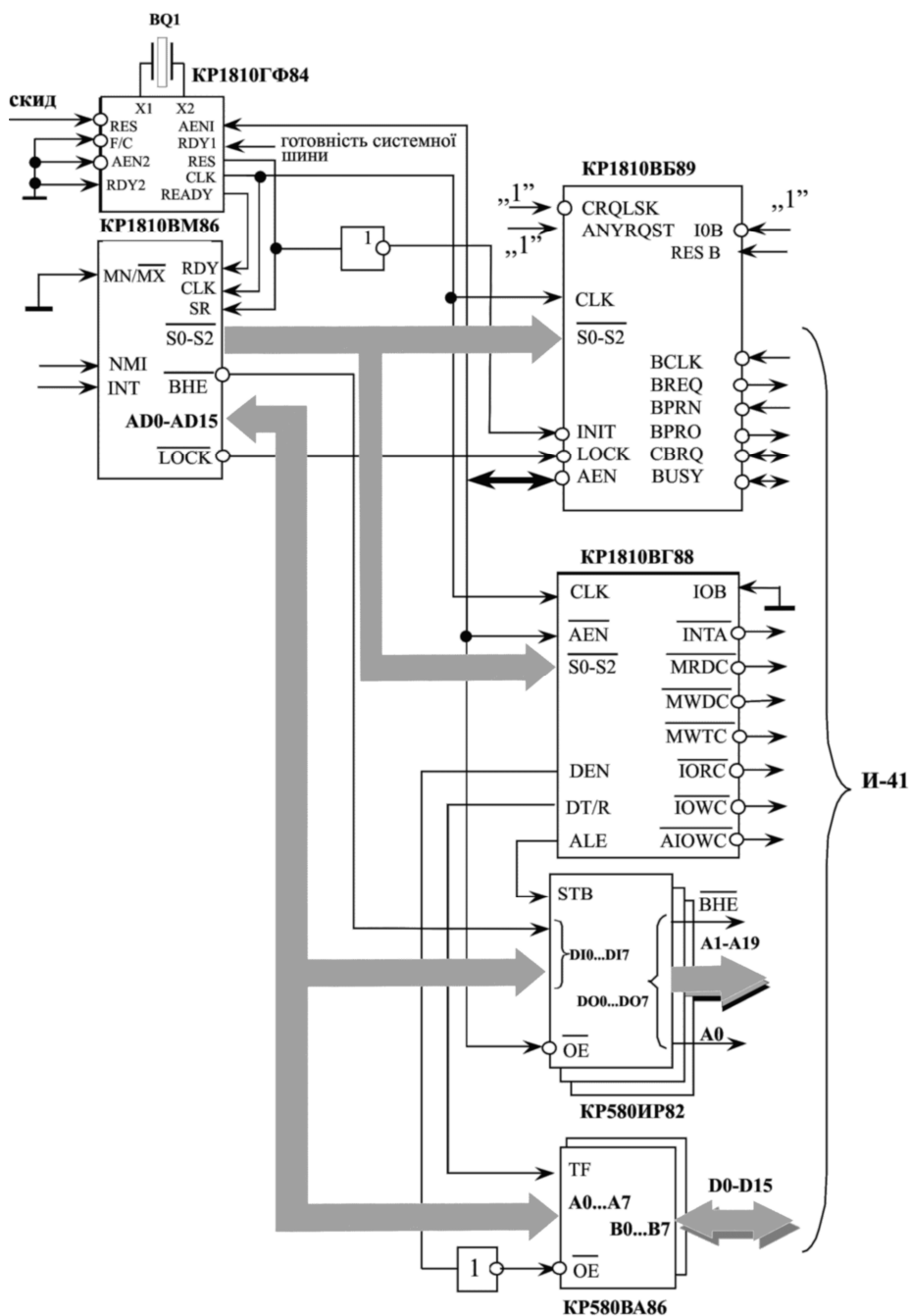


Рис. 41. МП система в максимальному режимі з однією системною магістраллю

У максимальному режимі вивід MN/MX МП під'єднаний до загальної шини. Режим орієнтований на застосування в складних одно- та мультипроцесорних системах. У такій МП-системі функції керування магістраллю бере на себе системний контролер шини KR1810BG88. Він дешифрує три сигнали стана ST0-ST2, що надходять від МП, і видає

розширений набір сигналів керування: IORC, IOWC, MRDC, MWTC, AIOWC, AMWC та INTA, а також сигнали DT/R, DEN, ALE, MCE/PDEN, які керують обміном. Мікропроцесор під'єднується до спільної мультипроцесорної магістралі за допомогою арбітра KP1810BB89.

МП-система (рис. 41), призначена для роботи з однією системною магістраллю. МП KM1810BM86 працює в максимальному режимі (MN/MX = 0) і до його локальної магістралі безпосередньо під'єднані арбітр магістралі KP1810BB59, контролер KP1810BG88, буферні регістри KP580IP82 та шинні формувачі KP580BA86. До системної магістралі під'єднані розділені ресурси (пам'ять та зовнішні пристрої), а також інші мікропроцесори.

Під'єднання мікросхем пам'яті, послідовних і паралельних інтерфейсів, пристроїв вводу-виводу та інш. до системної шини в схемах (рис. 40 та рис. 41) здійснюється аналогічно до схем, побудованих на МП KP580BM80A.

2.5.3. Арбітр магістралі KP1810BB89

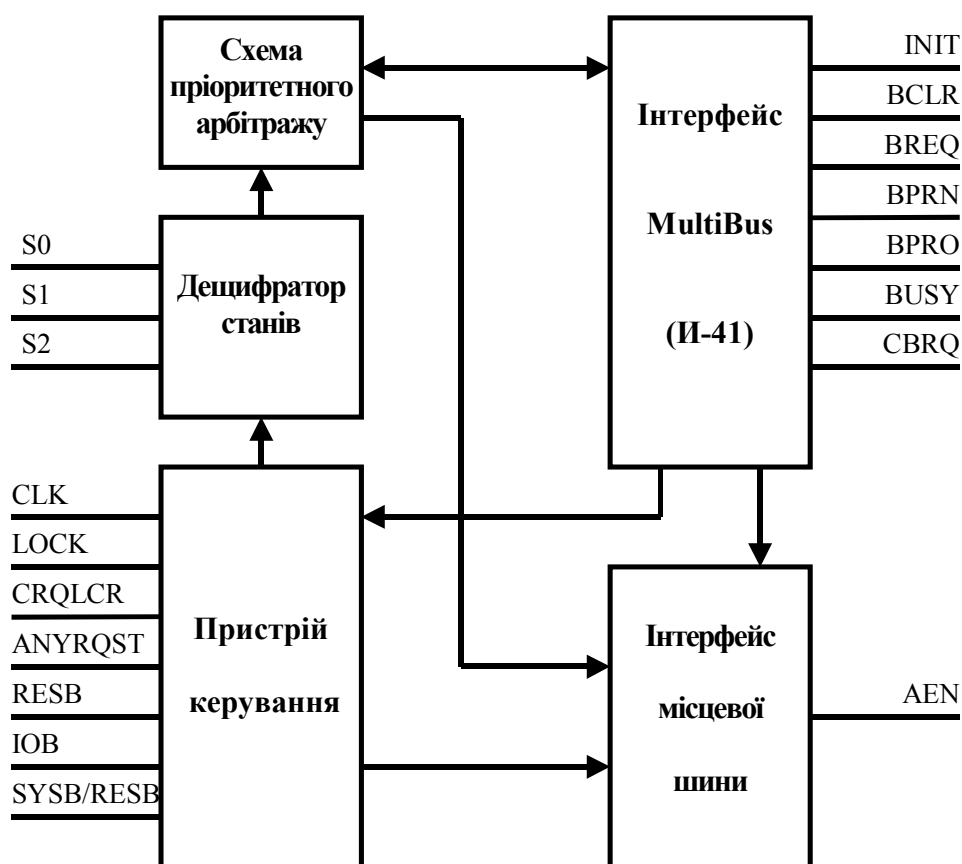


Рис. 42. Структурна схема АШ ВБ89

Арбітр шини забезпечує синхронізацію доступу кількох ведучих пристроїв

до системної шини відповідно до визначеного пріоритету доступу. Арбітр шин K1810B89 використовується в багатопроцесорних системах як пристрій, що здійснює синхронізацію доступу багатьох мікропроцесорів до системної шини.

Структурна схема арбітра шин (АШ) зображена на рис. 42. Дешифратор стана залежно від коду стана МП (VM86, VM87 або VM89) запускає схему пріоритетного арбітражу, інтерфейсу I-41 та місцевої шини на виконання дій по захопленню або звільненню системної шини. Схема пріоритетного арбітражу проводить арбітраж мікропроцесорів, що запитують керування системною шиною, та по тактовому сигналу мікропроцесора, що займає шину, здійснює дії по її звільненню. Інтерфейс I-41(Multibus) здійснює процедуру взаємодії арбітрів шини багатопроцесорної системи та синхронізує дії по захопленню системної шини.

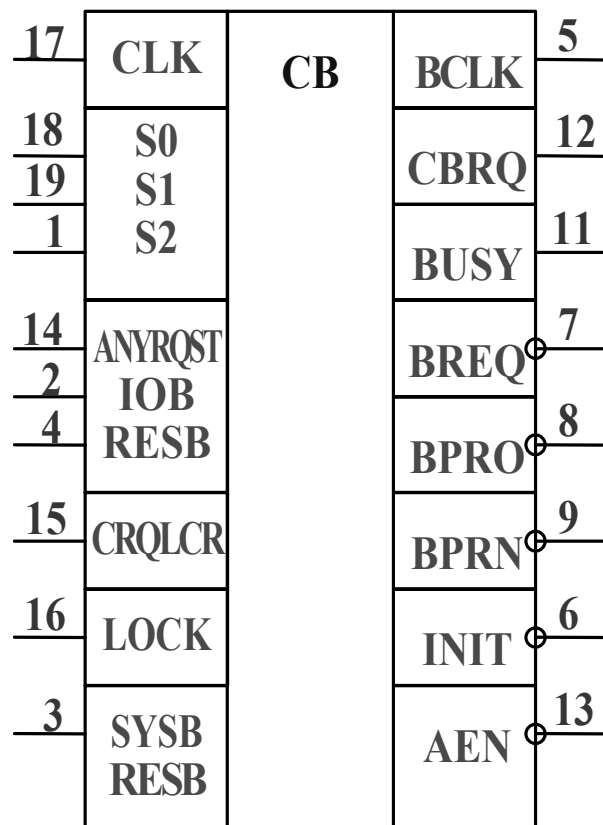


Рис. 43. Умовне графічне позначення АШ K1810B89

Інтерфейс локальної шини формує сигнал дозволу доступу до системної шини для шинного інтерфейсу МП. Пристрій керування здійснює синхронізацію та керування режимами роботи арбітра шин з боку мікропроцесора.

2.5.4. Призначення виводів мікросхеми

Призначення виводів мікросхеми наведено в таблиці 23.

Таблиця 23. Призначення виводів мікросхеми

Вивід	Номер	Призначення
1	2	3
S2, S1, S0	1,19,18	Входи сигналів стана мікропроцесора
CLK	17	Вхід сигналу синхронізації МП
LOCK	16	Сигнал заборони звільнення системної шини. Активний стан - лог. 0.
CRQLCK	15	Лог. 0 забороняє передавання шини арбітрам з нижчим пріоритетом
ANYRQST	14	Дозвіл доступу по текучому запиту. Вхід дозволу звільнення системної шини при будь-якому запиті. Активний сигнал ANYRQST = 1 дозволяє арбітру звільнити системну шину при запиті керування шиною арбітром з нижчим пріоритетом. Запитуючий арбітр здійснює захоплення шини в цьому випадку так, якщо б він був арбітром з вищим пріоритетом. Активний сигнал на вході ANYRQST в поєднанні з активним сигналом на вході CBRQ = 0 змушує арбітра звільняти системну шину після кожного циклу передавання
RESB	4	Режим роботи резидентної шини. Забезпечує роботу арбітра шин у багатопроцесорних системах, які мають як системну, так і резидентну шину. При вхідному сигналі RESB = 1 системна шина запитується або звільняється як функція вхідного сигналу SYSB/RESB. При RESB = 0 вхідний сигнал SYSB/RESB не впливає на роботу арбітра
SYSB/RESB	3	Вхід дозволу доступу до системної шини для арбітра, встановленого в режим роботи з системною та резидентною шиною. Сигнал SYSB/RESB зазвичай формується додатковим дешифратором, під'єднаним до резидентної шини адреси. Зміна значення сигналу SYSB/RESB дозволена в період часу від четвертого такту біжучого циклу до початку іншого такту наступного циклу МП. У період від іншого такту до початку четвертого такту циклу МП зміна сигналу SYSB/RESB призводить до невизначеності по захопленню та звільненню системної шини. При SYSB/RESB = 1 арбітр запитує системну шину, при SYSB/RESB = 0 він звільняє системну шину іншому арбітру, що запитав керування нею
BREQ	7	Сигнал запиту шини. Використовується у схемах паралельного та циклічного дозволу пріоритету для запиту керування системною шиною

1	2	3
IOB	2	Вхід вибору режиму роботи з периферійною шиною вводу-виводу. Забезпечує роботу арбітра шини в багатопроцесорних системах, що мають як системну, так і периферійну шину вводу-виводу. При IOB = 0 арбітр запитує та звільняє системну шину залежно від значення сигналу S2: при виконанні мікропроцесором команд вводу-виводу (S2 = 0) арбітр звільняє системну шину, а при виконанні команд звертання до пам'яті (S2=1) він захоплює керування системною шиною
AEN	13	Вихід дозволу доступу до системної шини. Сигнал AEN = 0 переводить фіксатори адреси, шинні формувачі та системний контролер МП в активний стан і дозволяє йому керувати системною шиною. Сигнал AEN = 1 знімає сигнал готовності з входу МП і переводить виходи фіксаторів адреси та шинних формувачів у високоомний стан
INIT	6	Початкове встановлення. Використовується для початкового встановлення всіх шинних арбітрів, пов'язаних із системною шиною. Після початкового встановлення ніякий арбітр не має доступу до керування системною шиною. Тривалість імпульсу INIT повинна бути не менше суми трьох періодів CLK та трьох періодів BCLK
CBRQ	12	Вхід/вихід загального запиту шини. В якості вхідного сигнал CBRQ = 0 повідомляє арбітру шини, що в системі є інші арбітри з нижчим пріоритетом, які запитують керування системною шиною. Сигнал CBRQ=1 повідомляє, що таких арбітрів немає і арбітр може продовжувати керування, що виключає процедуру захоплення шини. В якості вихідного сигнал CBRQ = 0 видається арбітром, який у даний момент не керує шиною, але хоче отримати доступ до керування. Виводи CBRQ всіх арбітрів системи, аналогічно виводам BUSY, мають вихід з відкритим колектором та об'єднані по лінії CBRQ
BPRO	8	Вихід пріоритетного дозволу доступу до шини, що використовується в системах зі схемою послідовного арбітражу, в яких вихід BPRO з'єднаний з входом BPRN арбітра з нижчим пріоритетом. Сигнал BPRO = 0 видається арбітром, що отримав сигнал BPRN = 0, але не запитав керування шиною, та подається на вхід BPRN наступного арбітра з нижчим пріоритетом

1	2	3
BCLK	5	Вхід синхронізації системної шини. На нього подаються тактові імпульси від шинного тактового генератора, що здійснює синхронізацію дій усіх арбітрів шин із захоплення системної шини. В якості шинного тактового генератора може служити генератор тактових імпульсів одного з мікропроцесорів багатопроцесорної системи
BUSY	11	Вхід/вихід зайнятості шини. Сигнал $BUSY = 0$ видається арбітром, що отримав керування системною шиною та служить для вказання іншим арбітрам, що системна шина зайнята. Виводи всіх шинних арбітрів системи мають вихід із відкритим колектором та об'єднані по лінії BUSY. Коли який-небудь арбітр захоплює керування шиною, він видає сигнал $BUSY = 0$ та забороняє доступ до керування шиною іншим арбітрам системи. Після закінчення роботи з шиною арбітр видає сигнал $BUSY=1$, дозволяючи іншим арбітрам захопити шину.
BPRN	9	Вхід пріоритетного дозволу доступу до шини. Сигнал $BPRN = 0$ вказує арбітру шини, що в системі немає іншого арбітра з вищим пріоритетом, який би запитав керування шиною. Сигнал $BPRN = 1$ вказує арбітру, що він втратив пріоритет відносно іншого арбітра. В подальшому поява сигналу $BPRN = 0$ дозволяє арбітру знову захопити системну шину при переході сигналу BCLK від високого рівня до низького.
5V	20	Напруга живлення
0V	10	Загальний

Арбітр магістралі KP1810B89 вводить і контролює сигнали стану S0 - S2 МП, щоб визначити, коли запитувати та звільняти шину. З метою виявлення початку циклу шини контролер KP1810B88 також контролює стан МП. Контролер використовується в режимі керування магістраллю (IOB=0). Коли арбітру KP1810B89 дозволено доступ до магістралі, він формує сигнали $BUSY=0$ та $AEN=0$. Сигнал AEN дозволяє передачу адреси та сигналів керування. Сигнал ALE контролера KP1810B88 фіксує адресу на регістрах KP580IP82. Сигналом $DEN=1$ дозволяється передавання даних через шинні формувачі KP580BA86. Після завершення передавання даних адресована комірка по лінії готовності RDY1 повертає сигнал підтвердження, по якому тактовий генератор KP1810GF84 формує сигнал READY. При отриманні цього

сигналу МП виходить із стану очікування і завершує поточний цикл шини.

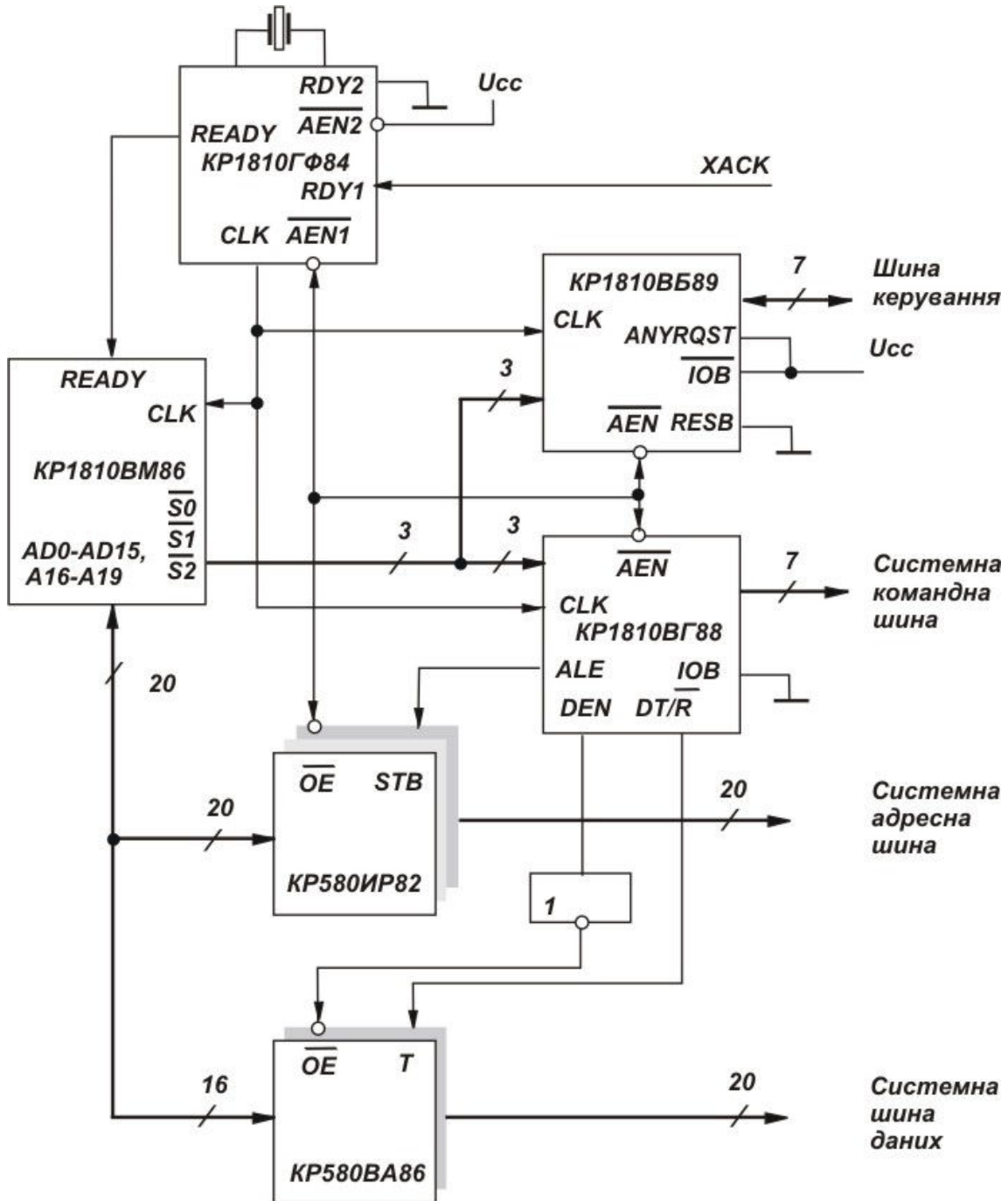


Рис. 44. Арбітр шини в конфігурації з системною шиною

Сигнал заборони звільнення системної магістралі LOCK з виходу МП безпосередньо надходить на арбітр магістралі. Сигнал LOCK=0 не дозволяє арбітру звільняти магістраль. Якщо ж арбітр у даний момент не керує магістраллю, він формує сигнал AEN=1, котрий переводить виходи контролера шини KP1810ВГ88 і виходи адресних регістрів KP580ИР82 у високоімпедансний стан. Так як сигнал AEN керує і виходом DEN контролера

шини, то забороняється передавання даних через шинні формувачі КР580ВА86.

Крім цього, сигнал AEN=1 арбітра КР1810ВБ89 забороняє тактовому генератору КР1810ГФ84 видавати сигнал готовності READY в МП.

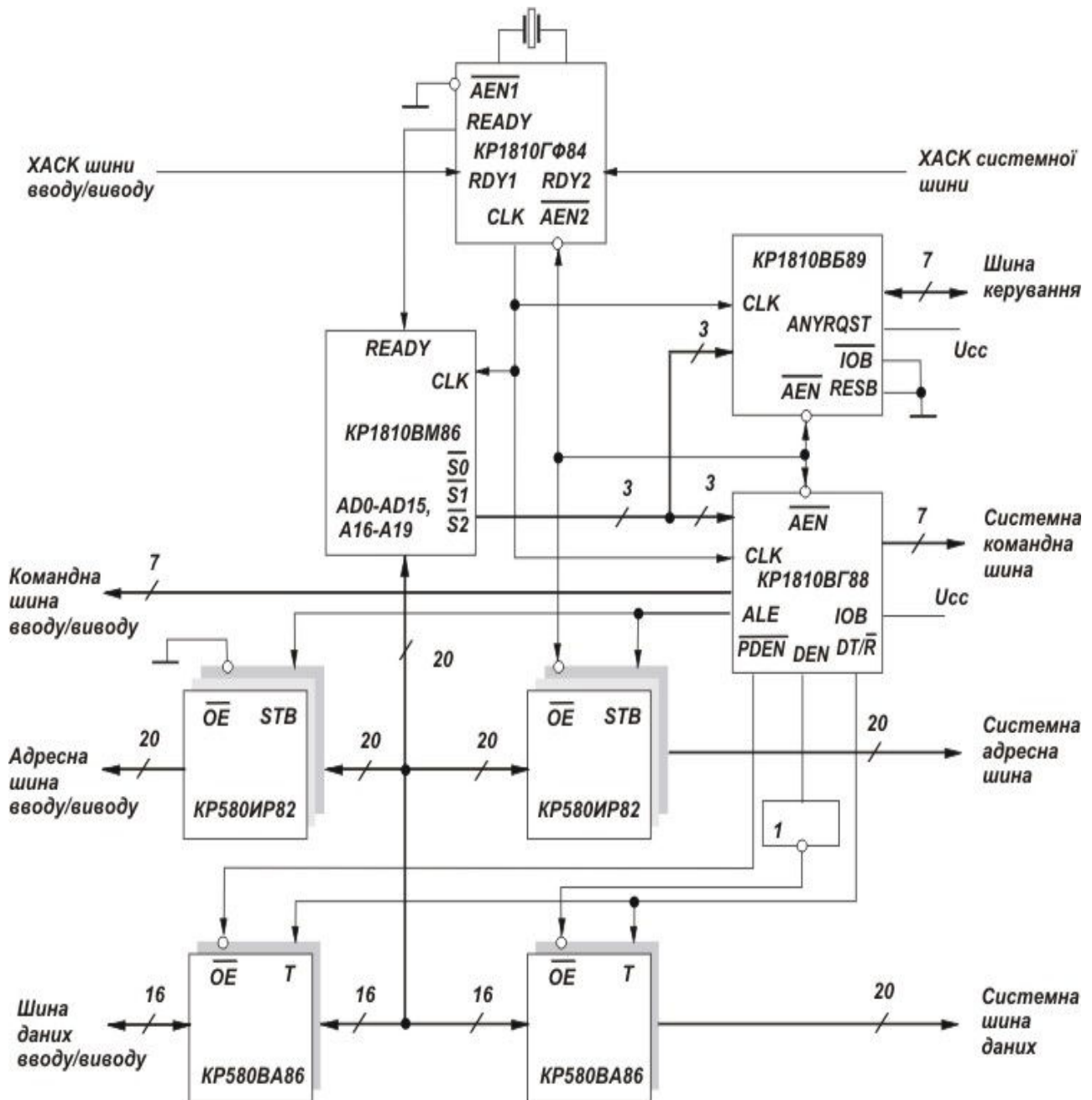


Рис. 45. Арбітр шини в конфігурації з системною та периферійною шинами вводу-виводу

Приклад побудови мікропроцесорної системи на базі МП-комплекта КР1810 у різних конфігураціях ввімкнення арбітра шини зображені на рис. 44, 45 та 46.

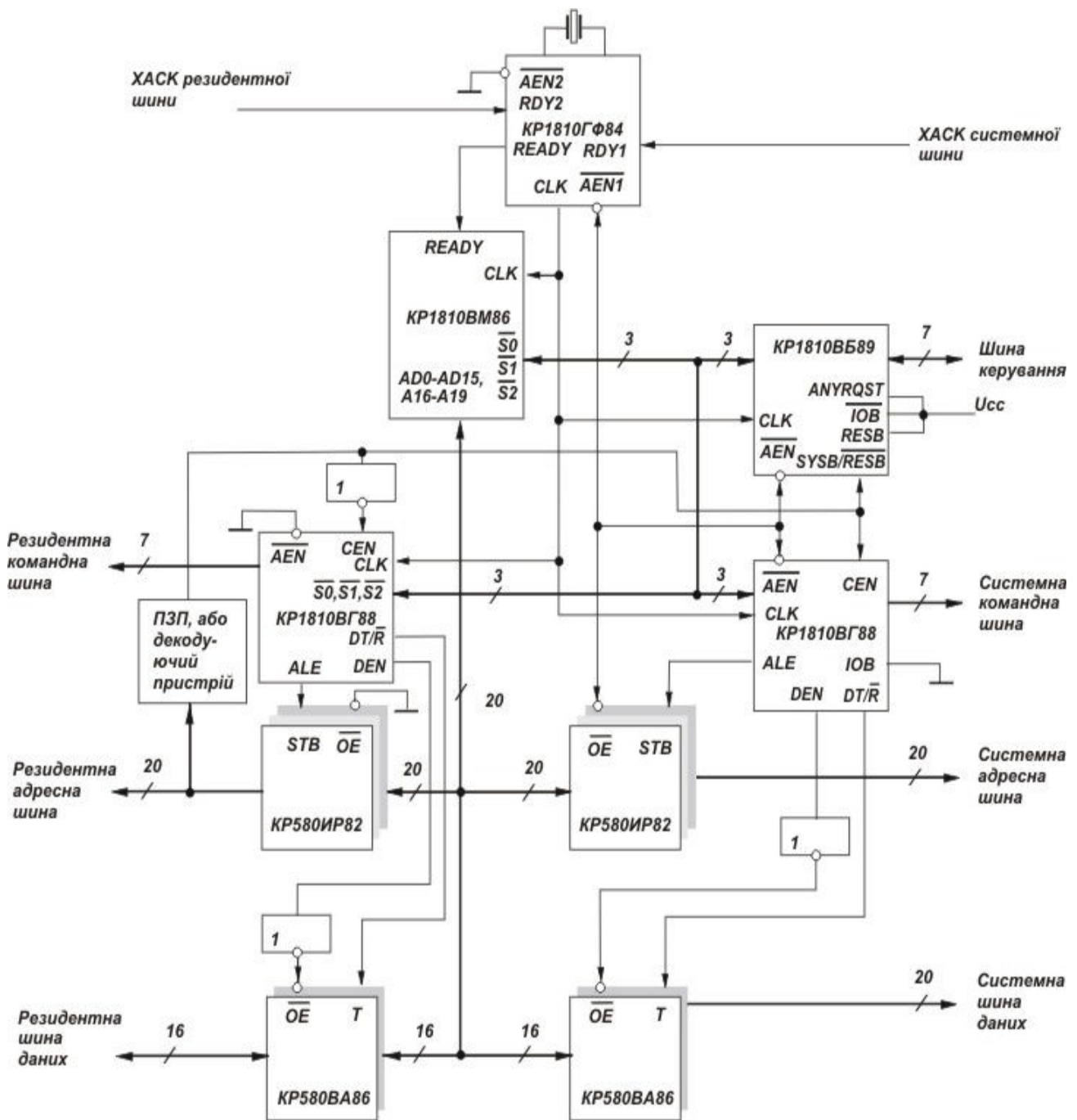


Рис. 46. Арбітр шини в конфігурації з системною та резидентною шинами

2.6. Організація переривань мікропроцесорної системи керування на основі мікропроцесорного комплекту К1810

При побудові МП систем на основі мікропроцесорного комплекту К1810 використовують неекторні інтерфейсні та векторні інтерфейсні переривання. При передаванні неекторного інтерфейсного переривання використовується тільки одна з ліній запиту переривання, одночасно при цьому інші пристрої можуть виконувати операції з передавання даних. Якщо по одній лінії запиту

переривань передаються сигнали від кількох джерел, то додаткова інформація може бути отримана шляхом читання байтів стана у програмі опрацювання переривань.

При векторному інтерфейсному перериванні проводиться передавання вектора переривань по лініях даних. Схема керування перериванням задавача перериває програму мікропроцесора і проводить захоплення керування інтерфейсом на весь час запиту на переривання. Після видавання першого сигналу INTA схема керування перериванням виставляє код переривання на лініях ADR8-ADR0. При наступному надходженні сигналу INTA в МП передається восьмибітний вектор переривання, що визначає початкову адресу пам'яті з програмою обслуговування переривання.

Для реалізації протоколу обміну в циклі переривання МП-систем на основі K1810BM86 (BM88) найчастіше використовується контролер переривань KP1810BH59A, що формує всі необхідні сигнали для опрацювання переривання.

2.6.1. Схема реалізації протоколу переривань

Процесор KM1810BM86 має два входи переривань:

- ◇ NMI – немасковане переривання;
- ◇ INTR – запит маскованих переривань.

Сигнал на вході NMI змушує процесор викликати процедуру для переривання з номером 2. Сигнал intr дозволяє зовнішнім пристроям передати в мікропроцесор по шині даних однобайтний номер переривання. Завдяки цьому останній може обслуговувати до 256 різних типів переривань, не потребуючи 256 входів запитів переривань.

На вхід INT подається вихідний сигнал програмованого контролера переривань KP1810BH59A. Контролер сприймає запити переривань від зовнішніх пристроїв (включаючи й інші мікросхеми KP1810BH59A), враховує їхні пріоритети, а потім сигналізує мікропроцесору по входу INT у випадку надходження неблокованого переривання. У випадку, коли переривання

дозволені, мікропроцесор закінчує виконання поточної команди і далі ініціює два цикли шини підтвердження переривання INTA.

Перший цикл шини INTA інформує контролер переривань про те, що мікропроцесор розпізнав запит переривання. Він також відводить час контролеру для взаємодії з під'єднаними до нього іншими контролерами. Протягом другого циклу шини INTA контролер видає в процесор по шині 8-бітний номер переривання. Після цього мікропроцесор викликає відповідну процедуру опрацювання переривання.

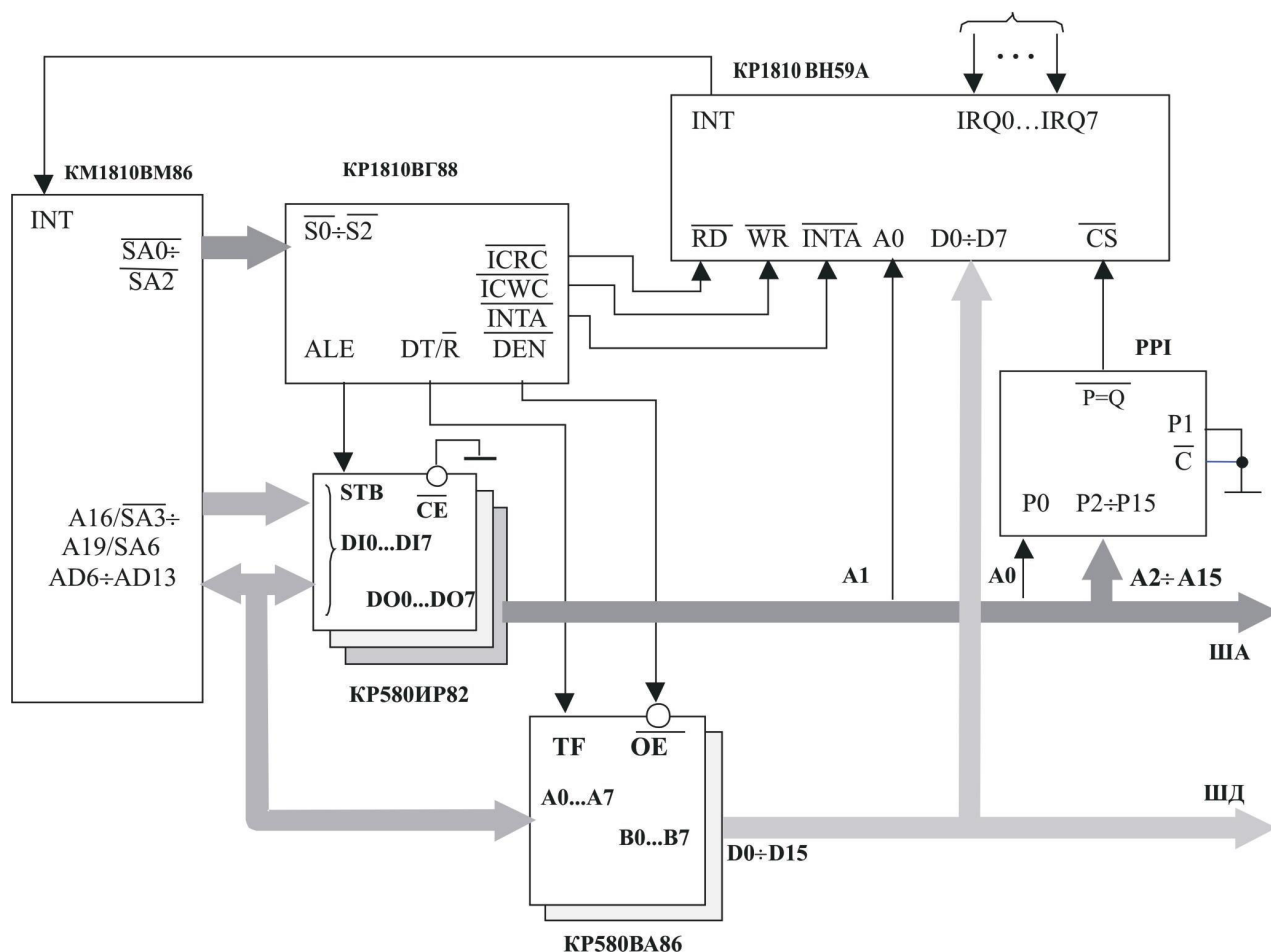


Рис. 47. Під'єднання контролера переривань

Під'єднання контролера переривань до МП системі зображено на рис. 47. Запити переривань подаються на входи IRQ0 – IRQ7. Один контролер може прийняти до восьми видів переривань. За наявності більше ніж восьми типів переривань необхідне встановлення декілька контролерів.

Контролер потребує дві адреси в просторі портів вводу-виводу, щоб програми могли зчитувати і встановлювати різні режими роботи, звертатися до

його внутрішніх регістрів і повідомляти про закінчення процедури переривання. Вхід A0 визначає, котрий з двох портів адресується. Сигнали RD, WR, CS означають відповідно дозвіл зчитування, записування та вибору мікросхеми.

2.7. Контрольні запитання

1. Розробка мікропроцесорної системи на базі МП К1810ВМ86 у мінімальному режимі роботи МП.
2. Робота МП в якості системного контролера.
3. Формування адресної шини, шини даних та шини керування.
4. Побудова оперативної та постійної пам'яті в системі на базі МП К1810ВМ86 в мінімальному режимі роботи.
5. Організація роботи МП-системи в циклі переривання.
6. Використання програмованого контролера переривання К1810ВН59А.
7. Організація режиму прямого доступу до пам'яті в МП-системі.
8. Структурна схема мікропроцесорної системи на базі МП К1810ВМ86 у максимальному режимі роботи МП.
9. Використання системного контролера К1810ВГ88. Формування адресної шини, шини даних та шини керування.
10. Побудова оперативної та постійної пам'яті в системі на базі МП К1810ВМ86 у максимальному режимі роботи. Організація шини вводу-виводу.

ТЕМА №3. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ ОДНОКРИСТАЛЬНИХ МІКРОЕОМ КМ1816ВЕ48

3.1. Призначення виводів та умовне позначення мікросхеми мікроЕОМ КМ1816ВЕ48

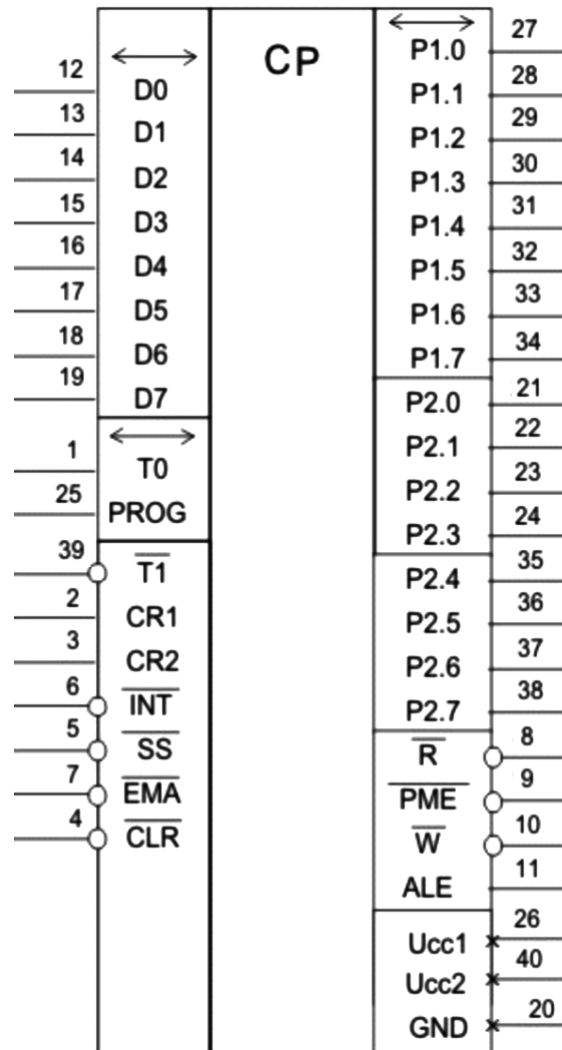


Рис. 48. Умовне позначення КМ1816ВЕ48

Однокристална мікроЕОМ КМ1816ВЕ48 призначена для побудови нескладних систем керування і містить 64 байти ОЗП даних, 1Кбайт ПЗП програм, багатоканальний інтерфейс вводу-виводу, восьмирозрядний лічильник-таймер. У конструкції мікроЕОМ передбачено розширення пам'яті програм до 4 КБайт, ОЗП даних до 256 Байт, а також збільшення числа ліній вводу-виводу за

рахунок під'єднання зовнішніх ПЗП, ОЗП та інтерфейсів вводу-виводу серії K580. Умовне позначення однокристальної мікроЕОМ KM1816BE48 зображено на рис. 48, призначення виводів вказано в таблиці 24.

Таблиця 24. Призначення виводів мікросхеми

Позначення	Номер	Призначення
1	2	3
GND	20	Загальний
U_{CC1}	26	Напруга живлення ОЗП у режимі нормальної роботи. У режимі програмування - напруга програмування
U_{CC2}	40	Напруга живлення решти блоків мікросхеми. Завжди дорівнює 5В
PROG	25	У режимі програмування – це вхід для подачі програмуючого імпульсу (+25 В). В нормальному режимі роботи використовується для подавання сигналу синхронізації для розширювача вводу-виводу
CR1, CR2	2, 3	Входи під'єднання кварцового резонатора, або резонатора іншого типу. Також на вхід X1 KM1816BE48 можна подавати сигнал синхронізації від зовнішнього джерела
\overline{CLR}	4	Сигнал скидання мікросхеми. Логічний нуль призводить до встановлення мікросхеми в початковий стан
SS	5	Покрокове виконання. Логічний нуль на лінії призупиняє виконання кожної команди на етапі видавання адреси для вибірки коду команди до тих пір, поки на лінію SS не надійде логічна 1. При цьому на лінії порту P2 та P0 виводиться адреса комірки пам'яті програм, код з якої має бути вибраний. Сигнал дозволяє виконувати програму з зупинкою після виконання кожної чергової команди
PME	9	Строб вибірки з пам'яті програм. Логічний нуль вказує, що відбувається вибірка коду команди з зовнішньої пам'яті
ALE	11	Строб адреси зовнішньої пам'яті. Використовується для вказання наявності та фіксації адреси зовнішньої пам'яті на зовнішньому регістрі. Логічна одиниця вказує на наявність на лініях порту P0 та P2 нової адреси. Фіксація адреси повинна відбуватись при переході сигналу з високого рівня в низький. Сигнал формується в кожному машинному циклі і може використовуватись в якості сигналу синхронізації з частотою, що дорівнює 1/15 частоти тактового сигналу
R	8	Читання. Логічний нуль вказує, що відбувається читання зовнішньої пам'яті даних.

1	2	3
W	10	Записування. Логічний нуль вказує, що відбувається записування зовнішньої пам'яті даних
T0	1	Вхідний сигнал, що опитується командами умовного переходу JT0 та JNT0. Також може працювати як вихід сигналу з частотою, що дорівнює 1/3 частоти тактового сигналу.
T1	39	Вхідний сигнал, що опитується командами умовного переходу JT1 та JNT1. Також може використовуватися в якості входу лічильника зовнішніх подій
INT	6	Вхід маскованого запиту переривання від зовнішнього джерела, може опитуватись командою умовного переходу JN1
EMA	7	Сигнал виключення внутрішнього ПЗП програм. Лог. 1 на вході вказує МК на необхідність виконувати вибірку всіх команд із зовнішньої пам'яті. Використовується для цілей налагодження і роботи з модифікаціями МК, які не мають внутрішньої пам'яті програм
P2.0 - P2.7	21-24, 35-38	Порт 2. Восьмибітний квазідвонаправлений порт вводу-виводу 2, кожен розряд якого може працювати як вхід або вихід незалежно від інших. Біти P2.3-P2.0 під час читання із зовнішньої пам'яті програм змінюють своє значення, на 4 старших розряди адреси коду команди, що виконується на весь час вибірки коду команди. Після завершення зчитування коду дані відновлюються із внутрішніх тригерів порту, крім того, вони можуть бути зчитані при виконанні відповідних команд
P1.0 - P1.7	27-34	Порт 1. Восьмибітний квазідвонаправлений порт вводу-виводу 1, кожен розряд якого може працювати як вхід або вихід незалежно від інших
D0-D7 (BUS)	12-19	Порт 0. Восьмибітний двонаправлений порт вводу-виводу інформації. Дані, що передаються, стробуються сигналами записування і читання

Лінії портів P1 та P2 є квазідвонаправленими тобто працюють як виводи з відкритим колектором, котрі мають внутрішні резистори, що підтягують лінію до високого рівня. Тому при встановленні відповідного біта порту в стан лог. 1 вивід видає невисокий струм, а при встановленні в стан лог. 0 може приймати досить значний струм. Така асиметрія полегшує узгодження виводу з входами і виходами ТТЛ мікросхем.

Для роботи з портами P1 та P2 використовують два типи команд:

- ◇ зчитування стана лінії (команди вводу з порту);

◇ зчитування стана внутрішнього регістра.

При зчитуванні стана лінії необхідно, щоб відповідний розрід порту був попередньо вставлений у лог. 1. У протилежному випадку лінія буде "посаджена" в стан лог. 0 і цей нуль і буде зчитаний.

Лінії P2.3 - P2.0 порту P2 мають внутрішній комутатор, що відмикає квазідвонаправлений вихід і підключає звичайний МОН вихід на час видавання адреси. Це збільшує швидкодію ліній порту P2 у режимі зчитування коду команд.

3.2. Структурна схема мікросхеми мікроЕОМ

КМ1816ВЕ48

Структурна схема мікросхеми зображена на рис. 49. Розглянемо основні функціональні блоки.

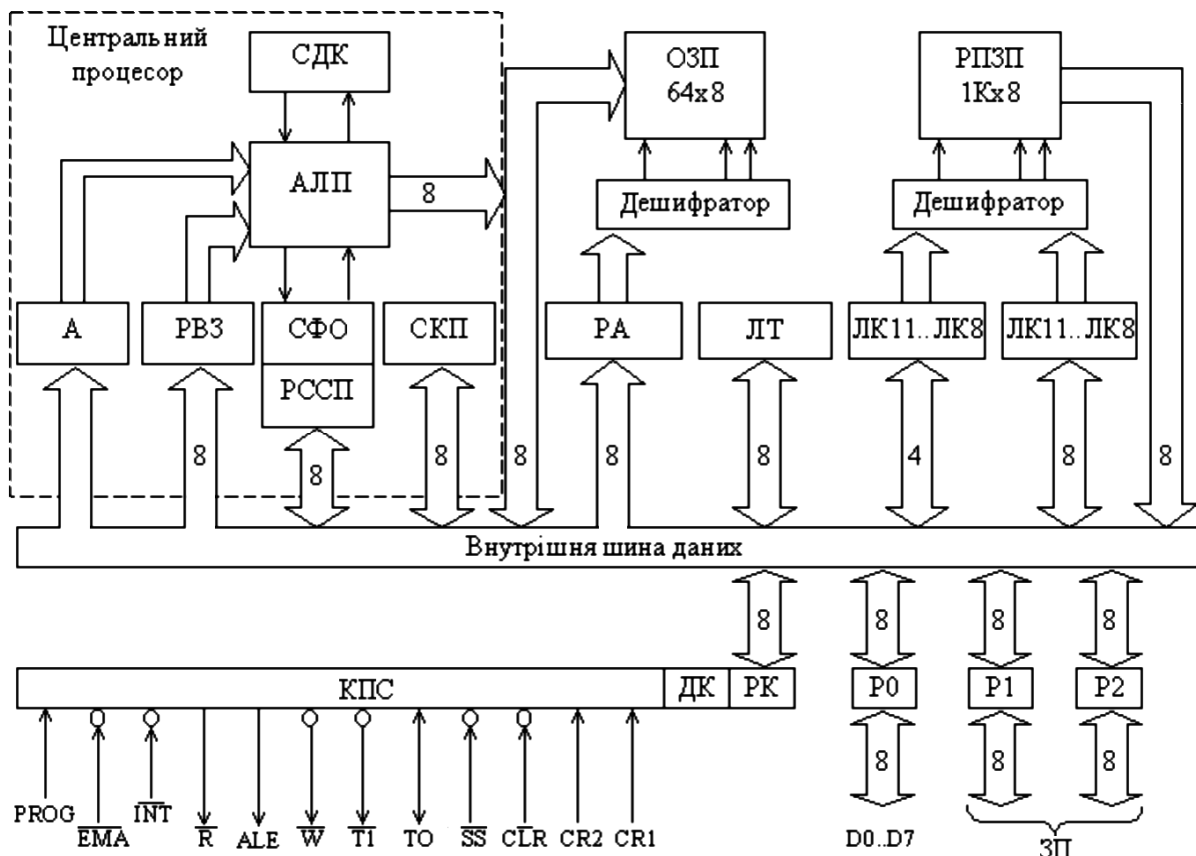


Рис. 49. Структурна схема мікросхеми

Блок "Центральний процесор". Центральний процесор має розрядність 8 біт і складається з шести підблоків:

1. Акумулятор (А). Регістр, призначений для збереження операндів і результатів виконання команд.

2. Регістр вибірки та зберігання (РВЗ) призначений для тимчасового зберігання другого операнда арифметичних та логічних команд.

3. Арифметико-логічний пристрій (АЛП), призначений для формування результатів арифметичних і логічних операцій, які виконує центральний процесор.

4. Схема десяткового коректора (СДК), призначена для десяткової корекції результату арифметичних операцій, виконаних над упакованими двійково-десятковими числами.

5. Схема формування ознак (СФО), призначена для формування ознак для команд переходів, що не фіксуються в регістрі стану процесора. До таких ознак належать ознака нульового вмісту та ознаки наявності 1 в обраному розряді акумулятора.

6. Регістр слова стану процесора (РССП), призначений для збереження ознак стану процесора. Формат регістра слова стану зображений на рисунку 50, призначення кожної ознаки наведені в таблиці 25.

Таблиця 25. Значення ознак слова стану

Ознака	Значення
С	Перенесення. Стан логічної одиниці означає, що остання арифметична операція викликала переповнення розрядної сітки
АС	Допоміжне перенесення. Логічна одиниця вказує, що в результаті виконання останньої арифметичної операції відбулося перенесення із третього в четвертий розряд. Ознака найчастіше використовується при роботі з упакованими двійково-десятковими числами.
F0	Ознака користувача - ознака може бути встановлена і скинута програмно за допомогою спеціальних команд встановлення і скидання
BS	Селектор банку регістрів загального призначення. Логічний нуль означає, що при роботі команд, які оперують регістрами R0-R7 дані вибираються з комірок ОЗП з адресами 0-7, одиниця вказує, що дані вибираються за адресами $18_{16} - 1F_{16}$.
S2, S1, S0	Вказівник стека. Регістр показує адресу біжучої вершини стека. Адреса вершини стека обчислюється за формулою $S2S1S0 \times 2 + 8$

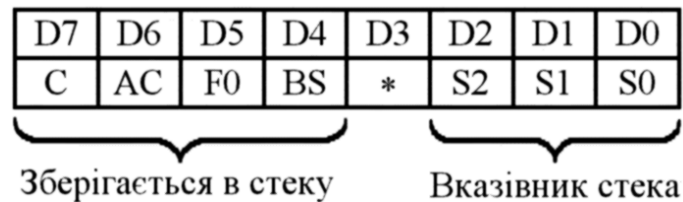


Рис. 50. Формат слова стана програми

Пристрій керування і синхронізації (ПКС), призначений для керування і синхронізації роботи інших блоків мікроЕОМ залежно від виконуваного коду команди та стана ліній мікросхеми. У ньому виділяють два підблоки.

1. Регістр команд (РК), котрий містить код виконуваної команди.
2. Дешифратор команд (ДК), призначений для перетворення коду команди в сигнали керування.

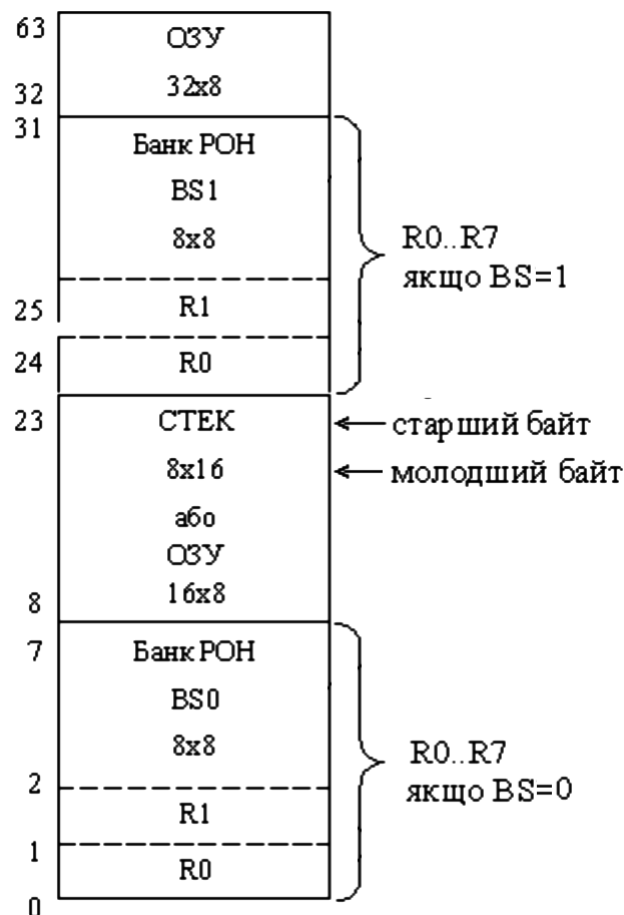


Рис. 51. Структура ОЗП

Блок пам'яті даних складається з ОЗП, дешифратора і регістра адреси (РА). Дешифратор призначений для визначення позиції комірки ОЗП, на основі адреси збереженої в РА. ОЗП містить пам'ять даних об'ємом 64 байти і служить для збереження даних. Структура ОЗП мікроЕОМ зображена на рис.

51. Комірки ОЗП з адресами 00-07 та 24-31₁₀ використовується в якості реєстрів загального призначення. Усі реєстри розбиті на два банки (BS0, BS1) по вісім у кожному. Перемикання банків здійснюється бітом BS реєстра слова стану. У структуру ОЗП також входить восьмирівневий 16-розрядний стек (адреси 8÷23₁₀). Решта комірок ОЗП використовується лише як ОЗП даних.

Блок РПЗП призначений для збереження та вибірки кодів команд, що виконуються мікроЕОМ. Він складається з таких підблоків:

1. РПЗП – постіна пам'ять програм ємністю 1 Кбайт.
2. Дешифратора, призначеного для дешифрування номера комірки РПЗП.
3. ЛК11 - ЛК8, ЛК7-ЛК0 лічильників команд старшого та молодшого байта адреси.

Блок вводу-виводу призначений для вводу і виводу інформації з мікроЕОМ через порти P0-P2. На схемі цьому блоку відповідають підблоки з однойменними назвами.

Останнім блоком, що розглядається, є блок лічильника-таймера (ЛТ), що являє собою 8-розрядний лічильник, якій може працювати в режимі таймера. ЛТ може генерувати переривання при переповненні.

3.3. Використання зовнішньої пам'яті та розширеного

вводу-виводу

Коли функціональних можливостей однокристалльної мікроЕОМ недостатньо, можна відносно простими засобами розширити МК-систему до таких розмірів:

- ◇ пам'ять програм – до 4 Кбайт;
- ◇ пам'ять даних – до 256 байт;
- ◇ лінії вводу-виводу – практично необмежено.

Крім того, шляхом під'єднання спеціалізованих ВІС, що входять у мікропроцесорний комплект К580, у системі можуть бути реалізовані різні допоміжні функції: зв'язок з дисплеєм і клавіатурою, багаторівнева система переривань, складна система таймування, зв'язок із телеграфно-телефонними

лініями передавання інформації і т.д. Із використанням засобів буферування і мультиплексування адреси/даних можна під керуванням програми створювати МК-системи будь-якої необхідної конфігурації.

3.3.1. МК-системи з зовнішньою пам'яттю програм

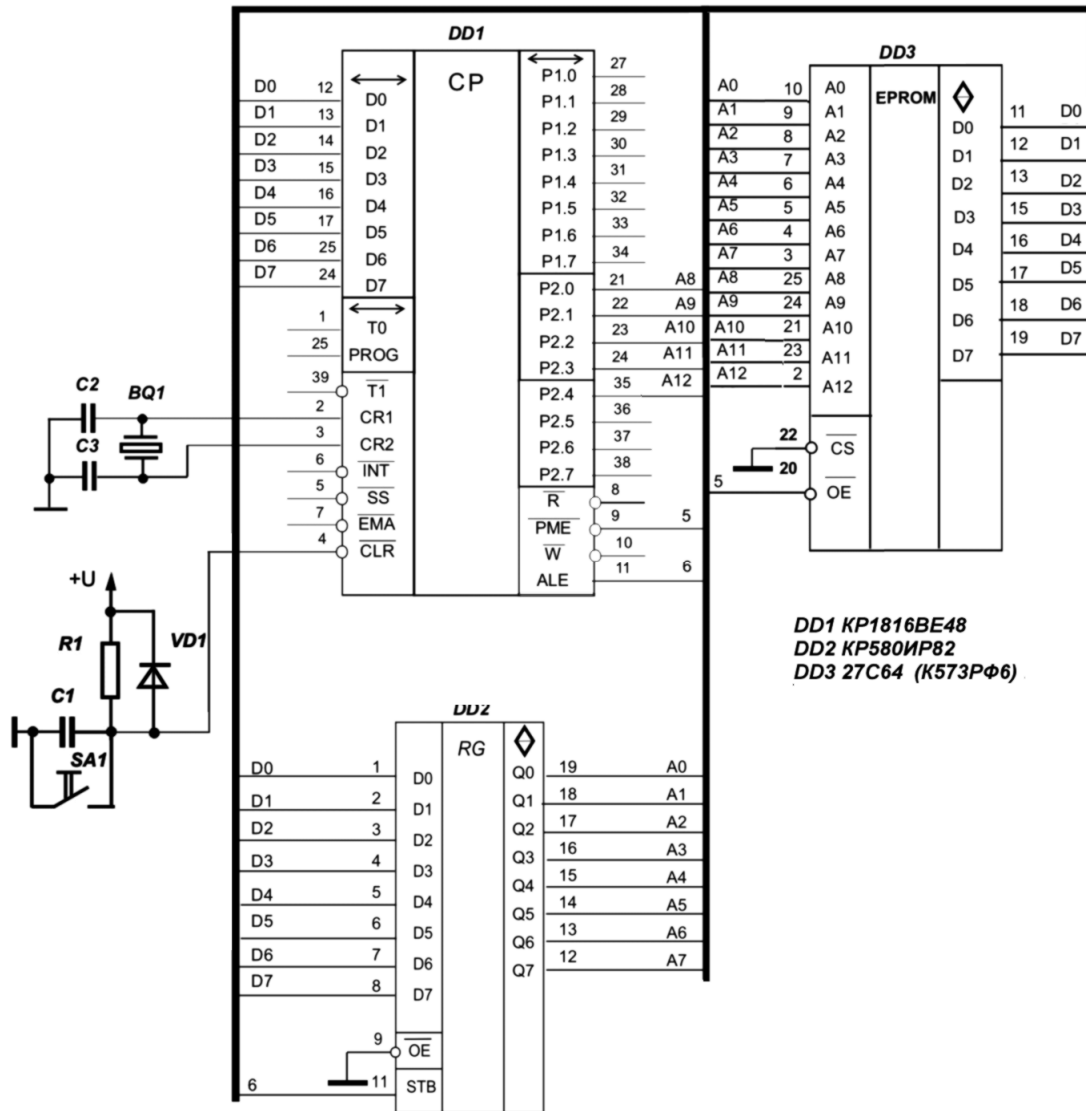


Рис. 52. Під'єднання ПЗП до ОМЕОМ

Шина BUS (P0) за своїми властивостями подібна до двонаправленої шини даних мікропроцесора KP580 і всі розширення МК використовуються й 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). При звертанні до комірок пам'яті програм, починаючи з адреси, що не відповідає області внутрішньої пам'яті програм 1024, МК автоматично формує керуючі сигнали, що забезпечують вибірку команд із

зовнішньої пам'яті програм. Послідовність процесу вибірки команди з зовнішньої пам'яті така:

- ◇ вміст лічильника команд виводиться через порт BUS і молодшу тетраду порту P2 (P2.0...P2.3);
- ◇ 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.

Комірки пам'яті з адресами, що лежать за межами банку пам'яті 0, будуть доступні після виконання команди перемикання банків пам'яті SEL MB1, лише після виконання команди переходу (JMP чи CALL).

На рис. 52 зображено структуру МК-системи з зовнішньою пам'яттю програм. Додаткова мікросхема пам'яті (DD3) ємністю 8 Кбайт під'єднується до шини BUS своїми інформаційними виходами. Молодший байт адреси за сигналом ALE фіксується на зовнішньому буферному регістрі DD2. Старша тетрада адреси, виведена через порт P2, не має потреби в буферизації, тому що вона зберігається протягом усього циклу вибірки. Найстарший біт адреси (P2.4) призначений для перемикання сторінок ПЗП після скидання рівний лог. 1 і може бути програмно скинутий в нуль для звертання до нульової програмної сторінки пам'яті. Для дозволу роботи шини не тільки на приймання даних на вхід ОЕ заведено сигнал PМЕ.

3.3.2. МК-система з зовнішньою пам'яттю даних

На рис. 53 зображено схему МК-системи, до складу якої входить додаткова мікросхема ОЗП (DD3), на основі якої реалізується пам'ять ємністю 256 байт.

Зовнішня оперативна пам'ять доступна МК за командами пересилання MOVX A,@R і MOVX @R,A, які за непрямою адресою (регістри R0 і R1) виконують операції передавання байта між внутрішньою пам'яттю даних (ВПД) і акумулятором. Сигналом ALE непряма адреса, виведена по шині BUS, фіксується в буферному регістрі DD2. Сигнали W і R визначають режим роботи

ОЗП. Схема на рис. 53 забезпечує адресацію 256 комірок ОЗП додатково до 64 комірок резидентної пам'яті даних МК48.

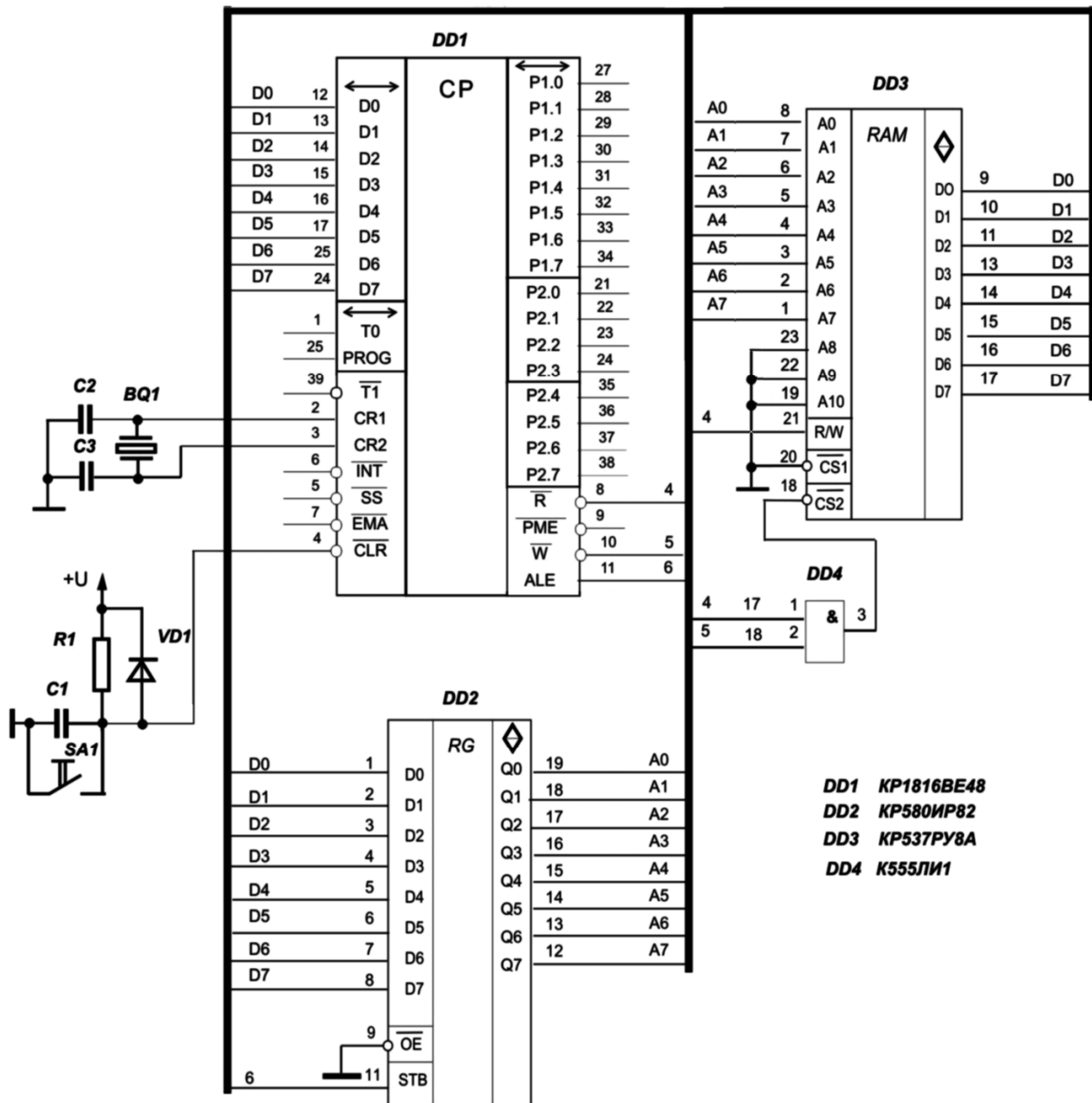


Рис. 53. Зовнішнє ОЗП даних

За необхідності подальшого нарощування об'єму зовнішнього ОЗП можна програмним способом реалізувати механізм сторінок пам'яті з використанням, наприклад, ліній порту P2. В такому випадку лінії A8 – A10 ОЗП під'єднуються до порту P2 і використовуються для перемикання сторінок, а молодші 8 розрядів адреси, що передаються через шину BUS, використовуються для визначення адреси в середині сторінки.

3.3.3. МК-система з розширеним вводом/виводом

Для з'єднання МК з об'єктом, що має велику кількість входів-виходів, кількість ліній вводу-виводу можна розширити, під'єднавши до МК необхідну кількість зовнішніх портів. Таке розширення може бути виконано з використанням:

- ◇ стандартного розширювача вводу-виводу (РВВ) КР580ВР43;
- ◇ інтерфейсних ВІС (КР580ВВ55, КР580ВВ51).

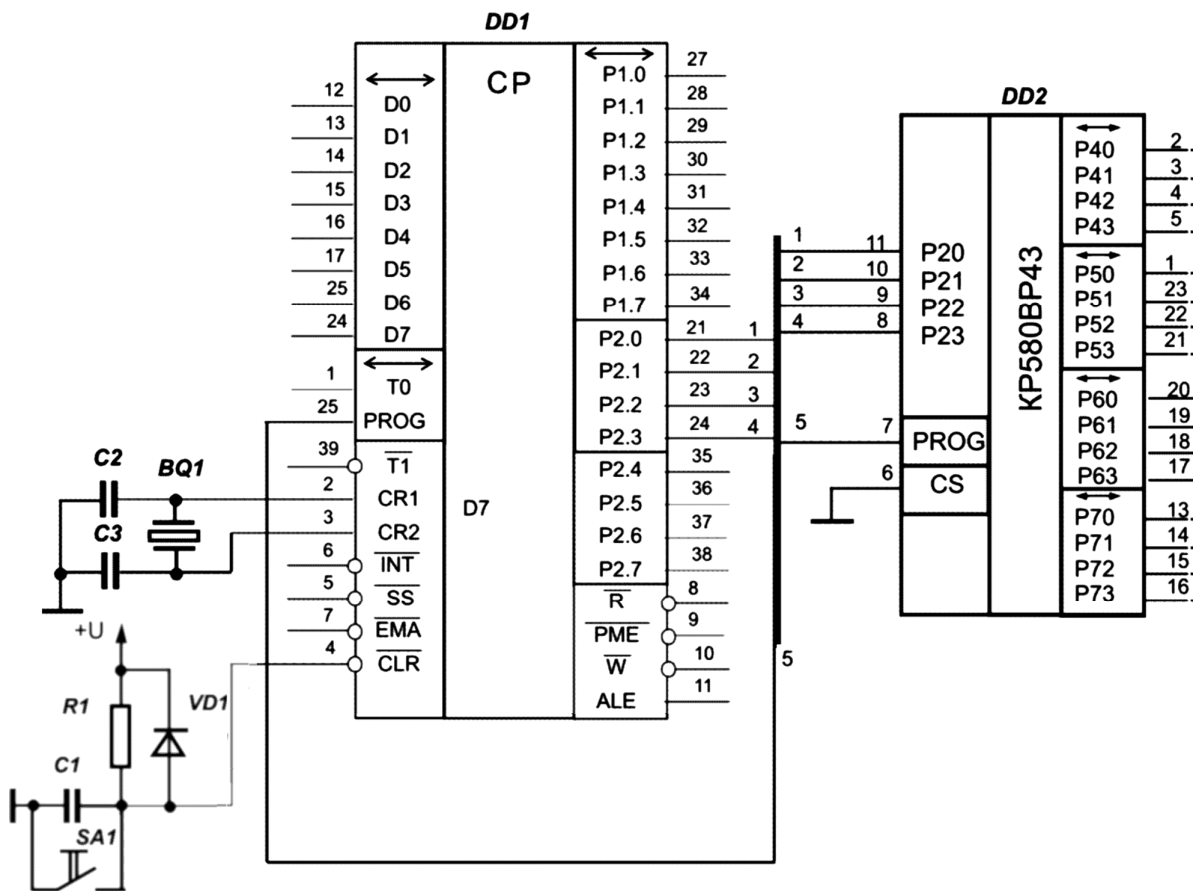


Рис. 54. Використання розширювача вводу-виводу

Розширювач під'єднується до МК48 так, як показано рис. 54. Кожен із чотирьох портів РВВ може використовуватися для вводу чи виводу інформації незалежно від інших і забезпечувати високу навантажувальну здатність. Для виводу байта даних у порти Р4 і Р5 розширювача можна скористатися такою послідовністю команд:

```

MOVD P4,A      ;Вивід А(0...3) в порт 4
SWAP A        ;Обмін тетрад акумулятора
MOVD P5,A     ;Вивід другої тетради в порт 5.
    
```

На рис. 55 зображено два варіанти розширення вводу-виводу з використанням периферійного адаптера KP580BB55. У першому варіанті (рис. 55а) порти адаптера адресуються як комірки ЗПД, доступ до яких здійснюється за командами MOVX. Наприклад, для виводу байта в порт В необхідно виконати дві команди:

```
MOV R1, #1      ; (R1) = Адреса порту В
MOVX @R1, A     ; Вивід в порт В з акумулятора.
```

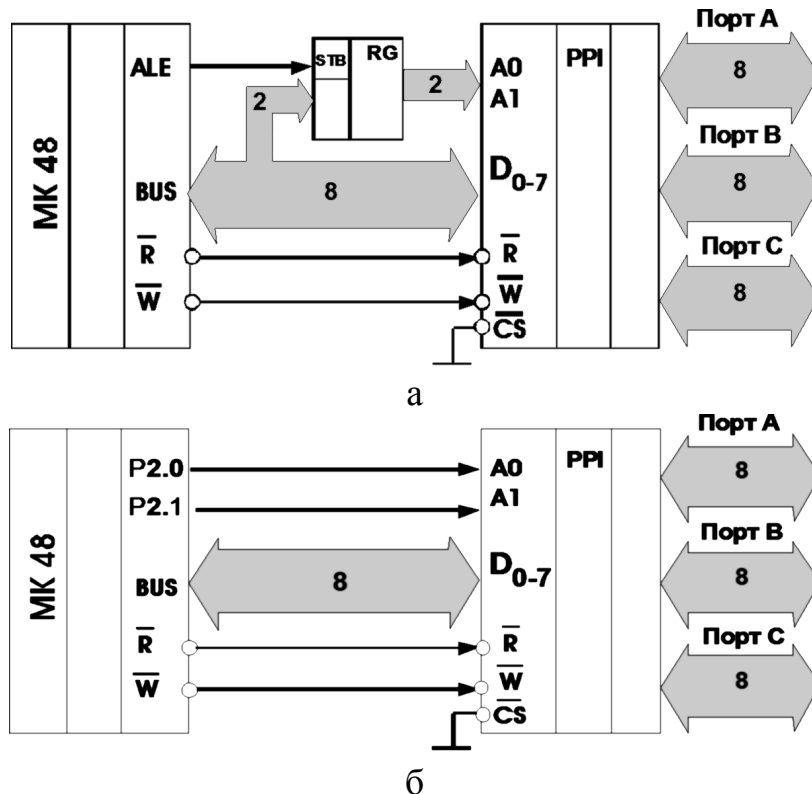


Рис. 55. Використання мікросхем комплексу KP580

Для другого варіанта під'єднання ППА (рис. 55б), вибір порту здійснюється через установку/скидання двох розрядів порту P2. Для виводу байта даних у порт А можна скористатися такими командами:

```
ANL P2, #0FCH   ; Скидання A0 і A1 адаптера
OUT BUS, A      ; Вивід байта в порт А
```

У розглянутих вище схемах розширення використовується тільки по одній зовнішній ВІС, тому їхній вхід CS під'єднується до землі, за необхідності до МК може бути під'єднано декілька РВВ і ППІ. У такому випадку визначення мікросхеми, з якою відбувається обмін, здійснюється за допомогою сигналу CS. Сигнал CS може формуватися на основі дешифрування

старших ліній адреси, не використаних на схемі рис. 55а, або за допомогою використання додаткових сигналів із вільних портів вводу-виводу.

3.4. Робота з клавіатурами

Як пристрої вводу інформації найширше використовуються в МК-системах цифрові, алфавітно-цифрові і спеціальні клавіатури.

3.4.1. Різновидності клавіатур.

У різних за складністю та призначенням керуючих системах використовуються різноманітні клавіатури для вводу інформації:

- ◇ найпростіші, що складаються з клавіш керування, наприклад, СКИДАННЯ, ПУСК, ЗУПИНКА і т.п.;
- ◇ цифрові, призначені для вводу даних і керування режимом роботи МП-системи і складаються із шістнадцяткової клавіатури і керуючих клавіш ЗАВАНТАЖЕННЯ, АДРЕСА/ДАНИ, ПОКРОКОВИЙ РЕЖИМ, ІНДИКАЦІЯ і т.д.;
- ◇ алфавітно-цифрові;
- ◇ спеціалізовані, у яких кожній клавіші відповідає деяка процедура процесу керування, наприклад, ПІДВИЩИТИ ТИСК У МАСЛЯНІЙ МАГІСТРАЛІ, ЗНИЗИТИ ТЕМПЕРАТУРУ ОХОЛОДЖУВАЛЬНОЇ РІДИНИ і т.п.;
- ◇ багатофункціональні, на основі сенсорних перемикачів, що доповнюються змінюваними лицьовими панелями з відповідними написами. Ці клавіатури за наявності відповідних програмних засобів дозволяють на тих самих апаратурних засобах реалізувати набір різноманітних технологічних мов і забезпечити їх оперативну заміну.

За способами апаратурної реалізації розрізняють два типи клавіатур: кодовані та некодовані.

У клавіатурах першого типу схемним шляхом на виході формується код, що відповідає натисненій клавіші. Через значний об'єм неуніфікованої апаратури схем перетворення кодів і високої вартості, що різко зростають із

зростанням кількості знаків, такі клавіатури в МК-системах застосовуються рідко. Значно ширше використовують дешеві некодуєчі (матричні) клавіатури, що є простою матрицею двійкових перемикачів, ввімкнених на перетинанні рядків і стовпців матриці. Ідентифікація натисненої клавіші в таких клавіатурах виконується програмою.

3.4.2. Ввід коду натисненої клавіші

Визначення стана клавіатури можна здійснити різними шляхами. По-перше, сама клавіатура при натисненні кнопки може формувати сигнал переривання, що повідомить МП про наявність натиснення клавіші і, можливо, видасть код натисненої кнопки.

Розглянутий метод спрощує програмування, проте призводить до ускладнення апаратних засобів системи, тому частіше використовують інший метод – опитування (поллінг). Через визначені інтервали часу опитується стан клавіатури для визначення та ідентифікації натисненої кнопки.

Суть третього методу полягає в звертанні основної програми до процедури вводу коду натисненої кнопки в момент, коли таке звертання необхідне. Процедура вводу проводить опитування клавіатури. Викликана з основної програми, процедура блокує процес керування об'єктом на час опитування клавіатури.

Процедуру вводу інформації з матричної некодованої клавіатури зручно розглянути на прикладі клавіатури 4×5, що включає 16 цифрових клавіш (0 - F) і 4 керуючих. Структура матриці клавіатури аналогічна структурі матриці двійкових датчиків, а схема під'єднання клавіатури до МК представлена на рис. 56.

Лінії порту P1 використовуються для сканування, а лінії порту P2 – для опитування матриці клавіш. Кожна клавіша в такій матриці має свій номер, якій відповідає місце її розташування. На цифрові клавіші можна нанести позначення, які відповідають їх кодам (від 0 по F). Коди керуючих клавіш більше числа 0F₁₆. Діоди забезпечують захист від замикання між собою скануючих ліній у випадку одночасного натискання декількох клавіш.

Процедура вводу коду натисненої клавіші складається з послідовності таких дій:

- ◇ сканування матриці клавіш;
- ◇ усунення деренчання контактів;
- ◇ очікування звільнення клавіші;
- ◇ ідентифікація коду натисненої клавіші.

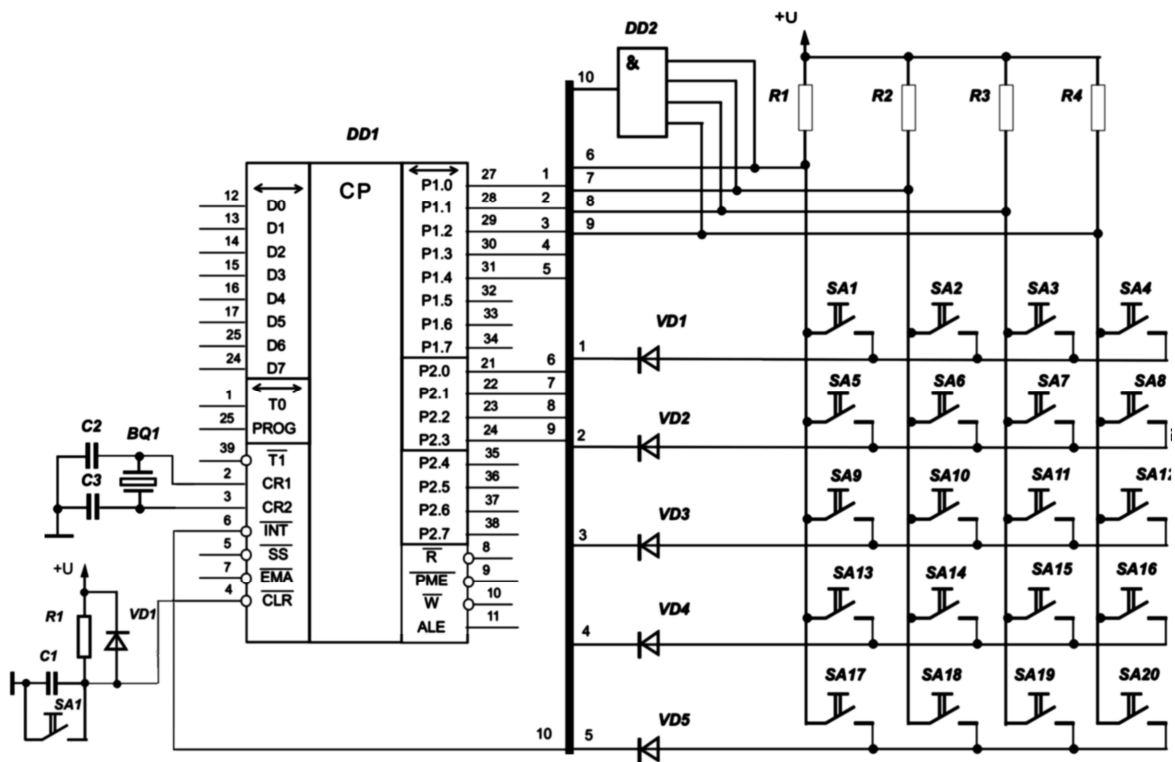


Рис. 56. Схема під'єднання клавіатури

Для деякого типу клавіатур може бути відсутня процедура усунення деренчання контактів. Процедуру сканування іноді буває зручно поєднати з процедурою ідентифікації. Розглянемо окремо кожну з перерахованих процедур.

3.4.3. Сканування

Процедура сканування служить для виявлення натисненої клавіші і наступної її ідентифікації. Процедура зводиться до почергового обнулювання кожної з ліній сканування й опитування ліній повернення. У порт P1 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 байта повернення (БВ), прийнятого в порт P2, 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 натисненої клавіші, то процедура сканування повторюється спочатку.

Після виходу з процедури SCAN у SCANCODE буде знаходитися код натисненої клавіші. Крім того, процедура SCAN здійснює захист від одночасного натискування кількох клавіш. Порядок аналізу клавіш такий, що при одночасному натискуванні клавіша з більшим кодом ігнорується:

;R4 - лічильник коду сканування

SCAN:

```
MOV R4,#0      ;скидання SCANCODE
MOV R7,#5      ;завантаження лічильника сканування
MOV R6,#11111110B ;завантаження вихідного байта
                ;сканування
```

LOOP:

```
MOV A,R6
OUTL P1,A      ;вивід поточного байта
                ;сканування
RL A           ;зсув байта сканування
MOV R6,A       ;збереження поточного байта
                ;сканування
IN A,P2        ;ввід байта повернення
MOV R5,#4      ;завантаження лічильника опитувань
```

ROTATE:

```
RRC A          ;зсув байта повернення
JNC DBNC       ;вихід із процедури при
                ;виявленні першої натисненої
```

```

INC R4 ;інкремент SCANCODE
DJNZ R5,ROTATE;декремент лічильника опитувань
;і цикл, якщо не нуль
DJNZ R7,LOOP ;декремент лічильника
;сканування і цикл, якщо не 0
JMP SCAN ;почати сканування спочатку

```

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

3.4.4. Усунення деренчання контактів

Усунення деренчання контактів при введенні символу з клавіатури здійснюється, як правило, програмною реалізацією очікування тривалістю 5-20 мс у залежності від механічних характеристик клавіатури:

```
DBNC : CALL DALEY ;затримка
```

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

3.4.5. Чекання звільнення клавіші

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 натиснена клавіша. Тому процедура чекання звільнення натисненої клавіші зводиться до зчитування й аналізу байта повернення:

```

WAITOP: IN A,P2 ;звід байта повернення
CPL A ;інверсія байта повернення
JNZ WAITOP ;якщо клавіша натиснена, то чекати

```

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

Знайти "залипання" контакту можна шляхом підрахунку "невдалих" опитувань. Якщо число "невдалих" опитувань перевищила N і клавіша не була

звільнена, видається повідомлення оператору про залипання контакту. Число N (і відповідно розрядність лічильника) визначається, виходячи з механічних властивостей клавіатури.

3.4.6. Ідентифікація натисненої клавіші

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 натисненої клавіші.

3.4.7. Оформлення процедури вводу

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

KEYBRD :

SCAN : ... ;сканування клавіатури

...

...

DBNC : CALL DELAY;усунення деренчання при
;натискуванні

WAITOP : ... ;чекання звільнення клавіші

...

...

DBNC : CALL DELAY;усунення деренчання контакту
;при звільненні клавіші може бути
;відсутня

KEYW : .. ;ідентифікація натисненої клавіші
;(може бути відсутня)

Процедура опитування стана клавіатури. Вище відзначалося, що в МК-системах реалізують безупинне керування, процедури KEYBRD повинна

передувати процедура опитування стана клавіатури ASK. Приклад програмної реалізації процедури ASK, оформленої у вигляді підпрограми, наведено нижче.

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

```
ASK : CLR A      ;скидання акумулятора
      CLR C      ;скидання ознаки перенесення
      OUTL P1,A  ;вивід байта "усі нулі"
                        ;для одночасного
                        ;опитування всіх клавіш
      IN A,P2    ;ввід байта повернення
      CPL A      ;інверсія байта повернення
      JZ EXIT    ;вихід, якщо немає натисненої клавіші
      CPL C      ;встановлення ознаки переносу
EXIT: RET        ;повернення
```

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

3.5. Вивід і відображення інформації. Індикатори

Як пристрої виводу інформації найширше застосовуються в МК-системах лінійні дисплеї на семисегментних і матричних світлодіодних індикаторах, алфавітно-цифрові і графічні дисплеї на газорозрядних, рідкокристалічних і світлодіодних панелях.

Багато МК-пристроїв вимагають наявності тільки найпростішої індикації на взірць ТАК/НІ, ВКЛ/ВИКЛ. Така індикація реалізується на основі окремих світлодіодів. Семисегментні індикатори (ССІ) використовуються для відображення цифрової і буквеної інформації. Сім елементів, які відображають, дозволяють висвічувати десяткові та шістнадцаткові цифри, деякі літери кирилиці і латинського алфавітів, а також деякі спеціальні знаки.

Для засвічування одного сегмента 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 великої потужності. Як проміжні підсилювачі струму можуть використовуватися логічні елементи ТТЛ або інтегральні схеми перетворювачів кодів для керування ССІ (рис. 57).

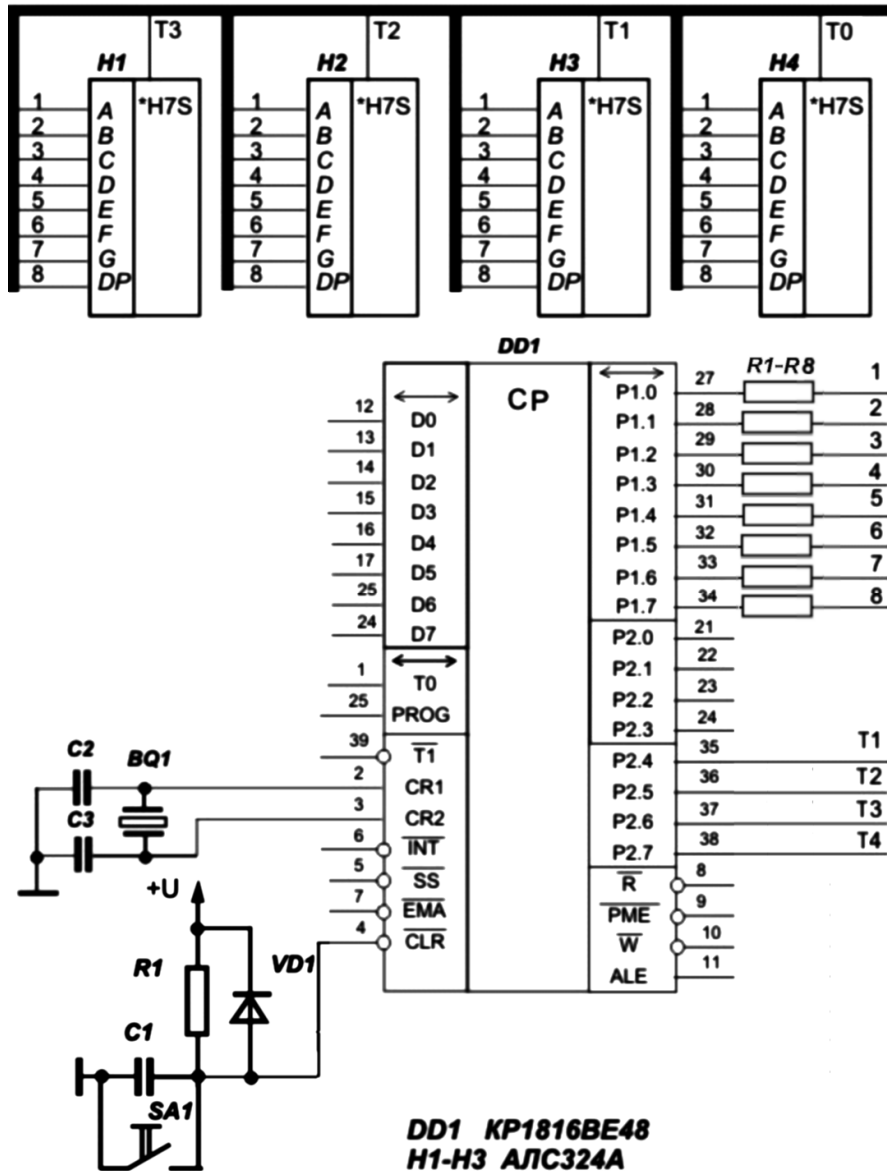


Рис. 57. Приклад під'єднання дисплея до КР1816ВЕ48

Перетворення двійкових кодів у коди для ССІ може здійснюватися програмно, або апаратно з використанням перетворювачів К514ИД1, К514ИД2, 133ПП4, 564ИД5.

Матричні світлодіодні індикатори (МСІ) використовуються для відображення алфавітно-цифрової інформації. Кожен з таких МСІ, виконаний у вигляді інтегральної мікросхеми, є матрицею світлодіодів розмірністю $m \times n$, де

n - кількість стовпчиків, m - кількість рядків матриці. Найбільше поширення набули МСІ з розмірністю матриці 7×5 і 9×7 (рис. 58).

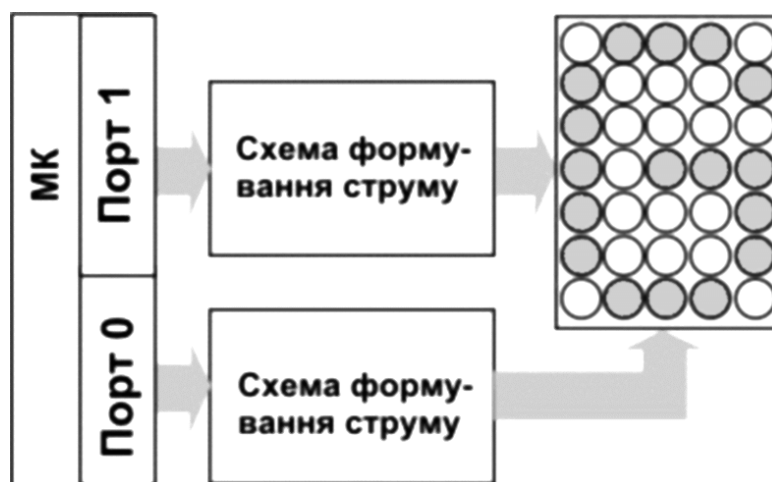


Рис. 58. Схема під'єднання матричного дисплея

Для ввімкнення одного світлодіода матриці необхідно забезпечити протікання через нього струму 10-15 мА при напрузі 2,0-2,5 В. Під'єднання матричного індикатора до МК здійснюється через керовані схеми формування струму стовпців і рядків.

Для відображення багатосимвольної інформації використовуються лінійні (однорядкові) дисплеї, які являють собою "лінійку", змонтовану з окремих ССІ чи МСІ. Кількість знакомісць дисплея визначається вимогами МК-системи. Існує два способи організації інтерфейсу МК із лінійним дисплеєм: статичний і динамічний.

Статичний вимагає наявності на входах кожного індикатора спеціальних буферних регістрів для збереження кодів виведених символів. Зі збільшенням розрядності дисплея зростає кількість додаткових мікросхем, отже, і вартість МК-системи. Проте символи на таких дисплеях залишаються розбірливими при високій амплітуді вібрацій і тому такий метод широко застосовується при роботі МП систем на рухомих об'єктах.

Динамічний спосіб заснований на тому, що будь-який світловий індикатор є інерційним приладом, а людському оку відображується на дисплеї інформація, якщо її поновлювати з частотою приблизно 20 разів за секунду, здається незмінною. Динамічний спосіб виводу інформації на дисплей вимагає

значно менших апаратурних витрат, але складнішого програмного забезпечення. Саме цей спосіб організації введення інформації набув переважного поширення в МК-системах.

3.5.1. Вивід символу на індикатори

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

При програмному перекодуванні МК повинен поставити у відповідність двійковому коду зображуваного символу визначений байт індикації (БІ), який і видається у вихідний порт. Перекодування найзручніше проводити табличним способом. Для цього байти індикації розміщуються в суміжних комірках РПП у порядку зростання вихідних двійкових кодів символів. Таке розташування байтів індикації спрощує процес перекодування, тому що в цьому випадку необхідний байт знаходиться в рядку таблиці з номером, що дорівнює двійковому коду зображуваного символу:

```
SYMBOL: MOV A, @R0      ;завантаження в акумулятор коду  
                          ;символу  
ADD A, #CODTBL ;формування адреси байта  
                          ;індикації  
MOVP3 A, @A      ;зчитування байта індикації з  
                          ;таблиці  
OUTL P1, A      ;видача байта індикації на  
                          ;індикатор через порт P1
```

Наведений фрагмент програми розрахований на те, що гасіння ССІ здійснюється при ініціалізації системи. Одночасно з цим у регістрі R0 формується адреса комірки CODE, у якій зберігається двійковий код символу, що виводиться на індикатор:

```
INIT : ...  
ORL P2, #0FFH      ;гасіння індикатора  
MOV R0, #CODE      ;завантаження в R0 адреси CODE
```

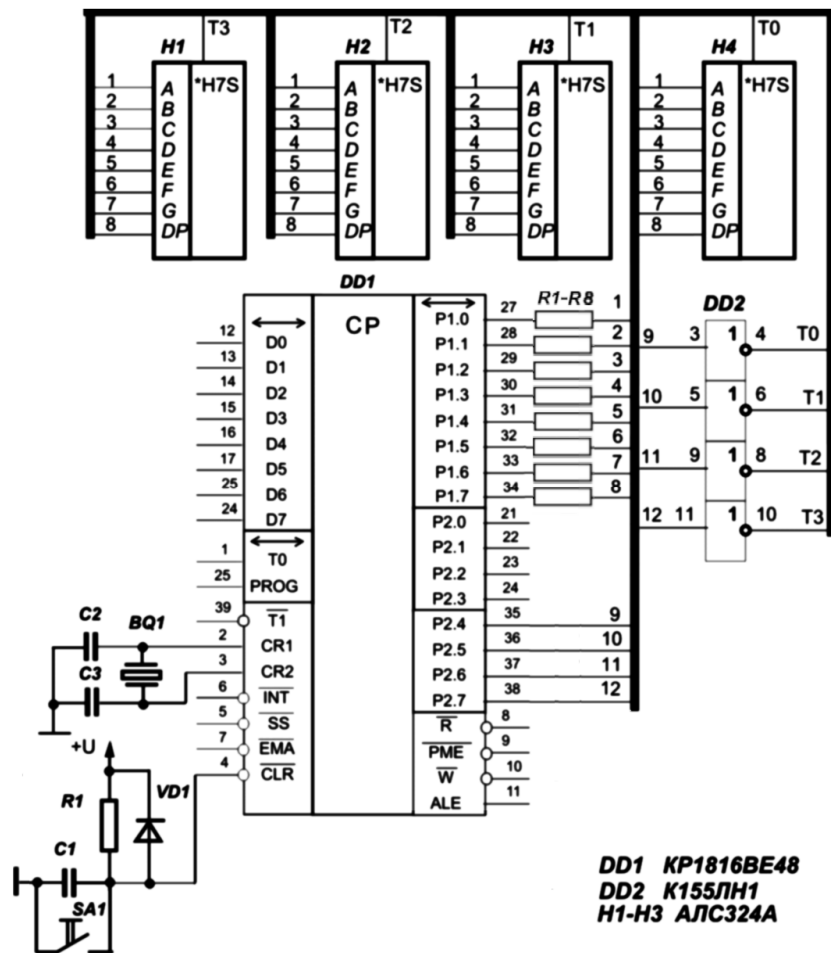


Рис. 59. Лінійний дисплей на семисегментних світлодіодних індикаторах

3.5.2. Виведення інформації на лінійний дисплей

При динамічній індикації байт індикації надходить одночасно на входи всіх ССІ, що утворюють лінійний дисплей, а вибір знакомісця здійснюється байтом вибірки, що являє собою код "біжучий нуль". Використання для формування коду "біжучий нуль" зовнішнього дешифратора дозволяє збільшити кількість знакомісць лінійного дисплея. Значне збільшення розрядності дисплея призводить до пропорційного збільшення скважності імпульсів, що надходять на світлодіоди і зменшення максимально можливої яскравості. Тому на практиці обмежуються 8–12 індикаторами. При бездешифраторному способі формування байта вибірки максимальна кількість індикаторів обмежується розрядністю порту з урахуванням обмеження на скважність.

Для динамічної індикації групи символів зручно скористатися

процедурою індикації символу, оформивши її у вигляді параметризованої підпрограми. Вхідними параметрами для підпрограми DSPLY є вихідний код відображуваного символу і номер знакомісця, на яке здійснюється виведення.

Вихідний код символу задається поточною адресою в масиві CODE (регістр R0), а номер знакомісця – поточним значенням байта вибірки (регістр R2) і значенням лічильника знакомісць (регістр R7):

```
DSPLY : MOV A,#OFFH ;бланкування(гасіння всіх
        OUTL BUS,A ;індикаторів)
        MOV A,@R0 ;вибірка коду символу
        ADD A,#CODTBL ;перекодування символу
        MOVP3 A,@A
        OUTL P1,A ;виведення байта індикації
        MOV A,R2
        OUTL BUS,A ;виведення байта вибірки
        RL A ;зсув байта вибірки убік
        MOV R2,A ;старших біт
        INC R0 ;просування по масиву CODE
        DJNZ R7,EXIT ;декремент лічильника
        ;знакомісць і вихід, якщо не
        ;нуль
INITD : MOV R2,#0FЕH ;завантаження в R2 вихідних
        ;байтів вибірки
        MOV R7,#8 ;завантаження лічильника
        ;знакомісць
        MOV R0,CODE ;завантаження в R0 початкової
        ;адреси і масиву CODE
EXIT : RET ;повернення
```

Підпрограма DSPLY реалізує вибірку коду чергового символу з РПД (вихідні коди символів повинні бути розміщені в послідовно розташованих комірках пам'яті), його перекодування і відображення на поточному знакомісці.

Для отримання яскравої і рівної індикації необхідно забезпечити:

- ◇ заборону вибірки знакомісць на час зміни байта індикації в порту P1 (бланкування);
- ◇ регенерацію зображення на кожному знакомісті з частотою не менше 50 разів за секунду, тобто звертання до підпрограми DSPLY через кожні $20/N$ мс., де N - число знакомісць на дисплеї.

У схемі бланкування здійснюється видачею байта вибірки "всі одиниці". Необхідна частота регенерації зображення забезпечується основною програмою МК-системи, якщо вона організована у відповідності зі структурою:

```
INIT : ... ; ініціалізація
      CALL INITD ; формування вихідних параметрів
      ... ; для підпрограми DSPLY
      CALL DSPLY ; вивід першого символу
      ...
      ; фрагмент основної програми
      ; тривалістю не більше  $20/N$  мс
      CALL DSPLY ; вивід другого символу
      ; фрагмент основної програми
      ; тривалістю не більше  $20/N$  мс
      ... ;
      CALL DSPLY ; вивід третього символу
      ...
      ...
      CALL DSPLY ; вивід N-го символу
      ; фрагмент основної програми
      ; тривалістю не більше  $20/N$  мс
      ...
```

Проте значно краще необхідну частоту звертань до підпрограми DSPLY забезпечити за допомогою переривання від таймера, що надходить через $20/N$ мс.

3.6. Введення інформації з датчиків

3.6.1. Опитування двійкового датчика. Чекування події

У пристроях і системах логічного керування об'єкти та події в об'єкті керування фіксуються з використанням різноманітних датчиків цифрового й аналогового типів. Найпоширенішими з двійкових датчиків є кінцеві вимикачі, що під'єднуються до МК так, як зображено на рис. 60.

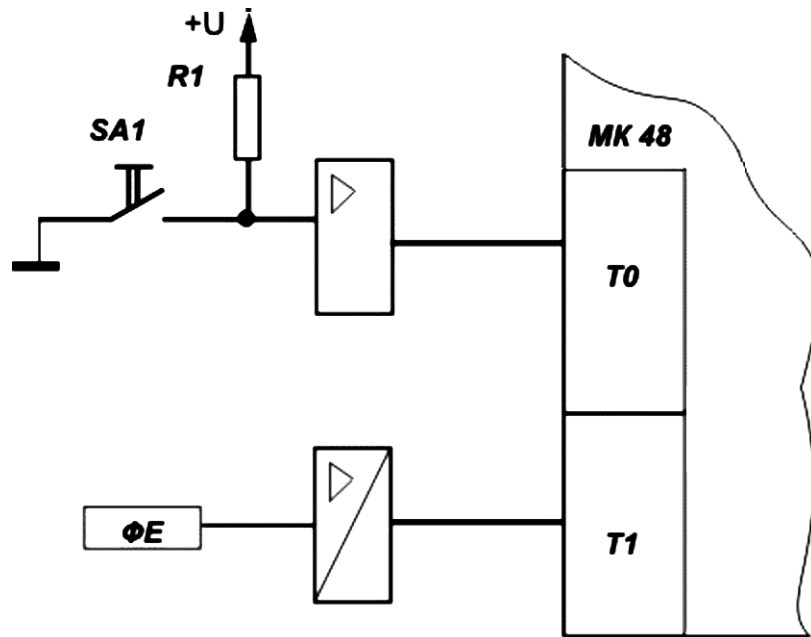


Рис. 60. Введення інформації з датчиків

3.6.2. Чекування статичного сигналу

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

- ◇ вводу сигналу від датчика;
- ◇ аналізу значення сигналу і передавання керування залежно від стану датчика.

Конкретна програмна реалізація процедури залежить не тільки від МК, але і від того, яким чином датчик під'єднаний до МК. Він може бути під'єднаний до однієї з ліній портів МК чи до спеціальних входів (T0, T1 для МК48).

Наприклад, при під'єднанні датчика до лінії 3 порту 1 програма процедури очікування замикання контакту буде мати вигляд:

```
WAITC: IN A, P1 ;ввід сигналу від датчика
        JB3 WAITC ;якщо контакт датчика розімкнений,
                ;то повторювати ввід, інакше вихід
                ;із процедури
```

При під'єднанні датчика до тестового входу T0 мікроконтролера МК48 програма буде мати вигляд:

```
WAITC: JT0 WAITC ;якщо контакт розімкнено, то цикл
                ;повторити
```

Інший варіант типової процедури чекання події є процедура чекання розмикання контакту:

```
WAIT0: IN A, P1;ввід байта
        CPL A ;інвертування
        JB3 WAIT0 ;якщо контакт замкнено, то повторити
```

Крім входів T0 і T1 для опитування датчика може використовуватися і вхід INT. У цьому випадку треба попередньо заборонити переривання і використовувати вхід INT як тестовий.

Режим переривання доцільно використовувати тільки для опитування особливо важливих датчиків з метою зменшення часу реакції на виняткову (аварійну) ситуацію в об'єкті керування.

3.6.3. Чекання імпульсного сигналу

Схема під'єднання датчика імпульсного сигналу аналогічна схемі, що на рис. 60. Особливість процедури чекання імпульсного сигналу полягає в тому, що МК повинний знайти не тільки факт появи, але й факт закінчення сигналу.

Оформляти процедури WAITC і WAIT0 у вигляді підпрограм недоцільно, тому що це подовжує програму, а довжина і час виконання програми визначають мінімальну тривалість імпульсу, що може бути виявлений програмою.

Послідовність склеювання процедур WAITC і WAIT0 залежить від форми імпульсу. Для "від'ємного" імпульсу ($1 \rightarrow 0 \rightarrow 1$) процедура WAITC передус

процедурі WAIT0, для "позитивного" ($0 \rightarrow 1 \rightarrow 0$) – слідує за нею.

Нижче наведено приклади програмної реалізації процедури чекання "від'ємного" імпульсного сигналу при під'єднанні датчика до біта 3 порту P1 за умови, що початковий стан входу – одиничний:

```
WAITC: IN A, P1 ;ввід байта
        JB3 WAITC ;якщо P1.3=1, то чекати
WAIT0: IN A, P1 ;ввід байта
        CPL A ;інверсія
        JB3 WAIT0 ;якщо P1.3=0, то чекати
```

Аналогічним чином будується програма при під'єднанні імпульсного "від'ємного" датчика:

```
WAITC: IN A, P1 ;ввід байта
        CPL A ;інверсія
        JB3 WAITC ;якщо P1.3=0, то чекати
WAIT0: IN A, P1 ;ввід байта
        JB3 WAIT0 ;якщо P1.3=0, то чекати
```

Програмна реалізація циклу чекання накладає обмеження на тривалість імпульсу: імпульси тривалістю менше часу виконання циклу чекання можуть бути "не помічені" МК.

3.6.4. Усунення деренчання контактів

При роботі МК із датчиками, що мають механічні чи електромеханічні контакти (кнопки, клавіші, реле і клавіатури), виникає явище, назване деренчанням. Це явище полягає в тому, що при замиканні контактів можлива поява відскоку контактів. При цьому сигнал з контакту може бути прочитаний МК як випадкова послідовність нулів і одиниць. Усунути це небажане явище можна схемотехнічними засобами з використанням буферного тригера, але частіше це робиться програмним шляхом.

Найпоширенішими є два програмних способи чекання сталого значення:

- 1) підрахунок заданого числа співпадаючих значень сигналу;
- 2) очікування на протязі інтервалу часу.

Суть першого способу полягає в багаторазовому зчитуванні сигналу з контакту. Підрахунок вдалих опитувань (тобто опитувань, які підтвердили, що контакт стійко замкнений) ведеться програмним лічильником. Якщо після серії вдалих опитувань зустрічається невдалий, то підрахунок починається спочатку.

Контакт вважається надійно замкненим (деренчання усунуто), якщо отримано N вдалих опитувань. Число N підбирається експериментально для кожного типу датчиків, що використовуються, і лежить у межах від 5 до 50.

Приклад програмного усунення деренчання контакту приводиться для випадку, коли датчик імпульсного сигналу під'єднаний до входу T0, підрахунок вдалих опитувань проводиться в регістрі R3, N=20:

```
DBNC:  MOV R3, #20      ;ініціалізація лічильника
DBNC1:  JT0 DBNC        ;якщо контакт розімкнений, то
                          ;почати відлік опитувань
                          ;спочатку
          DJNZ R3, DBNC1;декремент лічильника, і якщо
                          ;вміст лічильника не дорівнює 0,
                          ;то повторити аналіз стана
                          ;контакту
```

Усунення деренчання контакту шляхом введення очікування полягає в тому, що програма, знайшовши замикання контакту, забороняє опитування стана цього контакту на час, що є більший тривалості перехідного процесу. Програма написана для випадку під'єднання датчика до входу T0 і програмної реалізації очікування.

```
DBNCD:  JT0 DBNCD      ;чекання нуля на вході T0
CALL DELAY          ;виклик підпрограми очікування
JT0 DBNCD          ;перевірка нуля на вході T0
EXIT:          ;вихід із процедури
```

Тривалість очікування (у межах 1-10 мс) підбирається експериментально для кожного з датчиків і реалізується підпрограмою DELAY.

3.6.5. Підрахунок кількості імпульсів

Часто в керуючих програмах виникає необхідність чекання послідовності подій, що представляється послідовністю імпульсних сигналів від датчиків. Розглянемо дві типові процедури:

- ◇ підрахунок числа імпульсів між двома подіями;
- ◇ підрахунок числа імпульсів за заданий інтервал часу.

3.6.5.1. Підрахунок числа імпульсів між двома подіями

Цю типову процедуру зручно проілюструвати на конкретному прикладі. Припустимо, що необхідно підрахувати кількість деталей, що зійшли з конвеєра від моменту його включення до моменту вимикання. Факт сходу деталі з конвеєра фіксується фотоелементом, на виході якого формується імпульсний сигнал.

Для простоти реалізації програми вважаємо, що загальна кількість деталей не перевищує 99:

```
COUNT: CLR A ;скидання лічильника деталей
WAITC1: JTO WAITC1 ;чекання ввімкнення конвеєра
WAITC2: JTI WAITC2 ;чекання початку імпульсу
WAITO2: JNTI WAITO2 ;чекання кінця імпульсу
INC A; ;інкремент лічильника деталей
DA A ;десятькова корекція
JNTO WAITC2 ;якщо конвеєр не вимкнений, то
;продовжувати підрахунок
;інакше - вихід із процедури
```

Після закінчення виконання процедури в акумуляторі буде знаходитись кількість деталей, представлена в двійково-десятьковому коді.

Процедура підрахунку імпульсів може бути реалізована інакше, якщо використовувати вхід T0 не як тестовий, а як вхід лічильника подій:

```
COUNT2: CLR A
MOV T,A ;скидання лічильника
WAITC1: JTO WAITC1 ;чекання ввімкнення конвеєра
```

```

STRT CNT      ;запуск лічильника подій
JNT0 WAIT01 ;чекання виключення конвеєра
WAIT01: STOP TCNT      ;зупинка лічильника
MOV A,T       ;зчитування вмісту лічильника
. . . .

```

В акумуляторі фіксується кількість деталей, представлена в двійковому кодi (максимальна кількість деталей 255).

3.6.5.2. Підрахунок числа імпульсів за заданий проміжок часу

При вирішенні задачі перетворення число-імпульсного коду в двійковий, а також у ряді інших задач може виникнути необхідність підрахунку числа імпульсів за заданий інтервал часу. Ця процедура може бути реалізована трьома різними способами:

- ◇ програмною реалізацією часового інтервалу і програмним підрахунком числа імпульсів на вході МК;
- ◇ програмною реалізацією часового інтервалу й апаратурним підрахунком числа імпульсів (на внутрішньому таймері/лічильнику) ;
- ◇ апаратурною реалізацією часового інтервалу і програмним підрахунком числа імпульсів.

При апаратурній реалізації підрахунку числа подій імпульсний датчик повинен бути під'єднаний до входу T1 мікроконтролера:

```

STDLY: CLR A           ;скидання лічильника імпульсів
        MOV T,A
        STRT CNT       ;запуск лічильника
        CALL DELAY     ;виклик процедури очікування
        STOP TCNT      ;зупинка лічильника
        MOV A,T        ;фіксація результату

```

Підрахунок імпульсів виробляється лічильником подій, а відлік заданого часового інтервалу – підпрограмою DELAY.

При апаратурній реалізації відліку часового інтервалу імпульсні сигнали найзручніше приймати на вхід T0:

```

CTLDYT: MOV A, #0      ;настроювання таймера
        MOV T, A
        STRT T        ;запуск таймера
        MOV R1, A     ;скидання лічильника імпульсів
WAITC:  JTF STP       ;якщо часовий інтервал минув STP
        JTO WAITC    ;чекання імпульсу
WAITO:  JNTO WAITO
        JNC R1        ;інкремент лічильника
        JMP WAITC     ;перехід для продовження рахунку
STP:    STOP TCNT     ;зупинка таймера
EXIT:   ...           ;вихід із процедури

```

У наведеному прикладі таймер налаштований на максимально можливий часовий інтервал – 20 мс, лічильник імпульсів реалізований у регістрі R1. Перевірка завершення заданого часового інтервалу здійснюється за ознакою переповнення таймера (TF). Переривання від таймера повинно бути замасковане.

Оскільки апаратний таймер не може реалізувати інтервал очікування тривалістю понад 20 мс при збереженні швидкодії системи, "довгі" інтервали очікування повинні реалізовуватися програмно, наприклад, вони можуть набиратися з інтервалів у 20 мс із підрахунком числа переривань від таймера.

3.6.6. Опитування групи двійкових датчиків

Мікроконтролери найчастіше мають справу не з одним датчиком, як у розглянутих вище прикладах, а з групою автономних (логічно незалежних) чи взаємозалежних датчиків, що можуть формувати двійковий код. Групу взаємозалежних датчиків називають композицією. При цьому МК може виконувати процедуру опитування датчиків і передавання керування окремим фрагментам прикладної програми залежно від прийнятого коду.

Програмну реалізацію процедури чекання заданого коду (WTCODE) розглянемо для випадку під'єднання групи з восьми взаємозалежних статичних датчиків до входів порту P1 МК:

```

CODE EQU 10          ;визначення еталонного коду
WTCODE: IN A, P1     ;опитування групи датчиків
                XRL A, #CODE;порівняння прийнятого коду з
                ;заданим значенням CODE
                JNZ WTCODE ;якщо коди не співпали, то
                ;повторити ввід,
EXIT:    ...        ;інакше вихід із процедури

```

Порівняння прийнятого коду з заданим здійснюється операцією Виключне АБО. У наведеному прикладі число CODE дорівнює 10.

При опитуванні композиції двійкових датчиків умовний перехід зручно здійснювати по таблиці. Нижче наведено текст програми, що здійснює передавання керування одній з восьми прикладних програм PROG0-PROG7 (які розташовані в межах однієї сторінки пам'яті програм) залежно від кодової комбінації, набраної на перемикачах, під'єднаних до входів P1.0 - P1.2:

```

GOCODE:MOV R0,#LOW BASE ;завантаження в R0 початкової
                ;адреси таблиці переходів
                IN A, P1          ;ввід байта
                ANL A, #00000111B;виділення молодших бітів
                ADD A, R0         ;формування адреси рядка
                ;у таблиці переходів
                JMPP @A           ;передавання керування
BASE:    DB LOW PROG0           ;таблиця переходів
                ...
                DB LOW PROG7

```

Програма забезпечує опитування і виділення сигналів від трьох датчиків шляхом маскування старших бітів акумулятора. Адреса рядка таблиці, у якій зберігаються адреси переходів, обчислюється як сума відносного (усередині поточної сторінки РПП) початкової адреси таблиці BASE і коду, прийнятого від датчиків. Команда JMPP @A, таблиця BASE і програми PROG0 - PROG7 повинні розташовуватися в межах однієї сторінки пам'яті програм (ПП).

3.6.7. Опитування групи імпульсних датчиків

Ця процедура складається з послідовності таких дій:

- ◇ чекання замикання одного з контактів;
- ◇ усунення деренчання;
- ◇ чекання розімкнення замкненого контакту.

Програмна реалізація процедури для випадку під'єднання чотирьох імпульсних датчиків до входів 0-3 порту P1 буде мати вигляд:

```
KBRD:   IN A, P1           ;ввід коду
        CPL A             ;інверсія коду
        ANL A, #00001111B ;є замкнений контакт?
        JZ KBRD           ;якщо жоден контакт не
                           ;замкнений, то чекати
        MOV R2, A         ;пересилання прийнятого
                           ;коду в R2

DBNC:   CALL DELAY       ;усунення деренчання

WAIT:   IN A, P1         ;ввід коду
        CPL A             ;інверсія коду
        ANL A, #00001111B ;є замкнений контакт?
        JNZ WAIT         ;якщо контакт замкнений,
                           ;то чекати,

EXIT:   ...              ;інакше вихід із процедури
```

Аналіз стана контактів здійснюється накладанням маски на прийнятий від датчиків код. Для датчиків, що формують "від'ємний" імпульс, прийнятий код попередньо інвертують.

Для групи імпульсних датчиків, що являють собою клавiшний реєстр, процедура KBRD повинна бути доповнена процедурою ідентифікації натисненої клавiші і захисту від одночасного натискання двох і більше клавiш.

Ідентифікація натисненої клавiші може здійснюватися двома способами: за таблицею або програмно. За табличного способу перекодування в пам'яті програм повинна знаходитися таблиця двійкових еквівалентів кодів клавiш.

Програмне перетворення унітарного коду, прийнятого від клавіатури, в двійковий, може бути виконано методом зсуву вихідного унітарного коду і підрахунком числа зсувів лічильником до появи першого переносу.

```
IDNKEY: MOV A,R2
        MOV R1,#0           ;скидання лічильника зсувів
        CLR C               ;скидання ознаки переносу
ROTATE: RRC A               ;зсув унітарного коду
        JC CHECK           ;якщо виник перенос, то
                               ;перевірка
                               ;на одночасне натискання
        INC R1              ;інкремент лічильника зсувів
        JMP ROTATE         ;продовження зсуву
CHECK:  JZ EXIT            ;якщо (A)=0,то вихід із
                               ;процедури
        MOV R1, #0FFH      ;занесення в R1 коду
                               ;одночасного натискання
                               ;кількох клавіш
EXIT:   ...                ;вихід із процедури
```

Результат зберігається в регістрі R1. У програмі передбачається, що в R2 знаходиться інверсія унітарного коду, прийнятого від групи датчиків процедурою KBRD. У результаті роботи програми IDNKEY в R1 буде сформований двійковий код натисненої клавіші чи код (FFH) "натиснено кілька клавіш".

3.7. Контрольні запитання

1. Однокристальні мікроЕОМ сімейства МК48. Основні характеристики.
2. Архітектура ОМЕОМ К1816ВЕ48.
3. Арифметико-логічний пристрій.
4. Організація резидентної пам'яті даних та пам'яті програм.
5. Система переривань.
6. Лічильник-таймер.

7. Порти вводу-виводу.
8. Функціонування МК48.Режими роботи. Передача сигналів через порти.
9. Адресний простір пам'яті.
10. Під'єднання до МК48 зовнішньої пам'яті даних та програм.
11. Спряження МК48 з розширювачем вводу-виводу.
12. Під'єднання до МК48 розширеної кількості давачів. Під'єднання індикаторів та клавіатури.

ТЕМА №4. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ ОДНОКРИСТАЛІЧНИХ МІКРОЕОМ СЕРІЇ K1816BE751

4.1. Однокристальні мікроЕОМ сімейства МК51 (MCS51).

Основні характеристики

ОМЕОМ сімейства МК51 відносяться до 8-розрядних мікроЕОМ. Розроблені за n-мон технологією, 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 випускають тільки ОМЕОМ, виконані за КМОН технологією. Базовим кристалом сімейства є ОМЕОМ K1816BE751 (87C51), який має характеристики, наведені в таблиці 26.

Таблиця 26. Основні характеристики мікроЕОМ

Параметр	Значення
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 87C51 входять наступні додаткові пристрої:

- ◇ чотири восьмирозрядні паралельні порти вводу-виводу;
- ◇ два 16-ти розрядні таймери - лічильники;
- ◇ послідовний порт;
- ◇ тактовий генератор;
- ◇ блок регістрів спеціальних функцій;
- ◇ система захисту пам'яті програм від несанкціонованого доступу.

Система команд ОМЕОМ 87С51 містить 111 базових команд. Дворівнева система переривань підтримує переривання від 5 джерел. Керамічний корпус DIP має вбудоване вікно, закрите кварцовим склом, для стирання ультрафіолетовим випромінюванням записаної в ЗУПП програми.

Деякі виробники випускають мікросхеми, що мають додаткові функціональні можливості, а саме: збільшену, або зменшену кількість виводів, підвищену швидкодію, збільшену кількість перелічильників, можливість внутрисхемного програмування, тощо.

4.2. Архітектура ОМЕОМ 80С51

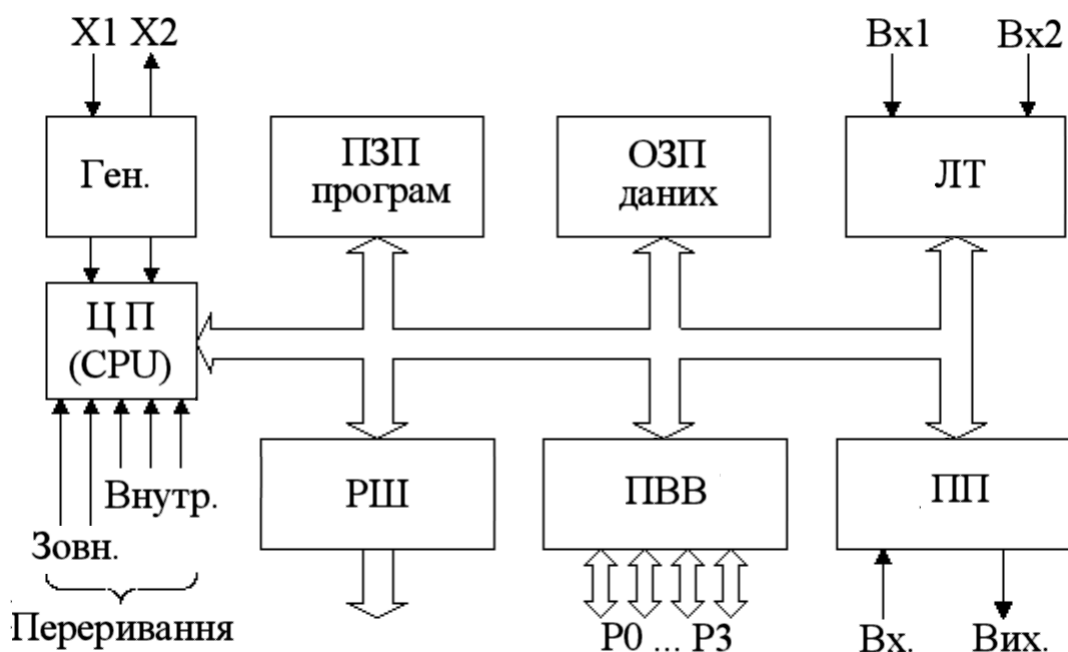


Рис. 61. Архітектура мікроЕОМ

Пам'ять програм і пам'ять даних в ОМЕОМ сімейства МК51 не тільки фізично і логічно розділені, але мають різні тип і систему адресації. Водночас, для звертання до інформації, що міститься в пам'яті даних і пам'яті програм, використовується та сама восьмирозрядна шина даних.

До складу ОМЕОМ входять такі функціональні вузли (рис. 61):

- ◇ ЦП – центральний процесор (англ. CPU – Central Processor Unit);
- ◇ ПЗП програм – постійний запам'ятовуючий пристрій призначений для збереження програми користувача;
- ◇ ОЗП даних – оперативний запам'ятовуючий пристрій, який

використовується для збереження даних;

- ◇ Ген. – генератор тактових сигналів;
- ◇ ПВВ – порти вводу-виводу;
- ◇ ПП – послідовний порт з програмованим режимом та швидкістю роботи;
- ◇ ЛТ – лічильники-таймери, два 16-розрядних лічильники з можливістю відліку часу;
- ◇ РШ – розширювач шини для роботи з зовнішньою пам'яттю ємністю до 64 Кбайт.

Усі вузли зв'язані між собою загальною восьмирозрядною шиною. ЦП є сукупністю операційного і керуючого пристроїв, що виконують програму, записану в ПЗП програм, ємність якого 4Кбайт. ЦПП забезпечує виконання таких груп операцій:

- ◇ арифметичні операції (додавання, додавання з урахуванням перенесення, віднімання з урахуванням позичання, беззнакове множення і ділення, інкремент і декремент, десяткова корекція).
- ◇ логічні операції (І, АБО, виключне АБО, інверсія);
- ◇ операції зсуву;
- ◇ операції пересилання;
- ◇ бітові операції;
- ◇ операції переходу та виклику підпрограм керування.

Сам ЦП розділений на керуючий пристрій КП і чотири блоки, що є операційним пристроєм: АЛУ, регістри тимчасового збереження операндів TR1 і TR2, та регістр ознак PSW (англ. Program Status Word).

Проміжні результати обчислень зберігаються в ОЗП ПД (RAM) ємністю 128 байт. Крім того, в адресному просторі ПД розміщені всі регістри спеціальних функцій, які задають стан портів, таймери, регістр акумулятор (А), регістр В, регістр вказівника стека SP, з блоку РШ в адресний простір ПД входить вказівник даних DPTR (англ. Data PointeR) з асоційованими шинами обміну і лічильником команд PC (англ. Program Counter), призначення якого

полягає у формуванні адреси команди. Крім того, в ПД розміщується стек, у якому можуть зберігатись адреси повернення з підпрограм та локальні змінні підпрограм.

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

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

Ввід у процесор інформації, що опрацьовується, може бути здійснений або в паралельній байтовій (ввід восьми розрядів однією командою), або в послідовній (по одному біту) формах також як і вивід результатів. 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 й економії обчислювальних ресурсів, необхідних для його реалізації, ОМЕОМ містить вбудований програмований послідовний порт, що дозволяє практично без витрат обчислювальних ресурсів організувати послідовний обмін по кількох видах протоколів.

Крім розглянутих вузлів, до складу ОМЕОМ входять два шістнадцятирозрядні таймери/лічильники, що можуть функціонувати або в режимі таймера, або в режимі лічильника зовнішніх подій. Режим таймера використовується, головним чином, коли необхідно організувати циклічні процеси з жорстко фіксованим і незалежним від часу виконання програми періодом циклу, наприклад, при опрацюванні сигналів, коли необхідно забезпечити необхідний інтервал дискретизації. Режим лічильника зовнішніх подій використовується, наприклад, при підрахунку кількості імпульсів,

вимірюванні 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, коли для розміщення програми чи даних при вирішенні якоїсь задачі внутрішніх ресурсів ОМЕОМ виявляється недостатньо. Режим роботи з зовнішньою пам'яттю не є типовим для ОМЕОМ.

Режим звертання до зовнішнього ЗП, використовується не тільки за прямим призначенням. Так як зовнішня шина універсальна, то до неї можна під'єднати будь-який спеціалізований пристрій, ВІС комплектів КР580 і КР1810 або навіть спеціалізовану мікроЕОМ і керувати їх роботою безпосередньо з 87С51.

4.2.1. Умовне позначення та призначення виводів мікроЕОМ

Слід зауважити, що на відміну від МК-48 в мікроЕОМ цього сімейства більшість виводів має подвійне функціональне призначення. Наприклад, при використанні зовнішнього ОЗП його адресація здійснюється через виводи портів P0 (молодший байт) і P2 (старший байт), а обмін інформацією (ввід кодів команд, ввід\вивід даних) – через виводи порту P0 при цьому керування обміном здійснюється через дві лінії порту P3. Тому, якщо не застосовувати додаткові апаратні засоби, ці порти за наявності зовнішнього ОЗП не можна використати для інших цілей.

Таймери/лічильники, послідовний порт і система переривань не мають своїх виводів з корпусу ОМЕОМ, а використовують виводи порту P3, і, таким чином, використання таймерів/лічильників у режимі лічильників зовнішніх подій, ліній послідовного порту і зовнішніх переривань знижує розрядність порту P3.

Специфічну для даної лінії функцію, яку виконує порт, називають альтернативною функцією лінії порту. А функцію простого вводу-виводу називають основною функцією порту.

Позначення ОМЕОМ на електричній схемі зображено на рис. 62, а

призначення виводів наведено в таблиці 27.

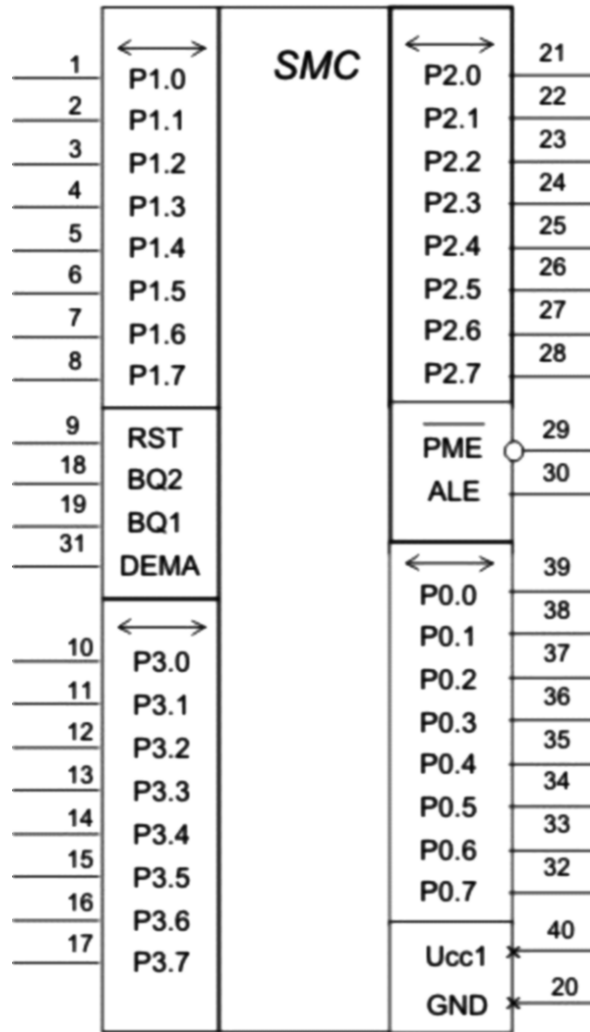


Рис. 62. Позначення КР1816BE751

Зауважимо, що лінії портів P1-P3 є квазідвонаправленими – фактично вони працюють як виводи з відкритим колектором, у яких внутрішні підтягуючі до високого рівня резистори знаходяться в середині мікроЕОМ. При встановленні відповідного біта порту в стан логічної 1 вивід видає невеликий струм, що обмежується підтягуючим резистором, а при лог. 0 вивід приймає досить значний струм. Це полегшує узгодження виводу з входами і виходами ТТЛ мікросхем.

Лінії порту P0 аналогічні лініям портів P1-P3, за винятком того, що внутрішні підтягуючі резистори в них взагалі відсутні – лінії можуть лише приймати струм у стані логічного нуля. Проте при використанні ліній як виходів навантажених на входи ТТЛ внутрішньої підтяжки ТТЛ-входу достатньо для нормальної роботи виводу.

При зчитуванні стана лінії можливі два типи команд, які зчитують стан лінії (команди вводу з порту), або зчитують стан внутрішнього регістра.

При зчитуванні стана лінії необхідно, щоб на відповідному виході ОМЕОМ був заздалегідь виставлений лог. 1. У протилежному випадку лінія буде "посаджена" в стан лог. 0 і цей нуль і буде зчитаний.

Таблиця 27. Призначення виводів мікросхеми

Позначення	Номер виводу	Призначення
1	2	3
P1.0-P1.7	1-8	Порт вводу-виводу 1. Виводи паралельного порту P1
RST	9	Сигнал скидання мікросхеми або вхід напруги програмування внутрішнього ПЗП. Логічний 0 дозволяє нормальну роботу мікросхеми, логічна 1 протягом двох машинних циклів переводить мікросхему у режим скидання, рівень напруги 12 В на виводі призводить до програмування мікросхеми
P3.0/RxD	10	Лінія 0 порту вводу-виводу P3 може працювати як вхід даних у послідовному коді
P3.1/TxD	11	Лінія 1 порту вводу-виводу P3 може працювати як вихід даних у послідовному коді
P3.2/ $\overline{\text{INT0}}$	12	Лінія 2 порту вводу-виводу P3 може працювати як вхід запиту переривання по переходу сигналу в стан лог. 0
P3.3/ $\overline{\text{INT0}}$	13	Лінія 3 порту вводу-виводу P3 може працювати як вхід запиту переривання по переходу сигналу в стан лог. 0
P3.4/T0	14	Лінія 4 порту вводу-виводу P3 може працювати як вхід відліку внутрішнього лічильника 0
P3.5/T1	15	Лінія 5 порту вводу-виводу P3 може працювати як вхід відліку внутрішнього лічильника 1
P3.6/ $\overline{\text{WR}}$	16	Лінія 6 порту вводу-виводу P3 може працювати як сигнал записування у зовнішній ЗП даних, або порт вводу-виводу
P3.7/ $\overline{\text{RD}}$	17	Лінія 7 порту вводу-виводу P3 може працювати як сигнал читання зовнішнього ЗП даних, або порту вводу-виводу
BQ2 (X2)	18	Вивід для під'єднання зовнішнього резонатора
BQ1(X1)	19	Вивід для під'єднання зовнішнього резонатора
GND	20	Напруга 0 В
P2.0/(A8)- P2.7/(A16)	21-28	Порт вводу-виводу 2, лінії порту також використовуються при адресації зовнішньої пам'яті та пристроїв вводу-виводу для виведення старшого байта адреси

1	2	3
PME	29	Сигнал читання зовнішньої пам'яті програм. Логічний 0 на лінії вказує, що відбувається читання зовнішньої пам'яті програм
ALE	30	При нормальній роботі це вихід стробуючого сигналу адреси. Перехід сигналу зі стана лог. 1 в стан лог. 0 вказує на наявність адреси на лініях порту 0. У режимі програмування лінія є входом, на який надходить імпульс програмування
DEMA	31	Вхід заборони роботи внутрішньої пам'яті програм. Логічна 1 на вході дозволяє звертання до внутрішньої пам'яті програм, при логічному 0 на вході при звертанні по адресах, що відповідають внутрішній пам'яті програм, звертання відбувається до зовнішньої пам'яті програм
P0.0-P0.7	39-32	Порт вводу-виводу 0, лінії порту також використовуються при звертанні до зовнішньої пам'яті програм та даних, а також до пристроїв вводу-виводу для виведення молодшого байта адреси і та обміну даними
Vcc1	40	Напруга живлення 5В

Мінімальна схема ввімкнення ОМЕОМ 87С51 зображена на рис. 63.

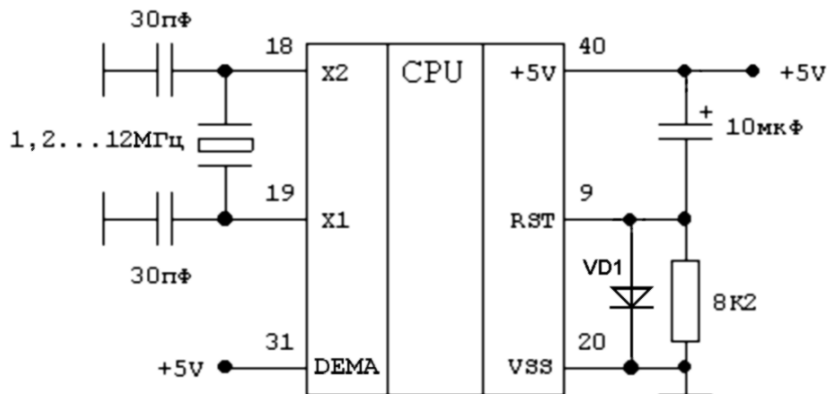


Рис. 63. Мінімальна схема ввімкнення ОМЕОМ 87С51

До виводів X1 і X2 ОМЕОМ під'єднана резонансна схема, що містить кварцовий резонатор. До входу скидання RST під'єднана схема автоматичного рестарту ОМЕОМ при ввімкненні живлення.

Коли функціонально-логічних можливостей однокристалного МК недостатньо, можливе розширення МК-системи відносно простими методами з можливістю отримання таких значень параметрів: пам'ять програм – до 64 Кбайт; пам'ять даних – до 64 Кбайт; лінії вводу-виводу – практично необмежено.

Крім того, шляхом під'єднання спеціалізованих ВІС, що входять до мікропроцесорного комплекту КР580, у МК-системі можуть бути реалізовані різні допоміжні функції: зв'язок з дисплеєм і клавіатурою, багаторівнева система переривань, зв'язок з телеграфно-телефонними лініями передавання інформації і т.д. за схемами, які використовуються для роботи з ОМЕОМ КР1816ВЕ48.

4.3. МК - системи з зовнішньою пам'яттю програм

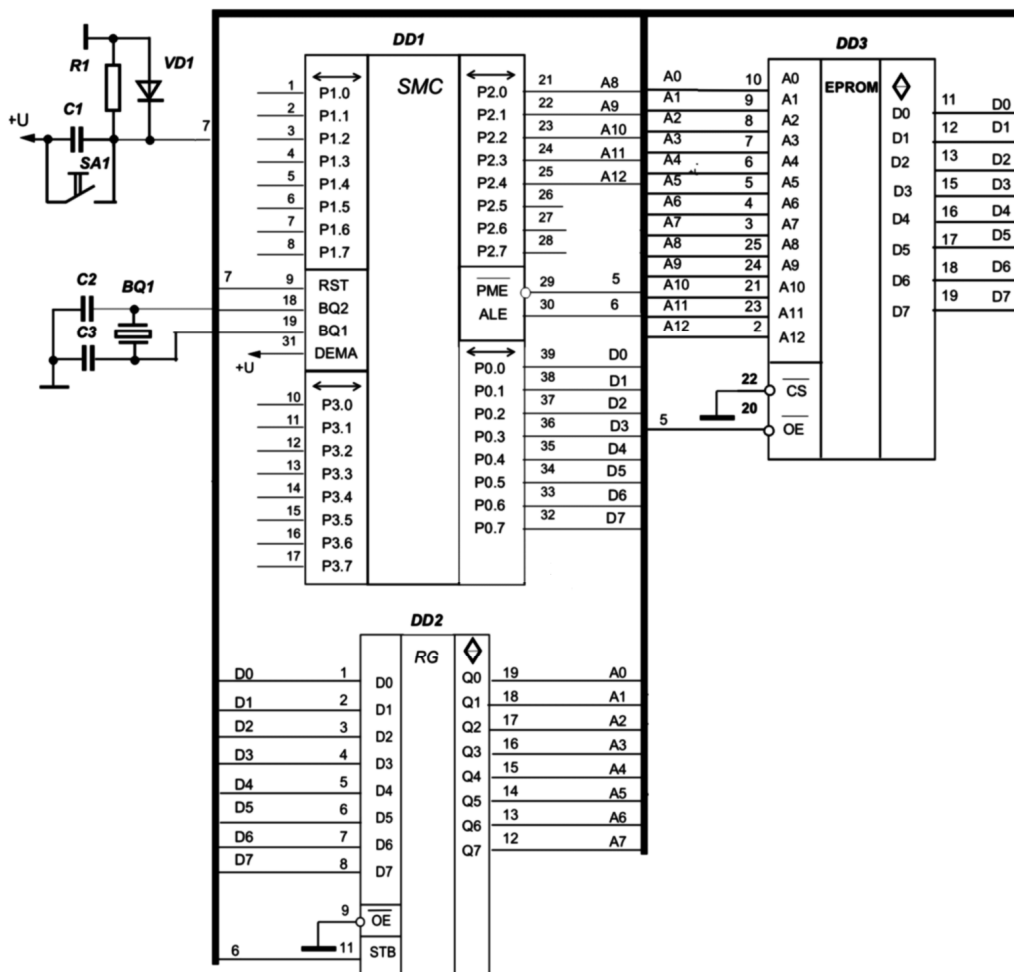


Рис. 64. Під'єднання ПЗП

На рис. 64 зображено структуру МК-системи з зовнішньою пам'яттю програм. Шина 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 ідентифікує кожен машинний цикл, проте в деяких випадках він теж може бути відсутній). 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 P0 переходить у режим вводу.

Додаткова мікросхема пам'яті (DD3) ємністю 8 Кбайт під'єднується до шини P0 своїми інформаційними виходами. Молодший байт адреси за сигналом ALE фіксується на зовнішньому буферному регістрі DD2. Старша тетрада адреси, виведена через порт P2, не має потреби в буферизації, тому що вона зберігається протягом усього циклу вибірки. У випадку використання великої кількості мікросхем ПЗП, ОЗП та інших пристроїв, під'єднаних до шини даних, шину даних однокристальної мікроЕОМ необхідно буферизувати за допомогою мікросхеми K555АП6, або аналогічної.

4.4. Розширення ОЗП

На рис. 65 зображено схему МК-системи, до складу якої входить додаткова мікросхема ОЗП (DD3), на основі якої реалізується пам'ять ємністю 2 Кбайт.

Сигналом ALE непряма адреса, виведена по шині P0, фіксується в буферному регістрі DD2. Сигнали W і R визначають режим роботи ОЗП. Схема, що на рис. 65, забезпечує адресацію 2Кбайт комірок ОЗП на додаток до 256 комірок резидентної пам'яті даних (ВПД) МК51.

Молодші вісім розрядів шини адреси при звертанні до ОЗП видаються на порт P0 і запам'ятовуються в зовнішньому регістрі DD2. Старші розряди шини адреси при використанні команд, що працюють із регістром DPTR, видаються на лінії порту P2 і утримуються там протягом усього циклу. Звертання до зовнішнього ОЗП з використанням 8 бітної адресації не змінює стан порту P2,

отже, вимагає попереднього встановлення потрібного значення старших розрядів адреси за допомогою порту P2.

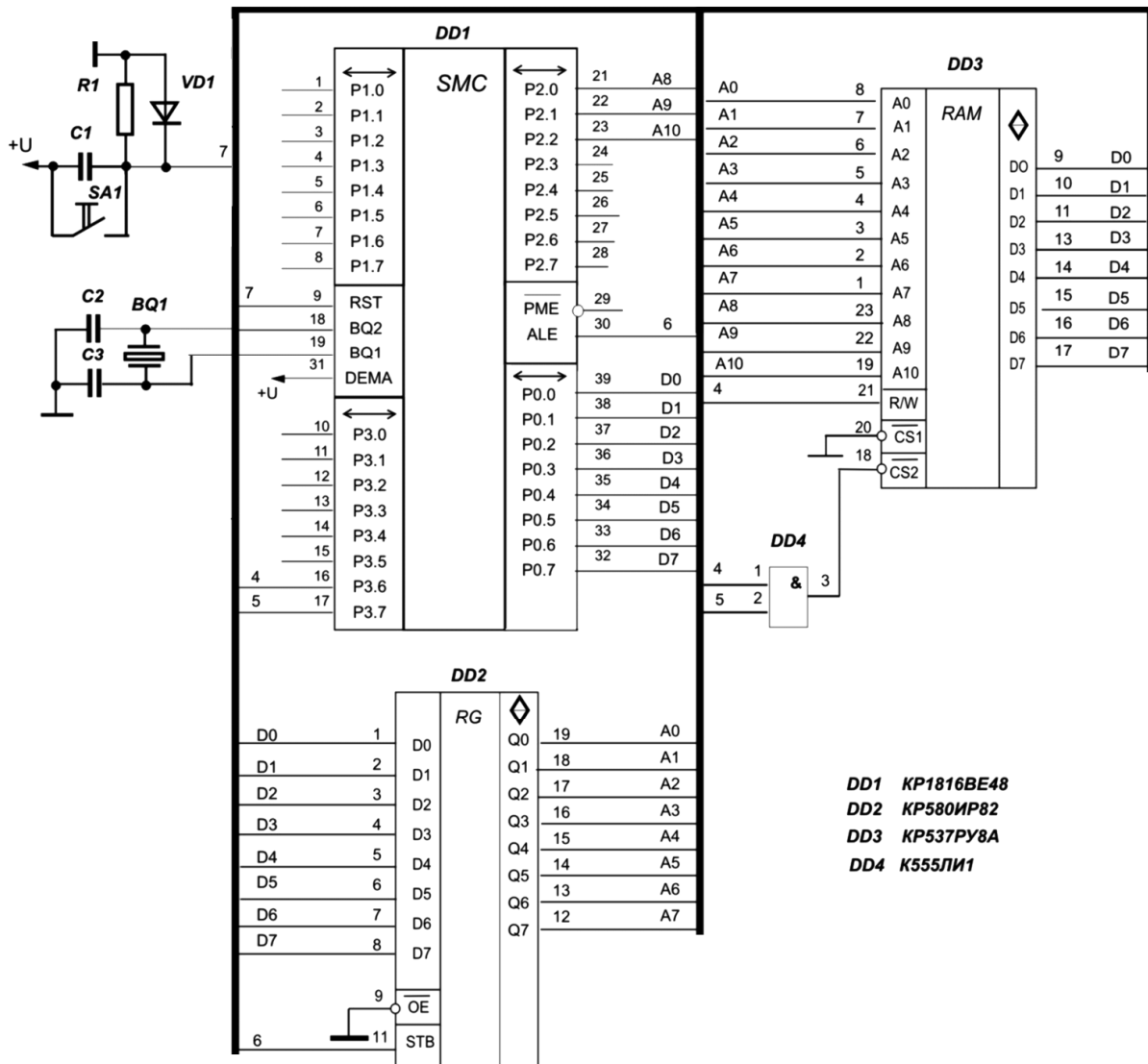


Рис. 65. Розширення пам'яті даних KP1816BE51

4.5. Ввід-вивід у МК-системах

Для з'єднання МК з об'єктом, що має велику кількість входів-виходів, можна розширити резидентну систему вводу-виводу за рахунок зовнішніх портів. Таке розширення, як і для МК48, можна виконати з використанням: стандартного розширювача вводу-виводу (РВВ) KP580BP43, або інтерфейсних ВІС (KP580BB55, KP580BB51).

Розширювач під'єднується до МК51 так, як показано рис. 66. Кожен із чотирьох портів РВВ може використовуватися для введення чи виведення інформації незалежно від інших і забезпечує високу навантажувальну

здатність.

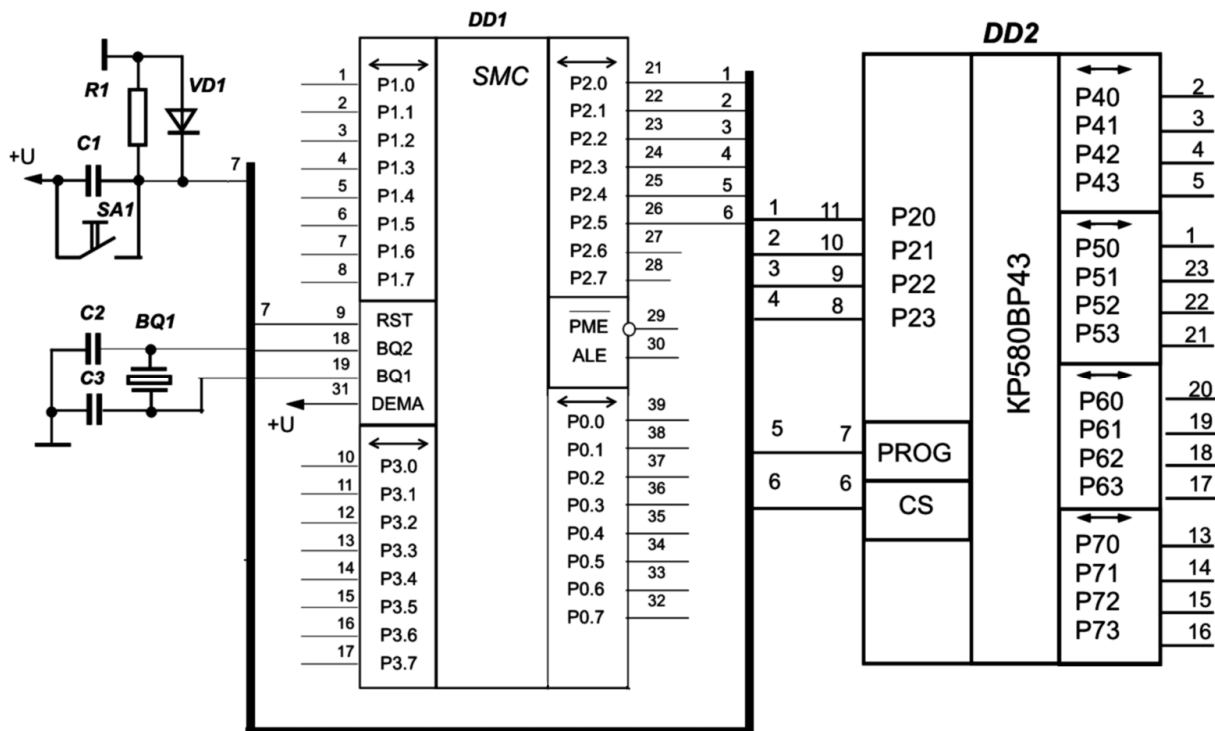


Рис. 66. Під'єднання розширювача вводу-виводу

Розширення ПВВ за допомогою ППІ серії КР580 здійснюється аналогічно схемам для МК48, зображених на рис. 55. Прикладом такої схеми може бути схема, що на рис. 67.

Мікросхема КР580ВВ55 для мікроЕОМ відображається на комірці зовнішнього ОЗП даних. Формування адреси здійснюється за допомогою регістра DD2, який фіксує адресу, котра видається мікросхемою в циклі звертання до зовнішньої пам'яті даних. Старший біт адреси надходить на сигнал вибірки мікросхеми. Це зроблено для спрощення системи. У випадку необхідності використання кількох мікросхем КР580ВВ55, їх вибірку можна здійснювати за допомогою розрядів адреси А2-А7, під'єднаних до сигналів вибірки відповідних мікросхем. У випадку великої кількості необхідних мікросхем формування сигналів вибірки можна здійснювати за допомогою дешифратора, на який подаються старші біти адреси, або з використанням для формування сигналів вибірки ліній порту P1 або P2.

Під'єднання клавіатури та дисплеїв у мікропроцесорних системах на основі МК51 виконується за тими ж правилами, що і для схем на основі МК48.

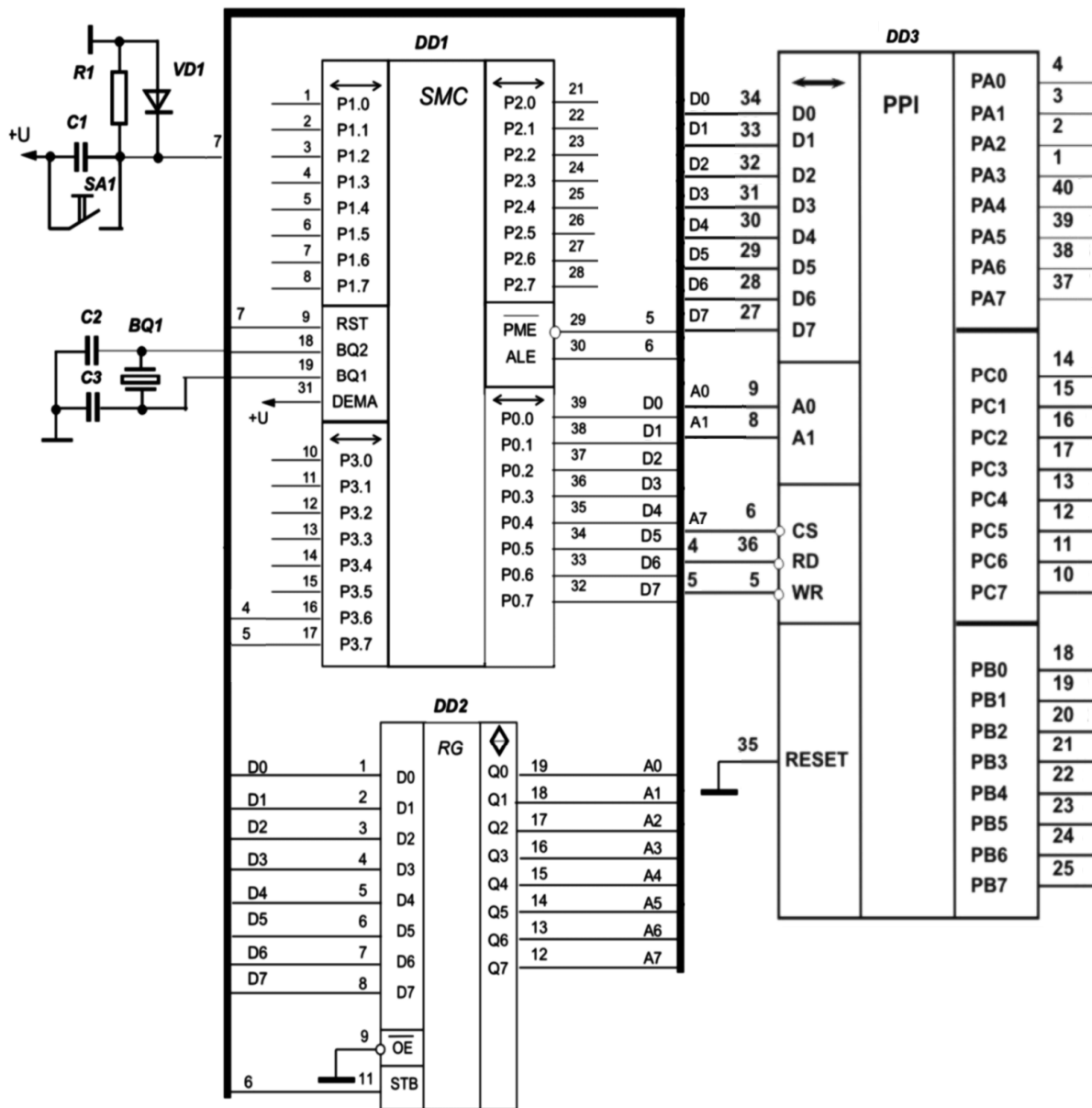


Рис. 67. Приклад під'єднання мікросхем серії КР580 до мікроЕОМ

Організація більшості програмних процедур для МК51 також є такими, як і для МК48. З урахуванням того, що в МК-51 кожен вивід кожного порта може бути перевірений одною командою без необхідності завантаження стану порту в акумулятор.

4.6. Лічильник-таймер у МК51

Два програмованих 16-бітних лічильники/таймери (Т/С0 і Т/С1) можуть бути використані в якості таймерів або лічильників зовнішніх подій. При роботі в якості таймера вміст Т/С інкрементується у кожному машинному циклі, тобто через кожні 12 періодів резонатора. При роботі в якості лічильника вміст Т/С

інкрементується під впливом переходу з 1 у 0 зовнішнього вхідного сигналу, що подається на відповідний вивід MCS51 (T0 або T1). Опитування значення зовнішнього вхідного сигналу виконується в момент часу S5P2 кожного машинного циклу. Вміст лічильника буде збільшений на 1 у тому випадку, якщо в попередньому циклі був зчитаний вхідний сигнал високого рівня (1), а в наступному – сигнал низького рівня (0). Нове (інкрементоване) значення лічильника буде сформоване в момент S3P1 у циклі, що йде за тим, у якому був виявлений перехід сигналу з 1 у 0. На розпізнавання переходу потрібно два машинних цикли, тому максимальна частота підрахунку вхідних сигналів дорівнює 1/24 частоти резонатора. На тривалість періоду вхідних сигналів обмежень зверху немає. Для гарантованого зчитування вхідного сигналу, що рахується, він повинен утримувати значення лог. 1 як мінімум протягом одного машинного циклу MCS51.

Старшій та молодшій байти лічильника носять назву T_{Nx} та T_{Lx}. Керування лічильниками здійснюється загальним регістром режиму TMOD, та регістром конфігурації TCON, стан лічильників відображається у регістрі TCON. Опис регістрів приведено в таблицях 28 та 30, структура регістрів зображена на рис. 68 та 69.

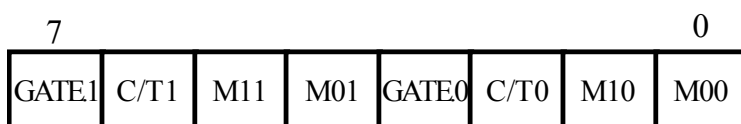


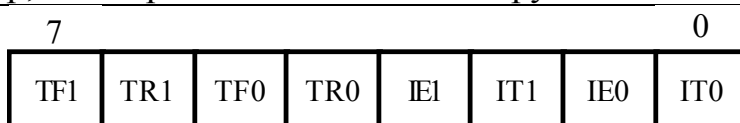
Рис. 68. Формат регістра режиму TMOD

Таблиця 28. Регістр режиму роботи таймера/лічильника TMOD

Ознака	Ім'я і призначення
GATE _x	Керування блокуванням. Якщо біт встановлено, то таймер/лічильник "x" дозволений доти, поки на вході "INT _x " високий рівень і біт керування "TR _x " встановлений. Якщо біт скинутий то T/C _x дозволяється, як тільки біт керування "TR _x " встановлюється
C/T _x	Біт вибору режиму таймера або лічильника подій. Якщо біт скинутий, то блок працює в режимі таймера від внутрішнього джерела сигналів синхронізації. Якщо біт встановлено, то блок працює в режимі лічильника від зовнішніх сигналів на вході "T _x "
M1. _x , M0. _x	Режим роботи, що вибирається згідно таблиці 29

Таблиця 29. Режим роботи таймера в залежності від біт настроювання

M1	M0	Режим роботи
0	0	Таймер сумісний з МК48. TL працює як 5-бітний попередній подільник, TH – у режимі сумісному з таймером K1816BE48
0	1	16-бітний лічильник/таймер. TL і TH включені послідовно
1	0	8-бітний лічильник/таймер, що перевантажується. TH зберігає значення, що повинно бути перезавантажене в TLx у момент переповнення
1	1	Таймер/лічильник 1 зупиняється. У таймері/лічильнику 0 TL0 працює як 8-бітний таймер/лічильник, і його режим визначається керуючими бітами таймера 0. TH0 працює тільки як 8-бітний таймер, і його режим визначається керуючими бітами таймера 1

**Рис. 69. Формат керування/статусу таймера**

Таблиця 30. Регістр керування/статусу таймера

Ознака	Ім'я і призначення
TFx	Ознака переповнення таймера x. Встановлюється апаратно при переповненні таймера/лічильника. Скидається при обслуговуванні переривання теж апаратно
TRx	Біт керування таймера x. Встановлюється/скидається програмою для його пуску (1) та зупинки (0)
IEx	Ознака запиту переривання x. Встановлюється апаратно, коли детектується активний перехід сигналу INTx. Скидається при обслуговуванні переривання
ITx	Біт керування типом переривання x. При ITx = 0 активним є перехід 1→0, при IT = 0 активним є рівень лог. 0 на вході INTx

Як впливає з опису керуючих бітів TMOD, для обох лічильників режими роботи 0, 1 і 2 однакові.

4.6.1. Режими роботи таймера

Режим 0. Переведення будь-якого лічильника/таймера в режим 0 робить його схожим на лічильник/таймер K1816BE48, на вхід котрого під'єднано подільник частоти на 32 (рис. 70a). У цьому режимі таймерний регістр має розрядність 13 бітів. При переході зі стану "всі одиниці" у стан "всі нулі" встановлюється ознака переривання від таймера TFx. Вхідний синхросигнал таймера дозволений (поступає на вхід Tx), коли керуючий біт TR1 встановлений в 1 і або керуючий біт GATE дорівнює 0, або на зовнішній вивід запиту

переривання INT1 надходить рівень 1.

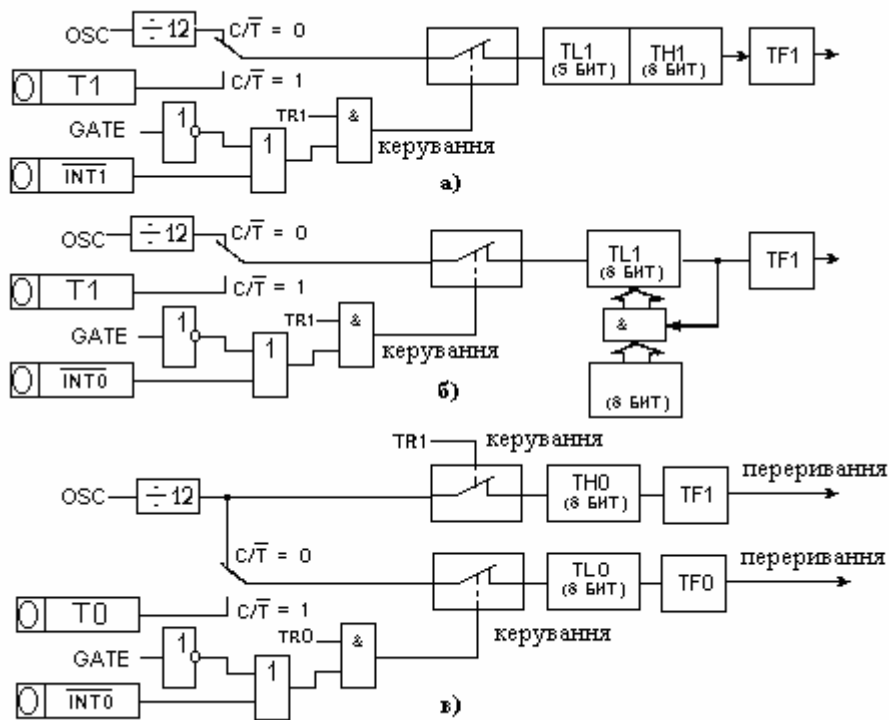


Рис. 70. Таймер/лічильник подій:

а – у режимі 0 13-бітний лічильник; б – у режимі 2: 8-бітний лічильник з перезавантаженням; в – T/C0 у режимі 3: два 8-бітних лічильники.

Встановлення біта GATE в 1 дозволяє використовувати таймер для вимірювання тривалості імпульсного сигналу, що подається на вхід запиту переривання.

Режим 1. Робота будь-якого T/C у режимі 1 така ж, як і в режимі 0, за винятком того, що таймерний регістр має розрядність 16 бітів.

Режим 2. У режимі 2 (рис. 70б) робота організована таким чином, що переповнення (перехід із стану "усі одиниці" у стан "усі нулі") 8-бітного лічильника TLx призводить не тільки до встановлення ознаки TF, але й автоматично перезавантажує у TLx вміст старшого байта (THx) таймерного регістра, що попередньо було задано програмним шляхом. Перезавантаження залишає вміст THx незмінним. У режимі 2 лічильники/таймери працюють ідентично.

Режим 3. У режимі 3 лічильники/таймери працюють по-різному. Лічильник/таймер 1 зберігає незмінним свій поточний вміст, іншими словами,

ефект такий же, як і при скиданні керуючого біта TR1 в нуль. Роботу лічильника/таймера 0 у режимі 3 проілюстровано на рис. 70в.

У режимі 3 TL0 і TH0 функціонують як два незалежних 8-бітних лічильники. Роботу TL0 визначають керуючі біти T/C0 (C/T, GATE, TR0), вхідний сигнал INT0 і ознака переповнення TF0. TH0 може виконувати тільки функції таймера, його роботу визначає керуючий біт TR1, TH0 використовує ознаку переповнення TF1.

Режим 3 використовується у випадках, коли потрібна наявність додаткового 8-бітного таймера або лічильника подій. Можна вважати, що в режимі 3 MCS51 має в своєму складі три лічильники/таймери. У тому випадку, якщо лічильник/таймер 0 використовується в режимі 3, лічильник/таймер 1 може бути або ввімкнений, або вимкнений, або переведений в свій власний режим, або може бути використаний послідовним портом у якості генератора частоти, або, нарешті, може бути використаний для виконання будь-якої функції, що не вимагає переривання.

4.7. Послідовний інтерфейс

4.7.1. Універсальний асинхронний прийомопередавач

Суттєвою відмінністю МК51 від розглянутих вище мікропроцесорів і мікроЕОМ є наявність на кристалі процесора блоків, що дозволяють апаратно реалізувати канал послідовного зв'язку.

Через послідовний порт, який часто називають універсальним асинхронним прийомопередавачем (УАПП), здійснюється приймання і передавання інформації, поданої послідовним кодом (молодшими бітами вперед), у повному дуплексному режимі обміну. До складу УАПП входять приймаючий та передаючий зсувні регістри, а також спеціальний буферний регістр (SBUF) прийомопередавача. Записування байта в буфер призводить до автоматичного переписування байта в зсувний регістр передавача й ініціює початок передавання байта. Наявність буферного регістра приймача дозволяє суміщати операцію читання раніше прийнятого байта з прийманням чергового

байта. Якщо до моменту закінчення приймання байта попередній байт не був зчитаний із SBUF, то він буде втрачений.

Послідовний порт MCS51 може працювати в чотирьох різних режимах. У режимі 0 інформація передається і приймається через зовнішній вивід входу приймача (RXD). Приймаються або передаються 8 біт даних. Через зовнішній вивід виходу передавача (TXD) видаються імпульси зсуву, що супроводжують кожен біт. Частота передавання біта інформації дорівнює 1/12 частоти резонатора.

У режимі 1 передаються через TXD або приймаються з RXD 10 біт інформації: старт-біт (0), 8 біт даних і стоп-біт (1). Швидкість приймання/передавання змінна і задається таймером.

У режимі 2 через TXD передаються або з RXD приймаються 11 біт інформації: старт-біт, 8 біт даних, дев'ятий біт, стан якого програмується, та стоп-біт. При передаванні дев'ятий біт даних може приймати значення 0 або 1, або, наприклад, для підвищення надійності в нього може бути поміщене значення ознаки паритету із слова стана програми (PSW.0). Швидкість обміну даними вибирається програмою і може дорівнювати або 1/32, або 1/64 частоти резонатора залежно від керуючого біта SMOD.

Режим 3 аналогічний режиму 2, проте швидкість обміну даними змінна і задається таймером.

4.7.2. Регістр керування/статусу УАПП

Керування режимом роботи УАПП здійснюється через спеціальний регістр SCON. Він містить не тільки керуючі біти, що визначають режим роботи послідовного порту, але й дев'ятий біт прийнятих або переданих даних (RB8 і TB8) і біти переривання прийомопередавача (RI і TI).

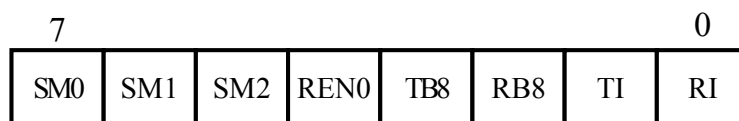


Рис. 71. Формат регістра керування та статусу УАПП (SCON)

Формат регістра, що керує роботою УАПП зображено на рис. 71.

Призначення бітів регістра керування/статусу УАПП приводиться у таблиці 31.

Таблиця 31. Регістр керування/статусу УАПП

Символ	Ім'я й призначення
SM0, SM1	Біти вибору режиму роботи УАПП. Установлюються та скидаються програмно. Записаний у біти код вказує номер режиму
SM2	Біт керування режимом УАПП. Встановлюється програмно для заборони приймання повідомлення, у якому дев'ятий біт має значення 0
REN	Біт дозволу приймання. Установлюється та скидається програмно для дозволу або заборони приймання послідовних даних
TB8	Стан восьмого біта передавача. Встановлюється та скидається програмно для завдання дев'ятого переданого біта в 9-бітному режимі УАПП
RB8	Стан восьмого біта приймача. Встановлюється/скидається апаратно для фіксації дев'ятого прийнятого біта в 9-бітному режимі
TI	Ознака переривання передавача. Встановлюється апаратно у момент закінченні передавання байта. Скидається програмно після обслуговування переривання
RI	Ознака переривання приймача. Установлюється апаратно при у момент закінчення приймання байта. Скидається програмно після обслуговування переривання

Прикладна програма шляхом завантаження в старші біти SCON 2-бітного коду визначає режим роботи УАПП. В усіх чотирьох режимах роботи передавання з УАПП ініціюється будь-якою командою, що записує дані в буферний регістр SBUF. Приймання даних УАПП у режимі 0 здійснюється за умови, що $RI = 0$ і $REN = 1$. У режимах 1, 2, 3 приймання даних починається з приходом старт-біта, якщо $REN = 1$.

У біті TB8 програмно встановлюється значення дев'ятого біта даних, який буде переданий у режимі 2 або 3. У біті RB8 фіксується в режимах 2 і 3 дев'ятий прийнятий біт даних. У режимі 1, якщо $SM2 = 0$, у біт RB8 заноситься стоп-біт. У режимі 0 біт RB8 не використовується.

Ознака переривання передавача TI встановлюється апаратно в кінці періоду передавання восьмого біта даних у режимі 0 і на початку періоду передавання стоп-біта в режимах 1, 2 і 3. Відповідна підпрограма обслуговування переривання повинна скидати біт TI.

Ознака переривання приймача RI встановлюється апаратно в кінці

періоду приймання восьмого біта даних у режимі 0 і в середині періоду приймання стоп-біта в режимах 1, 2 і 3. Підпрограма обслуговування переривання повинна скидати біт RI.

4.7.3. Робота УАПІ у мультимікроконтролерних системах

У системах децентралізованого керування, що використовуються для керування і регулювання в розподілених об'єктах (наприклад, прокатних станах, рухомому складі залізниці і метрополітену, складальних конвеєрах і лініях гнучких автоматизованих виробництв), виникає задача обміну інформацією між множиною мікроконтролерів, об'єднаних у локальну обчислювально-керуючу мережу. Як правило, локальні мережі на основі MCS51 мають магістральну архітектуру з розподіленим моноканалом (коаксіальний кабель, вита пара, оптичне волокно), по якому здійснюється обмін інформацією між МК.

У регістрі спеціальних функцій SCON мікроконтролера є керуючий біт SM2, що у режимах 2 і 3 УАПІ дозволяє відносно простими засобами реалізувати міжконтролерний обмін інформацією в локальних керуючих мережах.

Механізм межконтролерного обміну інформацією через послідовний порт MCS51 побудований на тому, що в режимах 2 і 3 програмований дев'ятий біт даних при прийманні фіксується в біті RB8. УАПІ може бути запрограмований таким чином, що при отриманні стоп-біта переривання від приймача буде можливо тільки за умови $RB8 = 1$. Це виконується встановленням керуючого біта SM2 у регістрі SCON.

Пояснимо процес міжконтролерного обміну інформацією на прикладі. Нехай головному МК потрібно передати блок даних деякому веденому МК. З цією метою головний МК у протокольному режимі "широкомовного" передавання (усім веденим МК) видає в моноканал байт-ідентифікатор абонента (код адреси МК - отримувача), який відрізняється від байтів даних тільки тим, що в його дев'ятому біті міститься 1. Програма реалізації протоколу мережного обміну інформацією повинна бути побудована таким чином, щоб при отриманні байта-

ідентифікатора ($RB8 = 1$) в усіх ведених МК відбулося переривання прикладних програм і виклик підпрограми порівняння байта-ідентифікатора з кодом власної мережної адреси. Адресований МК скидає свій керуючий біт SM2 і готується до приймання блока даних. Інші ведені МК, адреса яких не збіглася з кодом байта-ідентифікатора, залишають незмінним стан $SM2 = 1$. При $SM2 = 1$ інформаційні байти, передані по моноканалі і, ті що надходять в УАПП ведених МК, не викликають переривання, тобто ігноруються.

В режимі 1 УАПП автономного МК керуючий біт SM2 використовується для контролю істинності стоп-біта (при $SM2 = 1$ переривання не відбудеться до тих пір, поки не буде отримане істинне (одичне) значення стоп-біта). У режимі 0 біт SM2 не використовується і повинен бути скинутий.

4.7.4. Швидкість послідовного обміну

Швидкість послідовного обміну даними УАПП у різних режимах, визначається різними способами.

У режимі 0 швидкість обміну залежить лише від резонансної частоти кварцового резонатора ($f_{рез}$) і дорінює $f_0 = f_{рез}/12$. – За один машинний цикл послідовний порт передає один біт інформації. В режимах 1, 2 і 3 швидкість обміну даними залежить від значення керуючого біта SMOD у регістрі спеціальних функцій (табл 32, рис. 72).



Рис. 72. Формат регістра спеціальних функцій PCON

Таблиця 32. Біти регістра спеціальних функцій

Символ	Ім'я і призначення
SMOD	Подвоєна швидкість обміну. Якщо біт встановлений у 1, то швидкість обміну вдвічі більше, ніж при $SMOD = 0$
—	Не використовуються
GF1, GF0	Ознаки, що визначаються користувачем,
PD	Біт зниженої потужності. При установці біта в 1 МК переходить у режим зниженої споживаної потужності
IDL	Біт холостого ходу. Якщо біт встановлений в 1, то МК переходить у режим холостого ходу

В режимах 1 і 3 у формуванні швидкості обміну крім керуючого біта SMOD бере участь таймер 1. При цьому швидкість обміну залежить від частоти переповнення таймера (OVT1) і визначається в такий спосіб:

$$f_{1,3} = (2^{\text{SMOD}}/32) \cdot f_{\text{OVT1}}.$$

Переривання від таймера 1 у цьому випадку повинно бути заблоковано. Сам лічильник-таймер може працювати і як таймер, і як лічильник подій у будь-якому із трьох режимів. Однак найзручніше використовувати режим таймера з автоперезавантаженням. При цьому швидкість обміну визначається виразом

$$f_{1,3} = (2^{\text{SMOD}}/32) \cdot (f_{\text{рез}}/12)/(256 - (\text{TH1})).$$

У таблиці 33 наведено опис способів налаштування лічильника-таймера для отримання типових швидкостей обміну даними через УАПП.

Таблиця 33. Налаштування таймера 1 для керування частотою роботи УАПП

Швидкість обміну	Режим роботи	Частота резонатора, МГц	SMOD	Число, що заноситься в регістр TH1
1 МГц	0	12	X	X
375 кГц	2	12	1	X
62.5 кГц	1, 3	12	1	0FFH
19.2 кГц	1, 3	11.059	1	0FDH
9.6 кГц	1, 3	11.059	0	0FDH
4.8 кГц	1, 3	11.059	0	0FAH
2.4 кГц	1, 3	11.059	0	0F4H
1.2 кГц	1, 3	11.059	0	0E8H
137.5 Гц	1, 3	11.059	0	1DH
110 Гц	1, 3	6	0	72H

4.7.5. Особливості роботи УАПП у різних режимах

4.7.5.1. Режим 0

На рис. 73 зображено спрощену часову діаграму роботи УАПП в режимі 0. Дані передаються і приймаються через вивід RXD. Через вивід TXD видаються синхросигнали зсуву.

Передавання починається будь-якою командою, за якою у SBUF надходить байт даних. У момент часу S6P2 пристрій керування MCS51 за сигналом "Запис у буфер" записує байт у зсувний регістр передавача, установлює тригер дев'ятого біта і запускає блок керування обміном, що через

один машинний цикл виробляє сигнал дозволу посилання даних. При цьому в момент S6P2 кожного машинного циклу вміст зсувного регістра зсувається вправо і надходить на вивід RXD. У старші біти зсувного регістра передавача записуються нулі. При отриманні від детектора нуля сигналу "Передавач порожній" блок керування передавачем знімає сигнал "Посилка" і встановлює ознаку TI (момент S1P1 десятого машинного циклу після надходження сигналу "Запис у буфер").

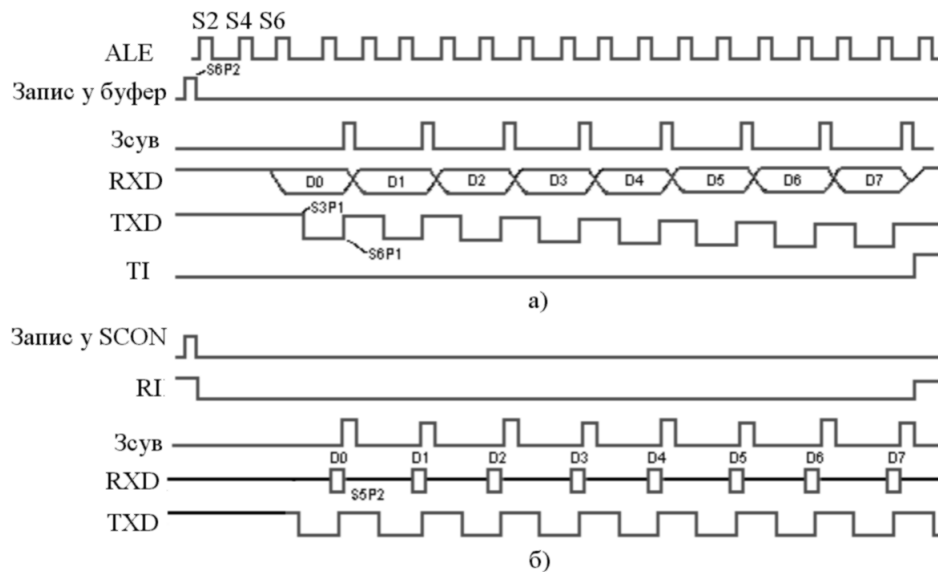


Рис. 73. Часова діаграма обміну в режимі 0:

а)передача даних, б)приймання даних

Приймання починається за умови $REN = 1$ і $RI = 0$. В момент S6P2 наступного машинного циклу блок керування приймачем формує сигнал дозволу прийому, по якому на вихід TXD передаються синхросигнали зсуву і в зсувному регістрі приймача починають формуватися значення бітів даних, що зчитуються з входу RXD в моменти S5P2 кожного машинного циклу. У момент S1P1 десятого машинного циклу після сигналу Запис у SCON блок керування приймачем переписує вміст зсувного регістра в буфер, знімає дозволяючий прийом сигнал і встановлює флаг RI.

4.7.5.2. Режим 1

Часова діаграма роботи УАПП в режимі 1 зображена на рис. 74. Через вивід TXD УАПП передає, а вивід RXD приймає 10 біт: старт-біт, 8 біт даних, можливо, дев'ятий біт даних і стоп-біт. При прийманні стоп-біт надходить в

біт RB8 регістра SCON.

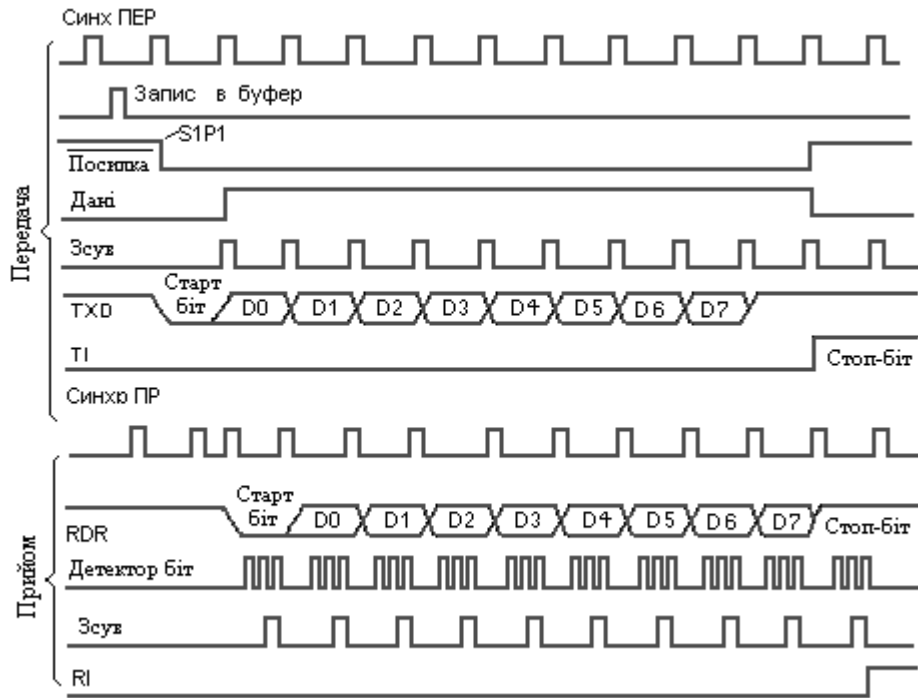


Рис. 74. Часова діаграма роботи УАПІ у режимі 1

Передавання ініціюється будь-якою командою запису у регістр SBUF. При цьому генерується сигнал "Запис у буфер", що завантажує 1 у дев'ятий біт зсувного регістра передавача. По цьому сигналу на вивід TXD спочатку надходить старт-біт, а потім по дозволяючому сигналу біти даних. Кожен період передавання біта дорівнює 16 тактам.

Прийом починається при виявленні переходу сигналу на вході RXD із стана 1 у стан 0. Для цього під керуванням внутрішнього лічильника вхід RXD опитується 16 разів за період представлення біта. Як тільки перехід із 1 у 0 на вході RXD виявлено, в зсувний регістр приймача завантажується код 1FFH, внутрішній лічильник по модулю 16 негайно скидається і перезапускається для вирівнювання його переходів із границями періодів представлення прийнятих бітів. Таким чином, кожен період представлення біта ділиться на 16 періодів внутрішнього лічильника. У станах 7, 8 і 9 лічильника в кожному періоді представлення біта опитується сигнал на вході RXD. Зчитаним вважається значення, що було отримано двічі з трьох вимірів. Якщо значення, прийняте в першому такті, не дорівнює 0, то блок керування прийомом знову повертається

до пошуку переходу з 1 у 0. Цей механізм забезпечує видалення помилкових старт-бітів. Блок керування прийомом формує сигнал "Завантаження буфера", встановлює ознаки RB8 та RI лише в тому випадку, якщо в останньому такті зсуву виконуються дві умови:

- 1) біт RI = 0;
- 2) або SM2 = 0, або прийнятий стоп-біт дорівнює 1.

Якщо одна з цих двох умов не виконується, то прийнята послідовність бітів втрачається. В цей час незалежно від того, виконуються вказані умови чи ні, блок керування прийомом знову починає відшукувати перехід із 1 у 0 на вході RXD.

4.7.5.2. Режими 2, 3

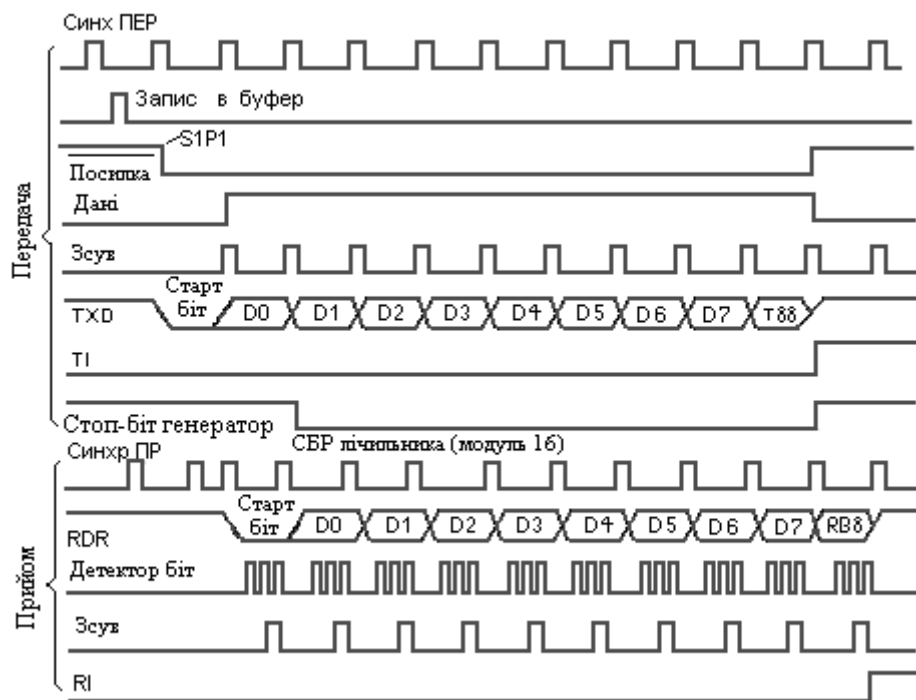


Рис. 75. Часова діаграма роботи УАПІ у режимах 2 і 3

Через вивід TXD УАПІ передає або з виводу RXD приймає 11 бітів: старт-біт (0), 8 бітів даних, програмований дев'ятий біт і стоп-біт (1). На часовій діаграмі (рис. 75) зображено роботу УАПІ при передаванні й прийманні даних у режимах 2 і 3. Режими 2 і 3 відрізняються від режиму 1 тільки наявністю дев'ятого програмованого біта. Внаслідок цього змінюються умови закінчення циклу прийому: блок керування приймачем сформує керуючий сигнал

"Завантаження буфера", завантажить RB8 і встановить ознаку RI тільки в тому випадку, якщо в останньому такті зсуву виконуються дві умови:

- 1) біт RI = 0;
- 2) SM2 = 0, або значення прийнятого дев'ятого біта даних дорівнює 1.

4.8. Система переривань

Спрощена схема переривань MCS51 зображена на рис. 76. Зовнішні переривання INT0 і INT1 можуть бути викликані або рівнем, або переходом сигналу з 1 у 0 на входах MCS51 залежно від значень керуючих бітів IT0 і IT1 у регістрі TCON. Від зовнішніх переривань встановлюються ознаки IE0 і IE1 у регістрі TCON, що ініціюють виклик відповідної підпрограми обслуговування переривання. Скидання цих ознак виконується апаратно тільки в тому випадку, якщо переривання було викликано по переходу (зрізу) сигналу. Якщо ж переривання викликане рівнем вхідного сигналу, то скиданням флага IE керує відповідна підпрограма обслуговування переривання шляхом впливу на джерело переривання.

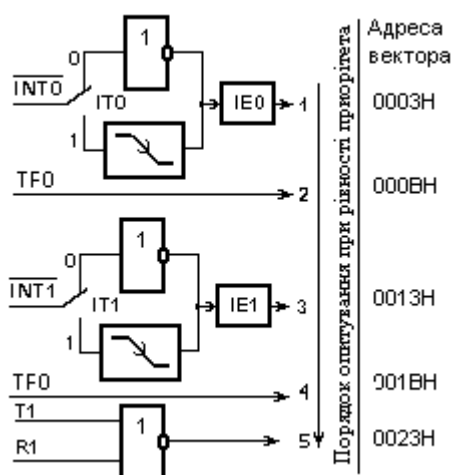


Рис. 76. Схема переривань MCS51

Ознаки запитів переривання від таймерів TF0 і TF1 скидаються автоматично при передаванні керування підпрограмою обслуговування. Ознаки запитів переривання RI і TI встановлюються блоком керування УАПП апаратно, але скидатися повинні програмою.

Переривання можуть бути дозволені або заборонені програмою за

допомогою скидання розрядів регістра дозволів переривань (EI), формат регістра EI зображено на рис. 77, опис розрядів наведено в таблиці 34.

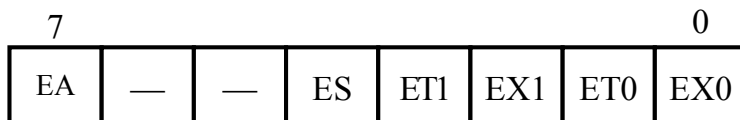


Рис. 77. Формат регістра дозволу переривань EI

Таблиця 34. Регістр дозволу переривання EI

Символ	Ім'я і призначення
EA	Блокування переривань. Скидається програмно для заборони всіх переривань незалежно від станів IE4 - IE0
ES	Біт дозволу переривання від УАПІ. Ознака встановлюється та скидається програмою для дозволу або заборони переривань від ознак TI або RI
ETx	Біт дозволу переривання від таймера x. Ознака встановлюється та скидається програмою для дозволу або заборони переривань від таймера
EXx	Біт дозволу зовнішнього переривання, що надходить на вхід INTx. Ознака встановлюється та скидається програмою для дозволу або заборони переривань від зовнішнього входу INTx

У випадку одночасного надходження переривань першим обробляється те переривання, пріоритет котрого вище, а при однаковому пріоритеті – у порядку опитування (рис. 76).

У блоці регістрів спеціальних функцій є рівнями пріоритету (IP), котрий вказує рівень пріоритету переривання. Його формат зображени рис. 78, опис розрядів наведено в таблиці 35.

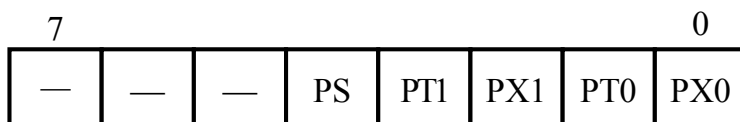


Рис. 78. Формат регістра пріоритетів переривань IP

Таблиця 35. Регістр пріоритетів переривань IP

Символ	Ім'я і призначення.
PS	Біт пріоритету УАПІ. Одиниця вказує на високий пріоритет переривання УАПІ, нуль – на низький.
PTx	Біт пріоритету таймера x. Одиниця вказує на високий пріоритет переривання таймера, нуль – на низький
PX	Біт пріоритету зовнішнього переривання (INTx). Одиниця вказує на високий пріоритет переривання таймера, нуль – на низький

Ознаки переривань опитуються в момент S5P2 кожного машинного

циклу. Ранжування переривань за рівнем пріоритету виконується протягом наступного машинного циклу. Система переривань сформує апаратно виклик (LCALL) відповідної підпрограми обслуговування, якщо вона не заблокована однією із наступних умов:

1) у даний момент обслуговується запит переривання рівного або більш високого рівня пріоритету;

2) поточний машинний цикл – не останній у циклі команди, що виконується;

3) виконується команда RETI або будь-яка команда, що зв'язана зі звертанням до регістрів IE або IP.

Якщо ознака переривання була встановлена, але по одній з перерахованих вище умов не одержала обслуговування і до моменту закінчення блокування вже була скинута, то запит переривання втрачається і ніде не запам'ятовується.

За апаратно-сформованим кодом LCALL система переривання заносить в стек тільки вміст лічильника команд (PC) і завантажує в лічильник команд адресу вектора відповідної підпрограми обслуговування. За адресою вектора повинна бути розташована команда безумовного переходу (JMP) до початкової адреси підпрограми обслуговування переривання. Підпрограма обслуговування у випадку необхідності повинна починатися командами запису в стек (PUSH) стану програми (PSW), акумулятора, розширювача, вказівника даних і т.д. і закінчуватися командами відновлення зі стека (POP) збережених даних. Підпрограми обслуговування переривання обов'язково завершуються командою RETI, за якою в лічильник команд перезавантажується зі стека збережена адреса повернення в основну програму. Команда RET також повертає керування перерваній основній програмі, але при цьому не знімає блокування переривань, що призводить до необхідності мати програмний механізм аналізу закінчення процедури обслуговування даного переривання.

Переривання можуть бути викликані або заборонені програмою, а всі перераховані ознаки доступні і можуть бути встановлені або скинуті з тим же результатом, якби вони були встановлені або скинуті апаратними засобами.

Наведемо приклад організації очікування тривалістю 50 мс за допомогою переривання. Вважаємо, що біт ІЕ.7 встановлено.

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

Для вимірювання тривалості сигналу може використовуватись таймер. Особливо ефективним є його використання в МК51, з входом дозволу відліку (альтернативна функція входу INT). Вимірюваний сигнал можна, наприклад, подавати на вхід INT0, вимірювання тривалості при цьому буде виконуватися в лічильнику-таймері 0. Програма вимірювання тривалості "додатного" імпульсу буде мати вигляд:

```
MOV TMOD, #00001001B          ;вибір режиму таймера
MOV TH0, #0                    ;скидання
MOV TL0, #0
```



```

SETB TCON.4           ;старт
WAITO:
    JNB P3.2, WAITO   ;чекання "1"
WAITC:
    JB P3.2, WAITC    ;чекання "0"
    CLR TCON.4        ;стоп таймера
EXIT:RET .            ;вихід з процедури

```

4.9. Контрольні запитання

1. Однокристальні мікроЕОМ сімейства MCS51. Основні характеристики.
2. Архітектура ОМЕОМ 87С51.
3. Будова арифметико-логічного пристрою.
4. Організація резидентної пам'яті даних та пам'ять програм.
5. Система переривань однокристальних мікроЕОМ сімейства MCS51.
6. Порти вводу-виводу.
7. Послідовний інтерфейс.
8. Таймери/ лічильники. Режими роботи.
9. Під'єднання до МК51 зовнішньої пам'яті даних та програм.
10. Проектування систем керування на базі МК51. Спряження МК51 з розширювачем вводу-виводу.
11. Проектування систем керування на базі МК51. Під'єднання до МК51 розширеної кількості давачів.
12. Виконання динамічної індикації на основі семисегментних індикаторів та клавіатури.

ТЕМА №5. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ НА БАЗІ PIC16X8X

5.1. Особливості контролерів PIC16X8X

Мікроконтролери підгрупи PIC 16x8x відносяться до сімейства 8-розрядних КМОП мікроконтролерів групи PIC16CXXX. Низька ціна, економічність, швидкодія, простота використання і гнучкість вводу-виводу робить PIC16X8X привабливим навіть у тих галузях, де раніше не застосовувалися мікроконтролери. Наприклад: таймери, заміна жорсткої логіки у великих системах, співпроцесори. Висока навантажувальна здатність спрощує зовнішні драйвери і тим зменшує загальну вартість системи. Розробки на базі контролерів PIC16X8X підтримуються асемблерами, симуляторами, схемним емуляторами і програматорами різних фірм.

Серія PIC16X8X підходить для широкого спектра схем від високошвидкісного керування автомобільними й електричними двигунами до економічних віддалених прийомопередавачів, вимірювальних приладів і процесорів вводу-виводу. Наявність EEPROM даних дозволяє розміщувати параметри настроювання в середині кристала, чим знижує загальну вартість системи. Малі розміри корпусів як для звичайного, так і для поверхневого монтажу, роблять цю серію мікроконтролерів придатною для портативних цілей.

Вбудований автомат програмування кристала PIC16X8X дозволяє легко змінювати програму і дані під конкретні вимоги, в тому числі з використанням внутрисхемного програмування – програмування кристала після встановлення його в цільову систему. Ця можливість може бути використана як для тиражування, так і для занесення каліброваних даних уже після остаточного тестування.

Всі регістри контролерів статичні, отже мінімальна тактова частота може дорівнювати 0 Гц, максимальна тактова частота – 10 МГц. Усі команди виконуються за один цикл, що становить 4 машинних такти, крім команд

переходів та виклику підпрограм, що виконуються за 2 цикли. Система команд включає 35 простих команд з ортогональною симетрією. Машинні коди всіх команд 14-бітні і обробляють 8-бітні дані, використовуючи пряму, непряму і відносну адресацію даних. При звертанні до підпрограм та виклику переривань використовується вісьмирівневий апаратний стек, що не відображається в область ОЗП, або в жодну іншу область пам'яті. Він призначений для збереження адрес повернення з підпрограм, у тому числі з підпрограм обслуговування переривань. У випадку необхідності глибшого стека в контролері передбачена можливість організації програмного стека в ОЗП будь-якої необхідної глибини.

До складу сімейства входять МК PIC16F83, PIC16CR83, PIC16F84 і PIC16CR84. Основні характеристики МК підгрупи PIC16F8X наведено в таблиці 36.

Таблиця 36. Основні характеристики МК підгрупи PIC16F8X

Параметр	PIC16F83	PIC16CR83	PIC16C84	PIC16F84	PIC16CR84
Максимальна частота, МГц	10	10	10	10	10
Flash-пам'ять програм, слів	512	-	-	1К	-
EEPROM пам'ять програм, слів	-	-	1К	-	-
ПЗП програм, слів	-	512	-	-	1К
Пам'ять даних, байт	36	36	36	68	68
Пам'ять даних у РПЗП (EEPROM), байт	64	64	64	64	64
Таймери	TMR0	TMR0	TMR0	TMR0	TMR0
Число джерел переривань	4	4	4	4	4
Число ліній вводу-виводу	13	13	13	13	13
Діапазон напруги живлення, У	2.0-6.0	2.0-6.0	2.0-6.0	2.0-6.0	2.0-6.0
Число виводів і тип корпусу	18 DIP, SOIC	18 DIP, SOIC	18 DIP, SOIC	18 DIP, SOIC	18 DIP, SOIC

Мікроконтролери підгрупи PIC16F8X різняться між собою тільки об'ємом ОЗП даних, а також об'ємом і типом пам'яті програм. Наявність у

складі підгрупи МК із Flash і EEPROM-пам'яттю програм полегшує створення і прототипів зразків виробів.

Підсистема пам'яті складається з:

- ◇ пам'яті програм об'ємом 1024 х 14, в якості якої використовується PROM (в PIC16C84), FLASH (в PIC16F84) або масочний постійний запам'ятовуючий пристрій (в PIC16CR84);
- ◇ пам'яті даних об'ємом 36х8 (в PIC16F83, PIC16CR83, PIC16C84), або 68х8 (в PIC16CR84 та в PIC16F84), що може використовуватись в якості регістрів загального призначення;
- ◇ EEPROM даних об'ємом 64 байт, що можуть бути запрограмовані як у процесі програмування мікросхеми, так і за нормальної роботи контролера при виконанні відповідних команд програми.

Мікросхема має можливість використовувати чотири джерела переривання:

- ◇ зовнішній вхід INT;
- ◇ переповнення таймера RTCC;
- ◇ переривання при зміні сигналів на лініях порту В;
- ◇ після завершення записування даних у пам'ять EEPROM.

Мікросхема має 13 ліній вводу-виводу з індивідуальним настроюванням на ввід або на вивід. Вхідний струм при видаванні на вихід логічного нуля становить 25 мА. Вихідний струм при видаванні логічної одиниці 20 мА. Існує можливість генерації переривання по зміні стана відповідного виводу, а також відлік кількості змін сигналів на визначеному виводі.

У мікросхемі присутній 8-бітний таймер/лічильник RTCC з 8-бітним запрограмованим попереднім дільником, який може працювати в режимі таймера та лічильника імпульсів, що надходять на відповідний вхід.

Для підвищення надійності в PIC16X8X присутні можливості автоматичного скидання при пропаданні та появі живлення, таймери

затримки виходу з режиму скидання, а також Watchdog-таймер (WDT).

Програмування здійснюється через вбудований пристрій програмування пам'яті програм і даних, що використовує лише два виводи кристала. При цьому саме програмування може виконуватись як в програматорі, так і в цільовій системі. Для захисту коду від несанкціонованої модифікації та копіювання можливе використання бітів захисту інформації. Також користувач при програмуванні може вибрати тип генератора синхронізації, що використовується з: RC-генератора (RC), звичайного кварцового резонатора (XT), високочастотного кварцового резонатора (HS), економічного низькочастотного кварцового резонатора (LP).

Також мікросхема володіє низьким енергоспоживанням:

- ◇ 3 мА при напрузі живлення 5В, і частоті 4 МГц;
- ◇ 50 мкА при напрузі живлення 2В і частоті 32 кГц.

5.2. Призначення виводів та позначення мікросхеми

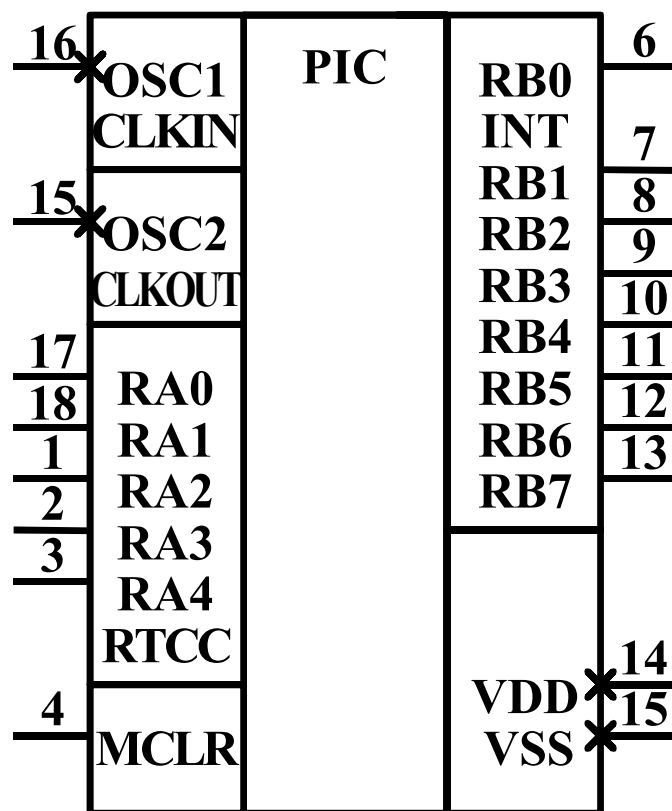


Рис. 79. Умовне позначення мікросхеми

Умовне позначення мікросхеми наведено на рис. 79, а призначення

выводів – в таблиці 37.

Таблиця 37. Призначення виводів мікросхеми

Позначення	Номер виводу	Призначення
RA0-RA3	17, 18, 1, 2	Двонаправлені лінії вводу-виводу. Вхідні та вихідні рівні відповідають TTL
RA4 RTCC	3	Двонаправлена лінія вводу-виводу. Вхід через тригер Шмітта. Вивід працює як вивід з відкритим стоком, також може використовуватись як вхід частоти для лічильника-таймера. В такому режимі роботи вивід має тригер Шмітта на вході
RB0/INT	6	Двонаправлена лінія вводу-виводу порту В. Лінія може використовуватись як вхід переривання, при цьому вхід буферизується тригером Шмітта.
RB1 - RB7	7-13	Двонаправлені лінії вводу-виводу порту В. Лінії можуть бути запрограмовані у режимі внутрішніх активних навантажень на лінію живлення по усіх виводах. Виводи RB4...RB7 можуть бути програмно налаштовані як входи переривання по зміні стану на кожному із входів. При програмуванні МК RB6 використовується як тактовий, а RB7 як вхід/вихід даних. У режимі програмування лінії RB6 та RB7 мають на вході тригер Шмітта. В інших режимах сигнали на лініях відповідають КМОП TTL логіці
MCLR	4	Сигнал скиду. Логічний нуль на лінії призводить до скидання мікросхеми. В режимі програмування призначений для подавання напруги програмування. Лінія на вході має тригер Шмітта
OSC1 CLKIN	16	Вхід під'єднання кварцу, зовнішньої RC-ланки або зовнішнього джерела синхронізації. Вхід має тригер Шмітта, коли конфігурується в режимі RC-генератора і КМОП в усіх інших випадках
OSC2 CLKOUT	15	Вхід під'єднання кварцу, вихід сигналу синхронізації в режимі синхронізації за допомогою RC ланки, або в режимі включення зовнішнього тактового сигналу. Частота сигналу у режимі роботи з RC-ланкою дорівнює 1/4 частоти OSC1 і вказує швидкість виконання команд
VDD	14	Напруга живлення
VSS	5	Загальний (земля)

Параметри сигналів не мають виходити за межі вказані в таблиці 38.

Таблиця 38. Граничні значення сигналів

Параметр	Граничні значення
Напруга на будь-якому виводі відносно землі (за винятком VDD та MCLR)	-0.6В VDD+0.6В
Напруга VDD (робоча) для PIC16C84, PIC16CR84, PIC16F84	4.0... 6 В
Напруга VDD (робоча) для PIC16LC84	2.0...6. В
Напруга VDD (у режимі SLEEP)	1.5...6 В
Напруга VDD (максимально допустима)	0...7.5 В
Струм логічного нуля на кожному виході, що може прийняти мікросхема	25 мА
Струм логічного нуля на кожному виході, що може видати мікросхема	20 мА
Струм логічної одиниці на кожному виході (у будь-якому напрямку)	20 мА
Сумарний струм, що приймається, через лінії порта А	80 мА
Сумарний струм, що видається, через лінії порта А	50 мА
Сумарний струм, що приймається, через лінії порта В	150 мА
Сумарний струм, що видається, через лінії порта В	100 мА
Максимальний струм через лінію VDD	100 мА
Максимальний струм через лінію VSS	150 мА

5.3. Архітектура PIC16X8X

Спрощена архітектура PIC16X8X зображена на рис. 80. Архітектура контролера базується на концепції роздільних шин та областей пам'яті для даних і команд. Шина даних і пам'ять даних мають ширину 8 бітів, а шина адреси програм і ПЗП програм мають ширину 13 і 14 бітів відповідно. Така концепція забезпечує просту, але потужну систему команд, розроблену так, що бітові, байтові і реєстрові операції працюють із високою швидкістю та з перекриттям за часом вибірок команд і циклів виконання. 14-бітова ширина програмної пам'яті забезпечує вибірку 14-бітової команди в один цикл. Двоступінчастий конвеєр забезпечує одночасну вибірку і виконання команди. Усі команди виконуються за один цикл, крім команд переходів.

Розглянемо основні функціональні блоки структурної схеми однокристалльної мікроЕОМ.

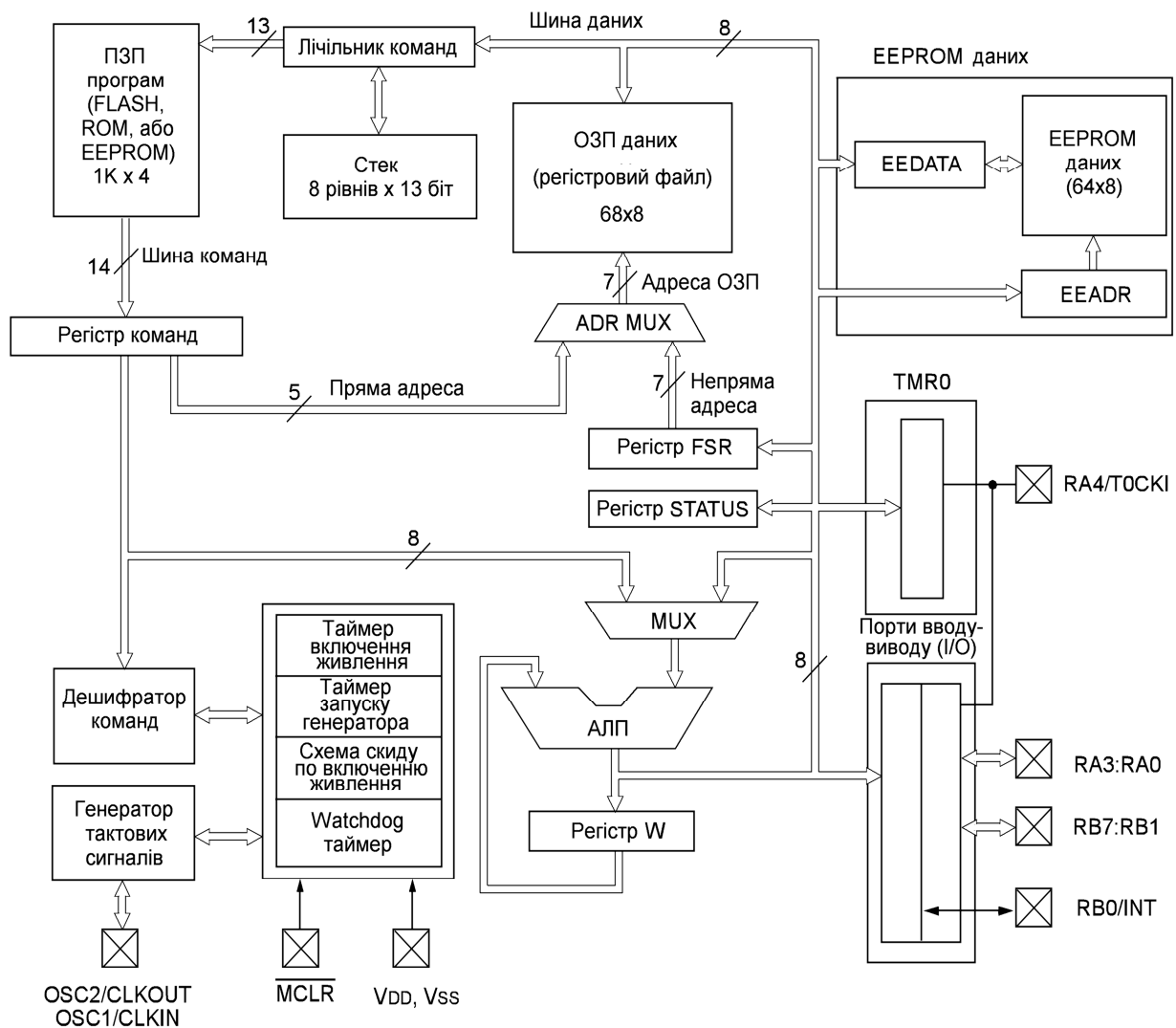


Рис. 80. Структурна схема мікросхеми

Блок вибірки команд призначений для вибірки команд, що виконуються в певній послідовності. До складу блока входять 4 підблоки:

1. ПЗП програм, призначений для збереження коду програми, під керуванням якої працює мікроконтролер. Структуру ПЗП програм та спеціалізовані адреси в ПЗП розглянемо пізніше.

2. Лічильник команд (ЛК), має розрядність 13 бітів і призначений для визначення адреси наступної команди, що буде виконуватись.

3. Стек, має 8 рівнів і розрядність 13 бітів, призначений для збереження тільки адрес повернення з підпрограм.

4. Регістр команд (РК), використовується для збереження коду команди, що виконується. Команда після вибірки з ПЗП програм записується в регістр і

зберігається протягом усього часу її виконання.

Блок ОЗП, призначений для збереження даних і складається із таких трьох підблоків:

1. ОЗП даних (регістровий файл 68x8), призначений для збереження даних, що використовуються при роботі програми.

2. Адресний мультиплексор (ADR MUX), визначає джерело адреси, за допомогою якої адресуються комірки ОЗП даних. Такими джерелами можуть бути регістр FSR при непрямій адресації та регістр команд – при прямій.

3. Регістр вибору адреси (Регістр FSR) використовується при непрямій адресації комірок ОЗП даних. Розрядність регістра 8 бітів, отже, за його допомогою можна адресувати будь-яку комірку ОЗП.

Блок синхронізації та керування. Призначений для синхронізації і керування іншими блоками мікросхеми. Він складається із двох наступних підблоків:

1. Генератора тактових сигналів (ГТС), який призначений для генерації сигналів синхронізації роботи окремих вузлів мікросхеми.

2. Дешифратора команд (ДК), котрий призначений для перетворення коду команди в сигнали керування іншими блоками мікросхеми. Синхронізація блока здійснюється генератором тактових сигналів.

Блок формування сигналу скидання призначений для формування сигналу скидання від різних можливих джерел. До блоку входять чотири підблоки:

1. Схема скидання по ввімкненню живлення (ССВЖ). Схема встановлює внутрішній сигнал скидання в активний рівень при ввімкненні живлення (коли на лінії VDD є передній фронт сигналу і напруга перевищує 1.2 – 1.7 В).

2. Таймер ввімкнення живлення (ТВЖ). Призначений для утримання сигналу скидання тривалістю 72 мс після того, як напруга живлення ввійде в робочий діапазон. Це знижує ймовірність невірної роботи мікросхеми у випадку нестабільного живлення.

3. Таймер ввімкнення генератора (ТВГ). Утримує мікроЕОМ у стані

скидання протягом 1024 цикли після виключення таймера ТВЖ.

4. Watchdog таймер (WT), називають також сторожовим таймером – це спеціальний таймер, що приводить мікроЕОМ в стан скидання у випадку збою програми.

Розглянуті таймери можуть бути використані вибірково, щоб уникнути небажаних очікувань як при ввімкненні, так і при виході з режиму SLEEP.

Операційний блок виконує всі арифметичні й логічні операції та складається із трьох підблоків:

1. Арифметико-логічного пристрою (АЛП), що виконує всі арифметичні та логічні операції.

2. Регістр W - регістр тимчасового зберігання. Розрядність регістра 8 бітів.

3. Регістр STATUS - регістр призначений для збереження ознак виконаних арифметичних і логічних операцій, а також визначення банку пам'яті даних і стана Watchdog таймера.

Блок EEPROM даних призначено для збереження даних, які необхідно зберегти після вимкнення живлення. До блоку відносять 3 підблоки:

1. Регістр EEADR, задає адресу комірки EEPROM, з якою відбувається обмін.

2. Регістр EEDATA служить для збереження даних, які записуються в EEPROM, а також для отримання даних з неї.

3. EEPROM даних – матриця пам'яті, в якій зберігається інформація.

Також на структурній схемі позначені:

TMRO – восьмирозрядний лічильник, що може працювати в режимі лічильника подій і таймера. До блоку входить також попередній подільник з коефіцієнтом поділу 1:2, 1:4 .. 1:256.

Порти вводу-виводу призначені для введення виведення інформації. В мікроЕОМ є 13 ліній вводу-виводу, що згруповані в 5-розрядний порт А та 8-розрядний порт В. Кожна лінія будь-якого порту може бути налаштована незалежно від інших на введення і на виведення інформації. При зміні стана ліній порту В можлива генерація переривань.

5.4. Робота мікроЕОМ

Схема тактування і виконання команди зображена на рис. 81. Вхідна тактова частота, що надходить з виводу OSC1/CLKIN, ділиться всередині блока на чотири і з її формуються чотири тактові послідовності Q1, Q2, Q3 і Q4, що не перекриваються одна з одною. Лічильник команд збільшується в такті Q1, команда зчитується з пам'яті програми і запам'ятовується в регістрі команд у такті Q4. Команда декодується і виконується протягом наступного циклу в тактах Q1...Q4. Протягом тактів Q2, Q3 і Q4 наступного циклу відбувається декодування і виконання команди. У такті Q2 зчитується пам'ять даних (читання операнда), а запис відбувається в такті Q4.

Цикл виконання команди складається з чотирьох тактів: Q1...Q4. Вибірка команди та її виконання сполучені за часом таким чином, що вибірка команди займає один цикл, а виконання – наступний. Ефективний час виконання команди складає один цикл.

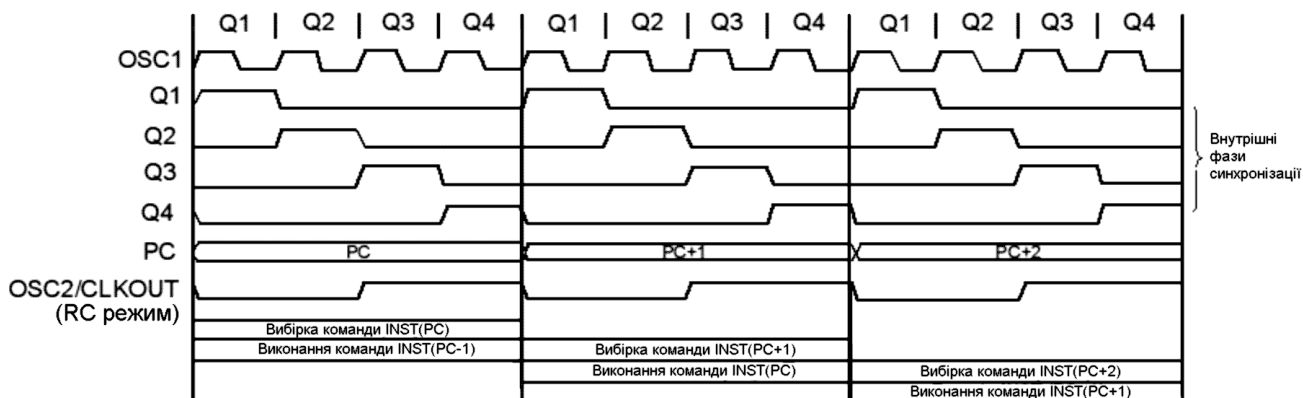


Рис. 81. Схема тактування і виконання команди

Якщо команда змінює лічильник команд (наприклад, команда GOTO), то для її виконання буде потрібно два цикли. Після переходу необхідно вибрати код команди за новою адресою, в цей час виконання наступної команди неможливе.

5.5. Структура та робота операційного блока

Мікроконтролер містить 8-розрядний арифметико-логічний пристрій (АЛП) і робочий регістр W. АЛП є арифметичним модулем загального призначення і виконує арифметичні та логічні функції над вмістом робочого регіст-

ра і кожного з регістрів контролера. АЛП може виконувати операції додавання, віднімання, зсуву і логічні операції. Якщо не зазначено інше, то арифметичні операції виконуються в додатковому двійковому коді.

Залежно від результату операції, АЛП може змінювати значення бітів регістра STATUS: C (Carry), DC (Digit carry) і Z (Zero). Регістр статусу (STATUS) містить ознаки операції (арифметичні прапори) АЛП, стан контролера при скиданні і біти вибору сторінок для пам'яті даних. Формат регістра STATUS зображено на рис. 82, а призначення бітів наведено у таблиці 39.

Таблиця 39. Призначення бітів регістра STATUS

Біт	Значення
IRP	Біт вибору сторінки банку даних (використовується при непрямій адресації): 0 = банк 0, 1 (00 _h -FF _h), 1 = банк 2, 3 (100 _h -1FF _h). Біт IRP не використовується в МК підгрупи PIC16F8X
RP1- RP0	Біти вибору сторінки банка даних (використовуються при прямій адресації): 00 = банк 0 (00 _h -7F _h); 01 = банк 1 (80 _h -FF _h); 10 = банк 2 (100 _h -17F _h); 11 = банк 3 (180 _h - IFF _h). У МК підгрупи PIC16F8X використовується тільки біт RP0
$\overline{T0}$	Біт спрацювання сторожового таймера. Контролює спрацювання сторожового таймера, встановлюється в одиницю командами CLRWDT і SLEEP, а також сигналом скидання (крім сигналу від сторожового таймера). Скидання відбувається при спрацюванні сторожового таймера
\overline{PD}	Біт зниження споживаної потужності. Логічна 1 вказує на режим зниження потужності споживання. Встановлюється при ввімкненні живлення, а також командою CLRWDT, скидання командою SLEEP
Z	Біт нульового результату. Одиниця вказує, що результат останньої арифметичної або логічної операції був нульовим, одиниця – не був.
DC	Біт десяткового (додаткового) перенесення/позичання. Одиничне значення вказує, що при виконанні останньої арифметичної операції відбулося перенесення з третього в четвертий розряд. Використовується при роботі з двійково-десятковими числами
C	Біт перенесення/позичання. Одиничне значення вказує, що при виконанні останньої арифметичної операції відбулось переповнення або переспустощення. Біт також встановлюється та скидається командами зсуву. При виконанні команд зсуву цей біт завантажується з молодшого чи старшого розряду джерела залежно від команди

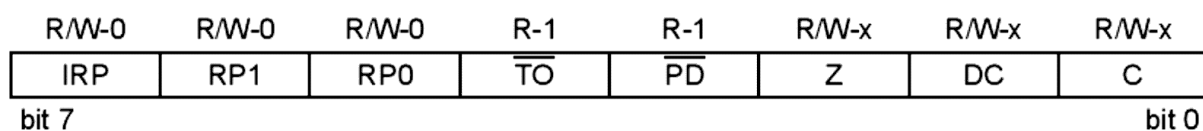


Рис. 82. Формат регістра STATUS

На рис. 82 прийняти наступні позначення: R - біт, що читається; W - записуваний біт; S - встановлюваний біт; U - біт, що використовується (читається як "0"), 0 або 1 - значення біта після скиду.

Регістр статусу доступний для будь-якої команди так само, як будь-який інший регістр. Однак якщо регістр STATUS є регістром призначення для команди, що впливає на біти Z, DC чи C, записування у них буде заборонено. Біти \overline{TO} і \overline{PD} встановлюються апаратно і не можуть бути записані в регістр статусу. Наприклад, команда CLRf STATUS обнулить усі біти, крім біт \overline{TO} і \overline{PD} , і встановить біт Z=1, хоча після виконання цієї команди регістр статусу буде мати ненульове значення. Рекомендується для зміни регістра статусу використовувати тільки команди встановлення біт BCF, BSF, MOVWF.

5.6. Структура ПЗП програм

ПЗП програм у різних типів контролерів сімейства побудовано на основі EEPROM, флеш-пам'яті, або масочного ПЗП. Об'єм ПЗП змінюється від 512 до 1024 слів, що мають розрядність 14 бітів. На ПЗП надходить 13-розрядна адресна шина, з якої використовуються лише 9 або 10 молодших розрядів. Старші розряди є зарезервованими. Організацію пам'яті програм і стека зображено на рис. 83.

У пам'яті програм є виділені адреси. Вектор скидання знаходиться за адресою 0000H, вектор переривання – за адресою 0004H. Звичайно за адресою 0004H розташовується підпрограма ідентифікації й опрацювання переривань, а за адресою 0000H - команда переходу на програму початкового встановлення, розташовану за підпрограмою обробки переривань. Вся пам'ять програм є внутрішньою. Запустити програму з зовнішньої пам'яті неможливо.

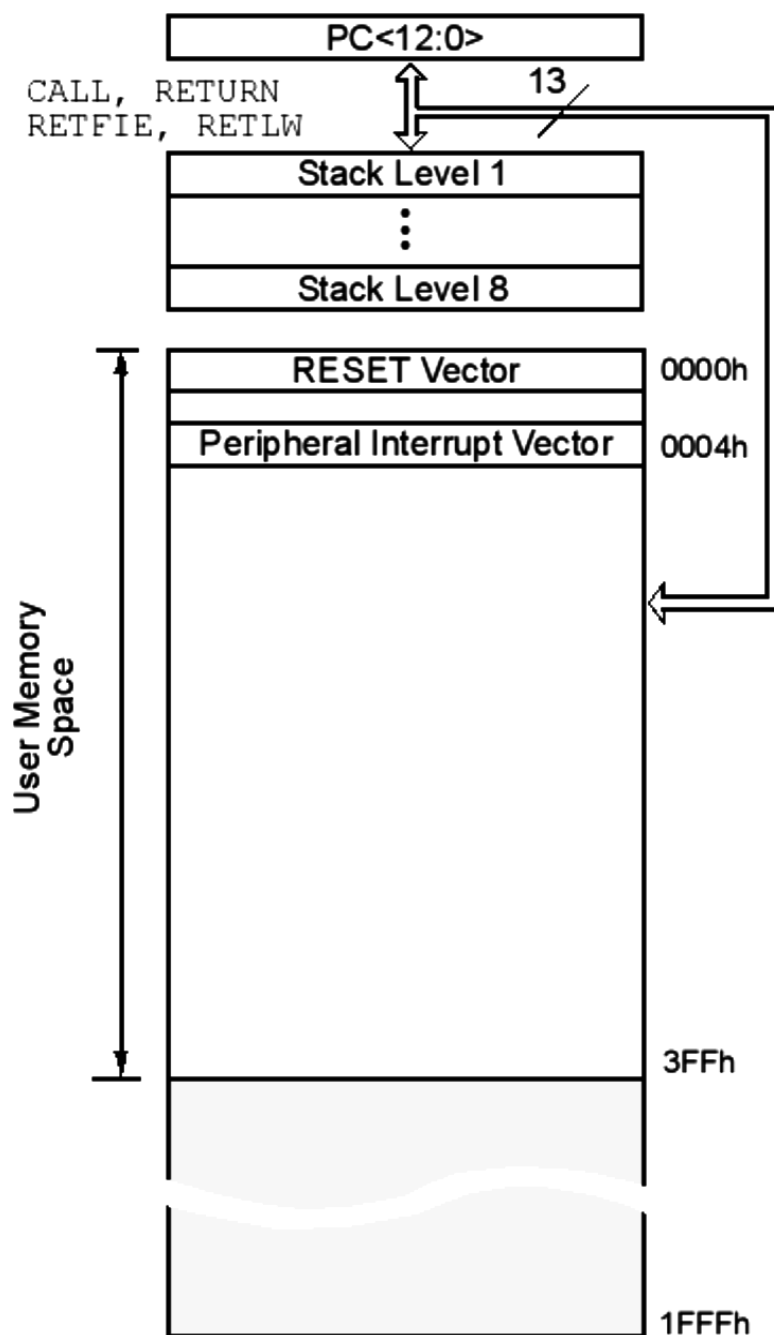


Рис. 83. Організація пам'яті програм і стека

Для адресації пам'яті програм використовується лічильник команд. Лічильник команд у МК PIC16F8X має ширину 13 бітів і здатний адресувати 8 К x 14 об'єму програмної пам'яті. Однак фактично на кристалах PIC16F83 і PIC16CR83 є тільки 512 × 14 пам'яті (адреси 0000_h-01FF_h), а в МК PIC16C84, PIC16F84 і PIC16CR84 - 1К×14 пам'яті (адреси 0000_h-03FF_h). Звертання до адрес вище 1FF_h (3FF_h) є адресацією перших 512 або 1К комірок пам'яті програм.

Молодший байт лічильника (PCL) доступний для читання й записування і

відображається на регістр з адресою 02_h. Старший байт лічильника команд не може бути прямо записаний чи зчитаний і береться з регістра PCLATH (PC latch high), адреса якого 0A_h. Уміст PCLATH передається в старший байт лічильника команд, коли він завантажується новим значенням.

Залежно від того, чи завантажується в лічильник команд нове значення під час виконання команд CALL, GOTO, чи в молодший байт лічильника команд (PCL) здійснюється запис, старші біти лічильника команд завантажуються з PCLATH різними способами, як зображено на рис. 84.

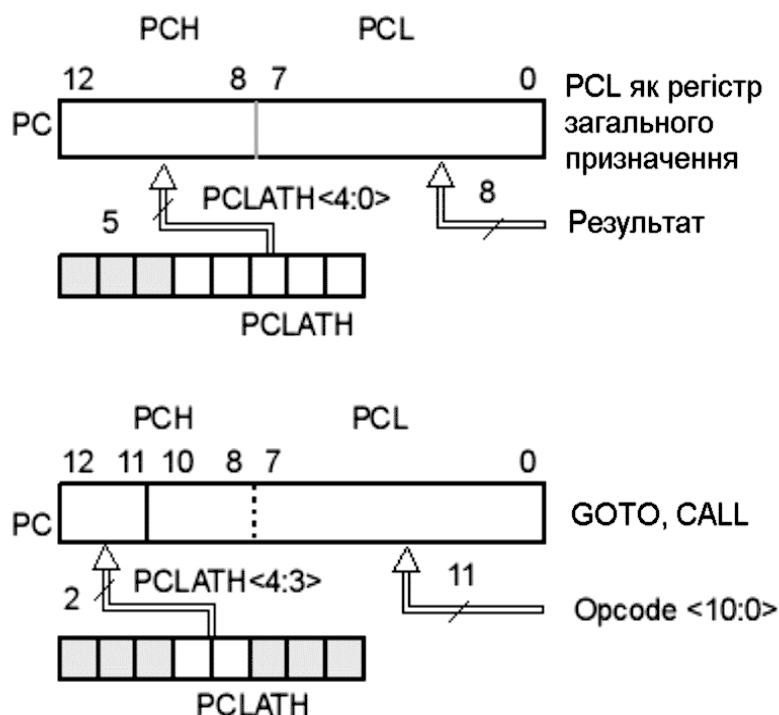


Рис. 84. Завантаження старших бітів лічильника команд

Можливість виконувати арифметичні операції безпосередньо над лічильником команд дозволяє дуже швидко й ефективно здійснити табличні перетворення в PIC-контролерах.

Команди CALL і GOTO оперують 11-розрядним адресним діапазоном, достатнім для зсуву в межах сторінки програмної пам'яті об'ємом 2К слів. Для МК підгрупи PIC16F8X цього вистачає. З метою забезпечення можливості розширення пам'яті команд для майбутніх моделей МК передбачене завантаження двох старших бітів лічильника команд із регістра PCLATH<4:3>. При використанні команд CALL і GOTO користувач повинний переконатися в

тому, що ці сторінкові біти запрограмовані для виходу на потрібну сторінку. При виконанні команди CALL чи виконанні переривання весь 13-бітний лічильник команд поміщається в стек, тому для повернення з підпрограми не потрібні маніпуляції з розрядами PCLATH<4:3>.

Мікроконтролери підгрупи PIC16F8X ігнорують значення бітів PCLATH<4:3>, що використовуються для звертання до сторінок 1, 2 і 3 програмної пам'яті. Однак застосовувати біти PCLATH<4:3> у якості комірок пам'яті загального призначення не рекомендується, тому що це може вплинути на сумісність із майбутніми поколіннями виробів.

Мікроконтролери підгрупи PIC16F8X мають восьмирівневий апаратний стек шириною 13 бітів. На відміну від попередньо розглянутих мікропроцесорів і мікроЕОМ у даному сімействі стек ніяк не відображається на область пам'яті даних. Стек у мікроЕОМ цього типу призначений лише для збереження адрес повернення з підпрограм.

Область стека не належить ні до програмної області, ні до області даних, а покажчик стека користувачу недоступний. Поточне значення лічильника команд посилається в стек, коли виконується команда CALL чи здійснюється опрацювання переривання. При виконанні процедури повернення з підпрограми (команди RETLW, RETF1E чи RETURN) уміст лічильника команд відновлюється зі стека. Регістр PCLATH при операціях зі стеком не змінюється.

Стек працює як циклічний буфер. Отже, після того, як стек був завантажений 8 разів, дев'яте завантаження переписує значення першого. Якщо стек був вивантажений 9 разів, лічильник команд стає таким же, як після першого вивантаження.

Ознак положення стека в контролері не передбачено, тому програміст повинен самостійно стежити за рівнем вкладення підпрограм.

5.7. Структура ОЗП

Мікроконтролер може за допомогою прямої, або непрямой адресації звертатися до регістрів пам'яті чи даних. Усі регістри спеціальних функцій,

включаючи лічильник команд, відображаються на пам'ять даних.



Рис. 85. Організація пам'яті даних

Пам'ять даних МК падіння на дві області. Перші 12 адрес – це область регістрів спеціальних функцій (SFR), друга область регістрів загального призначення (GPR). Область SFR керує роботою мікрохеми.

Обидві області поділені у свою чергу на банки 0 і 1. Банк 0 вибирається обнуленням біта RP0 регістра статусу (STATUS). Установка біта RP0 в одиницю вибирає банк 1. Кожен банк має довжину 128 байт. Однак для PIC16F83 і PIC16CR83 пам'ять даних існує тільки до адреси 02FH, а для PIC16F84 і PIC16CR84 – до адреси 04FH.

Деякі регістри спеціального призначення продубльовані в обох банках, а до деяких звертання можливе лише в одному банку.

Регістри з адресами 0CH-4FH (0CH-2FH) можуть використовуватися як регістри загального призначення, що являють собою статичне ОЗП. Адреси регістрів загального призначення банку 1 відображаються на банк 0. Отже, коли встановлений банк 1, то звертання до адрес 8CH-CFH фактично адресує банк 0.

У регістрі статусу крім біта RP0 є біт RB1, що дозволяє звертатися до чотирьох банків, які можуть бути присутні в майбутніх модифікаціях цього кристала.

До комірок ОЗП можна звертатись напряму, використовуючи абсолютну адресу кожного регістра, чи за допомогою непрямой адресації, через регістр-вказівник FSR. Непряма адресація використовує біжуче значення розрядів RP1:RP0 для доступу до банків.

5.7.1. Пряма та непряма адресація даних

Коли здійснюється пряма адресація даних, молодші 7 бітів беруться у з коду операції, а два біти покажчика сторінок (RP1, RP0) з регістра статусу, як зображено на рис. 86.

Ознакою непрямой адресації є звертання до регістру INDF. Будь-яка команда, що використовує INDF (адресу 00H) у якості операнду звертається до покажчика, що зберігається в FSR (адреса 04H). Непряме читання самого регістра INDF дасть результат 00H. Непряме записування у регістр INDF призведе до втрати даних, що записувались, хоч і може змінити біти стана. Необхідна 9-бітна адреса формується об'єднанням умісту 8-бітного FSR

регістра і біта IRP з регістра статусу.

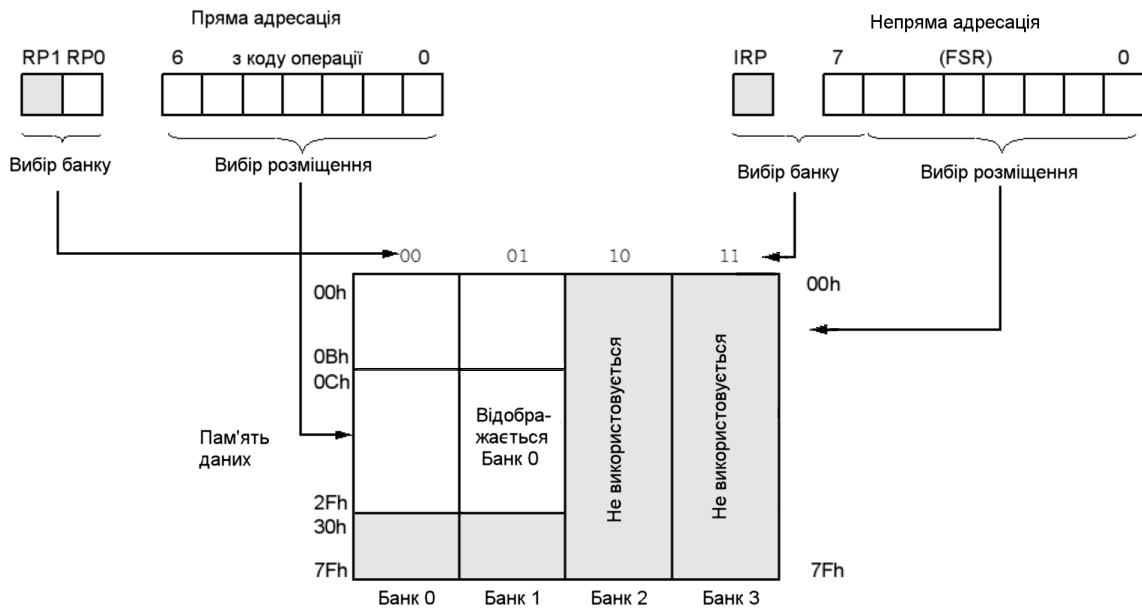


Рис. 86. Методи адресації даних

5.8. EEPROM даних PIC16X8X

Мікроконтролери підгрупи PIC16X8X мають енергонезалежну EEPROM пам'ять даних об'ємом 64x8біт, що допускає записування і читання під час нормальної роботи в усьому діапазоні напруги живлення. Ця пам'ять не належить області ОЗП даних. Доступ до неї здійснюється за допомогою непрямої адресації через регістри спеціальних функцій: EEDATA (з адресою 08H), через якій здійснюється запис і читання даних, і EEADR (адреса 09H), котрий визначає адресу комірки, до якої здійснюється звертання. Для керування процесом читання та записування використовуються регістри: EECON1 (адреса 88H) і EECON2 (адреса 89H).

При записуванні байта в комірку EEPROM попереднє значення стирається і записуються нові дані. Всі операції зі стирання і записування здійснює вбудований автомат запису EEPROM. Уміст комірок цієї пам'яті при вимиканні живлення зберігається.



Рис. 87. Формат регістра EECON1.

Регістр EEADR може адресувати до 256 байт даних EEPROM. У МК підгрупи PIC16X8X використовуються тільки перші 64 байт, адресовані шістьма молодшими бітами EEADR<5:0>. Однак старші два біти також декодуються. Тому ці два біти повинні бути встановлені в "0", щоб адреса потрапила в доступні 64 біта адресного простору.

Формат регістра EECON1 зображено на рис. 87, призначення бітів регістра наведено в таблиці 40.

Таблиця 40. Призначення біт регістра EECON1

Біт	Значення
EEIF	Біт запиту переривання по закінченню запису в EEPROM. 0 = операція запису незакінчена або не починалася. 1 = операція запису закінчена. Біт скидається програмно
WRERR	Біт ознаки помилки запису в EEPROM 0 = операція запису завершена 1 = операція запису передчасно перервана
WREN	Біт дозволу запису в EEPROM 0 = запис у EEPROM заборонений 1 = запис у EEPROM дозволений
WR	Біт керування записом. Встановлення біта в 1 ініціює цикл запису, біт скидається апаратно по завершенні запису і не може бути повернений програмно в 0
RD	Біт керування читанням. Встановлення біта в 1 ініціює цикл читання і займає один цикл. Біт скидається апаратно по завершенні читання і не може бути програмно скинутий в 0

Регістр EECON2 призначений для запобігання записування в EEPROM при збоях програми. Для дозволу запису в EEPROM перед записуванням кожного байта необхідно в регістр подати спеціальну ключову послідовність (спочатку байт 55h, потім AAh). Тоді в наступному машинному циклі при встановленні біта WR регістра EECON1 почнеться занесення даних у EEPROM. Регістр EECON2 використовується винятково при організації записування даних у EEPROM. Читання регістра EECON2 дає нулі.

При зчитуванні даних з пам'яті EEPROM необхідно записати потрібну адресу в регістр EEADR і потім встановити біт RD EECON1<0> в одиницю. Дані з'являться в наступному командному циклі в регістрі EEDATA і можуть бути прочитані. Зауважимо, що дані в регістрі EEDATA фіксуються. Немає

необхідності виконувати ще раз цикл читання у випадку повторного читання тієї ж комірки пам'яті.

При записуванні в пам'ять EEPROM необхідно спочатку записати адресу в регістр EEADR і дані в регістр EEDATA. Потім необхідно виконати спеціальну послідовність команд, що виконує записування:

```
movlw 55h      ;  
movwf EECON2  ; записати в регістр EECON2 перший  
               ; ключовий код  
movlw AAh  
movwf EECON2  ; записати в регістр EECON2 другий  
               ; ключовий код  
bsf EECON1.WR; встановити WR біт,  
               ; встановлення біта відразу після  
               ; подавання ключових кодів  
               ; починає процес записування.
```

Під час виконання цієї ділянки програми всі переривання повинні бути заборонені для точного виконання тимчасової діаграми. Час записування – приблизно 10 мс. Фактичний час записування може змінюватися залежно від напруги, температури та індивідуальних властивостей мікросхеми. В кінці записування біт WR автоматично обнулюється, а ознака завершення записування EEIF, (вона ж є запитом на переривання), встановлюється.

Для запобігання випадкових записів у пам'ять даних передбачено спеціальний біт WREN у регістрі EECON1. Рекомендується тримати біт WREN вимкненим, кодові сегменти, що встановлюють біт WREN, і ті, котрі записують дані у EEPROM, варто зберігати на різних адресах, щоб уникнути випадкового виконання їх обох при збої програми.

5.9. Регістри спеціальних функцій

Загальний опис регістрів спеціальних функцій наведено в таблиці 41. Більшість регістрів спеціальних функцій розглядаються у розділах присвячених відповідним функціональним вузлам, тому опис їх в цьому розділі не

приведено. Виключення становить регістр конфігурації.

Таблиця 41 Регістри спеціальних функцій

Адреса	Назва	Призначення
00H	INDF	Фіктивний регістр непрямої адресації. Звертання до регістра призводить звертання до комірки пам'яті з адресою, котра задана в регістрі FSR
01H	TMR0	Регістр-лічильник лічильника таймера. Зберігає біжуче значення лічильника
02H	PCL	Молодший байт лічильника команд
03H	STATUS	Регістр стана мікросхеми
04H	FSR	Вказівник при непрямій адресації
05H	PORTA	Регістр, що вказує стан порта А. Значення мають лише молодші 5 біт
06H	PORTB	Регістр, що вказує стан порта В
07H	-	Не використовується (зарезервований для порта С)
08H	EEDATA	Регістр даних для обміну з EEPROM
09H	EEADR	Регістр адреси для обміну з EEPROM
0AH	PCLATH	Старші 5 розрядів для записування в лічильник команд
0BH	INTCON	Регістр умов переривання
80H	INDF	Відображається регістр з адреси 00H
81H	OPTION	Регістр конфігурації
82H	PCL	Відображається регістр з адреси 02H
83H	STATUS	Відображається регістр з адреси 03H
84H	FSR	Відображається регістр з адреси 04H
85H	TRISA	Регістр вказує напрямок передачі порта А
86H	TRISB	Регістр вказує напрямок передачі порта В
87H	-	Не використовується (зарезервований для порта С)
88H	EECON1	Регістр режиму роботи EEPROM
89H	EECON2	Регістр ключа запису в EEPROM
8AH	PCLATH	Відображається регістр з адреси 0AH
8BH	INTCON	Відображається регістр з адреси 0BH

5.9.1. Регістр конфігурації (OPTION)

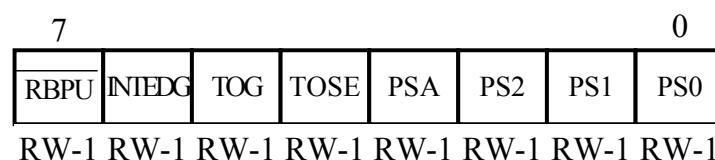


Рис. 88. Формат регістра конфігурації OPTION

Регістр конфігурації (OPTION) є доступним для читання і записування. Він містить керуючі біти для конфігурації попереднього подільника, зовнішніх

переривань, таймера, а також підтягуючих ("pull-up") резисторів на виводах PORTB. Призначення біт регістра приведено в таблиці 42, а формат регістра на рис. 88.

Таблиця 42 Формат регістра конфігурації (OPTION)

Назва	Призначення		
RBPUP	Біт установки підтягуючих «pull-up» резисторів на виводах PORTB: 0 – резистори під'єднані; 1 – резистори від'єднані		
INTEDG	Біт вибору переходу сигналу переривання: 0 – переривання по спаду сигналу на лінії RBO/INT; 1 – переривання по фронті сигналу на лінії RBO/INT		
TOCS	Біт вибору джерела сигналу таймера TMR0: 0 – внутрішній тактовий сигнал (CLKOUT); 1 – лінія RA4/TOCKI		
TOSE	Біт вибору переходу джерела сигналу для TMR0: 0 – збільшення по фронті сигналу на лінії RA4/TOCKI; 1 – збільшення по спаду сигналу на лінії RA4/TOCKI		
PSA	Біт призначення подільника: 0 – попередній подільник під'єднаний до TMR0; 1 – попередній подільник під'єднаний до сторожового таймера WDT		
PS2, PS1, PS0	Біти вибору коефіцієнта поділу попереднього подільника		
	PS2 PS1 PS0	Коефіцієнт поділу для TMR0	Коефіцієнт поділу для WDT
	000	2	1
	001	4	2
	010	8	4
	011	16	8
	100	32	16
	101	64	32
	110	128	64
111	256	128	

Коли попередній подільник непотрібний його необхідно, налаштувати на роботу з WDT з коефіцієнтом поділу 1 (PSA = 1, PS2 PS1 PS0 = 000).

5.10. Порти вводу-виводу

Контролери PIC16F8X мають два порти: PORTA (5 бітів) і PORTB (8 бітів) із побітовим індивідуальним настроюванням на введення чи на виведення.

Порт А (PORTA) – це 5-бітовий фіксатор, що відповідає виводам

контролера RA<4:0>. Лінія RA4 має вхід тригера Шмитта і вихід з відкритим стоком. Усі інші лінії порту мають ТТЛ вхідні рівні і КМОП вихідні буфери. Адреса регістра порту А – 05Н.

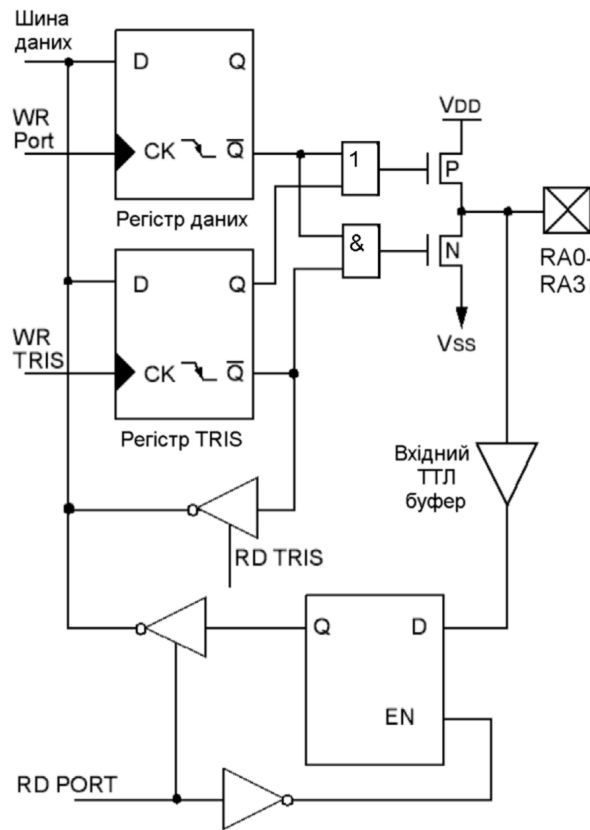


Рис. 89. Схема ліній RA<3:0> порта А

Кожній лінії порту поставлений у відповідність біт напрямку передавання даних, у керуючому регістрі TRISA, розташованому за адресою 85Н. Якщо біт регістра TRISA має значення 1, то відповідна лінія буде встановлюватися на ввід. Нуль перемикає лінію на вивід і одночасно виводить на неї вміст відповідного регістра-фіксатора порту. При ввімкненні живлення всі лінії порту за замовчуванням налаштовані на ввід. На рис. 89 зображена схема ліній RA<3:0> порту А. Зауважимо, що виводи порту мають захисні діоди до VDD і VSS.

Операція читання порту А зчитує стан виводів порту, а запис у порт змінює стан тригерів регістра даних. При скиданні та встановленні ліній порту слід проявляти обережність. При читанні порту зчитується стан вихідний лінії, а не внутрішньої засувки, одже, якщо в якийсь момент часу вихід буде тимчасово "посаджений" у нуль, цей нуль і буде зчитаним навідь коли лінія

повинна виводити логічну одиницю.

Вивід RA4 мультиплексований з тактовим входом таймера TMR0. Схема лінії RA4 порту A приведена на рис. 90.

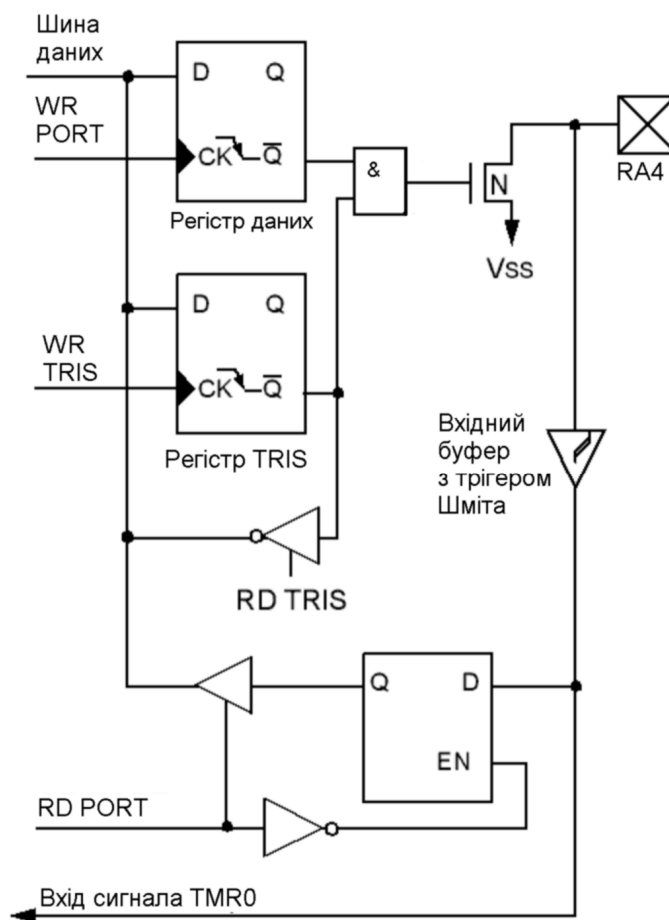


Рис. 90. Схема лінії RA4 порту A

Порт В (PORTB) – це двонаправлений 8-бітовий порт, що відповідає виводам $RB\langle 7:0 \rangle$ контролера і розташований за адресою 06H. Кожній лінії порту поставлений у відповідність біт напрямку передавання даних, що зберігається в керуючому регістрі TRISB, розташованому за адресою 86H. Якщо біт керуючого регістра TRISB має значення 1, то відповідна лінія буде встановлюватися на ввід. Нуль перемикає лінію на вивід і виводить на неї вміст відповідного регістра засувки. При вмиканні живлення всі лінії порту за замовчуванням налаштовані на ввід. Вивід порту має захисний діод тільки до Vss.

До кожної ніжки порту В підключене невелике активне навантаження (таке, що дає струм коло 10 мкА). Воно автоматично від'єднується, якщо цей

вивід запрограмований як вихід. Керуючий біт \overline{RBPU} регістра OPTION<7> може від'єднати (при $\overline{RBPU} = 1$) всі навантаження. Скидання при вмиканні живлення також відключає всі навантаження.

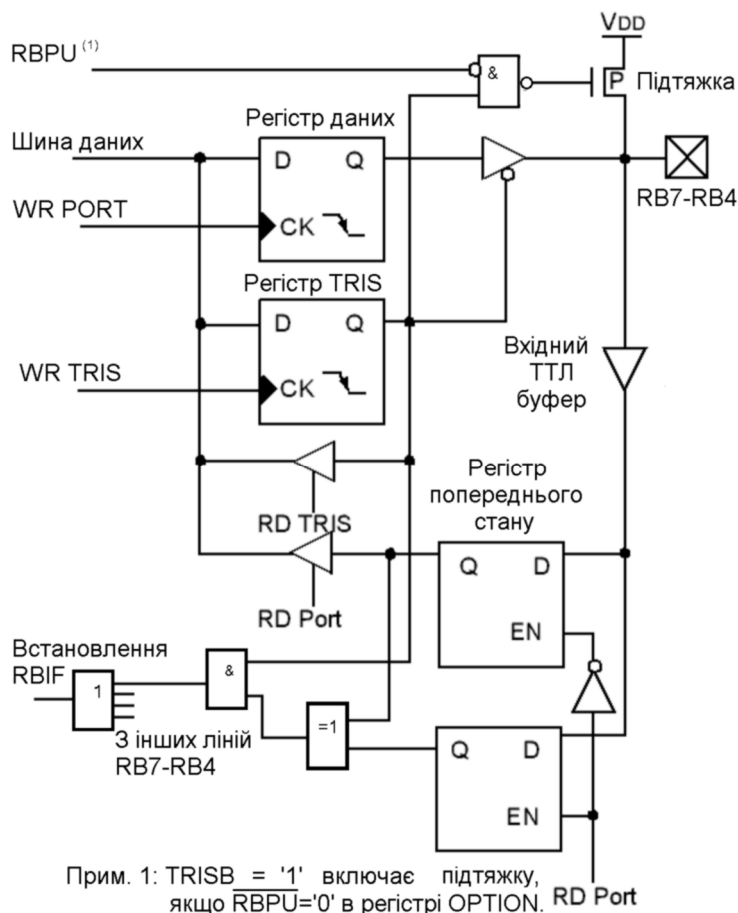


Рис. 91. Схема ліній RB<7:4> порту В

Чотири лінії порту В (RB<7:4>) можуть викликати переривання при зміні значення сигналу на кожній з них. Якщо ці лінії налаштовані на ввід, то вони опитуються і запам'ятовуються в циклі читання Q1. Нове значення вхідного сигналу порівнюється зі старим у кожному командному циклі. При розбіжності значення сигналу на ніжці й у фіксаторі генерується запит переривання.

Виходи детекторів різниці ліній RB4, RB5, RB6, RB7 поєднуються по АБО і генерують переривання RBIF (запам'ятовується в регістрі INTCON<0>). Будь-яка лінія, налаштована на вивід, участі в порівнянні не бере.

У підпрограмі обробки переривання варто скинути запит переривання одним із таких способів:

- ◇ прочитати (чи записати) порт В. Це зніме стан порівняння;

◇ обнулити біт RBIF регістра INTCON<0>.

Водночас слід мати на увазі, що умова зміни буде продовжувати встановлювати ознаку RBIF. Тільки читання порту В може усунути розбіжність і дозволить обнулити біт RBIF. Переривання по зміні стану і програмно установлені внутрішні активні навантаження на цих чотирьох лініях можуть забезпечити простий інтерфейс, наприклад, із клавіатурою, з виходом із режиму SLEEP по натисканню клавіш. Схеми ліній порту В приведені на рис. 91 і рис. 92.

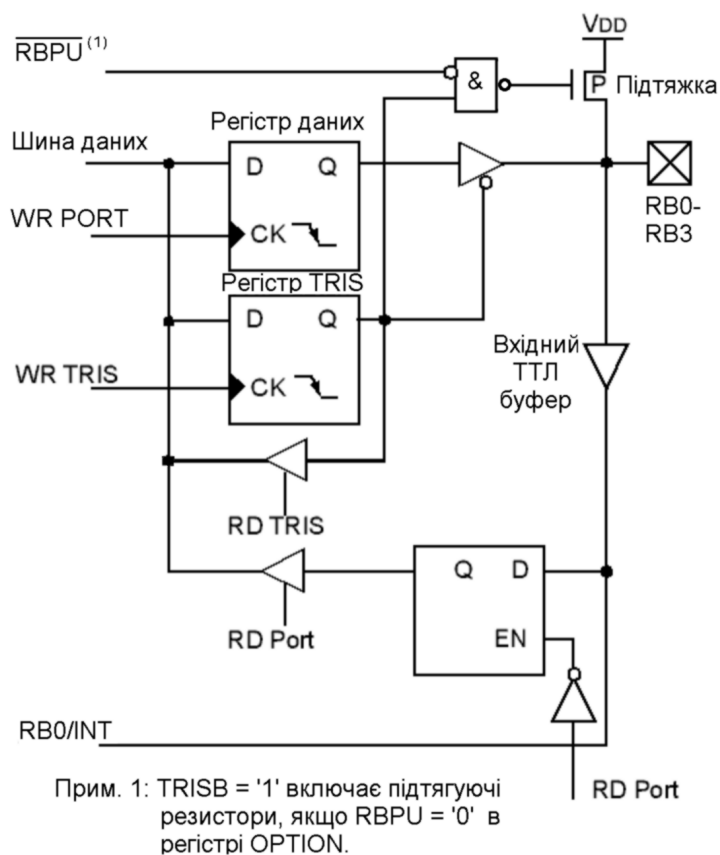


Рис. 92. Схема ліній RB<3:0> порту В

При організації двонаправлених портів необхідно враховувати особливості організації вводу-виводу даних МК. Будь-яка команда, що здійснює встановлення або скидання ліній порту, виконується як команда "читання-модифікація-запис". Наприклад, команди BCF і BSF зчитують порт повністю, модифікують один біт і виводять результат назад. Наприклад, команда BSF PORTB, 5 (установити в одиницю біт 5 порту В) спочатку зчитує значення всіх сигналів, що є в даний момент на виводах порту. Потім виконують дії над бітом 5, і нове значення байта записується у вихідні

фіксатори. Якщо інший біт регістра PORTB використовується в якості двонаправленого вводу-виводу (наприклад, біт 0) і в даний момент він визначений як вхідний, то вхідний сигнал на цьому виводі буде зчитаний і записаний назад у вихідний тригер-фіксатор цього ж виводу, стираючи попередній стан. Доти, поки ця лінія залишається в режимі вводу, ніяких проблем не виникає. Якщо ж пізніше лінія 0 переключиться в режим виводу, її стан буде змінено порівняно з попереднім.

5.11. Модуль таймера PIC16X8X

Структура модуля таймера-лічильника TIMER0 і його взаємозв'язок з регістрами TMR0 і OPTION зображені на рис. 93. TIMER0 є програмованим модулем і містить такі компоненти:

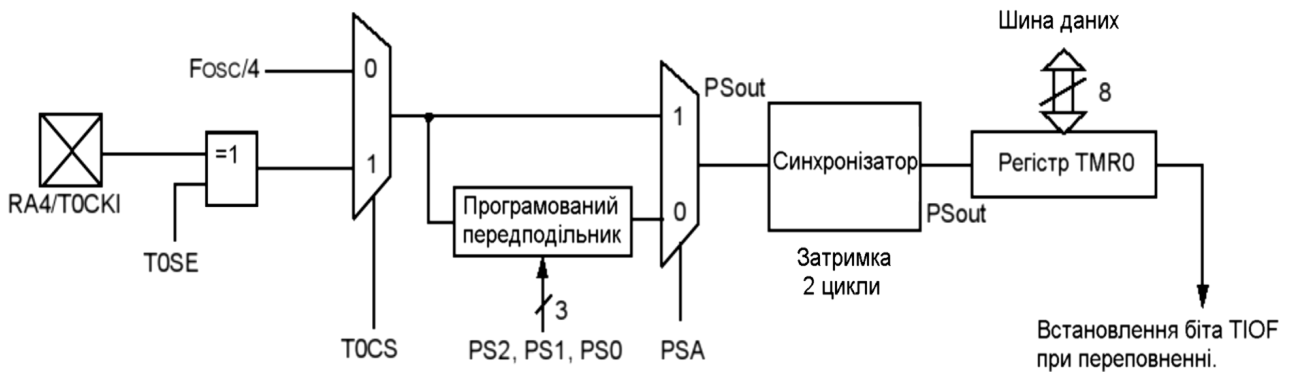
- ◇ 8-розрядний таймер-лічильник TMR0 з можливістю читання і записування;
- ◇ 8-розрядний програмно керований попередній подільник;
- ◇ мультиплексор для вибору внутрішнього чи зовнішнього тактового сигналу;
- ◇ схему вибору фронту зовнішнього тактового сигналу;
- ◇ формувач запиту переривання по переповненню регістра TMR0 (з стану FFH у стан 00H).

Режим таймера вибирається шляхом скидання в нуль біта TOCS регістра OPTION <5>. У режимі таймера TMR0 інкрементується кожен командний цикл (без подільника). Після записування інформації в TMR0 інкрементування його почнеться після двох командних циклів. Це відбувається з усіма командами, що проводять записування чи модифікацію TMR0 (наприклад, MOVF TMR0, CLRF TMR0). Якщо потрібно перевірити чи дорівнює TMR0 нулю без зупинки відліку, слід використовувати інструкцію MOVF TMR0,W.

Режим лічильника вибирається шляхом установки в одиницю біта TOCS регістра OPTION<5>. У цьому режимі регістр TMR0 буде інкрементуватись зростаючим або спадаючим фронтом на виводі RA4/ТОСКІ. Напрямок фронту

визначається керуючим бітом TOSE у регістрі OPTION<4>. При TOSE = 0 буде обраний зростаючий фронт.

Попередній дільник може використовуватися разом з TMR0, чи з Watchdog- таймером. Під'єднання попереднього подільника контролює біт PSA регістра OPTION<3>. При PSA = 0 попередній подільник буде приєднаний до TMR0, вміст попереднього подільника програмі недоступний. Коефіцієнт поділу попереднього подільника програмується бітами PS2...PS0 регістра OPTIOM<2:0>.



Прим. 1: Біти TOCS, TOSE, PS2, PS1, PS0 та PSA знаходяться в регістрі OPTIONS.

2: Передподільник спільний для TMR0 та Watchdog-таймера.

Рис. 93. Структурна схема таймера-лічильника TMR0

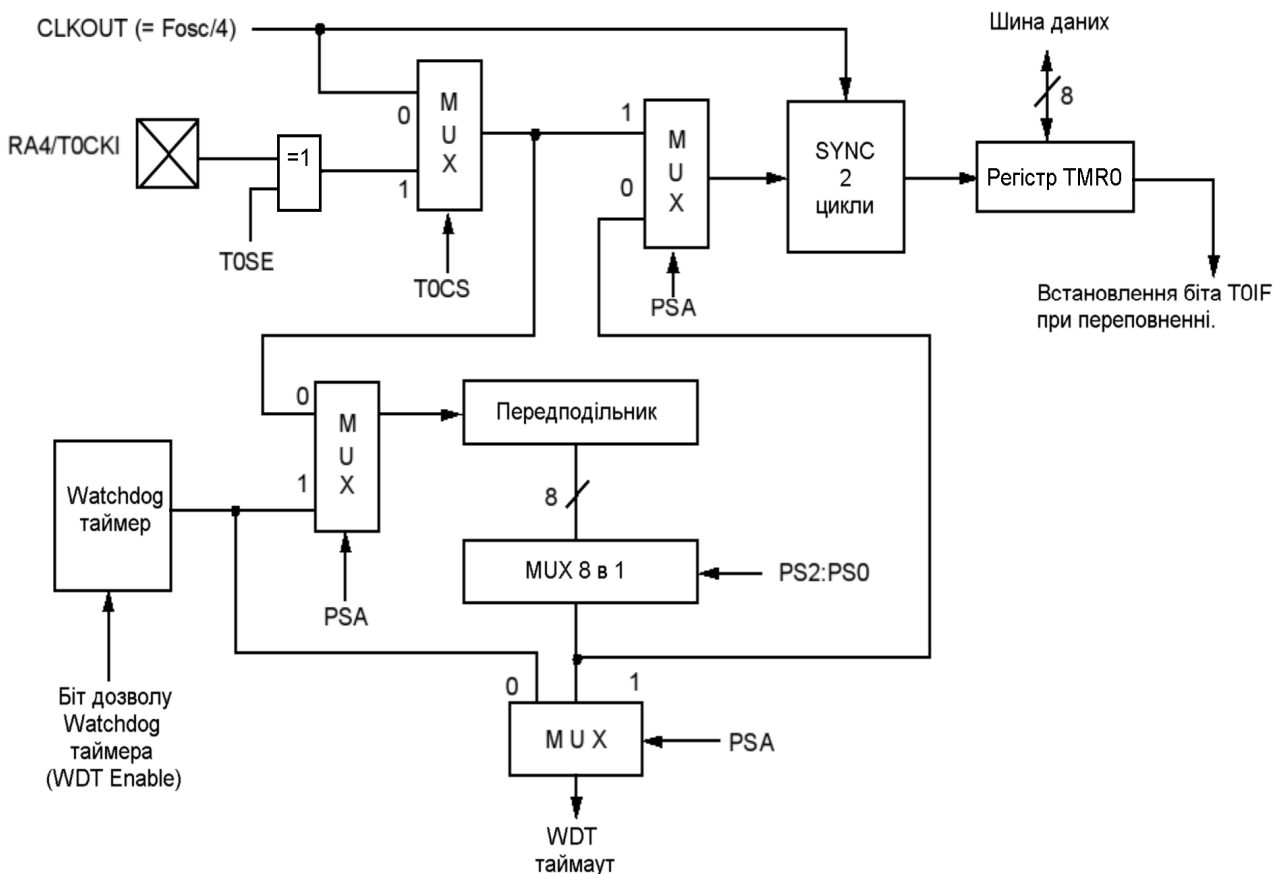


Рис. 94. Структура і можливі варіанти використання подільника

Переривання по TMR0 здійснюється, коли відбувається переповнення регістра таймера/лічильника при переході від FFH до 00H. У момент переповнення встановлюється біт запиту TOIF у регістрі INTCON<2>. Дане переривання можна замаскувати бітом TOIE у регістрі INTCON<5>. Біт запиту TOIF повинен бути скинутий програмно при обслуговуванні переривання. Переривання по TMR0 не може вивести процесор із режиму SLEEP тому, що таймер у цьому режимі не функціонує.

При PSA=1 попередній подільник буде приєднаний на вихід Watchdog-таймера. Можливі варіанти використання подільника зображені на рис. 94.

При використанні попереднього подільника разом із TMR0, усі команди, що змінюють вміст TMR0, обнулюють передподільник. Якщо попередній подільник використовується разом з WDT, команда CLR WDT обнуляє вміст попереднього подільника разом з WDT.

При використанні модуля TIMER0 у режимі лічильника зовнішніх подій необхідно враховувати, що зовнішній тактовий сигнал синхронізується внутрішньою частотою F_{osc} . Це призводить до появи затримки в часі фактичного інкрементування вмісту TMR0.

Синхронізація відбувається у момент закінчення 2-го і 4-го тактів роботи МК. Якщо передподільник не використовується, то для фіксації вхідної події необхідно, аби тривалості високого і низького станів сигналу на вході RA4/ТОСКИ були не менше двох періодів тактової частоти T_{osc} плюс деяка затримка (≈ 20 нс).

Якщо модуль TIMER0 використовується разом із попереднім подільником, то частота вхідного сигналу ділиться асинхронним лічильником так, що сигнал на виході попереднього подільника стає симетричним. При цьому необхідно, щоб тривалості високого і низького рівнів сигналу на вході RA4/ТОСКИ були не менше 10 нс. Синхронізація сигналу відбувається на виході подільника, тому існує невелика затримка між фронтом зовнішнього сигналу і часом фактичного інкременту таймера-лічильника. Ця затримка

знаходиться в діапазоні від 3 до 7 періодів тактового генератора.

5.12. Організація переривань PIC16F8X

МК підгрупи PIC16X8X мають чотири джерела переривань:

- ◇ зовнішнє переривання з виводу RB0/INT;
- ◇ переривання від переповнення лічильника/таймера TMR0;
- ◇ переривання від зміни сигналів на лініях порту RB<7:4>;
- ◇ переривання в момент закінченні запису даних у EEPROM.

Усі переривання мають той самий вектор та адресу початку процедури обслуговування (0004H). Однак у керуючому регістрі переривань INTCON відповідним бітом-ознакою записується, від якого саме джерела надійшов запит переривання. Вийняток складає переривання в момент завершення записування в EEPROM, ознака якого знаходиться в регістрі EECON1.

Регістр умов переривання (INTCON) є доступним для читання та записування. Призначення бітів регістра наведено у таблиці 43, а формат регістра – на рис. 95.

Таблиця 43 Біти регістра умов переривання

Назва	Призначення
GIE	Біт дозволу всіх переривань. Нуль вказує, що всі переривання заборонені. 1 дозволяє незамасковані переривання
EEIE	Біт дозволу переривання запису в EEPROM. Нуль вказує на заборону переривання в момент закінчення записування в EEPROM, одиниця – на дозвіл
TOIE	Біт дозволу переривання в момент переповнення TMR0. Нуль вказує на заборону переривання від TMR0, 1 – на дозвіл
INTE	Біт дозволу переривань з входу RB0/INT. Нуль вказує на заборону переривання, 1 – на дозвіл
RBIE	Біт дозволу переривань по зміні PORTB. Нуль вказує на заборонену переривань, 1 – на дозвіл
TOIF	Біт переривання по переповненню TMR0. Нуль вказує що переповнення TMR0 не було, один – переповнення TMR0 має місце
INTF	Біт запиту переривання по входу RB0/INT. Нуль – переривання по входу RB0/INT відсутнє, 1 – переривання по входу RB0/INT має місце
RBTF	Біт запиту переривання по зміні PORTB: нуль вказує, що на жодному з входів RB7-RB4 стан не змінився з звертання, 1 – хоча б на одному з входів RB7-RB4 змінився стан.

Переривання INT може вивести процесор з режиму SLEEP, якщо перед

входом у цей режим біт INTE був встановлений в одиницю. Стан біта GIE також визначає чи буде процесор переходити на підпрограму переривання після виходу з режиму SLEEP. Варто звернути увагу, що скидання біт запитів переривань здійснюється відповідною програмою обробки переривання.

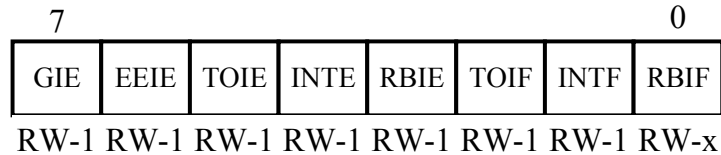


Рис. 95. Формат регістра умов переривання

Біт загального дозволу/заборони переривання GIE (INTCON <7>) дозволяє (GIE = 1) всі індивідуально незамасковані переривання, або забороняє їх (у випадку GIE = 0). У момент скидання біт GIE обнулюється. Кожне переривання окремо може бути додатково дозволено встановленням відповідного біта в регістрі INTCON. Скидання відповідного біту забороняє кожне переривання окремо.

Біт дозволу всіх переривань GIE скидається автоматично при наступних обставинах:

- ◇ по включенню живлення;
- ◇ по зовнішньому сигналі /MCLR при нормальній роботі;
- ◇ по зовнішньому сигналі /MCLR у режимі SLEEP;
- ◇ по закінченню затримки таймера WDT при нормальній роботі;
- ◇ по закінченню затримки таймера WDT у режимі SLEEP.

Також біт GIE обнулюється на початку процедури обслуговування переривання, щоб заборонити повторне входження у переривання. Адреса повернення посилається в стек, а в програмний лічильник завантажується адреса 0004H. Час реакції на переривання для зовнішніх подій, таких, як переривання від лінії INT чи порту B, складає програмних п'ять циклів. Це на один цикл менше, ніж для внутрішніх подій, таких, як переривання у зв'язку з переповненням таймера TMRO.

У підпрограмі обслуговування переривання джерело переривання може бути визначене за відповідним бітом у регістрі INTCON. Ознака джерела

переривання повинна бути програмно скинута у підпрограми обслуговування. Ознаки запитів переривань не залежать від відповідних маскуючих бітів і біта загального маскування GIE.

Команда повернення з переривання RETFIE завершує підпрограму, що перериває, і встановлює біт GIE, щоб знову дозволити переривання. Логіка переривань контролера зображена на рис. 96.

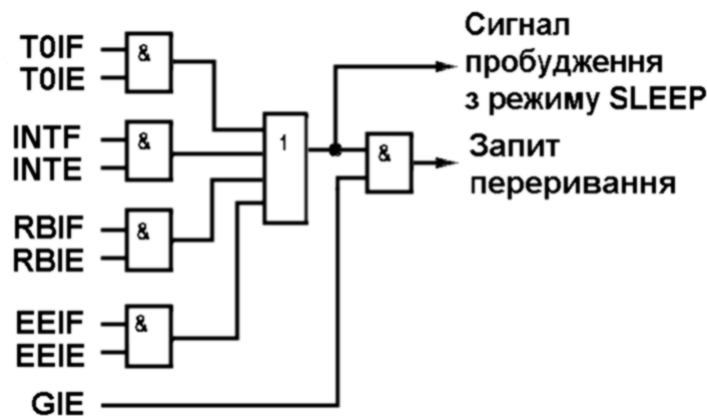


Рис. 96. Логіка переривань мікроконтролера

Зовнішнє переривання на лінії RB0/INT здійснюється або по наростаючому (якщо в регістрі OPTION біт INTEDG=1), або по спадаючому (якщо INTEDG=0) фронту. Коли фронт виявляється на лінії INT, біт запиту INTF встановлюється в одиницю (INTCON <1>). Це переривання може бути замасковано скиданням керуючого біта INTE у нуль (INTCON <4>). Біт запиту INTF необхідно очистити підпрограмою обслуговування переривання перед тим, як знову дозволити його. Переривання INT може вивести процесор з режиму SLEEP, якщо перед входом у цей режим біт INTE був встановлений в одиницю. Стан біта GIE також визначає, чи буде процесор переходити на підпрограму переривання після виходу з режиму SLEEP.

Переповнення лічильника TMR0 (перехід з стану FFH в стан 0H) встановлює в одиницю біт запиту T0IF (INTCON<2>). Це переривання може бути дозволено або заборонено установкою біта маски TOIE (INTCON<5>). Скидання запиту T0IF – справа підпрограми обслуговування переривання.

Будь-яка зміна сигналу на одному з чотирьох входів порту RB<7:4> встановлює в одиницю біт RBIF (INTCON<0>). Це переривання може бути

дозволено або заборонено встановленням біта маски RBIE (INTCON<3>). Скидання запиту RBIF – справа підпрограми обслуговування переривання.

Ознака запиту переривання в момент завершення записування даних в EEPROM, EEIF (EECON1<4>) встановлюється в одиницю в момент закінчення записування даних. Це переривання може бути замасковано скиданням біта EEIE (INTCON<6>).

5.13. Початкова ініціалізація та встановлення у початковий стан

5.13.1. Джерела скидання

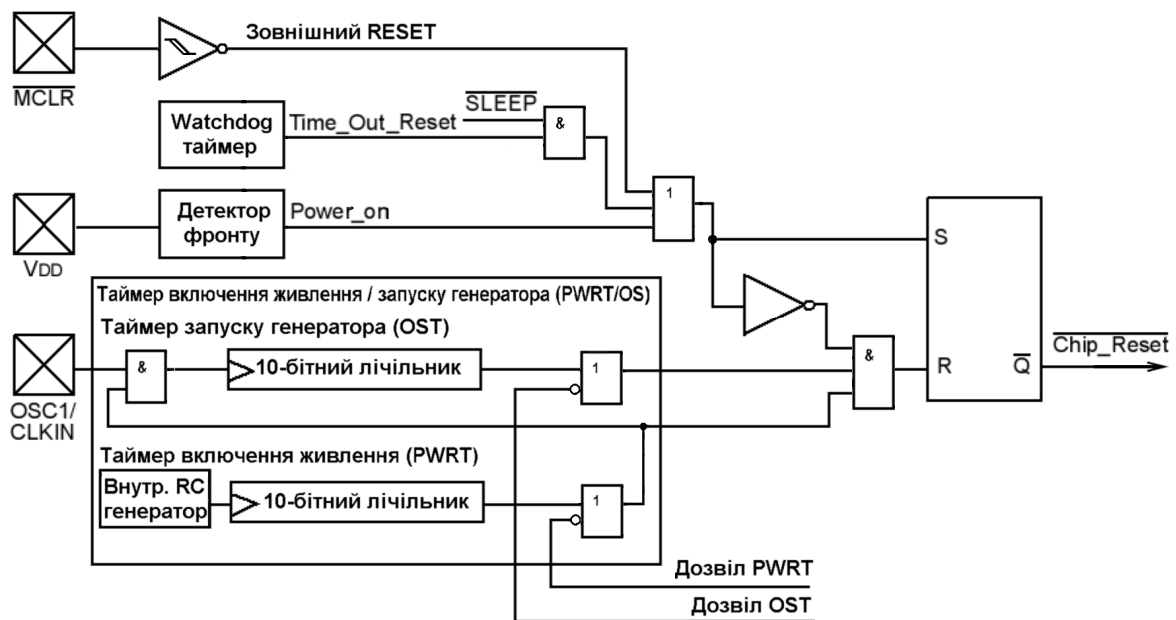


Рис. 97. Схема формування сигналу скидання

PIC16X8X має кілька джерел сигналу скидання:

- ◇ при ввімкненню живлення (Power-on Reset) POR;
- ◇ по зовнішньому сигналу MCLR;
- ◇ по сигналу Watchdog таймера (WDT Reset);
- ◇ за таймером ввімкнення живлення;
- ◇ за таймером запуску генератора.

Сигнал скидання з джерел скидання формується за допомогою схеми, зображеної на рис. 97.

Таймер ввімкнення живлення (PWRT) дає фіксовану витримку часу в

72 мс (номінальне значення) при ввімкненні живлення. Таймер працює на внутрішньому RC-генераторі. Він дозволяє дочекатись стабілізації напруги живлення в процесі запуску контролера.

Таймер запуску генератора дозволяє після ввімкнення живлення або при виході із режиму зниженого енергоспоживання (SLEEP) дочекатись стабілізації частоти та режиму генерації кварцового чи керамічного резонатора. Таймер очікує надходження з генератора 1024 імпульсів перед зняттям сигналу скидання. Таймер ввімкнення живлення і таймер запуску генератора можуть бути вимкнені у разі необхідності. Таймер запуску генератора, найчастіше, не використовується при роботі з RC генератором, і при роботі з зовнішнім джерелом синхросигналів.

Таймер ввімкнення живлення може не використовуватись у системах з зовнішнім формувачем сигналу скидання, наприклад, у системах з супервізором живлення.

5.13.2. Скидання при ввімкненні живлення

Кристал PIC16X8X має вбудований детектор ввімкнення живлення. Коли напруга живлення перевищить рівень 1,2...1,7 В, формується сигнал скидання, що включає таймер ввімкнення живлення запуску. Після закінчення витримки (близько 72 мс) вважається, що напруга досягла номіналу і запускається інший таймер витримки для стабілізації кварцового генератора. Програмований біт конфігурації дозволяє або забороняє витримку від згаданих вбудованих таймерів запуску. Витримка запуску змінюється залежно від кристала, живлення і температури.

Таймер для стабілізації генератора відраховує 1024 імпульси від генератора, який почав роботу. Вважається, що кварцовий генератор за цей час ввійшов у нормальний режим роботи. При використанні RC-генераторів витримка часу для стабілізації частоти не використовується.

Далі вмикається таймер чекання зовнішнього скидання MCLR. Це необхідно для тих випадків, коли потрібно синхронно запуснути в роботу

кілька PIC-контролерів через загальний для всіх сигнал MCLR. Якщо такого сигналу немає, то через час Tost формується внутрішній сигнал скидання і контролер починає працювати за програмою. Час Tost програмується бітами конфігурації в EEPROM.

Коли VDD наростає надто повільно і, попри всі затримки на запуск, живлення ще не досягло свого мінімального значення VDD(min) для нормального функціонування, рекомендується використовувати зовнішні RC-ланки для скидання по сигналу MCLR (рис. 98). В інших випадках необхідності в такому колі немає.

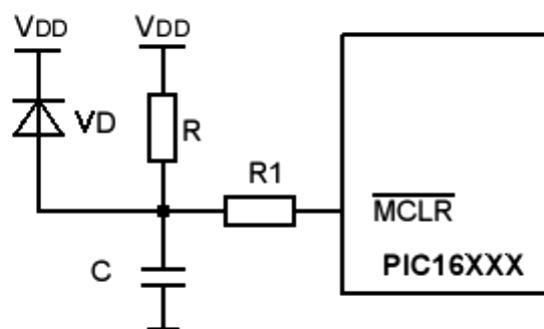


Рис. 98. Зовнішнє коло скидання

Рекомендується вибирати опір резистора R не більше 40 кОм. Опір резистора R1, що обмежує струм через лінію вибирають у межах 100 Ом - 1 кОм.

5.13.3. Watcdog-таймер

Watchdog-таймер (WDT) є повністю незалежним таймером, що працює незалежно від інших блоків мікросхеми. Робота таймера полягає в очікуванні протягом певного періоду звертання від основної програми і формуванні сигналу скидання після закінчення заданого інтервалу часу у випадку відсутності звертань.

Команди "CLRWDT" і "SLEEP" обнулюють WDT і попередній подільник, якщо він під'єднаний до WDT. Якщо сигнал скидання від WDT все ж відбувся, то одночасно обнулюється біт "TO" в регістрі статусу (STATUS). Формування сигналу скидання може бути заборонено записом у біт конфігурації WDTE. Така операція виконується на етапі програмування мікросхеми.

Номінальна тривалість витримки часу WDT складає 18 мс (без

використання попереднього подільника). Вона залежить від температури, напруги живлення та особливостей кристала. Якщо потрібна більший інтервал витримки часу, то до WDT може бути під'єднаний внутрішній попередній подільник з коефіцієнтом поділу до 128, що програмується шляхом запису в регістр OPTION. У результаті можуть бути реалізовані витримки часу до 2.5 секунди. Зауважимо, що такий метод формування затримок не є точним – параметри генератора WDT залежать від напруги живлення, температури та інших факторів.

5.14. Режим зниженого енергоспоживання

Вхід у режим SLEEP здійснюється командою SLEEP. За цією командою, якщо WDT дозволений, він скидається і починає відлік часу, скидається також біт "PD" у регістрі статусу (STATUS), біт "TO" встановлюється в "1", а генератор тактових сигналів вимикається. Порти вводу-виводу зберігають стан, який вони мали до входу в режим SLEEP.

Вихід з режиму SLEEP здійснюється в результаті:

- ◇ зовнішнього скидання – імпульсу низького рівня на виводі MCLR;
- ◇ скидання при спрацюванні WDT (якщо він дозволений);
- ◇ переривання з виводу INT, або при зміні стана лінії порту B, або при завершенні записування даних в EEPROM.

При першій вказаній події відбувається скидання всього пристрою. Дві інші події допускають продовження виконання програми.

Розглянемо вихід за зовнішнім скиданням. При ввімкненні живлення біт "PD" у регістрі статусу (STATUS) встановлюється в 1, а при формуванні скидання іншого виду він залишається в попередньому стані. Обнулення біта може здійснюватися командою "SLEEP". Отже він може бути використаний для ідентифікації стана процесора до скидання: чи процесор знаходився в режимі "SLEEP" (гарячий старт), чи було виключене живлення (холодний старт). Біт "TO" дозволяє визначити, чим був викликаний вихід із режиму SLEEP: зовнішнім сигналом на виводі MCLR, чи спрацюванням WDT. Проте такий

метод виходу призводить до втрати попереднього (до засинання) стана деяких регістрів, у тому числі лічильника команд. Тому ним не варто користуватися без особливої необхідності.

Для виходу з режиму SLEEP за допомогою переривань переривання повинні бути дозволені встановленням відповідної маски в регістрі INTCON. При виході з режиму SLEEP буде виконуватися команда, що йде за командою SLEEP, якщо біт загального дозволу переривань GIE обнулений, у протилежному випадку керування буде передано в підпрограму обслуговування переривань зі збереженням у стеку адреси команди, наступної за командою SLEEP.

5.15. Генератор та синхронізація

5.15.1. Типи генераторів

PIC16X8X може працювати з чотирма типами генераторів. Програміст може, користуючись двома конфігураційними бітами (FOSC1 і FOSC0), обрати один із режимів:

- ◇ RC-генератор.
- ◇ LP – низькочастотний кварцовий (керамічний) резонатор. Частоти резонансу до 100-200 кГц. Найчастіше використовують резонатори на 32 кГц та 200 кГц.
- ◇ XT – середньочастотний резонатор. Частоти резонансу від 100-200 кГц до 4МГц.
- ◇ HS – високочастотний резонатор. Частоти резонансу від 4МГц і вище.

5.15.2. Робота з кварцовим резонатором

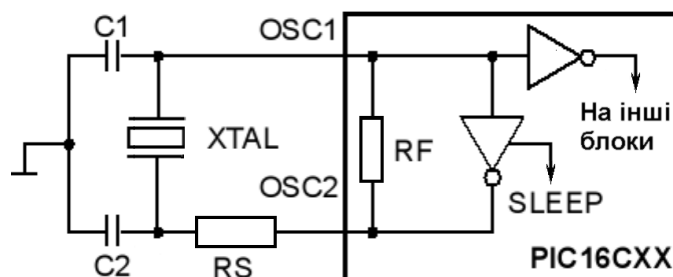


Рис. 99. Підключення кварцового резонатора

Схема включення кварцового резонатора показана на рис. 99. При роботі в режимах XT, LP, HS керамічний, або кварцовий резонатор під'єднується між виводами OSC1/CLKIN та OSC2/CLKOUT. Ємності C1 та C2 призначені для підвищення надійності запуску і роботи генератора: збільшення ємності призводить до підвищення стабільності, проте збільшує час запуску, який слід враховувати при виході з режиму зниженого енергоспоживання.

Типове значення ємності для резонаторів на частоту від 100 кГц і вище коло 33 пФ, для резонаторів на 32 кГц – 68 пФ. Резистор RS може бути потрібним для деяких типів резонаторів для гасіння вищих гармонік коливань.

Залежно від режиму роботи XT, LP, HS змінюється значення внутрішнього резистора RF, тому підбір типу резонатора необхідно здійснювати одночасно з вибором режиму роботи, користуючись при цьому документацією виробника мікросхеми.

5.15.3. Синхронізація від зовнішніх джерел

PIC16X8X може також синхронізуватися від зовнішніх джерел сигналу. Типова схема ввімкнення в такому режимі роботи зображена на рис. 100. При роботі в цьому режимі обов'язково повинен бути обраний режим синхронізації від кварцового резонатора (XT, LP або HS), у випадку вибору RC-резонатора можливе пошкодження кристала.

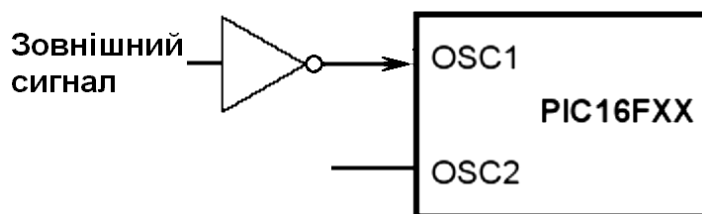


Рис. 100. Синхронізація від зовнішніх джерел

5.15.4. RC-генератор

Коли не пред'являються високі вимоги до точності відліку часу, зручно використовувати для тактування мікросхеми RC-генератор. Схема включення RC-генератора зображена на рис. 101. На виводі OSC2/CLKOUT видається сигнал з частотою, що дорівнює 1/4 частоти генератора. Сигнал може бути використаний для синхронізації інших мікросхем.

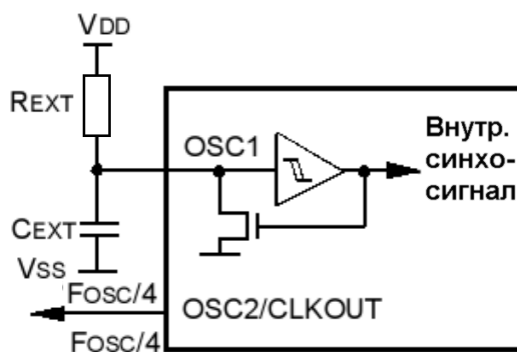


Рис. 101. RC генератор

Типові значення $R_{ext} = 5 - 100$ кОм, $C_{ext} \geq 20$ пФ. Наближені середні значення частоти синхронізації F_{osc} , похибка встановлення частоти при напрузі живлення 5В та температурі 25°C наведені в таблиці 44.

Таблиця 44. Середні значення частоти синхронізації

C_{ext}	R_{ext}	F_{osc}	Похибка	C_{ext}	R_{ext}	F_{osc}	Похибка
20 пФ	3.3к	4.68 МГц	± 27%	100 пФ	10к	620 кГц	± 30%
	5.1к	3.94 МГц	± 25%		100к	90.25 кГц	± 26%
	10к	2.34 МГц	± 29%	300пФ	3.3 к	524 кГц	± 28%
	100к	250 кГц	± 33%		5.1к	415 кГц	± 30%
100 пФ	3.3к	1.49 МГц	± 25%		10к	270 кГц	± 26%
	5.1к	1.12 МГц	± 25%		100к	25.4 кГц	± 25%

Для значень R_{ext} , нижчих 2.2 кОм, генератор може працювати нестабільно або не запускатися взагалі. При дуже великих значеннях R_{ext} (наприклад, 1 МОм) генератор стає чутливим до завад, вологості і струму через ізоляцію корпусу.

5.16. Конфігурація та захист

5.16.1. Ідентифікаційний код

PIC16X8X має чотири слова, розташовані за адресами 2000H-2003H. Вони призначені для збереження ідентифікаційного коду (ID), контрольної

суми або іншої інформації. Разом із конфігураційним словом, вони можуть бути прочитані та записані лише за допомогою програматора. Доступу за командами програми до них немає.

Якщо кристал захищений, користувачу рекомендується використовувати для ідентифікації тільки молодші сім бітів кожного слова ідентифікаційного коду, а в старші біти записувати 0.

5.16.2. Конфігураційне слово

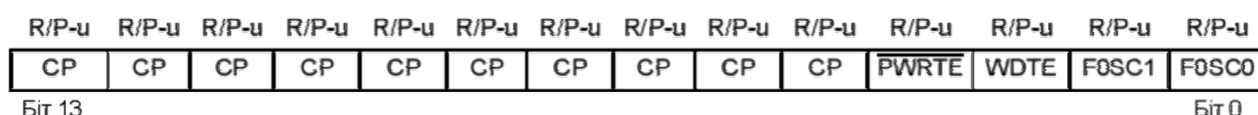


Рис. 102. Формат слова конфігурації

Призначення бітів слова конфігурації наведено в таблиці 45, а його формат – на рис. 102. PIC16F84 має п'ять бітів конфігурації, що зберігаються в пам'яті програм і встановлюються на етапі програмування кристала. Ці біти можуть бути запрограмовані (читаються як 0) або залишені незапрограмованими (читаються як 1) для вибору необхідного варіанта конфігурації мікросхеми. Вони розташовані в пам'яті програм за адресою 2007H. Слід зауважити, що ця адреса знаходиться поза областю кодів і програмно недоступна.

Програмний код, записаний на кристалі, може бути захищений від зчитування за допомогою встановлення біта захисту (CP) у слові конфігурації в нуль. Якщо встановлено захист, то біт CP можна стерти тільки разом зі стиранням умісту кристала.

Таблиця 45. Призначення біт слова конфігурації

Позначення	Призначення.
CP	Біт захисту коду. Одиниця вказує, що захист коду ввімкнений, нуль – на відсутність захисту
PWRTE	Біт дозволу витримки часу після включення живлення. Одиниця вказує на наявність витримки в 72 мс при включенні, нуль – на відсутність витримки. Біт не впливає на роботу таймера запуску генератора
WDTE	Біт дозволу роботи Watchdog таймера. Одиниця вказує на дозвіл роботи Watchdog-таймера, нуль – на заборону скидання по Watchdog-таймеру

FOSC1, FOSC0	Біти вибору типу генератора. Усі комбінації крім FOSC1,FOSC0 = 11 указують на роботу з кварцовим або керамічним резонатором: FOSC1, FOSC0 = 00 задає режим з низькочастотним резонатором (LP); FOSC1, FOSC0 = 01 задає режим з середньочастотним резонатором (XT); FOSC1, FOSC0 = 01 задає режим з високочастотним резонатором (HS); FOSC1, FOSC0 = 01 задає роботу RC-генератором.
-----------------	---

5.17. Система команд PIC-контролерів серії PIC16X8X

Кожна команда являє собою 14-розрядне слово, що містить поле коду операції OPCODE і поле операндів. Система команд включає команди роботи з байтами і бітами, команди керування й операції з константами.

Усі команди виконуються протягом одного командного циклу крім, таких двох випадків:

- ◇ Виконання команд умовних переходів, якщо результат перевірки умови – істина.
- ◇ Лічильник команд змінив значення у результаті виконання команди.

У цих випадках команда виконується за два цикли з виконанням другого циклу як NOP.

Команди однокристалної мікроЕОМ наведено в таблиці 46. У таблиці для команд роботи з байтами F позначає регістр, з яким виконується дія, W операційний регістр. У форматі кода команди використовуються позначення:

fff ffff – адреса регістра, над умістом котрого виконується дія.

d – вид регістра призначення. Якщо d = 0 результат записується в регістр W, якщо d = 1 – у регістр F, заданий у команді.

bbb –номер біта, з яким відбувається робота.

kkkk kkkk, kkk kkkk kkkk – 8- або 11-бітову константу чи ідентифікатор.

Таблиця 46. Система команд мікроЕОМ PIC16x8x

Позначення	Функція	Цикли	Код команди	Біти стана	Прим.
1	2	3	4	5	6

ADDLW	Додавання константи і W	1	11 111x kkkk kkkk	C, DC, Z	–
ADDWF	Додавання W с F	1	00 0111 dfff ffff	C, DC, Z	1, 2
ANDLW	Логічне І константи і W	1	11 1001 kkkk kkkk	Z	–
ANDWF	Логічне І W і регістра F	1	00 0101 dfff ffff	Z	1, 2
BCF	Скидання біта в регістрі F	1	01 00bb bfff ffff	–	1, 2

Продовження таблиці 46

1	2	3	4	5	6
BSF	Встановлення біта в регістрі F	1	01 01bb bfff ffff	–	1, 2
BTFSC	Пропустити команду, якщо біт у F дорівнює нулю	1 (2)	01 10bb bfff ffff	–	3
BTFSS	Пропустити команду, якщо біт у F дорівнює одиниці	1 (2)	01 11bb bfff ffff	–	3
CALL	Виклик підпрограми	2	10 0kkk kkkk kkkk	–	–
CLRF	Скидання регістра F	1	00 0001 1fff ffff	Z	2
CLRW	Скидання регістра W	1	00 0001 0xxx xxxx	Z	–
CLRWDT	Скидання Watchdog-таймера WDT	1	00 0000 0110 0100	–	–
COMF	Інверсія регістра F	1	00 1001 dfff ffff	Z	1, 2
DECF	Декремент регістра F	1	00 0011 dfff ffff	Z	1, 2
DECFSZ	Декремент F, пропустити команду, якщо F став рівний 0	1 (2)	00 1011 dfff ffff	–	1, 2, 3
GOTO	Перехід за адресою	2	10 1kkk kkkk kkkk	–	–
INCF	Інкремент регістра F	1	00 1010 dfff ffff	Z	1, 2
INCFSZ	Інкремент F, пропустити команду, якщо F став рівний 0	1 (2)	00 1111 dfff ffff	–	1, 2, 3
IORLW	Логічне АБО константи і W	1	11 1000 kkkk kkkk	Z	–
IORWF	Логічне АБО W і F	1	00 0100 dfff ffff	Z	1, 2
MOVF	Пересилання регістра F	1	00 1000 dfff ffff	Z	1, 2

MOVLW	Пересилання константи в W	1	11 00xx kkkk kkkk	–	–
MOVWF	Пересилання W у F	1	00 0000 1fff ffff	–	–
NOP	Команда "немає операції"	1	00 0000 0xx0 0000	–	–
OPTION	Завантаження реєстра OPTION	1	00 0000 0110 0010	–	–
RETFIE	Повернення з переривання	2	00 0000 0000 1001	–	–

Закінчення таблиці 46

1	2	3	4	5	6
RETLW	Повернення з підпрограми з завантаженням константи у W	2	11 01xx kkkk kkkk	–	–
RETURN	Повернення з підпрограми	2	00 0000 0000 1000	–	–
RLF	Зсув F вліво через ознаку переноса	1	00 1101 dfff ffff	C	1, 2
RRF	Зсув F вправо через ознаку переноса	1	00 1100 dfff ffff	C	1, 2
SLEEP	Перехід у режим SLEEP	1	00 0000 0110 0011	–	–
SUBLW	Віднімання W з константи	1	11 110x kkkk kkkk	C, DC, Z	–
SUBWF	Віднімання W з F	1	00 0010 dfff ffff	C, DC, Z	1, 2
SWAPF	Обмін місцями тетрад в F	1	00 1110 dfff ffff	–	1, 2
TRIS	Завантаження реєстра TRIS	1	00 0000 0110 0fff	–	–
XORLW	Виключаюче АБО константи і W	1	11 1010 kkkk kkkk	Z	–
XORWF	Виключаюче АБО W і F	1	00 0110 dfff ffff	–	1, 2

Примітки:

1. Якщо модифікується регістр вводу-виводу (наприклад, MOVF PORTB,1), то використовується значення, що зчитується з виходів. Наприклад, якщо у вихідній засувці порту, ввімкненого на введення даних, знаходиться 1, а зовнішній пристрій формує на цьому виході 0, то в цьому розряді даних буде записаний 0.
2. Якщо операндом команди є вміст регістра TMR0 (i, якщо припустимо, d=1), то попередній подільник, якщо він під'єднаний до TMR0, буде скинутий.
3. Якщо в результаті виконання команди змінюється лічильник команд або виконується перехід у результаті перевірки умови, то команда виконується за два цикли. Другий цикл виконується як NOP

5.18. Приклади застосування

5.18.1. Універсальний таймер

Пристрій призначений для реалізації широкого діапазону витримки часу від одиниць секунд до 13 діб. Таймер має можливість працювати у циклічному та одиничного режимах роботи, з можливістю вимикання індикації для зменшення енергоспоживання. Встановлене значення часу спрацювання може бути збережено у пам'яті. Таймер указує свій стан на світлодіодному трицифровому індикаторі зі спільним анодом. Схема пристрою зображена на рис. 103.

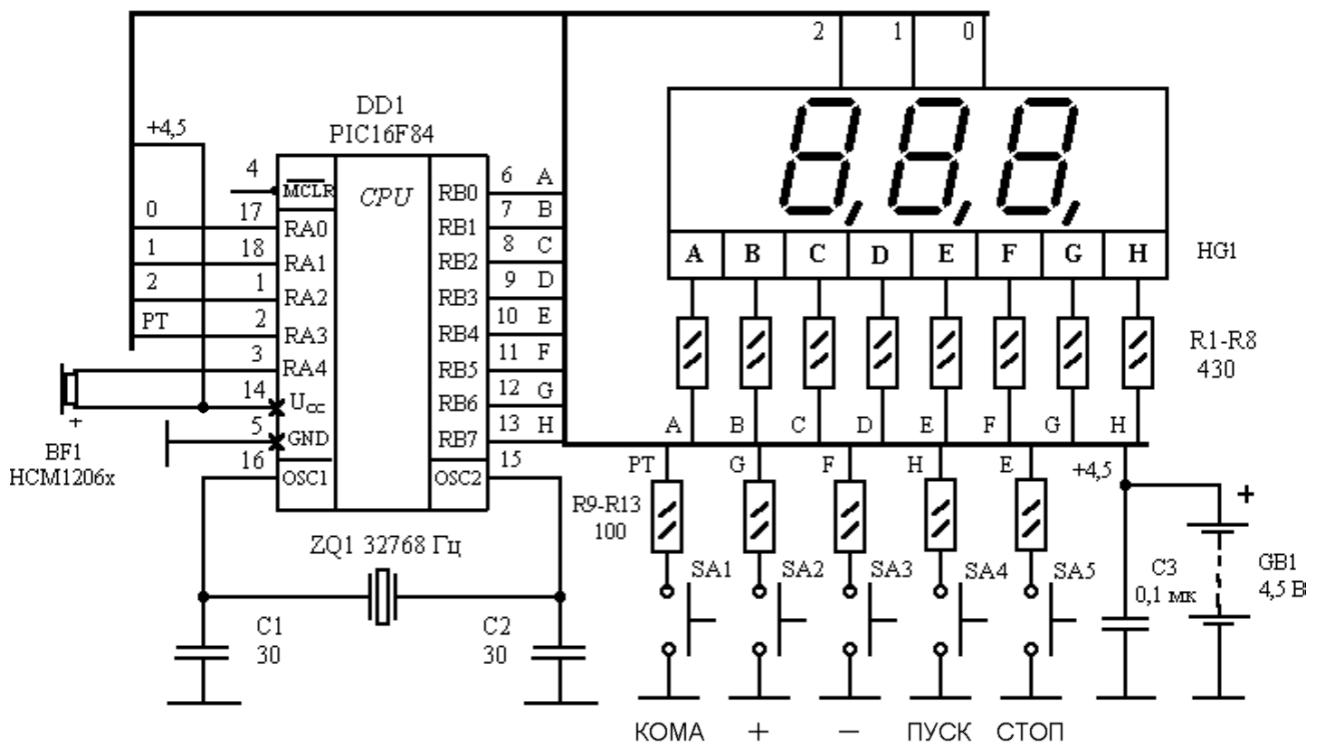


Рис. 103. Універсальний таймер

Входи RB0-RB7 використовуються як виходи значень сегментів світлодіодів. Під час переривання входи RB4-RB7 навантажуються кнопками SA2-SA5: "+", «Мінус», «Пуск», «Стоп». Кнопка SA1 ("кома") під'єднана до постійного входу RA3. Призначення кнопок подана у таблиці 47.

Таблиця 47. Призначення кнопок та їх комбінацій

Комбінація кнопок	Призначення
КОМА	Перемикання позиції вводу розрядів індикатора
ПЛЮС	Додавання одиниці до обраного розряду часу роботи
МІНУС	Віднімання одиниці з обраного розряду часу роботи
ПУСК	Вмикання індикації і таймера
СТОП	Вимикання індикації й зупинка таймера
СТОП та ПЛЮС	Запис в енергонезалежну пам'ять значення, що відображається на індикаторі
СТОП та МІНУС	Зчитування з пам'яті збереженого значення
СТОП та КОМА	Включення циклічного режиму роботи

Виводи RA0-RA3 використовуються, як виходи анодів світлодіодів.

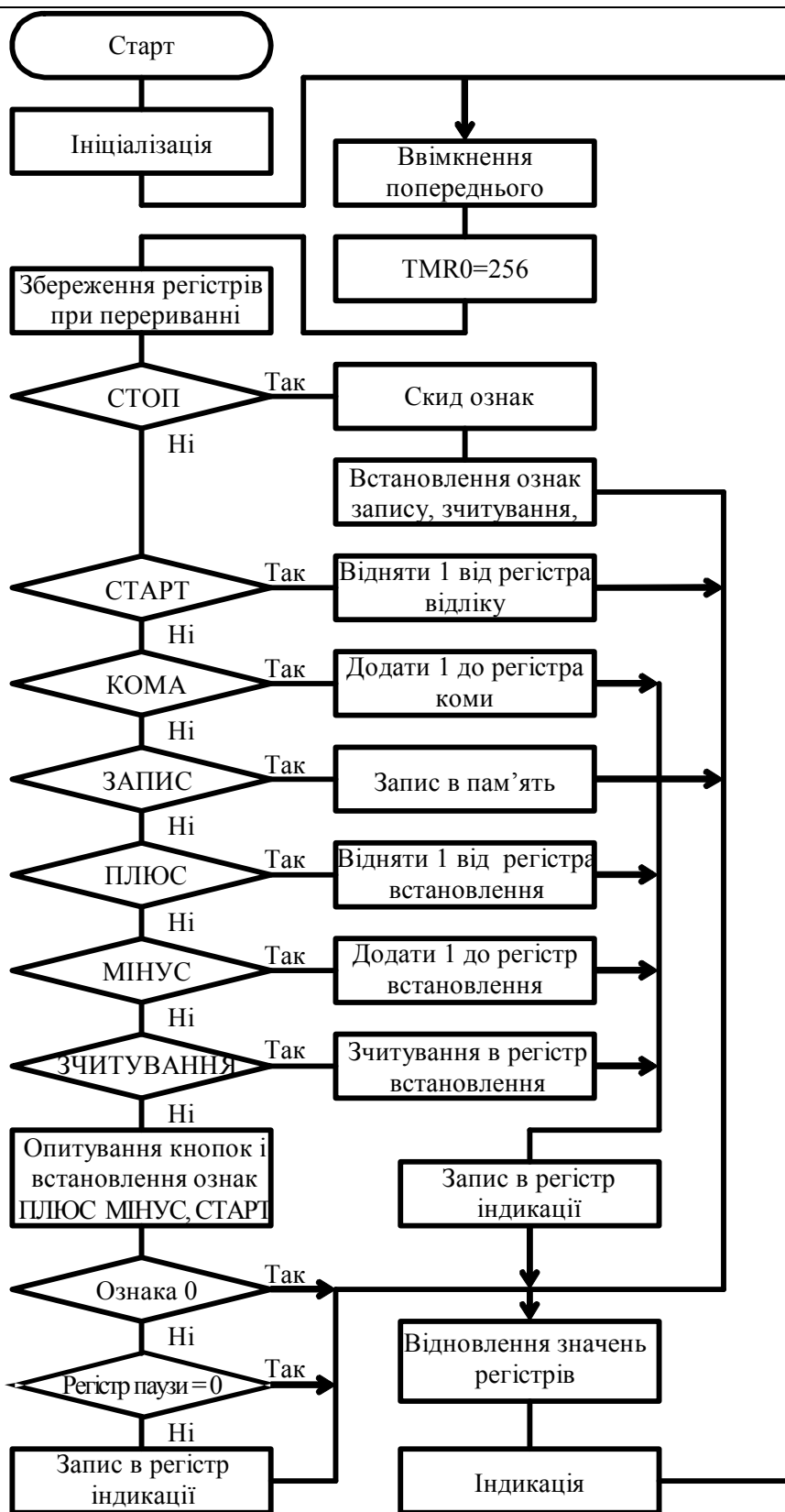


Рис. 104. Алгоритм роботи

Вивід RA4 на схемі навантажений звуковим динаміком HCM1206x з вбудованим генератором частотою 2 кГц. Цей вивід включає навантаження нульовим потенціалом і може бути використаний для під'єднання потужного

навантаження звичайними способами, тобто за допомогою реле, оптопари або тиристора.

Алгоритм робот таймера зображений на рис. 104. Після пуску і початкового встановлення реєстрів вмикається таймер із попереднім подільником, які мають такий коефіцієнт поділу, що таймер переповняється щосекунди. В момент переповнення таймера відбувається переривання. Під час переривання відбувається опитування стана кнопок та ознак режиму роботи. У першу секунду за натисненням кнопки встановлюється відповідна ознака. У наступну секунду, за встановленою ознакою, виконується необхідна дія. Кожна дія встановлення закінчується відновленням значень індикації після переривання. Після завершення процедури обробки переривання виконує основну роботу по організації динамічної індикації.

Ввід значення часового інтервалу здійснюється по одному знаку за допомогою кнопок ПЛЮС та МІНУС. Перехід до наступного знаку за допомогою кнопки КОМА. Включення коми коло знаку вказує на можливість його зміни за допомогою кнопок ПЛЮС та МІНУС. Для спрощення схеми кнопка КОМА також визначає діапазон відліку (0 – 999 секунд, 0 – 999 хвилин, або 0 – 999 годин). Встановлення діапазону відбувається в залежності від положення коми у момент запуску відліку. Індикація безперервного режиму здійснюється чергуванням включення всіх ком і коми в обраному розряді.

5.18.2. Частотомір на PIC-контролері.

Прилад призначений для вимірювання частоти логічних сигналів, а також періодичних сигналів непрямокутної форми позитивної полярності. Частотомір, принципова схема якого зображена на рисунку 105, дозволяє вимірювати частоту періодичних сигналів у діапазоні 250 Гц...50 МГц. Похибка вимірювань і відліку для кожного інтервалу частот наведені в таблиці 48. Практично єдиний елемент частотоміра – мікроконтролер PIC16F84. Він здійснює відлік імпульсів зовнішнього сигналу, що надходить на вхід приладу, опрацювання отриманих значень і виведення результатів вимірювання на індикаційне табло.

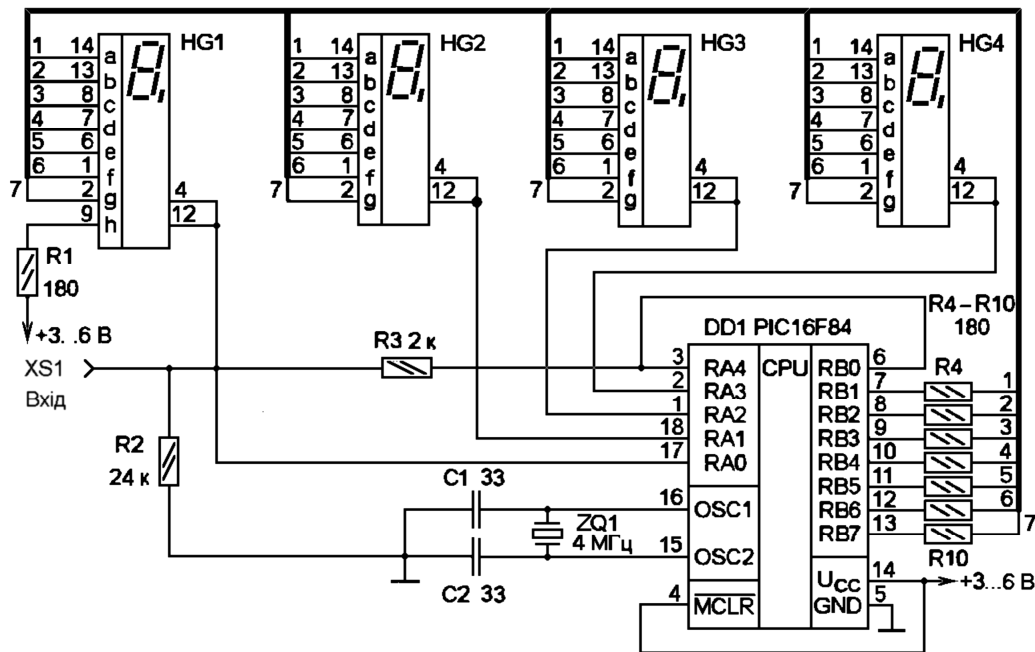


Рис. 105 Схема частотоміра на PIC-контролері

Таблиця 48. Параметри приладу

Інтервал частот, кГц (МГц)	Час вимірювань, мс	Похибка, Гц	
		Вимірювань	Відліку
0,25... 0,999 кГц	500	±2	±2
1...9.99 кГц	500	±2	±5
10... 69.9 кГц	500	±2	±50
100... 127 кГц	500	±2	±500
128...999 кГц	1	±1000	±1000
1...9.99 МГц	1	±1000	±5000
10... 50 МГц	1	±1000	±5000

Частота (у герцах) відображається індикаторами HG1-HG4 у форматі X.YZ F Гц, де X.YZ – десяткове значення частоти сигналу, а F - порядок числа (наприклад, значення "2.25 3" відповідає частоті $2,25 \cdot 10^3 = 2250$ Гц; "4.32 5" – $4,32 \cdot 10^5 = 432\,000$ Гц = 432 кГц і т.д.).

Мікроконтролер PIC16F84 має у своєму складі восьмирозрядний модуль таймера (TMR0), що використовується з восьмирозрядним передподільником. Останній функціонує асинхронно, тому таймер може рахувати частоту сигналів значно вище частоти генератора мікроконтролера, що у даному випадку дорівнює 4 МГц. Мінімальний час високого і низького рівнів вхідного сигналу — 10 нс, це дозволяє модулеві TMR0 функціонувати від зовнішнього сигналу частотою до 50 МГц.

Вимірюваний сигнал через резистор R2 надходить на вивід RA4 DD1, що є входом зовнішнього сигналу (ТОСКИ) таймера TMR0. Цей вивід з'єднаний з RB0, перемиканням якого здійснюється керування режимом відліку. Перед вимірюванням відбувається скидання TMR0 (при цьому скидається і попередній подільник). Для вимірювання частоти вивід RB0 конфігурується як вхід на точні інтервали часу, що дозволяє зовнішньому сигналові надходити на вхід таймера. Відлік тривалості інтервалів здійснюється програмою і виконується як точна часова затримка. Після закінчення її вивід RB0 конфігурується як вихід, TMR0 припиняє роботу, оскільки на RA4 встановлюється низький рівень, і зовнішній сигнал перестає надходити на його вхід. Потім зчитується накопичене 16-розрядне значення числа періодів вхідного сигналу: у старші вісім розрядів записується вміст TMR0, а в молодші – попереднього подільника. Для отримання значення попереднього подільника виконується додаткова підпрограма (з цією метою на виводі RA4 командами BSF і BCF перемикається вихідний рівень, тобто програмно формується послідовність коротких імпульсів). Кожен імпульс інкрементує попередній подільник і лічильник імпульсів N, після чого перевіряється вміст TMR0, щоб визначити, чи збільшився він. Якщо він зріс на 1, восьмирозрядне значення попереднього подільника визначається за вмістом лічильника імпульсів N як $256 - N$. Далі 16-розрядне двійкове значення частоти перетворюється в шестирозрядне десяткове, котре заокруглюється до тризначного, а потім формується зазначений вище експоненціальний формат для виведення на табло в динамічному режимі. Сканування індикаторів відбувається з частотою приблизно 80 Гц. Висока навантажувальна здатність мікроконтролера дозволила підключити індикатори безпосередньо до його виводів.

Вимірювання частоти відбувається в два етапи. Спочатку формується інтервал часу (програмна затримка) тривалістю 1 мс, що відповідає області високих частот. Якщо отримане значення частоти більше 127 (старший байт – значення TMR0, старший розряд молодшого байта – значення попереднього подільника – не дорівнюють “0”), воно перетворюється, і результат виведеться на

індикатори. Після цього цикл повторюється. Якщо ж значення частоти менше 127, виконується друге вимірювання (для низьких частот), при якому формується інтервал часу тривалістю 0,5 с. Для оптимізації роботи мікроконтролера він об'єднаний з циклом виведення результату попереднього вимірювання на індикатори. Якщо отримане в результаті вимірювання значення частоти більше 127, воно перетворюється для індикації, при меншому – проводить третій етап вимірювань. Під його час програмно визначається тривалість періоду імпульсу за допомогою визначення часу, необхідного для надходження 10 імпульсів на вхід частотоміра. Отримане значення перетворюється для індикації за відповідною програмою. Якщо за 0.2 с 10 імпульсів на вхід не надійшли вважається, що імпульси взагалі відсутні і покази індикаторів обнулюються. Після цього повний цикл вимірювань повторюється.

5.19. Контрольні запитання

1. Основні технічні характеристики мікроконтролерів PIC16X8X.
2. Система команд PIC 16x8x.
3. Проектування мікропроцесорних систем на базі PIC 16x8x.
4. PIC-контролер 16C84. Основні характеристики.
5. Архітектура PIC-контролера PIC16F84. Регістри контролера, їх призначення. Регістр статусу. RTCC таймер/ лічильник.
6. Організація резидентної пам'яті даних та пам'яті програм. PC і адресація ПЗП.
7. Система переривань. Зовнішнє переривання.
8. Переривання від RTCC. Переривання від порту RB. Переривання від EEPROM.
9. Стек і повернення з підпрограм.
10. Порти вводу-виводу.
11. Регістри портів.
12. Схема ліній порту A.
13. Схема ліній порту B.

14. Байт-орієнтовані команди асемблера PIC16F84 .
15. Операції з бітами асемблера PIC16F84.
16. Команди переходів асемблера PIC16F84.
17. Команди викликів та повернення з підпрограм асемблера PIC16F84.

СПИСОК ЛІТЕРАТУРИ

1. Сташин В. В., Урсов А. В., Мологонцева О. Ф. Проектирование цифровых устройств на однокристальных микроконтролерах. /Сташин В. В., Урсов А. В., Мологонцева О. Ф. – М.: Энергоатомиздат, 1990. – 224 с.
2. Предко М. Руководство по микроконтролерам. /Предко М. – М.: Постмаркет, 2001–.
4. Зубчук В.И., Сигорский В.П., Шкуро А.Н. Справочник по цифровой схемотехнике. /Зубчук В.И., Сигорский В.П., Шкуро А.Н. – К.: Техніка, 1990.– 448 с.
5. Проектирование микропроцессорной электронно-вычислительной аппаратуры. Справочник. [Артюхов В. Г., Будняк А. А., Лапий В. Ю., Молявко С. М., Петренко А. И.] – К.: Техніка, 1988. – 263 с.
6. Микропроцессорный комплект К1810. Структура , программирование, применение./ [Казаринов Ю. М., Номоконов В. Н., Подклетнов Г. С., Филиппов Ф. В.] – М. Высшая школа., 1990. – 269 с.
7. Мікропроцесорна техніка. [Якименко Ю. І., Терещенко Т. О., Сокол Є. І., Жуйков В. Я., Петергера Ю. С.]; за ред. Терещенко Т. О., – К.: "Видавництво Політехніка", "Кондор" 2004. – 440 с.
8. Бродин В. Б., Калинин А. В. Системы на микроконтролерах и БИС программируемой логики./ Бродин В. Б., Калинин А. В. – М.: ЭКОМ, 2002. – 400 с.
9. Коффрон Дж. Технические средства микропроцессорных систем. Практический курс./ Коффрон Дж.; пер. с англ. Балыбердина А. В.– М. Мир,– 1983. – 344 с.
10. Однокристальные микроЭВМ./ [Боборыкин А.В., Липовецкий Г.П., Литвинский Г.В., Оксинь О.Н., Прохорчик С.В., Проценко Л.В., Пертенко Н.В., Сергеев А.А., Сивобород. П.В.]. – М.: МИКАП, 1994. – 400 с.
11. Ульрих В.А. Микроконтролеры PIC16X7XX./ Ульрих В.А. – [2-е изд. перераб и доп.]. – СПб.: Наука и Техника, 2002. – 320 с.

12. Тавернье К. PIC-Микроконтролер. Практика применения./ Тавернье К. – М.: ДМКПресс, 2004. – 272 с.

13. PIC16F84 Data Sheet. 18-pin 8-Bit CMOS EEPROM Microcontroller. – Microchip Technology Inc. 1996. – 108 с. – Режим доступа: <http://ww1.microchip.com/downloads/en/DeviceDoc/30445c.pdf>.

14. PIC16F84A Data Sheet 18-pin Enhanced FLASH/EEPROM 8-bit Microcontroller. – Microchip Technology Inc. 2001. – 86 с. – Режим доступа: <http://ww1.microchip.com/downloads/en/DeviceDoc/35007b.pdf>.